

Advances in Intelligent Systems and Computing 1230

Kohei Arai  
Supriya Kapoor  
Rahul Bhatia *Editors*

# Intelligent Computing

Proceedings of the 2020 Computing  
Conference, Volume 3

 Springer

# Advances in Intelligent Systems and Computing

Volume 1230

## Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,  
Warsaw, Poland

## Advisory Editors

Nikhil R. Pal, Indian Statistical Institute, Kolkata, India

Rafael Bello Perez, Faculty of Mathematics, Physics and Computing,  
Universidad Central de Las Villas, Santa Clara, Cuba

Emilio S. Corchado, University of Salamanca, Salamanca, Spain

Hani Hagras, School of Computer Science and Electronic Engineering,  
University of Essex, Colchester, UK

László T. Kóczy, Department of Automation, Széchenyi István University,  
Gyor, Hungary


Vladik Kreinovich, Department of Computer Science, University of Texas  
at El Paso, El Paso, TX, USA

Chin-Teng Lin, Department of Electrical Engineering, National Chiao  
Tung University, Hsinchu, Taiwan

Jie Lu, Faculty of Engineering and Information Technology,  
University of Technology Sydney, Sydney, NSW, Australia

Patricia Melin, Graduate Program of Computer Science, Tijuana Institute  
of Technology, Tijuana, Mexico

Nadia Nedjah, Department of Electronics Engineering, University of Rio de Janeiro,  
Rio de Janeiro, Brazil

Ngoc Thanh Nguyen , Faculty of Computer Science and Management,  
Wrocław University of Technology, Wrocław, Poland

Jun Wang, Department of Mechanical and Automation Engineering,  
The Chinese University of Hong Kong, Shatin, Hong Kong



The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within “Advances in Intelligent Systems and Computing” are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

**\*\* Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink \*\***

More information about this series at <http://www.springer.com/series/11156>

Kohei Arai · Supriya Kapoor ·  
Rahul Bhatia  
Editors

# Intelligent Computing

Proceedings of the 2020 Computing  
Conference, Volume 3

 Springer

*Editors*

Kohei Arai  
Faculty of Science and Engineering  
Saga University  
Saga, Japan

Supriya Kapoor  
The Science and Information  
(SAI) Organization  
Bradford, West Yorkshire, UK

Rahul Bhatia  
The Science and Information  
(SAI) Organization  
Bradford, West Yorkshire, UK

ISSN 2194-5357                      ISSN 2194-5365 (electronic)  
Advances in Intelligent Systems and Computing  
ISBN 978-3-030-52242-1              ISBN 978-3-030-52243-8 (eBook)  
<https://doi.org/10.1007/978-3-030-52243-8>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Editor's Preface

On behalf of the Committee, we welcome you to the Computing Conference 2020.

The aim of this conference is to give a platform to researchers with fundamental contributions and to be a premier venue for industry practitioners to share and report on up-to-the-minute innovations and developments, to summarize the state of the art and to exchange ideas and advances in all aspects of computer sciences and its applications.

For this edition of the conference, we received 514 submissions from 50+ countries around the world. These submissions underwent a double-blind peer review process. Of those 514 submissions, 160 submissions (including 15 posters) have been selected to be included in this proceedings. The published proceedings has been divided into three volumes covering a wide range of conference tracks, such as technology trends, computing, intelligent systems, machine vision, security, communication, electronics and e-learning to name a few. In addition to the contributed papers, the conference program included inspiring keynote talks. Their talks were anticipated to pique the interest of the entire computing audience by their thought-provoking claims which were streamed live during the conferences. Also, the authors had very professionally presented their research papers which were viewed by a large international audience online. All this digital content engaged significant contemplation and discussions amongst all participants.

Deep appreciation goes to the keynote speakers for sharing their knowledge and expertise with us and to all the authors who have spent the time and effort to contribute significantly to this conference. We are also indebted to the Organizing Committee for their great efforts in ensuring the successful implementation of the conference. In particular, we would like to thank the Technical Committee for their constructive and enlightening reviews on the manuscripts in the limited timescale.

We hope that all the participants and the interested readers benefit scientifically from this book and find it stimulating in the process. We are pleased to present the proceedings of this conference as its published record.

Hope to see you in 2021, in our next Computing Conference, with the same amplitude, focus and determination.

Kohei Arai

# Contents

<b>Preventing Neural Network Weight Stealing via Network Obfuscation</b> . . . . .	1
Kálmán Szentannai, Jalal Al-Afandi, and András Horváth	
<b>Applications of Z-Numbers and Neural Networks in Engineering</b> . . . . .	12
Raheleh Jafari, Sina Razvarz, and Alexander Gegov	
<b>5G-FOG: Freezing of Gait Identification in Multi-class Softmax Neural Network Exploiting 5G Spectrum</b> . . . . .	26
Jan Sher Khan, Ahsen Tahir, Jawad Ahmad, Syed Aziz Shah, Qammer H. Abbasi, Gordon Russell, and William Buchanan	
<b>Adaptive Blending Units: Trainable Activation Functions for Deep Neural Networks</b> . . . . .	37
Leon René Sütthfeld, Flemming Brieger, Holger Finger, Sonja Füllhase, and Gordon Pipa	
<b>Application of Neural Networks to Characterization of Chemical Sensors</b> . . . . .	51
Mahmoud Zaki Iskandarani	
<b>Application of Machine Learning in Deception Detection</b> . . . . .	61
Owolafe Otasowie	
<b>A New Approach to Estimate the Discharge Coefficient in Sharp-Crested Rectangular Side Orifices Using Gene Expression Programming</b> . . . . .	77
Hossein Bonakdari, Bahram Gharabaghi, Isa Ebtehaj, and Ali Sharifi	
<b>DiaTTroD: A Logical Agent Diagnostic Test for Tropical Diseases</b> . . . . .	97
Sandra Mae W. Famador and Tardi Tjahjadi	
<b>A Weighted Combination Method of Multiple K-Nearest Neighbor Classifiers for EEG-Based Cognitive Task Classification</b> . . . . .	116
Abduljalil Mohamed, Amer Mohamed, and Yasir Mustafa	

<b>Detection and Localization of Breast Tumor in 2D Using Microwave Imaging</b> . . . . .	132
Abdelfettah Miraoui, Lotfi Merad Sidi, and Mohamed Meriah	
<b>Regression Analysis of Brain Biomechanics Under Uniaxial Deformation</b> . . . . .	142
O. Abuomar, F. Patterson, and R. K. Prabhu	
<b>Exudate-Based Classification for Detection of Severity of Diabetic Macula Edema</b> . . . . .	150
Nandana Prabhu, Deepak Bhoir, Nita Shanbhag, and Uma Rao	
<b>Analysis and Detection of Brain Tumor Using U-Net-Based Deep Learning</b> . . . . .	161
Vibhu Garg, Madhur Bansal, A. Sanjana, and Mayank Dave	
<b>Implementation of Deep Neural Networks in Facial Emotion Perception in Patients Suffering from Depressive Disorder: Promising Tool in the Diagnostic Process and Treatment Evaluation</b> . . . . .	174
Krzysztof Michalik and Katarzyna Kucharska	
<b>Invisibility and Fidelity Vector Map Watermarking Based on Linear Cellular Automata Transform</b> . . . . .	185
Saleh Al-Ardhi, Vijey Thayananthan, and Abdullah Basuhail	
<b>Implementing Variable Power Transmission Patterns for Authentication Purposes</b> . . . . .	198
Hosam Alamlah, Ali Abdullah S. Alqahtani, and Dalia Alamlah	
<b>SADDLE: Secure Aerial Data Delivery with Lightweight Encryption</b> . . . . .	204
Anthony Demeri, William Diehl, and Ahmad Salman	
<b>Malware Analysis with Machine Learning for Evaluating the Integrity of Mission Critical Devices</b> . . . . .	224
Robert Heras and Alexander Perez-Pons	
<b>Enhanced Security Using Elasticsearch and Machine Learning</b> . . . . .	244
Ovidiu Negoita and Mihai Carabas	
<b>Memory Incentive Provenance (MIP) to Secure the Wireless Sensor Data Stream</b> . . . . .	255
Mohammad Amanul Islam	
<b>Tightly Close It, Robustly Secure It: Key-Based Lightweight Process for Propping up Encryption Techniques</b> . . . . .	278
Muhammed Jassem Al-Muhammed, Ahmad Al-Daraiseh, and Raed Abuzitar	

**Statistical Analysis to Optimize the Generation of Cryptographic Keys from Physical Unclonable Functions** . . . . . 302  
 Bertrand Cambou, Mohammad Mohammadi, Christopher Philabaum, and Duane Booher

**Towards an Intelligent Intrusion Detection System: A Proposed Framework** . . . . . 322  
 Raghda Fawzey Hriez, Ali Hadi, and Jalal Omer Atoum

**LockChain Technology as One Source of Truth for Cyber, Information Security and Privacy** . . . . . 336  
 Yuri Bobbert and Nese Ozkanli

**Introduction of a Hybrid Monitor for Cyber-Physical Systems** . . . . . 348  
 J. Ceasar Aguma, Bruce McMillin, and Amelia Regan

**Software Implementation of a SRAM PUF-Based Password Manager** . . . . . 361  
 Sareh Assiri, Bertrand Cambou, D. Duane Booher, and Mohammad Mohammadinodoushan

**Contactless Palm Vein Authentication Security Technique for Better Adoption of e-Commerce in Developing Countries** . . . . . 380  
 Sunday Alabi, Martin White, and Natalia Beloff

**LightGBM Algorithm for Malware Detection** . . . . . 391  
 Mouhammd Al-kasassbeh, Mohammad A. Abbadi, and Ahmed M. Al-Bustanji

**Exploiting Linearity in White-Box AES with Differential Computation Analysis** . . . . . 404  
 Jakub Klemsa and Martin Novotný

**Immune-Based Network Dynamic Risk Control Strategy Knowledge Ontology Construction** . . . . . 420  
 Meng Huang, Tao Li, Hui Zhao, Xiaojie Liu, and Zhan Gao

**Windows 10 Hibernation File Forensics** . . . . . 431  
 Ahmad Ghafarian and Deniz Keskin

**Behavior and Biometrics Based Masquerade Detection Mobile Application** . . . . . 446  
 Pranieth Chandrasekara, Hasini Abeywardana, Sammani Rajapaksha, Sanjeevan Parameshwaran, and Kavinga Yapa Abeywardana

**Spoofed/Unintentional Fingerprint Detection Using Behavioral Biometric Features** . . . . . 459  
 Ammar S. Salman and Odai S. Salman

**Enabling Paratransit and TNC Services with Blockchain Based Smart Contracts** . . . . . 471  
 Amari N. Lewis and Amelia C. Regan



<b>A Review of Cyber Security Issues in Hospitality Industry</b> . . . . .	482
Neda Shabani and Arslan Munir	
<b>Extended Protocol Using Keyless Encryption Based on Memristors</b> . . . . .	494
Yuxuan Zhu, Bertrand Cambou, David Hely, and Sareh Assiri	
<b>Recommendations for Effective Security Assurance of Software-Dependent Systems</b> . . . . .	511
Jason Jaskolka	
<b>On Generating Cancelable Biometric Templates Using Visual Secret Sharing</b> . . . . .	532
Manisha and Nitin Kumar	
<b>An Integrated Safe and Secure Approach for Authentication and Secret Key Establishment in Automotive Cyber-Physical Systems</b> . . . . .	545
Naresh Kumar Giri, Arslan Munir, and Joonho Kong	
<b>How Many Clusters? An Entropic Approach to Hierarchical Cluster Analysis</b> . . . . .	560
Sergei Koltcov, Vera Ignatenko, and Sergei Pashakhin	
<b>Analysis of Structural Liveness and Boundedness in Weighted Free-Choice Net Based on Circuit Flow Values</b> . . . . .	570
Yojiro Harie and Katsumi Wasaki	
<b>Classification of a Pedestrian's Behaviour Using Dual Deep Neural Networks</b> . . . . .	581
James Spooner, Madeline Cheah, Vasile Palade, Stratis Kanarachos, and Alireza Daneshkhah	
<b>Towards Porting Astrophysics Visual Analytics Services in the European Open Science Cloud</b> . . . . .	598
Eva Sciacca, Fabio Vitello, Ugo Becciani, Cristobal Bordiu, Filomena Bufano, Antonio Calanducci, Alessandro Costa, Mario Raciti, and Simone Riggi	
<b>Computer Graphics-Based Analysis of Anterior Cruciate Ligament in a Partially Replaced Knee</b> . . . . .	607
Ahmed Imran	
<b>An Assessment Algorithm for Evaluating Students Satisfaction in e-Learning Environments: A Case Study</b> . . . . .	613
M. Caramihai, Irina Severin, and Ana Maria Bogatu	
<b>The Use of New Technologies in the Organization of the Educational Process</b> . . . . .	622
Y. A. Daineko, N. T. Duzbayev, K. B. Kozhaly, M. T. Ipalakova, Zh. M. Bekaulova, N. Zh. Nalgozhina, and R. N. Sharshova	

**Design and Implementation of Cryptocurrency Price Prediction System . . . . . 628**  
Milena Karova, Ivaylo Penev, and Daniel Marinov

**Strategic Behavior Discovery of Multi-agent Systems Based on Deep Learning Technique . . . . . 644**  
Boris Morose, Sabina Aledort, and Gal Zaidman

**Development of Prediction Methods for Taxi Order Service on the Basis of Intellectual Data Analysis. . . . . 652**  
N. A. Andriyanov

**Discourse Analysis on Learning Theories and AI. . . . . 665**  
Rosemary Papa, Karen Moran Jackson, Ric Brown, and David Jackson

**False Asymptotic Instability Behavior at Iterated Functions with Lyapunov Stability in Nonlinear Time Series . . . . . 673**  
Charles Roberto Telles

**The Influence of Methodological Tools on the Diagnosed Level of Intellectual Competence in Older Adolescents . . . . . 694**  
Sipovskaya Yana Ivanovna

**The Automated Solar Activity Prediction System (ASAP) Update Based on Optimization of a Machine Learning Approach . . . . . 702**  
Ali K. Abed and Rami Qahwaji

**Author Index. . . . . 719**



# Preventing Neural Network Weight Stealing via Network Obfuscation

Kálmán Szentannai, Jalal Al-Afandi, and András Horváth<sup>(✉)</sup>

Faculty of Information Technology and Bionics, Peter Pazmany Catholic University,  
Práter u. 50/A, Budapest 1083, Hungary  
horvath.andras@itk.ppke.hu

**Abstract.** Deep Neural Networks are robust to minor perturbations of the learned network parameters and their minor modifications do not change the overall network response significantly. This allows space for model stealing, where a malevolent attacker can steal an already trained network, modify the weights and claim the new network his own intellectual property. In certain cases this can prevent the free distribution and application of networks in the embedded domain. In this paper, we propose a method for creating an equivalent version of an already trained fully connected deep neural network that can prevent network stealing, namely, it produces the same responses and classification accuracy, but it is extremely sensitive to weight changes.

**Keywords:** Neural networks · Networks stealing · Weight stealing · Obfuscation

## 1 Introduction

Deep neural networks are employed in an emerging number of tasks, many of which were not solvable before with traditional machine learning approaches. In these structures, expert knowledge which is represented in annotated datasets is transformed into learned network parameters known as network weights during training.

Methods, approaches and network architectures are distributed openly in this community, but most companies protect their data and trained networks obtained from tremendous amount of working hours annotating datasets and fine-tuning training parameters.

Model stealing and detection of unauthorized use via stolen weights is a key challenge of the field as there are techniques (scaling, noising, fine-tuning, distillation) to modify the weights to hide the abuse, while preserving the functionality and accuracy of the original network. Since networks are trained by stochastic optimization methods and are initialized with random weights, training on a dataset might result various different networks with similar accuracy.

There are several existing methods to measure distances between network weights after these modifications and independent trainings: [1–3] Obfuscation of

neural networks was introduced in [4], which showed the viability and importance of these approaches. In this paper the authors present a method to obfuscate the architecture, but not the learned network functionality. We would argue that most ownership concerns are not raised because of network architectures, since most industrial applications use previously published structures, but because of network functionality and the learned weights of the network.

Other approaches try to embed additional, hidden information in the network such as hidden functionalities or non-plausible, predefined answers for previously selected images (usually referred as watermarks) [5, 6]. In case of a stolen network one can claim ownership of the network by unraveling the hidden functionality, which can not just be formed randomly in the structure. A good summary comparing different watermarking methods and their possible evasions can be found in [7].

Instead of creating evidence, based on which relation between the original and the stolen, modified model could be proven, we have developed a method which generates a completely sensitive and fragile network, which can be freely shared, since even minor modification of the network weights would drastically alter the networks response.

In this paper, we present a method which can transform a previously trained network into a fragile one, by extending the number of neurons in the selected layers, without changing the response of the network. These transformations can be applied in an iterative manner on any layer of the network, except the first and the last layers (since their size is determined by the problem representation). In Sect. 2 we will first introduce our method and the possible modifications on stolen networks and in Sect. 3 we will describe our simulations and results. Finally in Sect. 4 we will conclude our results and describe our planned future work.

## 2 Mathematical Model of Unrobust Networks

### 2.1 Fully Connected Layers

In this section we would like to present our method, how a robust network can be transformed into a non-robust one. We have chosen fully connected networks because of their generality and compact mathematical representation. Fully connected networks are generally applied covering the whole spectrum of machine learning problems from regression through data generation to classification problems. The authors can not deny the fact, that in most practical problems convolutional networks are used, but we would like to emphasize the following properties of fully connected networks: **(1)** in those cases when there is no topographic correlation in the data, fully connected networks are applied **(2)** most problems also contain additional fully connected layers after the feature extraction of the convolutional or residual layers **(3)** every convolutional network can be considered as a special case of fully connected ones, where all weights outside the convolutional kernels are set to zero.

A fully connected deep neural network might consist of several hidden layers each containing certain number of neurons. Since all layers have the same

architecture, without the loss of generality, we will focus here only on three consecutive layers in the network ( $i - 1$ ,  $i$  and  $i + 1$ ). We will show how neurons in layer  $i$  can be changed, increasing the number of neurons in this layer and making the network fragile, meanwhile keeping the functionality of the three layers intact. We have to emphasize that this method can be applied on any three layers, including the first and last three layers of the network and also that it can be applied repeatedly on each layer, still without changing the overall functionality of the network.

The input of the layer  $i$ , the activations of the previous layer ( $i - 1$ ) can be noted by the vector  $x_{i-1}$  containing  $N$  elements. The weights of the network are noted by the weight matrix  $W_i$  and the bias  $b_i$  where  $W$  is a matrix of  $N \times K$  elements, creating a mapping  $\mathbb{R}^N \mapsto \mathbb{R}^K$  and  $b_i$  is a vector containing  $K$  elements. The output of layer  $i$ , also the input of layer  $i + 1$  can be written as:

$$x_i = \phi(W_{i_{N \times K}} x_{i-1} + b_i) \quad (1)$$

where  $\phi$  is the activation function of the neurons.

The activations of layer  $i + 1$  can be extended as using Eq. 1:

$$x_{i+1} = \phi(\phi(xW_{i-1_{N \times K}} + b_{i-1})W_{i_{K \times L}} + b_i) \quad (2)$$

Creating a mapping  $\mathbb{R}^N \mapsto \mathbb{R}^L$ .

One way of identifying a fully connected neural network is to represent it as a sequence of synaptic weights. Our assumption was that in case of model stealing certain application of additive noise on the weights would prevent others to reveal the attacker and conceal thievery. Since fully connected networks are known to be robust against such modifications, the attacker could use the modified network with approximately the same classification accuracy. Thus, our goal was to find a transformation that preserves the loss and accuracy rates of the network, but introduces a significant decrease in terms of the robustness against parameter tuning. In case of a three-layered structure one has to preserve the mapping between the first and third layers (Eq. 2) to keep the functionality of this three consecutive layers, but the mapping in Eq. 1 (the mapping between the first and second, or second and third layers), can be changed freely.

Also, our model must rely on an identification mechanism based on a representation of the synaptic weights. Therefore, the owner of a network should be able to verify the ownership based on the representation of the neural network, examining the average distance between the weights [7].

## 2.2 Decomposing Neurons

We would like to find such  $W'_{i-1_{N \times M}}$  and  $W'_{i_{M \times L}}$  ( $M \in \mathbb{N}, M > K$ ) matrices, for which:

$$\begin{aligned} & \phi(\phi(xW_{i-1_{N \times K}} + b_{i-1})W_{i_{K \times L}} + b_i) \\ &= \phi(\phi(xW'_{i-1_{N \times M}} + b'_{i-1})W'_{i_{M \times L}} + b_i) \end{aligned} \quad (3)$$

Considering the linear case when  $\phi(x) = x$  we obtain the following form:

$$\begin{aligned} & xW_{i-1N \times K} W_{iK \times L} + b_{i-1}W_{iK \times L} + b_i \\ &= xW'_{i-1N \times M} W'_{iM \times L} + b'_{i-1}W'_{iM \times L} + b_i \end{aligned} \quad (4)$$

The equation above holds only for the special case of  $\phi(x) = x$ , however in most cases nonlinear activation functions are used. We have selected the rectified linear unit (ReLU) for our investigation ( $\phi(x) = \max(0, x)$ ). This non-linearity consist of two linear parts, which means that a variable could be in a linear domain of Eq. 3 resulting selected lines of 4 (if  $x \geq 0$ ), or the equation system is independent from the variable if the activation function results a constant zero (if  $x \leq 0$ ). This way ReLU gives a selection of given variables (lines) of 4. However, applying the ReLU activation function has certain constraints.

Assume, that a neuron with the ReLU activation function should be replaced by two other neurons. This can be achieved by using an  $\alpha \in (0, 1)$  multiplier:

$$\phi\left(\sum_{i=1}^n W_{ji}^l x_i + b_j^l\right) = N_j^l \quad (5)$$

$$N_j^l = \alpha N_j^l + (1 - \alpha)N_j^l \quad (6)$$

where  $\alpha N_j^l$  and  $(1 - \alpha)N_j^l$  correspond to the activation of the two neurons. For each of these, the activation would only be positive if the original neuron had a positive activation, otherwise it would be zero, this means that all the decomposed neuron must have the same bias.

After decomposing a neuron, it is needed to choose the appropriate weights on the subsequent layer. A trivial solution is to keep the original synaptic weights represented by the  $\overline{W}_j^{l+1}$  column vector. This would lead to the same activation since

$$N_j^l \overline{W}_j^{l+1} = \alpha N_j^l \overline{W}_j^{l+1} + (1 - \alpha)N_j^l \overline{W}_j^{l+1} \quad (7)$$

A fragile network can be created by choosing the same synaptic weights for the selected two neurons, but it would be easy to spot by the attacker, thus another solution is needed. In order to find a nontrivial solution we constructed a linear equation system that can be described by equation system  $Ap = c$ , where  $A$  contains the original, already decomposed synaptic weights of the first layer, meanwhile,  $p$  represents the unknown synaptic weights of the subsequent layer. Vector  $c$  contains the corresponding weights from the original network multiplied together: each element represents the amount of activation related to one input. Finally the non-trivial solution can be obtained by solving the following non-homogeneous linear equation system for each output neuron where index  $j$  denotes the output neuron.

$$\begin{bmatrix} w_{11}^1 & w_{21}^1 & \dots & w_{m1}^1 \\ w_{12}^1 & w_{22}^1 & \dots & w_{m2}^1 \\ \vdots & \vdots & \ddots & \vdots \\ w_{1n}^1 & w_{2n}^1 & \dots & w_{mn}^1 \\ b_1^1 & b_2^1 & \dots & b_m^1 \end{bmatrix} \times \begin{bmatrix} w_{j1}^{2'} \\ w_{j2}^{2'} \\ \vdots \\ w_{jm}^{2'} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^k w_{ji}^2 w_{i1}^1 \\ \sum_{i=1}^k w_{ji}^2 w_{i2}^1 \\ \vdots \\ \sum_{i=1}^k w_{ji}^2 w_{ik}^1 \\ \sum_{i=1}^k w_{ji}^2 b_i^1 \end{bmatrix} \quad (8)$$

It is important to note, that all the predefined weights on layer  $l + 1$  might change. In summary, this step can be considered as the replacement of a layer, changing all synaptic weights connecting from and to this layer, but keeping the biases of the neurons and the functionality of the network intact.

The only constraint of this method is related to the number of neurons regarding the two consecutive layers. It is known, that for matrix  $A$  with the size of  $M \times N$ , equation  $Ap = c$  has a solution if and only if  $\text{rank}(A) = \text{rank}[A|c]$  where  $[A|c]$  is the extended matrix. The decomposition of a neuron described in Eq. 7 results in linearly dependent weight vectors on layer  $l$ , therefore when solving the equation system the rank of the matrix  $A$  is less than or equal to  $\min(N + 1, K)$ . If the rank is equal to  $N + 1$  (meaning that  $K \geq N + 1$ ) then vector  $c$  with the dimension of  $N + 1$  would not introduce a new dimension to the subspace defined by matrix  $A$ . However if  $\text{rank}(A) = K$  (meaning that  $K \leq N + 1$ ) then vector  $c$  could extend the subspace defined by  $A$ . Therefore, the general condition for solving the equation system is:  $K \geq N + 1$ .

This shows that one could increase the number of the neurons in a layer, and divide the weights of the existing neuron in that layer. We have used this derivation and aim to find a solution of Eq. 7 where the order of magnitudes are significantly different (in the range of  $10^6$ ) for both the network parameters and for the eigenvalues of the mapping  $\mathbb{R}^N \mapsto \mathbb{R}^L$ .

### 2.3 Introducing Deceptive Neurons

The method described in the previous section results a fragile neural network, but unfortunately it is not enough to protect the network weights, since an attacker could identify the decomposed neurons based on their biases or could fit a new neural network on the functionality implemented by the layer. To prevent this we will introduce deceptive neurons in layers. The purpose of these neurons is to have non-zero activation in sum if and only if noise was added to their weights apart from this all these neurons have to cancel each others effect out in the network, but not necessarily in a single layer.

The simplest method is to define a neuron with an arbitrary weight and a bias of an existing neuron resulting a large activation and making a copy of it with the difference of multiplying the output weights by  $-1$ . As a result, these neurons do not contribute to the functionality of the network. However, adding noise to the weights of these neurons would have unexpected consequences depending on the characteristics of the noise, eventually leading to a decrease of classification accuracy.

One important aspect of this method is to hide the generated neurons and obfuscate the network to prevent the attacker to easily filter our deceptive neurons in the architecture. Choosing the same weights again on both layers would be an obvious sign to an attacker, therefore this method should be combined with decomposition described in Sect. 2.2.

Since decomposition allows the generation of arbitrarily small weights one can select a suitably small magnitude, which allows the generation of  $R$  real (non deceptive) neurons in the system, and half of their weights ( $\alpha$  parameters)

can be set arbitrarily, meanwhile the other half of the weights will be determined by Eq. 8. For each real neuron one can generate a number ( $F$ ) of fake neurons forming groups of  $R$  number of real and  $F$  number of fake neurons. These groups can be easily identified in the network since all of them will have the same bias, but the identification of fake and real neurons in a group is non-polynomial.

The efficiency of this method should be measured in the computational complexity of successfully finding two or more corresponding fake neurons having a total activation of zero in a group. Assuming that only one pair of fake neurons was added to the network, it requires  $\sum_{i=0}^L \binom{R_i+F_i}{2}$  steps to successfully identify the fake neurons, where  $R_i + F_i$  denotes the number of neurons in the corresponding hidden layer, and  $L$  is the number of hidden layers. This can be further increased by decomposing the fake neurons using Eq. 8: in that case the required number of steps is  $\sum_{i=0}^L \binom{R_i+F_i}{d+2}$ ,  $d$  being the number of extra decomposed neurons. This can be maximized if  $d + 2 = R_i + F_i/2$ , where  $i$  denotes the layer, where the fake neurons are located. However, this is true only if the attacker has information about the number of deceptive neurons. Without any prior knowledge, the attacker has to guess the number of deceptive neurons as well ( $0, 1, 2 \dots R_i + F_i - 1$ ) which leads to exponentially increasing computational time.

### 3 Experiments

#### 3.1 Simulation of a Simple Network

As a case study we have created a simple fully connected neural network with three layers, each containing two neurons to present the validity of our approach. The functionality of the network can be considered as a mapping  $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$ .

$$w_1 = \begin{bmatrix} 6 & -1 \\ -1 & 7 \end{bmatrix}, b_1 = [1 \ -5] \quad w_2 = \begin{bmatrix} 5 & 3 \\ 9 & -1 \end{bmatrix}, b_2 = [7 \ 1]$$

We added two neurons to the hidden layer with decomposition, which does not modify the input and output space and no deceptive neurons were used in this experiment. After applying the methods described in Sect. 2.1, we obtained a solution of:

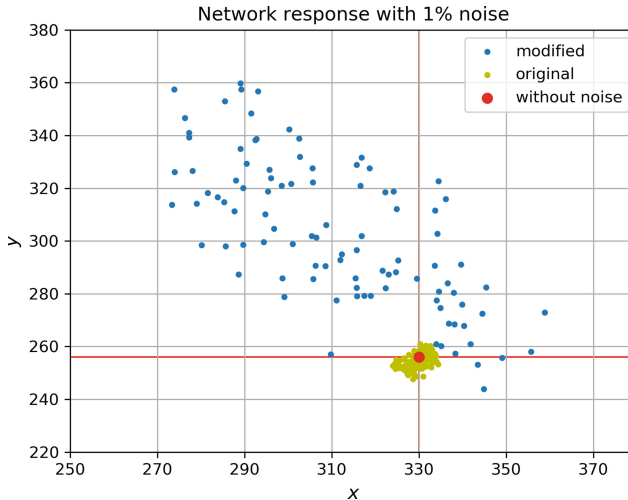
$$w_1 = \begin{bmatrix} 0.0525 & -0.4213 & 6.0058 & -0.5744 \\ -0.0087 & 2.9688 & -0.9991 & 4.0263 \end{bmatrix}$$

$$b_1 = [0.0087 \ -2.1066 \ 1.0009 \ -2.8722]$$

$$w_2 = \begin{bmatrix} 4.1924e + 03 & -5.4065e + 03 \\ -2.3914 & 7.3381 \\ -3.2266 & 5.7622 \\ 6.9634 & -7.0666 \end{bmatrix}$$

$$b_2 = [7 \ 1]$$





**Fig. 1.** This figure depicts the response of a simple two-layered fully connected network for a selected input (red dot) and the response of its variants with %1 noise (yellow dots) added proportionally to the weights. The blue dots represent the responses of the transformed MimosaNets under the same level of noise on their weights, meanwhile the response of the transformed network (without noise) remained exactly the same.

In the following experiment we have chosen an arbitrary input vector: [7, 9]. We have measured the response of the network for this input, each time introducing 1% noise to the weights of the network. Figure 1 shows the response of the original network and the modified network after adding 1% noise. The variances of the original network for the first output dimension is 6.083 and 8.399 for the second, meanwhile the variances are 476.221 and 767.877 for the decomposed networks respectively. This example demonstrates how decomposition of a layer can increase the networks dependence on its weights.

### 3.2 Simulations on the MNIST Dataset

We have created a five layered fully connected network containing 32 neurons in each hidden-layer (and 728 and 10 neurons in the input and output layers) and trained it on the MNIST [8] dataset, using batches of 32 and Adam Optimizer [9] for 7500 iterations. The network has reached an accuracy of 98.4% on the independent test set.

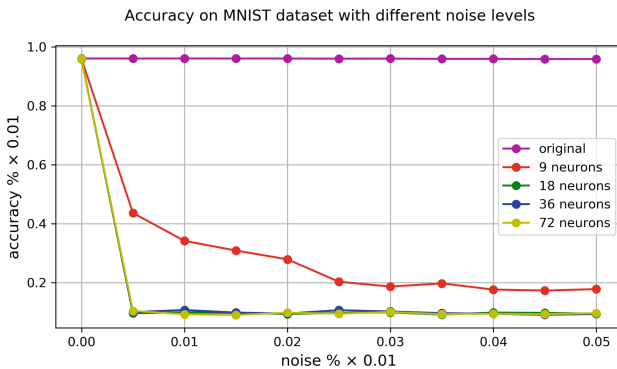
We have created different modifications of the network by adding 9, 18, 36, 72 extra neurons. These neurons were divided equally between the three hidden-layers and 2/3 of them were deceptive neurons (since they were always created in pairs) and 1/3 of them were created by decomposition. This means that in case of 36 additional neurons  $2 \times 4$  deceptive neurons were added to each layer and four new neurons per layer were created by decomposition.

In our hypothetical situations these networks (along with the original) could be stolen by a malevolent attacker, who would try to conceal his thievery by using the following three methods: adding additive noise proportionally to the network weights, continuing network training on arbitrary data and network knowledge distillation. All reported datapoints are an average of 25 independent measurements.

**Dependence on Additive Noise.** We have investigated network performance using additive noise to the network weights. The decrease of accuracy which depends on the ratio of the additive noise can be seen in Fig. 2.

At first we have tested a fully connected neural network trained on the MNIST dataset without making modifications to it. The decrease of accuracy was not more than 0.2% even with a relatively high 5% noise. This shows the robustness of a fully connected network.

After applying the methods described in Sect. 2 network accuracy retrogressed to 10% even in case of noise which was less than 1% of the network weights, as Fig. 2 depicts. This alone would reason the applicability of our method, but we have investigated low level noises further, which can be seen on Fig. 3. As it can be seen from the figure, accuracy starts to drop when the ratio of additive noise reaches the level of  $10^{-7}$ , which means the attacker can not significantly modify the weights. This effect could be increased by adding more and more neurons to the network.



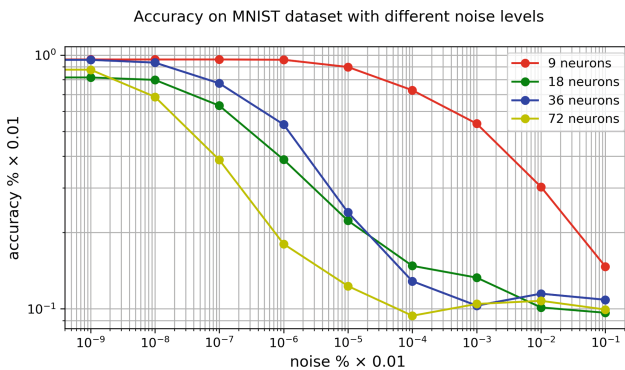
**Fig. 2.** This figure depicts accuracy changes on the MNIST dataset under various level of additive noise applied on the weights. The original network (purple) is not dependent on these weight changes, meanwhile accuracies retrogress in the transformed networks, even with the lowest level of noise.

**Dependence on Further Training Steps.** Additive noise randomly modifies the weights, but it is important to examine how accuracy changes in case of structured changes exploiting the gradients of the network. Figure 4 depicts accuracy changes and average in weights distances by applying further training

steps in the network. Further training was examined using different step sizes and optimizers (SGD, AdaGrad and ADAM) training the network with original MNIST and randomly selected labels and the results were qualitatively the same in all cases.

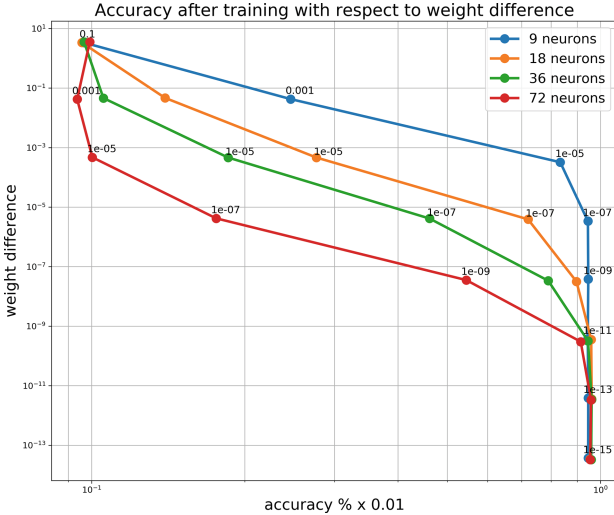
**Dependence on Network Distillation.** We have tried to distill the knowledge in the network and train a new neural network to approximate the functionality of previously selected layers, by applying the method described in [10].

We have generated one million random input samples with their outputs for the modified networks and have used this dataset to approximate the functionality of the network.



**Fig. 3.** A logarithmic plot depicting the same accuracy dependence as on Fig. 2, focusing on low noise levels. As it can be seen from the plot, accuracy values do not change significantly under  $10^{-7}$  percent of noise, which means the most important values of the weights would remain intact to proof connection between the original and modified networks.

We have created three-layered neural networks containing 32, 48, 64, 128 neurons in the hidden layer (The number of neurons in the first and last layer were determined by the original network) and tried to approximate the functionality of the hidden layers of the original structure. Since deceptive neurons have activations in the same order of magnitude as the original responses, these values disturb the manifold of the embedded representations learned by the network and it is more difficult to be approximated by a neural network. Table 1 contains the maximum accuracies which could be reached with knowledge distillation, depending on the number of deceptive neurons and the neurons in the architecture used for distillation. This demonstrates, that our method is also resilient towards knowledge distillation.



**Fig. 4.** The figure plots accuracy dependence of the networks in case of further training (applying further optimization steps). As it can be seen from the plot weights had to be kept in  $10^{-7}$  average distance to keep the same level of accuracy.

**Table 1.** The table displays the maximum accuracies reached with knowledge distillation. The different rows display the number of extra neurons which were added to the investigated layer, and the different columns show the number of neurons in the hidden layer of the fully connected architecture, which was used for distillation.

#Deceptive N.	#N. = 32	#N. = 48	#N. = 64	#N. = 128
9	0.64	0.65	0.69	0.71
18	0.12	0.14	0.15	0.17
36	0.10	0.11	0.10	0.13
72	0.11	0.09	0.10	0.10

## 4 Conclusion

In this paper, we have shown a transformation method which can significantly increase a network’s dependence on its weights, keeping the original functionality intact. We have also presented how deceptive neurons can be added to a network, without disturbing its original response. Using these transformations iteratively one can create and openly share a trained network, where it is computationally extensive to reverse engineer the original network architecture and embeddings in the hidden layers. The drawback of the method is the additional computational need for the extra neurons, but this is not significant, since computational increase is polynomial.

We have tested our method on simple toy problems and on the MNIST dataset using fully-connected neural networks and demonstrated that our approach results non-robust networks for the following perturbations: additive noise, application of further training steps and knowledge distillation.

**Acknowledgments.** This research has been partially supported by the Hungarian Government by the following grant: 2018-1.2.1-NKP-00008: Exploring the Mathematical Foundations of Artificial Intelligence also the funds of grant EFOP-3.6.2-16-2017-00013 are gratefully acknowledged.

## References

1. Koch, E., Zhao, J.: Towards robust and hidden image copyright labeling. In: IEEE Workshop on Nonlinear Signal and Image Processing, vol. 1174, pp. 185–206, Greece, Neos Marmaras (1995)
2. Wolfgang, R.B., Delp, E.J.: A watermark for digital images. In: Proceedings of the International Conference on Image Processing, vol. 3, pp. 219–222. IEEE (1996)
3. Zarrabi, H., Hajabdollahi, M., Soroushmehr, S., Karimi, N., Samavi, S., Najarian, K.: Reversible image watermarking for health informatics systems using distortion compensation in wavelet domain (2018) *arXiv preprint* arXiv:1802.07786
4. Xu, H., Su, Y., Zhao, Z., Zhou, Y., Lyu, M.R., King, I.: Deepobfuscation: securing the structure of convolutional neural networks via knowledge distillation (2018) *arXiv preprint* arXiv:1806.10313
5. Namba, R., Sakuma, J.: Robust watermarking of neural network with exponential weighting (2019) *arXiv preprint* arXiv:1901.06151
6. Gomez, L., Ibarrondo, A., Márquez, J., Duverger, P.: Intellectual property protection for distributed neural networks (2018)
7. Hitaj, D., Mancini, L.V.: Have you stolen my model? evasion attacks against deep neural network watermarking techniques (2018) *arXiv preprint* arXiv:1809.00615
8. LeCun, Y., Cortes, C., Burges, C.: MNIST handwritten digit database. AT&T Labs **2** (2010). <http://yann.lecun.com/exdb/mnist>
9. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014) *arXiv preprint* arXiv:1412.6980
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015) *arXiv preprint* arXiv:1503.02531



# Applications of Z-Numbers and Neural Networks in Engineering

Raheleh Jafari<sup>1</sup>(✉), Sina Razvarz<sup>2</sup>, and Alexander Gegov<sup>3</sup>

<sup>1</sup> School of design, University of Leeds, Leeds LS2 9JT, UK  
r.jafari@leeds.ac.uk

<sup>2</sup> Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute), Mexico City, Mexico  
srazvarz@yahoo.com

<sup>3</sup> School of Computing, University of Portsmouth, Buckingham Building, PO1 3HE  
Portsmouth, UK  
alexander.gegov@port.ac.uk

**Abstract.** In the real world, much of the information on which decisions are based is vague, imprecise and incomplete. Artificial intelligence techniques can deal with extensive uncertainties. Currently, various types of artificial intelligence technologies, like fuzzy logic and artificial neural network are broadly utilized in the engineering field. In this paper, the combined Z-number and neural network techniques are studied. Furthermore, the applications of Z-numbers and neural networks in engineering are introduced.

**Keywords:** Artificial intelligence · Fuzzy logic · Z-number · Neural network

## 1 Introduction

Intelligent systems are composed of fuzzy systems and neural networks. They have particular properties such as the capability of learning, modeling and resolving optimizing problems, suitable for specific kind of applications. The intelligent system can be named hybrid system in case that it combines a minimum of two intelligent systems. For example, the mixture of the fuzzy system and neural network causes the hybrid system to be called a neuron-fuzzy system.

Neural networks are made of interrelated groups of artificial neurons that have information which is obtainable by computations linked to them. Mostly, neural networks can adapt themselves to structural alterations while the training phase. Neural networks have been utilized in modeling complicated connections among inputs and outputs or acquiring patterns for the data [1–12].

Fuzzy logic systems are broadly utilized to model the systems characterizing vague and unreliable information [13–29]. During the years, investigators have proposed extensions to the theory of fuzzy logic. Remarkable extension includes Z-numbers [30]. The Z-number is defined as an ordered pair of fuzzy numbers

$(C, D)$ , such that  $C$  is a value of some variables and  $D$  is the reliability which is a value of probability rate of  $C$ . Z-numbers are widely applied in various implementations in different areas [31–36].

In this paper, the basic principles and explanations of Z-numbers and neural networks are given. The applications of Z-numbers and neural networks in engineering are introduced. Also, the combined Z-number and neural network techniques are studied. The rest of the paper is organized as follows. The theoretical background of Z-numbers and artificial neural networks are detailed in Sect. 2. Comparison analysis of neural networks and Z-number systems is presented in Sect. 3. The combined Z-number and neural network techniques are given in Sect. 4. The conclusion of this work is summarized in Sect. 5.

## 2 Theoretical Background

In this section, we provide a brief theoretical insight of Z-numbers and artificial neural networks.

### 2.1 Z-Numbers

**Mathematical Preliminaries.** Here some necessary definitions of Z-number theory are given.

**Definition 1.** If  $q$  is: 1) normal, there exists  $\omega_0 \in \mathfrak{R}$  where  $q(\omega_0) = 1$ , 2) convex,  $q(v\omega + (1 - v)\omega) \geq \min\{q(\omega), q(\tau)\}$ ,  $\forall \omega, \tau \in \mathfrak{R}, \forall v \in [0, 1]$ , 3) upper semi-continuous on  $\mathfrak{R}$ ,  $q(\omega) \leq q(\omega_0) + \epsilon$ ,  $\forall \omega \in N(\omega_0)$ ,  $\forall \omega_0 \in \mathfrak{R}, \forall \epsilon > 0$ ,  $N(\omega_0)$  is a neighborhood, 4)  $q^+ = \{\omega \in \mathfrak{R}, q(\omega) > 0\}$  is compact, so  $q$  is a fuzzy variable,  $q \in E : \mathfrak{R} \rightarrow [0, 1]$ .

The fuzzy variable  $q$  is defined as below

$$q = (\underline{q}, \bar{q}) \quad (1)$$

such that  $\underline{q}$  is the lower-bound variable and  $\bar{q}$  is the upper-bound variable.

**Definition 2.** The Z-number is composed of two elements  $Z = [q(\omega), p]$ .  $q(\omega)$  is considered as the restriction on the real-valued uncertain variable  $\omega$  and  $p$  is considered as a measure of the reliability of  $q$ . The Z-number is defined as  $Z^+$ -number, when  $q(\omega)$  is a fuzzy number and  $p$  is the probability distribution of  $\omega$ . If  $q(\omega)$ , and  $p$ , are fuzzy numbers, then the Z-number is defined as  $Z^-$ -number.

The  $Z^+$ -number has more information in comparison with the  $Z^-$ -number. In this work, we use the definition of  $Z^+$ -number, i.e.,  $Z = [q, p]$ ,  $q$  is a fuzzy number and  $p$  is a probability distribution.

The triangular membership function is defined as

$$\mu_q = G(a, b, c) = \begin{cases} \frac{\omega-a}{b-a} & a \leq \omega \leq b \\ \frac{c-\omega}{c-b} & b \leq \omega \leq c \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

and the trapezoidal membership function is defined as

$$\mu_q = G(a, b, c, d) = \begin{cases} \frac{\omega-a}{b-a} & a \leq \omega \leq b \\ \frac{d-\omega}{d-c} & c \leq \omega \leq d \\ 1 & b \leq \omega \leq c \end{cases} \text{ otherwise } \mu_q = 0 \quad (3)$$

The probability measure of  $q$  is defined as

$$P(q) = \int_{\Re} \mu_q(\omega)p(\omega)d\omega \quad (4)$$

such that  $p$  is the probability density of  $\omega$ . For discrete  $Z$ -numbers we have

$$P(q) = \sum_{j=1}^n \mu_q(\omega_j)p(\omega_j) \quad (5)$$

**Definition 3.** The  $\alpha$ -level of the  $Z$ -number  $Z = (q, p)$  is stated as below

$$[Z]^\alpha = ([q]^\alpha, [p]^\alpha) \quad (6)$$

such that  $0 < \alpha \leq 1$ .  $[p]^\alpha$  is calculated by the Nguyen's theorem

$$[p]^\alpha = p([q]^\alpha) = p([\underline{q}^\alpha, \bar{q}^\alpha]) = [\underline{P}^\alpha, \bar{P}^\alpha] \quad (7)$$

such that  $p([q]^\alpha) = \{p(\omega) | \omega \in [q]^\alpha\}$ . Hence,  $[Z]^\alpha$  is defined as

$$[Z]^\alpha = \left( [\underline{Z}^\alpha, \bar{Z}^\alpha] \right) = \left( ([\underline{q}^\alpha, \underline{P}^\alpha], [\bar{q}^\alpha, \bar{P}^\alpha]) \right) \quad (8)$$

such that  $\underline{P}^\alpha = \underline{q}^\alpha p(\underline{\omega}_j^\alpha)$ ,  $\bar{P}^\alpha = \bar{q}^\alpha p(\bar{\omega}_j^\alpha)$ ,  $[\omega_j]^\alpha = (\underline{\omega}_j^\alpha, \bar{\omega}_j^\alpha)$ .

Let  $Z_1 = (q_1, p_1)$  and  $Z_2 = (q_2, p_2)$ , we have

$$Z_{12} = Z_1 * Z_2 = (q_1 * q_2, p_1 * p_2) \quad (9)$$

where  $*$   $\in$   $\{\oplus, \ominus, \odot\}$ .  $\oplus$ ,  $\ominus$  and  $\odot$ , indicate sum, subtract and multiply, respectively.

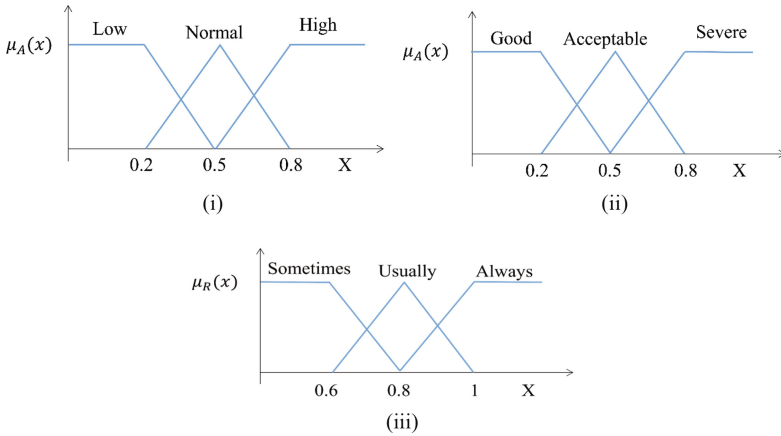
The operations utilized for the fuzzy numbers  $[q_1]^\alpha = [q_{11}^\alpha, q_{12}^\alpha]$  and  $[q_2]^\alpha = [q_{21}^\alpha, q_{22}^\alpha]$  are defined as [37],

$$\begin{aligned} [q_1 \oplus q_2]^\alpha &= [q_1]^\alpha + [q_2]^\alpha = [q_{11}^\alpha + q_{21}^\alpha, q_{12}^\alpha + q_{22}^\alpha] \\ [q_1 \ominus q_2]^\alpha &= [q_1]^\alpha - [q_2]^\alpha = [q_{11}^\alpha - q_{22}^\alpha, q_{12}^\alpha - q_{21}^\alpha] \\ [q_1 \odot q_2]^\alpha &= \left( \begin{array}{l} \min\{q_{11}^\alpha q_{21}^\alpha, q_{11}^\alpha q_{22}^\alpha, q_{12}^\alpha q_{21}^\alpha, q_{12}^\alpha q_{22}^\alpha\} \\ \max\{q_{11}^\alpha q_{21}^\alpha, q_{11}^\alpha q_{22}^\alpha, q_{12}^\alpha q_{21}^\alpha, q_{12}^\alpha q_{22}^\alpha\} \end{array} \right) \end{aligned} \quad (10)$$

For the discrete probability distributions, the following relation is defined for all  $p_1 * p_2$  operations

$$p_1 * p_2 = \sum_l p_1(\omega_{1,j})p_2(\omega_{2,(n-j)}) = p_{12}(\omega) \quad (11)$$





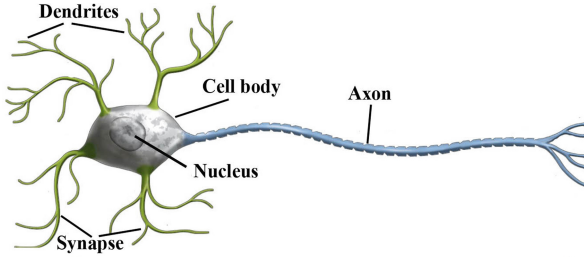
**Fig. 1.** Membership functions applied for (a) cereal yield, cereal production, economic growth, (b) threat rate, and (c) reliability

**Background and Related Work.** The implementations of Z-numbers based techniques are bounded because of the shortage of effective approaches for calculation with Z-numbers.

In [38], the capabilities of the Z-numbers in the improvement of the quality of risk assessment are studied. Prediction equal to (High, Very Sure) is institutionalized in the form of Z-evaluation “ $y$  is  $Z(c, p)$ ”, such that  $y$  is considered as a random variable of threat probability,  $c$  and  $p$  are taken to be fuzzy sets, demonstrating soft constraints on a threat probability and a partial reliability, respectively. The likelihood of risk is illustrated by Z-number as: Probability =  $Z_1(\text{High, Very Sure})$ , such that  $c$  is indicated through linguistic terms High, Medium, Low, also,  $p$  is indicated through terms Very Sure, Sure, etc. Likewise, consequence rate is explained as: Consequence measure =  $Z_2(\text{Low, Sure})$ . Threat rates ( $Z_{12}$ ) is computed as the product of the probability ( $Z_1$ ) and consequence measure ( $Z_2$ ).

In [39], Z-number-based fuzzy system is suggested to determine the food security risk level. The proposed system is relying on fuzzy If-Then rules, which applies the basic parameters such as cereal production, cereal yield, and economic growth to specify the threat rate of food security. The membership functions applied to explain input, as well as output variables, are demonstrated in Fig. 1.

In [40], the application of the Z-number theory to selection of optimal alloy is illustrated. Three alloys named Ti12Mo2Sn alloy, Ti12Mo4Sn alloy, and Ti12Mo6Sn alloy are examined and an optimal titanium alloy is selected using the proposed approach. The optimality of the alloys is studied based on three criteria: strength level, plastic deformation degree, and tensile strength.



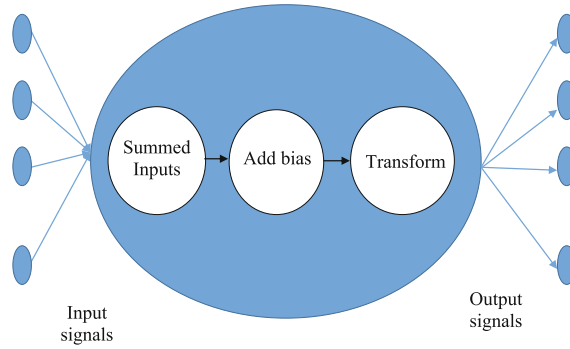
**Fig. 2.** The structure of a biological neuron

## 2.2 Neural Networks

Neural networks are constructed from neurons and synapses. They alter their rates in reply from nearby neurons as well as synapses. Neural networks operate similar to computer as they map inputs to outputs. Neurons, as well as synapses, are silicon members, which mimic their treatment. A neuron gathers the total incoming signals from other neurons, afterward simulate its reply represented by a number. Signals move among the synapses, which contain numerical rates. Neural networks learn once they vary the value of their synapsis. The structure of a biological neuron or nerve cell is shown in Fig. 2. The processing steps inside each neuron is demonstrated in Fig. 3.

**Background and Related Work.** In [41], artificial neural network technique is utilized for modeling the void fraction in two-phase flow inside helical vertical coils with water as work fluid. In [42] artificial neural network and multi-objective genetic algorithm are applied for optimizing the subcooled flow boiling in a vertical pipe. Pressure, the mass flux of the water, inlet subcooled temperature, as well as heat flux are considered as inlet parameters. The artificial neural network utilizes inlet parameters for predicting the objective functions, which are the maximum wall surface temperature as well as averaged vapor volume fraction at the outlet. The optimization procedure of design parameters is shown in Fig. 4.

In [43], artificial neural network technique is applied for predicting heat transfer in supercritical water. The artificial neural network is trained on the basis of 5280 data points gathered from experimental results. Mass flux, heat flux, pressure, tube diameter, as well as bulk specific enthalpy are taken to be the inputs of the proposed artificial neural network. The tube wall temperature is taken to be the output, see Fig. 5.



**Fig. 3.** Processing steps inside each neuron

### 3 Comparison Analysis of Neural Networks and Z-Number Systems

Neural networks and Z-number systems can be considered as a part of the soft computing field. The comparison of Neural networks and Z-number systems is represented in Table 1. Neural networks have the following advantageous:

**Table 1.** The comparison of neural networks and Z-number systems.

	Z-number systems	Neural networks
Knowledge presentation	Very good	Very bad
Uncertainty tolerance	Very good	Very good
Inaccuracy tolerance	Very good	Very good
Compatibility	Bad	Very good
Learning capability	Very bad	Very good
Interpretation capability	Very good	Very bad
Knowledge detection and data mining	Bad	Very good
Maintainability	Good	Very good

- i Adaptive Learning: capability in learning tasks on the basis of the data supplied to train or initial experience.
- ii Self-organization: neural networks are able to create their organization while time learning.
- iii Real-time execution: the calculations of neural networks may be executed in parallel, also specific hardware devices are constructed, which can capture the benefit of this feature.

Neural networks have the following drawbacks:

- i The utilization of neural networks is in direct connection with the availability of the training data.
- ii The acquired solution from the learning procedure may not be often explained.
- iii Almost all the neural network systems contain black boxes such that the ultimate state may not be explained.

Fuzzy logic has the following advantageous:

- i Simple to learn and apply.
- ii A user-friendly procedure to produce.
- iii Generation of more effective performance.

Fuzzy logic has the following drawbacks:

- i Constructing an uncertain system is complex.
- ii It is not easy to define proper membership values for uncertain systems.

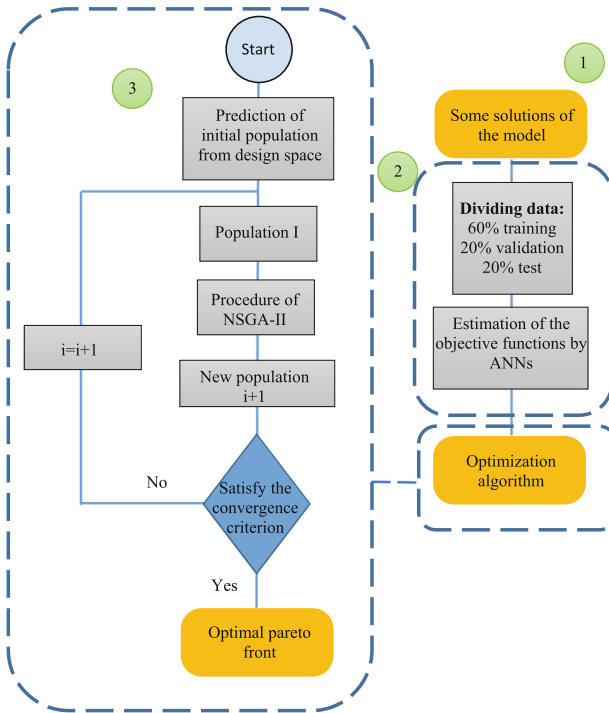


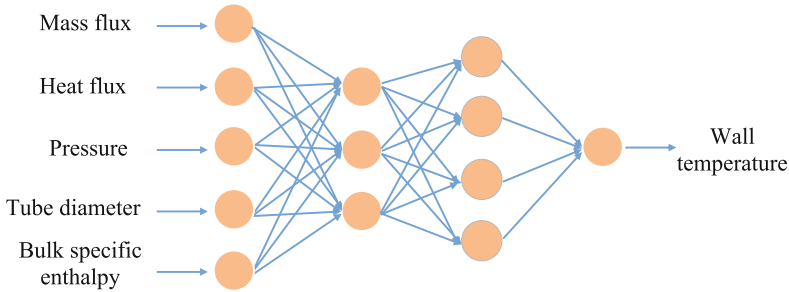
Fig. 4. The optimization procedure of input parameters

## 4 Combined Z-Number and Neural Network Techniques

### 4.1 Why Apply Z-Numbers in Neural Networks

Each neuron in the artificial neural network is linked with another neuron via a connection link in such a manner that the connecting link is related to a weight with the information regarding the input signal. Therefore, the weights contain beneficial information regarding input to resolve the problems. Some reasons for applying Z-numbers in neural networks are as follows:

- i In a case that crisp values cannot be implemented, uncertain values such as Z-numbers are utilized.
- ii Since the training, as well as learning, assist neural network to have a high performance in unanticipated status, therefore in such status, uncertain values like Z-numbers are more suitable than crisp values.
- iii In neural networks, Z-numbers are more applicable than fuzzy numbers. Z-numbers are more precise when compared with fuzzy numbers. Also, Z-numbers have less difficulty in computation in comparison with nonlinear system modeling approaches.



**Fig. 5.** Proposed artificial neural network for predicting heat transfer in supercritical water

### 4.2 Complexity in Applying Z-Numbers in Neural Networks

There exist some troubles when utilizing Z-numbers in neural networks. The complexity is associated with membership rules, the requirement to construct an uncertain system since it is often difficult to derive it by supplied set of complicated data.

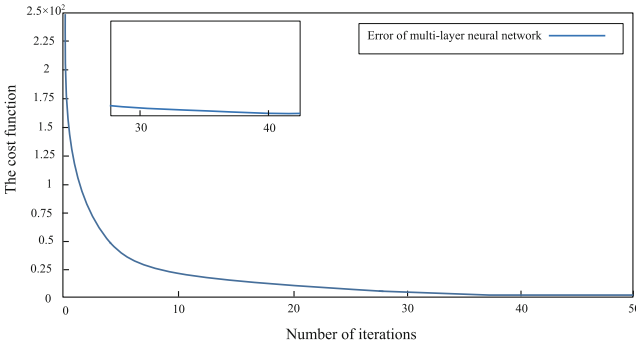
Neural networks can be used to train Z-numbers. The advantageous of using neural networks for training Z-numbers are as follows:

- i Novel patterns of data may be learned simply using neural networks therefore, it may be utilized for preprocessing data in uncertain systems.
- ii Neural networks due to their abilities in learning new relation with new input data may be utilized for refining fuzzy rules to generate the fuzzy adaptive system.

### 4.3 Examples of Combined Z-Number and Neural Network Techniques

*Example 1.* The following system is designed such that inputs and outputs are in the form of Z-numbers [44],

$$\begin{aligned} \zeta(t) &= \vartheta \cos(\varphi \Delta k t) \\ v(t+1) &= \frac{\Delta k^2 [\zeta(t) - \psi v^3(t)] - v(t-1) + \rho v(t)}{(1 + \omega \Delta k)} \end{aligned} \quad (12)$$



**Fig. 6.** Approximated error of multi-layer neural network

such that  $\rho = \omega \Delta k - \theta \Delta k^2 + 2$ .  $\Delta k, \omega, \theta, \psi, \vartheta$  are Z-number parameters.  $\varphi$  is taken to be a random variable uniformly distributed in the interval  $[0.1, 2.9]$  with mean  $E\{\varphi\} = 1.5$ , as well as the initial conditions being  $v(0) = v(1) = 1$ . The following are assumed,

$$\begin{aligned} \Delta k &= [(0.03, 0.05, 0.06), p(0.6, 0.8, 0.86)] \\ \omega &= [(0.1, 0.3, 0.5), p(0.6, 0.7, 0.87)] \\ \theta &= [(-4.2, -4, -3.8), p(0.6, 0.8, 86)] \\ \psi &= [(0.8, 1, 1.2), p(0.7, 0.8, 0.85)] \\ \vartheta &= [(0.2, 0.5, 0.7), p(0.7, 0.8, 0.85)] \end{aligned} \quad (13)$$

In order to model the uncertain nonlinear system (12), a multi-layer neural network is used such that obtains the Z-number coefficients of (12). The error plot is demonstrated in Fig. 6.

*Example 2.* A liquid tank system is demonstrated in Fig. 7, which is modeled as below

$$\frac{d}{dt}v(t) = -\frac{1}{SO}v(t) + \frac{d}{S} \quad (14)$$

where  $d = t + 1$  is inflow disturbances of the tank that generates vibration in liquid level  $v$ ,  $O = 1$  is the flow obstruction which can be curbed utilizing the

valve, also  $S = 1$  is the cross-section of the tank. Two types of neural networks, static Bernstein neural network (SBNN) and dynamic Bernstein neural network (DBNN) [45], are used to estimate the Z-number solutions of (14). The error plots of SBNN and DBNN are demonstrated in Fig. 8.

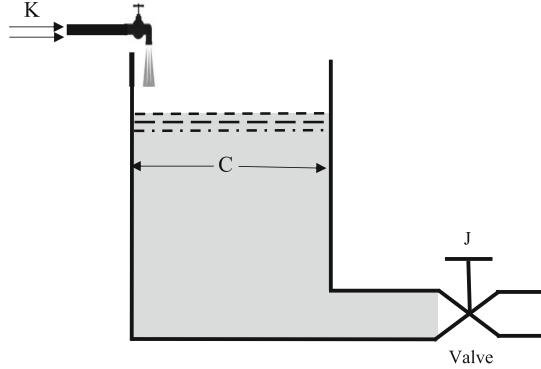


Fig. 7. Liquid tank system

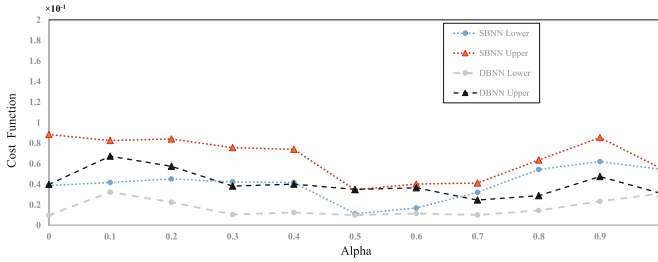
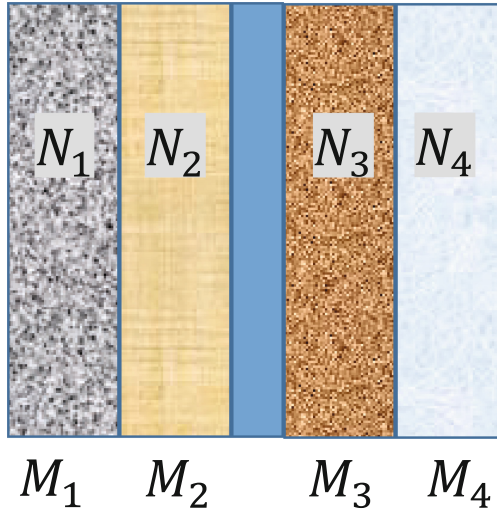


Fig. 8. Approximated errors of SBNN and DBNN

Example 3. The heat source by insulating materials is demonstrated in Fig. 9, which is modeled as below

$$\frac{M_1}{N_1} \oplus \frac{M_2}{N_2} = \frac{M_3}{N_3} \oplus \frac{M_4}{N_4} \oplus J \tag{15}$$

A heat source is placed in the center of insulating materials. The widths of the insulating materials are in the form of Z-numbers. The coefficients of conductivity materials are  $N_1 = h, N_2 = h\sqrt{h}, N_3 = h^2, N_4 = \sqrt{h}$ , such that  $h$  is elapsed time.  $J$  is thermal resistance. Neural network technique is used to approximate Z-number solutions of (15) [46].



**Fig. 9.** The heat source

## 5 Conclusion

The notion of Z-numbers is rather naturally obtained while gathering vague information in a linguistic appearance. In this paper, the combined Z-number and neural network techniques are studied. Furthermore, the applications of Z-numbers and neural networks in engineering are introduced. As some researchers have effectively used Z-numbers, in-depth discussions are given for stimulating future studies.

## References

1. Dote, Y., Hoft, R.G.: Intelligent Control Power Electronics Systems. Oxford University Press, Oxford (1998)
2. Mohanty, S.: Estimation of vapour liquid equilibria for the system carbon dioxide- difluoromethane using artificial neural networks. *Int. J. Refrig.* **29**, 243249 (2006)
3. Razvarz, S., Jafari, R., Yu, W., Khalili, A.: PSO and NN modeling for photocatalytic removal of pollution in wastewater. In: 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE) Electrical Engineering, pp. 1–6 (2017)
4. Jafari, R., Yu, W.: Artificial neural network approach for solving strongly degenerate parabolic and burgers-fisher equations. In: 12th International Conference on Electrical Engineering, Computing Science and Automatic Control (2015). <https://doi.org/10.1109/ICEEE.2015.7357914>
5. Jafari, R., Razvarz, S., Gegov, A.: A new computational method for solving fully fuzzy nonlinear systems. In: Computational Collective Intelligence, ICCCI 2018. Lecture Notes in Computer Science, vol. 11055, pp. 503–512. Springer, Cham (2018)



6. Razvarz, S., Jafari, R.: ICA and ANN modeling for photocatalytic removal of pollution in wastewater. *Math. Computat. Appl.* **22**, 38–48 (2017)
7. Razvarz, S., Jafari, R., Gegov, A., Yu, W., Paul, S.: Neural network approach to solving fully fuzzy nonlinear systems. In: *Fuzzy Modeling and Control Methods Application and Research*, pp. 45–68. Nova Science Publisher, Inc., New York (2018). ISBN: 978-1-53613-415-5
8. Razvarz, S., Jafari, R.: Intelligent techniques for photocatalytic removal of pollution in wastewater. *J. Electr. Eng.* **5**, 321–328 (2017). <https://doi.org/10.17265/2328-2223/2017.06.004>
9. Graupe, D.: Statistical training. In: Chen, W., Mlynski, D.A. (eds.) *Principles of Artificial Neural Networks*. Advanced Series in Circuits and Systems, vol. 7, pp. 311–327. World Scientific (2013)
10. Jafari, R., Yu, W., Li, X.: Solving fuzzy differential equation with Bernstein neural networks. In: *IEEE International Conference on Systems, Man, and Cybernetics*, Budapest, Hungary, pp. 1245–1250 (2016)
11. Jafari, R., Yu, W.: Uncertain nonlinear system control with fuzzy differential equations and Z-numbers. In: *18th IEEE International Conference on Industrial Technology*, Canada, pp. 890–895 (2017). <https://doi.org/10.1109/ICIT.2017.7915477>
12. Jafarian, A., Measoomy Nia, S., Jafari, R.: Solving fuzzy equations using neural nets with a new learning algorithm. *J. Adv. Comput. Res.* **3**, 33–45 (2012)
13. Werbos, P.J.: Neuro-control and elastic fuzzy logic: capabilities, concepts, and applications. *IEEE Trans. Ind. Electron.* **40**, 170180 (1993)
14. Jafari, R., Yu, W., Razvarz, S., Gegov, A.: Numerical methods for solving fuzzy equations: a survey. *Fuzzy Sets Syst.* (2019). <https://doi.org/10.1016/j.fss.2019.11.003>. ISSN 0165–0114
15. Kim, J.H., Kim, K.S., Sim, M.S., Han, K.H., Ko, B.S.: An application of fuzzy logic to control the refrigerant distribution for the multi type air conditioner. In: *Proceedings IEEE International Fuzzy Systems Conference*, vol. 3, pp. 1350–1354 (1999)
16. Wakami, N., Araki, S., Nomura, H.: Recent applications of fuzzy logic to home appliances. In: *Proceedings IEEE International Conference Industrial Electronics, Control, and Instrumentation*, Maui, HI, p. 155160 (1993)
17. Jafari, R., Razvarz, S.: Solution of fuzzy differential equations using fuzzy Sumudu transforms. In: *IEEE International Conference on Innovations in Intelligent Systems and Applications*, pp. 84–89 (2017)
18. Jafari, R., Razvarz, S., Gegov, A., Paul, S.: Fuzzy modeling for uncertain nonlinear systems using fuzzy equations and Z-numbers. In: *Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence*, 5–7 September 2018, Nottingham, UK. *Advances in Intelligent Systems and Computing*, vol. 840, pp. 66–107 (2018)
19. Jafari, R., Razvarz, S.: Solution of fuzzy differential equations using fuzzy Sumudu transforms. *Math. Comput. Appl.*, 1–15 (2018)
20. Jafari, R., Razvarz, S., Gegov, A.: Solving differential equations with z-numbers by utilizing fuzzy Sumudu transform. In: *Intelligent Systems and Applications, IntelliSys 2018. Advances in Intelligent Systems and Computing*, vol. 869, pp. 1125–1138. Springer, Cham (2019)
21. Yu, W., Jafari, R.: Fuzzy modeling and control with fuzzy equations and Z-number. In: *IEEE Press Series on Systems Science and Engineering*. Wiley-IEEE Press (2019). ISBN-13: 978-1119491552
22. Negoita, C.V., Ralescu, D.A.: *Applications of Fuzzy Sets to Systems Analysis*. Wiley, New York (1975)

23. Zadeh, L.A.: Probability measures of fuzzy events. *J. Math. Anal. Appl.* **23**, 421–427 (1968)
24. Zadeh, L.A.: *Calculus of Fuzzy Restrictions. Fuzzy sets and Their Applications to Cognitive and Decision Processes*, pp. 1–39. Academic Press, New York (1975)
25. Zadeh, L.A.: Fuzzy logic and the calculi of fuzzy rules and fuzzy graphs. *Multiple-Valued Logic* **1**, 1–38 (1996)
26. Razvarz, S., Jafari, R.: Experimental study of Al<sub>2</sub>O<sub>3</sub> nanofluids on the thermal efficiency of curved heat pipe at different tilt angle. *J. Nanomater.*, 1–7 (2018)
27. Razvarz, S., Vargas-Jarillo, C., Jafari, R.: Pipeline monitoring architecture based on observability and controllability analysis. In: *IEEE International Conference on Mechatronics (ICM)*, Ilmenau, Germany, vol. 1, pp. 420–423 (2019). <https://doi.org/10.1109/ICMECH.2019.872287>
28. Razvarz, S., Vargas-jarillo, C., Jafari, R., Gegov, A.: Flow control of fluid in pipelines using PID controller. *IEEE Access* **7**, 25673–25680 (2019). <https://doi.org/10.1109/ACCESS.2019.2897992>
29. Razvarz, S., Jafari, R.: Experimental study of Al<sub>2</sub>O<sub>3</sub> nanofluids on the thermal efficiency of curved heat pipe at different tilt angle. In: *2nd International Congress on Technology Engineering and Science, ICONTES*, Malaysia (2016)
30. Zadeh, L.A.: A note on Z-numbers. *Inform. Sci.* **181**, 29232932 (2011)
31. Jiang, W., Xie, C., Luo, Y., Tang, Y.: Ranking Z-numbers with an improved ranking method for generalized fuzzy numbers. *J. Intell. Fuzzy Syst.* **32**, 1931–1943 (2017)
32. Yaakob, A.M., Gegov, A.: Fuzzy rule based approach with Z-numbers for selection of alternatives using TOPSIS. In: *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1–8 (2015)
33. Zamri, N., Ahmad, F., Rose, A.N.M., Makhtar, M.: A fuzzy TOPSIS with Z-numbers approach for evaluation on accident at the construction site. In: *Proceedings of the International Conference on Soft Computing and Data Mining*, pp. 41–50 (2016)
34. Azadeh, A., Saberi, M., Atashbar, N.Z., Chang, E., Pazhoheshfar, P.: Z-AHP: a Z-number extension of fuzzy analytical hierarchy process. In: *Proceedings 7th IEEE International Conference on Digital Ecosystems and Technologies*, pp. 141–147 (2013). <https://doi.org/10.1109/DEST.2013.6611344>
35. Aliev, R.A., Alizadeh, A.V., Huseynov, O.H.: The arithmetic of discrete Z-numbers. *Inform. Sci.* **290**, 134–155 (2015). <https://doi.org/10.1016/j.ins.2014.08.024>
36. Aliev, R.A., Huseynov, O.H., Zeinalova, L.M.: The arithmetic of continuous Z-numbers. *Inform. Sci.* **373**, 441–460 (2016). <https://doi.org/10.1016/j.ins.2016.08.078>
37. De Barros, L.C., Bassanezi, R.C., Lodwick, W.A.: The extension principle of Zadeh and fuzzy numbers. In: *A First Course in Fuzzy Logic, Fuzzy Dynamical Systems, and Biomathematics. Studies in Fuzziness and Soft Computing*, vol. 347, pp. 23–41. Springer, Heidelberg (2017)
38. Nuriyev, A.M.: Application of Z-numbers based approach to project risks assessment. *Eur. J. Interdisc. Stud.* **5**, pp. 67–73 (2019). ISSN 2411-4138
39. Abiyev, R.H., Uyar, K., Ilhan, U., Imanov, E., Abiyeva, E.: Estimation of food security risk level using Z-number-based fuzzy system. *J. Food Q.*, 9 (2018). Article ID 2760907. <https://doi.org/10.1155/2018/2760907>
40. Babanlia, M.B., Huseynov, V.M.: Z-number-based alloy selection problem. In: *12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016*, 29–30 August 2016, Vienna, Austria, *Procedia Computer Science*, vol. 102, pp. 183–189 (2016)

41. Parrales, A., Colorado, D., Diaz-Gomez, J.A., Huicochea, A., Alvarez, A., Hernandez, J.A.: New void fraction equations for two-phase flow in helical heat exchangers using artificial neural networks. *Appl. Therm. Eng.* **130**, 149–160 (2018)
42. Alimoradi, H., Shams, M.: Optimization of subcooled flow boiling in a vertical pipe by using artificial neural network and multi objective genetic algorithm. *Appl. Therm. Eng.* **111**, 1039–1051 (2017)
43. Chang, W., Chu, X., Fareed, A.F.B.S., Pandey, S., Luo, J., Weigand, B., Laurien, E.: Heat transfer prediction of supercritical water with artificial neural networks. *Appl. Therm. Eng.* **131**, 815–824 (2018)
44. Jafari, R., Razvarz, S., Gegov, A.: Neural network approach to solving fuzzy non-linear equations using Z-numbers. *IEEE Trans. Fuzzy Syst.* (2019). <https://doi.org/10.1109/TFUZZ.2019.2940919>
45. Jafari, R., Yu, W., Li, X., Razvarz, S.: Numerical solution of fuzzy differential equations with Z-numbers using bernstein neural networks. *Int. J. Comput. Intell. Syst.* **10**, 1226–1237 (2017)
46. Jafari, R., Yu, W., Li, X.: Numerical solution of fuzzy equations with Z-numbers using neural networks. *Intell. Auto. Soft Comput.*, 1–7 (2017)



# 5G-FOG: Freezing of Gait Identification in Multi-class Softmax Neural Network Exploiting 5G Spectrum

Jan Sher Khan<sup>1</sup>, Ahsen Tahir<sup>2</sup>, Jawad Ahmad<sup>3(✉)</sup>, Syed Aziz Shah<sup>4</sup>, Qammer H. Abbasi<sup>5</sup>, Gordon Russell<sup>3</sup>, and William Buchanan<sup>3</sup>

<sup>1</sup> University of Gaziantep, Gaziantep, Turkey

<sup>2</sup> Glasgow Caledonian University, Glasgow, UK

<sup>3</sup> Edinburgh Napier University, Edinburgh, UK

J. Ahmad@napier.ac.uk

<sup>4</sup> Manchester Metropolitan University, Manchester, UK

<sup>5</sup> University of Glasgow, Glasgow, UK

**Abstract.** Freezing of gait (FOG) is one of the most incapacitating and disconcerting symptom in Parkinson's disease (PD). FOG is the result of neural control disorder and motor impairments, which severely impedes forward locomotion. This paper presents the exploitation of 5G spectrum operating at 4.8 GHz (a potential Chinese frequency band for Internet of Things) to detect the freezing episodes experienced by PD patients. The core idea is to utilize wireless devices such as network interface card (NIC), radio frequency (RF) signal generator and dipole antennas to extract the wireless channel characteristics containing the variances amplitude information that can be integrated into the 5G communication system. Five different human activities were performed including sitting on chair, slow-walk, fast-walk, voluntary stop and FOG episodes. A multi-class, multilayer full softmax neural network was trained on the obtained data for classification and performance evaluation of the proposed system. A high classification accuracy of 99.3% was achieved for the aforementioned activities, compared with the existing state-of-the-art detection systems.

**Keywords:** Parkinson's disease · FOG · Classification · Softmax neural network

## 1 Introduction

Parkinson's disease (PD) is a progressive neurodegenerative disease described by Parkinson in 1817 [1]. Over time, PD effectively progresses and worsen and hence called a progressive disease. A specific type of neuron known as dopamine neuron losses during PD that causes FOG. FOG is a serious gait disorder which interrupts walking with a transient and sudden nature. Due to sudden and serious debilitating nature, FOG disturbs the balance of PD patients and therefore

causes falls that may lead to mortality [2,3]. The pathophysiology of FOG is still under research and its treatment is still an open clinical challenge [4]. However, recently, authors in [5] reported the impact of levodopa-carbidopa intestinal gel (LCIG) FOG and concluded that a long term control over FOG is possible via LCIG if FOG is detected correctly. Furthermore, it is suggested in [5] that a number of experiments are required with correct identification of FOG in patients.

Authors in [6] reduced FOG and improved mobility via simultaneously targeting motor and cognitive regions through transcranial direct current stimulation. Though, authors [6] reported the improvement of mobility but correctly predicting the state of freezing was overlooked. Therefore, to decrease the fall rate and before providing a solution for FOG, a system must be developed to detect FOG with higher accuracy. FOG can be detected through numerous detection systems such as wearable devices and camera etc [7–11]. However, there are several limitations associated with camera-based and wearable based systems. For instance, the camera-based system works raise privacy concerns due to the constant recording of images or videos. In addition, they are computationally expensive as well since processing images or videos require dedicated hardware. On the other hand, wearable devices have to be worn by the subject's all the time due to which the patients might feel uncomfortable. Moreover, more often than not, the patients forget to wear the devices after changing clothes or taking a shower. Due to aforementioned issues, it is evident that other digital medium should be investigated. This paper presents a wireless channel information (WCI) based new detection method. A device free wireless sensing method is developed and the accuracy of the proposed scheme is tested using artificial neural network (ANN).

Over the past few years, ANN has been applied in a number of areas including speech recognition [12], image classification [13], and energy demand prediction. Rahim et al. [12] and Chu et al. [14] applied ANN to the speech recognition. Moreover, ANN-based algorithms have also been used in image classification and recognition [13,15,16]. Previously, Neural network based schemes are applied to chemical-related research, molecular biology, medicines, environmental sciences and ecosystems [17–20]. This paper exploits the application of multi-class, full softmax multilayer feedforward neural network (ML-FFNN) using WCI and 5G spectrum for FOG detection.

The core idea of the proposed work is to detect the FOG episode by classifying various human activities such as sitting/standing on chair, slow-walk, fast-walk, voluntary stop. The classification performed using variations in WCI data is received through wireless devices including RF signal generator, networks interface card (NIC) and dipole antenna [21–24].

## 2 Experimental Setup

The general experimental setup for FOG detection is shown in Fig. 1. The experiment was conducted in a room with dimensions (15 m × 15 m) in New Science Building, Xidian University, China. The experimental settings included an RF

generator (DSG3000 Series), two dipole antennas, TP-link (PCE-AC68) next generation dual-band wireless AC1900 PCIe adapter NIC, and HP desktop computer with Ubuntu 14.10 (64 bits) and 4 GB RAM. The RF signal generator connected with the dipole antenna operating at 4.8 GHz was set as an Access Point (AP) to generate RF signals at multiple frequencies. The network interface card wired with dipole antenna embedded in a desktop computer received the seamless WCI data. The transmitter and receiver were kept 10 m apart from each other.

A total number of 15 volunteers took part in the experimental campaign and were asked to perform the aforementioned five activities. Each human activity constantly disturbed the wireless medium and the unique WCI imprint induced was used for activity recognition.

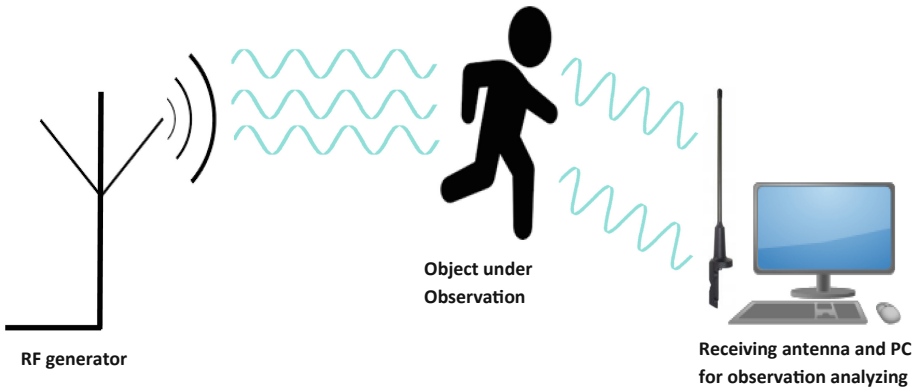


Fig. 1. General setup of the experiment.

### 3 FOG Detection Methodology

The design of the FOG system is presented in Fig. 2 which consists of three main parts:

- 1). Wireless channel information
- 2). Feature extraction
- 3). Multi-class softmax ML-FFNN training & classification for FOG detection

**Step 1:** Exploiting the IEEE 802.11n standards for orthogonal frequency division multiplexing (OFDM), which divide a single channel carrier into several subcarriers and enables the data to be transmitted in parallel to solve multipath fading problem [25]. The signal received using network interface card can be computed as:

$$Y = (H \times X) + N \quad (1)$$

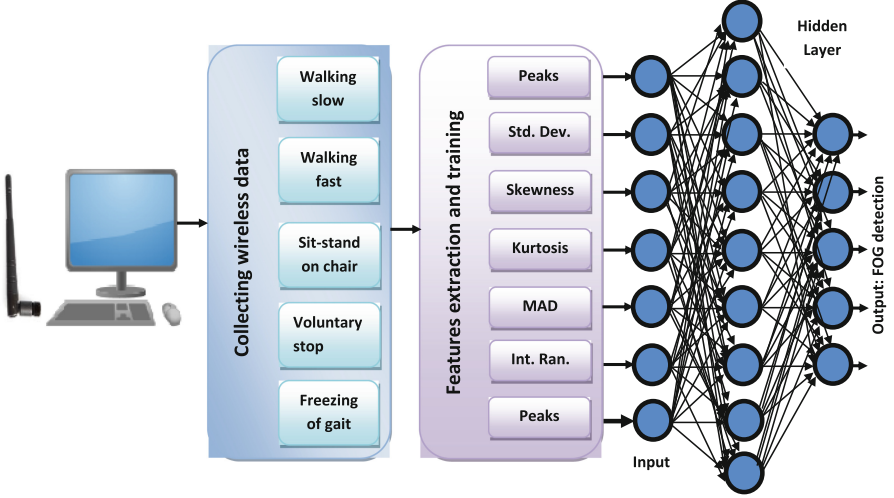


Fig. 2. Flowchart of the proposed FOG detection methodology.

Here  $\mathbf{X}$  and  $\mathbf{Y}$  are the transmitted and received signals, respectively.  $\mathbf{N}$  denotes the channel noise while  $\mathbf{H}$  demonstrates channel frequency response (CFR) of the wireless channel data which is a complex number.

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \dots, \mathbf{h}_n] \quad (2)$$

$$\mathbf{h}_n = \|h_n\| \exp^{j\angle h_n} \quad (3)$$

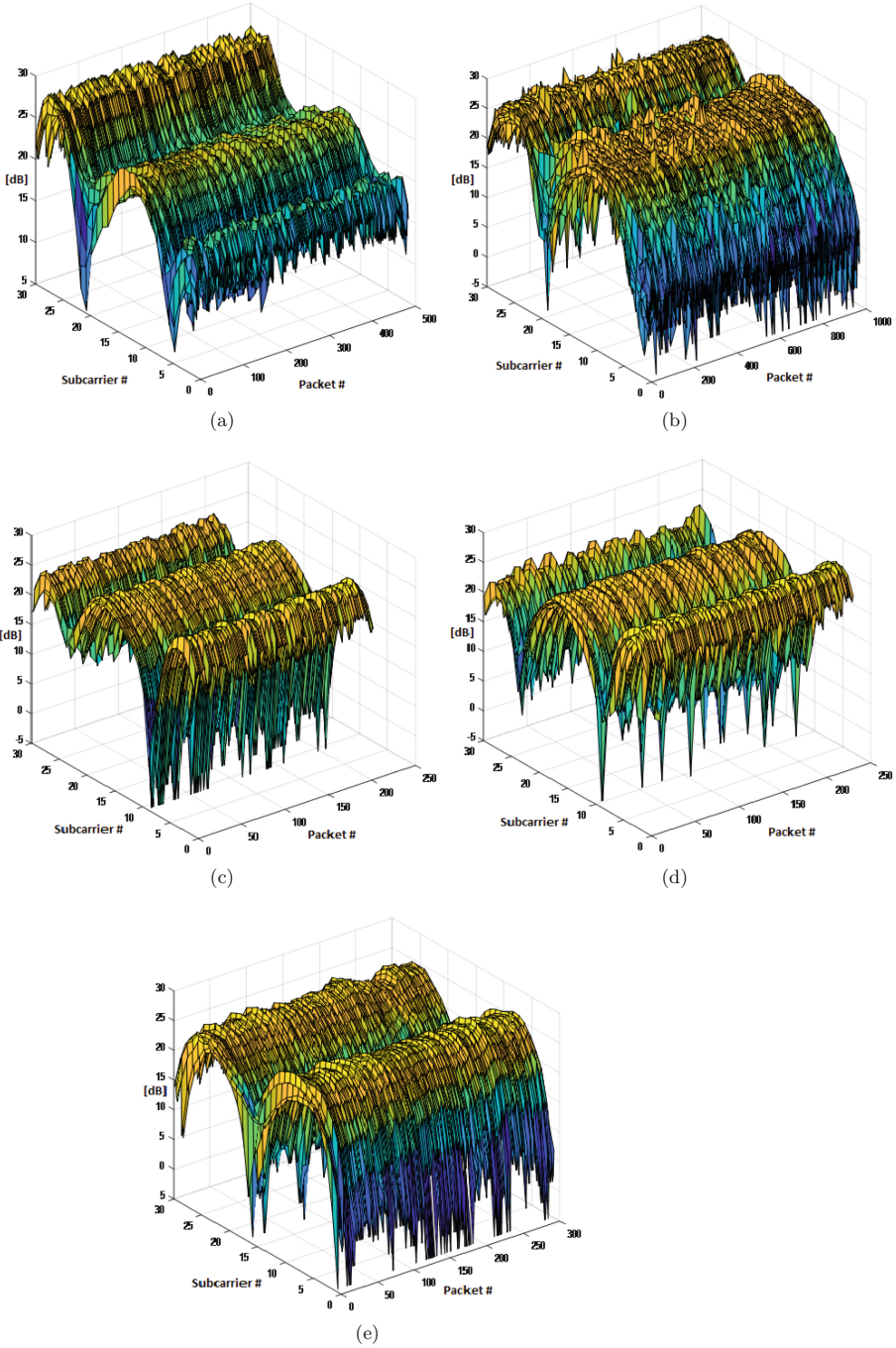
In Eq. 3,  $\|h_n\|$  represents the variance of amplitude information and  $\angle h_n$  describes phase information for  $n$  sub-carrier. It should be noted that the phase information obtained via NIC is extremely random and cannot be used for any application. Therefore, in this paper we have used the variances of amplitude information for training and testing the ANN algorithm to classify FOG from other daily life activities in an accurate and efficient way.

**Step 2:** In this step, time domain features such as mean, standard deviation, skewness, kurtosis, mean absolute deviation (MAD), interquartile range (IQR) and peaks are extracted from the WCI data and plugged into the levenberg-marquardt (LM) training algorithm. Features extraction is primarily data reduction by finding the most informative variables-based subset of the same dataset.

Mean is defined as the average of all data points in a data matrix and specify the variability around a distinct value in some data matrix. Mean can be more effective in case of relatively uniformly spread data with no extraordinarily high or low values. Mathematically, mean is defined as:

$$\mu_x = \frac{\sum_{i=1}^{a \times b} x_i}{a \times b}, \quad (4)$$





**Fig. 3.** Perturbations of amplitude information of 30 subcarriers. (a) Walking slow. (b) Walking fast. (c) Sit-stand on chair. (d) Voluntary stop. (e) FOG.



where  $a$  and  $b$  represent the number of rows and columns of a data matrix, respectively.  $x_i$  is a data point at index  $i$ . Standard deviation is known as the spread (variability) of data points in a data matrix. Mathematically standard deviation  $sd_X$  can be measured as [26]:

$$sd_x = \frac{1}{a \times b} \sum_{i=1}^{a \times b} (x_i - \mu_x)^2 \quad (5)$$

Information about the spread of data can also be obtained via interquartile range [26].  $q_u$  computes the middle value above the median, while  $q_l$  computes the middle value below the median of a data set. Mathematically interquartile range can be written as:

$$igr = q_u - q_l \quad (6)$$

Skewness  $s_x$  computes the asymmetry of the probability distribution while kurtosis  $k_x$  computes the shape of the probability distribution of a real-valued random variable. Skewness  $s_x$  and kurtosis  $k_x$  can be used to make judgments about image surfaces. Mathematically skewness and kurtosis can be computed as [27]:

$$s_x = \frac{1}{a \times b} \times \sum_{i=1}^{a \times b} \left( \frac{x_i - \mu_x}{sd_x} \right)^3 \quad (7)$$

$$k_x = \frac{1}{a \times b} \times \sum_{i=1}^{a \times b} \left( \frac{x_i - \mu_x}{sd_x} \right)^4 \quad (8)$$

The mean absolute deviation about mean measure the dispersion of X about its mean and can be mathematically written as [28]:

$$mad_x = \frac{\sum_{i=1}^{a \times b} |x_i - \mu_x|}{a \times b} \quad (9)$$

**Step 3:** Due to the faster operations, smaller training datasets requirement, easy implementation and ability to learn quickly, we have utilized a multi-layer perceptron neural network (MLPNN) with a single input layer, single hidden and single output layer as shown in Fig. 4. Levenberg-Marquardt (LM) [29] training algorithm is used during feature training process. LM is an iterative method that is used for solving non-linear minimization problem. The proposed classifier identify FOG episodes which is distinguishable from other routine activities using the proposed method. The input layer consists of seven neurons while the output layer consists of five neurons since we are classifying five different activities. Sigmoid activation function is used for input and output layers. Hidden layer which consist of ten neurons uses linear softmax activation function. Sigmoid function maps the interval  $(-\infty, \infty)$  onto  $(0, 1)$  while softmax function squashes an  $x$  size vector between 0 and 1. Furthermore, softmax function normalized the exponential function to make the sum of whole vector equal to 1. Therefore, the output softmax function interpret a set of specific features belong to certain

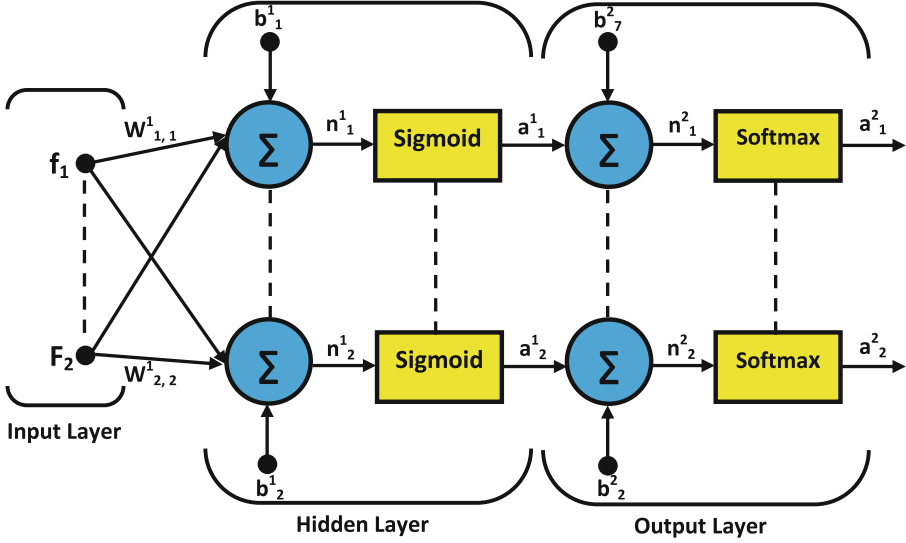


Fig. 4. MLPNN schematic diagram.

class. Mathematically sigmoid ( $\phi$ ) and softmax ( $\Phi$ ) functions can be computed as in [30] and in [31], respectively:

$$\phi(x) = \frac{1}{1 + \exp^{-x}} \quad (10)$$

$$\Phi(x_i) = \frac{\exp^{x_i}}{\sum_n^N \exp^{x_n}} \quad (11)$$

## 4 Result and Discussion

The variances of amplitude information for 30 subcarriers obtained using wireless devices exploiting 5G spectrum of five different activities is presented in Fig. 3, respectively. In Fig. 3, x-axis indicate total number of subcarriers, y-axis shows the total number of packets and z-axis is the relative power in dB indicating the variations in amplitude information. It can be observed that each human activity has resulted in a unique WCI signature which can be classified using multi-class ML-FFNN with the LM learning algorithm. Figure 5 shows the overall time history of five human activities for subcarrier number 6<sup>th</sup>. It illustrates the relative power level fluctuated between 4 dB to 16 dB for packet number 1 to 220. However, there is a shift in power level when the subject stands stationary (with small scale body movements such as breathing or small limb movements). Moreover, the power level varied around 24 dB when person was asked to walk slowly within area of interest. An increase in the variances of power level in

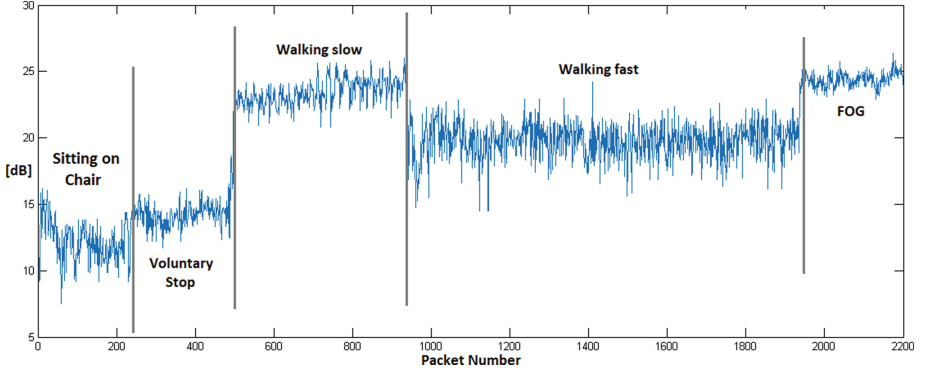


Fig. 5. Amplitude variation of a random subcarrier.

packet numbers 900 to 1800 is observed when the person walks at a fast pace. While, for FOG episodes, the variations power level fluctuations between 24 dB to 26 dB are observed.

Table 1 illustrates the performance of our system as compared to the state-of-the-art latest works [7–11, 32–36] in the domain of FOG detection leveraging traditional systems, such as wearable devices, smart phone sensors and vision based systems. The proposed system exploits 5G spectrum to detect and classify FOG with a high accuracy of 99.3% (see confusion matrix, Table 2) with an increase of approximately 6% over the second best method [33].

Table 1. Comparison of FOG detection systems

Authors	Detection system	Types of sensors	Algorithm	Accuracy
Prateek et al. [7]	Wearable devices	Inertial measurement unit	Generalized likelihood ratio test (GLRT)	81.03%
Camps et al. [8]	Wearable devices	Inertial measurement unit	Convolution neural network (CNN)	89%
Samà et al. [9]	Wearable devices	Accelerometer	Support vector machine	89.6%
Rodríguez et al. [32]	Wearable devices	Accelerometer	Support vector machine	76.8%
Aminis et al. [11]	Vision based	Camera, depth sensor	position/head offset & angle tracking	86.6%*
Bigy et al. [10]	Vision based	Camera, depth sensor	subject/body joint positions	92%
Kim et al. [33]	Smart phone	Accelerometer, gyroscope	Convolution neural network (CNN)	93.8%
Capecci et al. [34]	Smart phone	Accelerometer	Power spectrum and cadence measures	92.86%
Kim et al. [35]	Smart phone	Accelerometer, gyroscope	AdaBoost.M1	86%
Pepa et al. [36]	Smart phone	Accelerometer	Fuzzy inference system	89%
<b>Proposed</b>	<b>5G spectrum</b>	<b>Wireless sensing</b>	<b>Multi-class softmax FFNN</b>	<b>99.3%</b>

**Table 2.** Confusion matrix

Output Class 1	231	0	0	1	0	99.6%
	10.5%	0.0%	0.0%	0.0%	0.0%	0.4%
2	2	285	0	3	0	98.3%
	0.1%	13.0%	0.0%	0.1%	0.0%	1.7%
3	2	0	200	1	0	98.5%
	0.1%	0.0%	9.1%	0.0%	0.0%	1.5%
4	0	3	0	976	1	99.6%
	0.0%	0.1%	0.0%	44.4%	0.0%	0.4%
5	0	0	0	3	492	99.4%
	0.0%	0.0%	0.0%	0.1%	22.4%	0.6%
	98.3%	99.0%	100%	99.2%	99.8%	99.3%
	1.7%	1.0%	0.0%	0.8%	0.2%	0.7%
	1	2	3	4	5	
	Target Class					

## 5 Conclusion

This study presented the design and implementation of an FOG system leveraging wireless devices operating at 4.8 GHz (compatible with 5G spectrum for IoTs) in conjunction with multi-class softmax feedforward neural networks. The wireless channel information was extracted for five different human activities in indoor settings to classify the FOG episodes from sitting on chair, walking slowly, walking with fast pace and voluntary stop. The multi-class ML-FFNN leveraging features such as mean, standard deviation, skewness, kurtosis and peaks of power spectrum were used to classify the particular human activities. It was observed that the system provided an average accuracy of 99.3% for various subjects under test.

## References

1. Parkinson, J.: An essay on the shaking palsy (printed by whittingham and rowland for sherwood, neely, and jones) (1817)
2. Bloem, B.R., Hausdorff, J.M., Visser, J.E., Giladi, N.: Falls and freezing of gait in Parkinson's disease: a review of two interconnected, episodic phenomena. *Mov. Disord. Official J. Mov. Disord. Soc.* **19**(8), 871–884 (2004)
3. Canning, C.G., Paul, S.S., Nieuwboer, A.: Prevention of falls in Parkinson's disease: a review of fall risk factors and the role of physical interventions. *Neurodegener. Dis. Manage.* **4**(3), 203–221 (2014)
4. Amboni, M., Barone, P., Picillo, M., Cozzolino, A., Longo, K., Erro, R., Iavarone, A.: A two-year follow-up study of executive dysfunctions in Parkinsonian patients with freezing of gait at on-state. *Mov. Disord.* **25**(6), 800–802 (2010)
5. Zibetti, M., Angrisano, S., Dematteis, F., Artusi, C.A., Romagnolo, A., Merola, A., Lopiano, L.: Effects of intestinal levodopa infusion on freezing of gait in parkinson disease. *J. Neurol. Sci.* **385**, 105–108 (2018)

6. Dagan, M., Herman, T., Harrison, R., Zhou, J., Giladi, N., Ruffini, G., Manor, B., Hausdorff, J.M.: Multitarget transcranial direct current stimulation for freezing of gait in Parkinson's disease. *Mov. Disord.* **33**(4), 642–646 (2018)
7. Prateek, G.V., Skog, I., McNeely, M.E., Duncan, R.P., Earhart, G.M., Nehorai, A.: Modeling, detecting, and tracking freezing of gait in Parkinson disease using inertial sensors. *IEEE Trans. Biomed. Eng.* **65**(10), 2152–2161 (2018)
8. Camps, J., Samà, A., Martín, M., Rodríguez-Martín, D., Pérez-López, C., Moreno Arostegui, J.M., Cabestany, J., Català, A., Alcaine, S., Mestre, B., Prats, A., Crespo-Maraver, M.C., Counihan, T.J., Browne, P., Quinlan, L.R., Laighin, G., Sweeney, D., Lewy, H., Vainstein, G., Costa, A., Annicchiarico, R., Bayés, À., Rodríguez-Moliner, A.: Deep learning for freezing of gait detection in Parkinson's disease patients in their homes using a waist-worn inertial measurement unit. *Knowl.-Based Syst.* **139**, 119–131 (2018)
9. Samà, A., Rodríguez-Martín, D., Pérez-López, C., Català, A., Alcaine, S., Mestre, B., Prats, A., Crespo, M.C., Bayés, À.: Determining the optimal features in freezing of gait detection through a single waist accelerometer in home environments. *Pattern Recogn. Lett.* **105**, 135–143 (2018)
10. Bigy, A.A.M., Banitsas, K., Badii, A., Cosmas, J.: Recognition of postures and Freezing of Gait in Parkinson's disease patients using Microsoft Kinect sensor. In: 2015 7th International IEEE/EMBS Conference on Neural Engineering (NER), pp. 731–734. IEEE (2015)
11. Amini, A., Banitsas, K., Young, W.R.: Kinect4fog: monitoring and improving mobility in people with parkinson's using a novel system incorporating the Microsoft Kinect v2. *Disabil. Rehabil. Assist. Technol.* 1–8 (2018)
12. Rahim, M.G., Goodyear, C.C., Kleijn, W.B., Schroeter, J., Sondhi, M.M.: On the use of neural networks in articulatory speech synthesis. *J. Acoust. Soc. Am.* **93**(2), 1109–1121 (1993)
13. Kung, S.-Y., Taur, J.-S.: Decision-based neural networks with signal/image classification applications. *IEEE Trans. Neural Netw.* **6**(1), 170–181 (1995)
14. Chu, W., Bose, N.: Speech signal prediction using feedforward neural network. *Electron. Lett.* **34**(10), 999–1001 (1998)
15. DeKruger, D., Hunt, B.R.: Image processing and neural networks for recognition of cartographic area features. *Pattern Recogn.* **27**(4), 461–483 (1994)
16. Cosatto, E., Graf, H.P.: A neural network accelerator for image analysis. *IEEE Micro* **3**, 32–38 (1995)
17. Kvasnička, V.: An application of neural networks in chemistry. *Chem. Pap.* **44**(6), 775–792 (1990)
18. Lerner, B., Levinstein, M., Rosenberg, B., Guterman, H., Dinstein, L., Romem, Y.: Feature selection and chromosome classification using a multilayer perceptron neural network. In: 1994 IEEE International Conference on Neural Networks. IEEE World Congress on Computational Intelligence (1994)
19. Lek, S., Guégan, J.-F.: Artificial neural networks as a tool in ecological modelling, an introduction. *Ecol. Model.* **120**(2–3), 65–73 (1999)
20. Colasanti, R.: Discussions of the possible use of neural network algorithms in ecological modeling. *Binary Comput. Microbiol.* **3**(1), 13–15 (1991)
21. Yang, X., Shah, S.A., Ren, A., Zhao, N., Zhao, J., Hu, F., Zhang, Z., Zhao, W., Rehman, M.U., Alomainy, A.: Monitoring of patients suffering from REM sleep behavior disorder. *IEEE J. Electromagn. RF Microwaves Med. Biol.* **2**(2), 138–143 (2018)

22. Yang, X., Shah, S.A., Ren, A., Zhao, N., Fan, D., Hu, F., Ur-Rehman, M., von Deneen, K.M., Tian, J.: Wandering pattern sensing at s-band. *IEEE J. Biomed. Health Inform.* (2017)
23. Wang, X., Yang, C., Mao, S.: Tensorbeat: Tensor decomposition for monitoring multiperson breathing beats with commodity wifi. *ACM Trans. Intell. Syst. Technol. (TIST)* **9**(1), 8 (2017)
24. Zhang, D., Wang, H., Wu, D.: Toward centimeter-scale human activity sensing with wi-fi signals. *Computer* **50**(1), 48–57 (2017)
25. Cao, W., Li, X., Hu, W., Lei, J., Zhang, W.: OFDM reference signal reconstruction exploiting subcarrier-grouping-based multi-level lloyd-max algorithm in passive radar systems. *IET Radar Sonar Navig.* **11**(5), 873–879 (2016)
26. Kreyszig, E.: *Advanced Engineering Mathematics*. Wiley, Hoboken (2010)
27. Kumar, V., Gupta, P.: Importance of statistical measures in digital image processing. *Int. J. Emerg. Technol. Adv. Eng.* **2**(8), 56–62 (2012)
28. El Amir, E.A.H.: On uses of mean absolute deviation: decomposition, skewness and correlation coefficients. *Metron* **70**(2–3), 145–164 (2012)
29. Hagan, M.T., Demuth, H.B., Beale, M.H., De Jesús, O.: *Neural Network Design*, vol. 20. Pws Pub., Boston (1996)
30. Karlik, B., Olgac, A.V.: Performance analysis of various activation functions in generalized mlp architectures of neural networks. *Int. J. Artif. Intell. Expert Syst.* **1**(4), 111–122 (2011)
31. Tang, Y.: Deep learning using linear support vector machines. arXiv preprint [arXiv:1306.0239](https://arxiv.org/abs/1306.0239) (2013)
32. Rodríguez-Martín, D., Samà, A., Pérez-López, C., Català, A., Moreno Arostegui, J.M., Cabestany, J., Bayés, À., Alcaine, S., Mestre, B., Prats, A., Crespo, M.C., Counihan, T.J., Browne, P., Quinlan, L.R., ÓLaighin, G., Sweeney, D., Lewy, H., Azuri, J., Vainstein, G., Annicchiarico, R., Costa, A., Rodríguez-Moliner, A.: Home detection of freezing of gait using support vector machines through a single waist-worn triaxial accelerometer. *PLoS ONE* **12**(2), e0171764 (2017). <http://dx.plos.org/10.1371/journal.pone.0171764>
33. Kim, H.B., Lee, H.J., Lee, W.W., Kim, S.K., Jeon, H.S., Park, H.Y., Shin, C.W., Yi, W.J., Jeon, B. and Park, K.S.: Validation of freezing-of-gait monitoring using smartphone. *Telemed. e-Health* **24**(12) (2018). [tmj.2017.0215](https://www.liebertpub.com/doi/10.1089/tmj.2017.0215). <https://www.liebertpub.com/doi/10.1089/tmj.2017.0215>
34. Capecci, M., Pepa, L., Verdini, F., Ceravolo, M.G.: A smartphone-based architecture to detect and quantify freezing of gait in Parkinson's disease. *Gait Posture* **50**, 28–33 (2016). <http://dx.doi.org/10.1016/j.gaitpost.2016.08.018>
35. Kim, H., Lee, H.J., Lee, W., Kwon, S., Kim, S.K., Jeon, H.S., Park, H., Shin, C.W., Yi, W.J., Jeon, B.S., Park, K.S.: Unconstrained detection of freezing of Gait in Parkinson's disease patients using smartphone. In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, November 2015, pp. 3751–3754 (2015)
36. Pepa, L., Ciabattini, L., Verdini, F., Capecci, M., Ceravolo, M.G.: Smartphone based Fuzzy Logic freezing of gait detection in Parkinson's Disease. In: *MESA 2014 - 10th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, Conference Proceedings (2014)



# Adaptive Blending Units: Trainable Activation Functions for Deep Neural Networks

Leon René Sützelfeld<sup>1</sup>(✉), Flemming Brieger<sup>2</sup>, Holger Finger<sup>1</sup>, Sonja Füllhase<sup>1</sup>,  
and Gordon Pipa<sup>1</sup>

<sup>1</sup> Institute for Cognitive Science, Wachsbleiche 27, 49090 Osnabrück, Germany  
lsuetfel@uos.de

<sup>2</sup> Ulm University, Helmholtzstraße 16, 89081 Ulm, Germany  
flemming.brieger@uni-ulm.de

**Abstract.** The most widely used activation functions in current deep feed-forward neural networks are rectified linear units (ReLU), and many alternatives have been successfully applied, as well. However, none of the alternatives have managed to consistently outperform the rest and there is no unified theory connecting properties of the task and network with properties of activation functions for most efficient training. A possible solution is to have the network learn its preferred activation functions. In this work, we introduce Adaptive Blending Units (ABUs), a trainable linear combination of a set of activation functions. Since ABUs learn the shape, as well as the overall scaling of the activation function, we also analyze the effects of adaptive scaling in common activation functions. We experimentally demonstrate advantages of both adaptive scaling and ABUs over common activation functions across a set of systematically varied network specifications. We further show that adaptive scaling works by mitigating covariate shifts during training, and that the observed advantages in performance of ABUs likewise rely largely on the activation function's ability to adapt over the course of training.

**Keywords:** Adaptive Blending Units · Trainable · Activation functions · Deep learning · Convolutional networks

## 1 Introduction

Deep neural networks are structured around layers, each of which performs a linear transformation of its input before feeding the signal through a scalar non-linear activation function. Chaining larger numbers of non-linear functions then allows the networks to find and extract complex features in the input. Activation functions thus have a key function in deep neural networks: Without intermittent non-linearities, these networks could only perform linear operations on the input. But despite a large number of activation functions proven successful in the literature, it remains unclear, what properties of an activation function are most

desirable, given a particular task and network configuration. Ideally, the network would sort this issue out by itself, but most common activation functions are fixed during training, i.e., their shape and scaling are treated as hyperparameters. We suggest changing this practice by making an activation function’s shape and scaling a trainable parameter of the network. Our main contribution in this work is the Adaptive Blending Unit (ABU), a linear combination of a set of basic activation functions that allows the shape and scaling of the resulting activation function to be learned during training. In an effort to separate and understand the effects of the activation function’s shape and its scaling, we also examine the effect of adaptive scaling on common activation functions without adaptation of the shape, as well as normalizing the blending weights in ABUs, thus learning its shape without learning any scaling. Throughout this work, we apply one scaling weight, or one set of blending weights (i.e., one ABU) per layer of the network. This way, the network is free to learn the activation function and/or scaling that best suits the computations performed in any given layer, while the number of parameters in the network is kept low enough, as not to require regularization. The remainder of this work is structured as follows. In Sect. 2, we will review related work, before comparing and analyzing common activation functions, their adaptively scaled counterparts and ABUs on CIFAR image classification tasks in Sect. 3. In Sect. 4, we examine multiple ways of normalizing ABUs, to provide an account of adaptive shape without adaptive scaling. Finally, in Sect. 5, we examine pre-training of the scaling and blending weights, to examine the role of adaptiveness over the course of training. We conclude the paper by discussing limitations of the chosen approach, and providing an outlook on future work on this topic.

## 2 Related Work

The most prevalent activation function in modern neural networks is the *Rectified Linear Unit (ReLU)* [10, 21], a piecewise-linear function returning the identity for all positive inputs and zero otherwise. Its constant derivative of 1 on the positive part helps alleviating the vanishing gradient problem [7], making it the first activation function allowing for a large number of stacked layers to be trained efficiently. With this, ReLU was partly responsible for the breakthrough of deep neural networks around 2012, marked by AlexNet’s victory in the annual ILSVRC challenge [16]. *Leaky ReLU (LReLU)* [19], *Parametric ReLU (PReLU)* [11], and *Randomized Leaky ReLU (RRReLU)* [25] are all based on ReLU, but replace the zero-output for negative values by a linear function. In PReLU, the slope of the negative part of the function is controlled by a trainable parameter. Exponential Linear Units (ELU) [4] like ReLU, return the identity for positive values, but  $\alpha(\exp(x) - 1)$  for negative values, with  $\alpha$  typically set to 1. Scaled Exponential Linear Units (SELU) [14] are identical to ELU, except for an additional scaling parameter  $\lambda$  acting upon the function as a whole. The values for  $\alpha$  and  $\lambda$  in SELUs are analytically derived to ensure convergence of activations towards unit mean and variance across layers. In a more empirical approach,



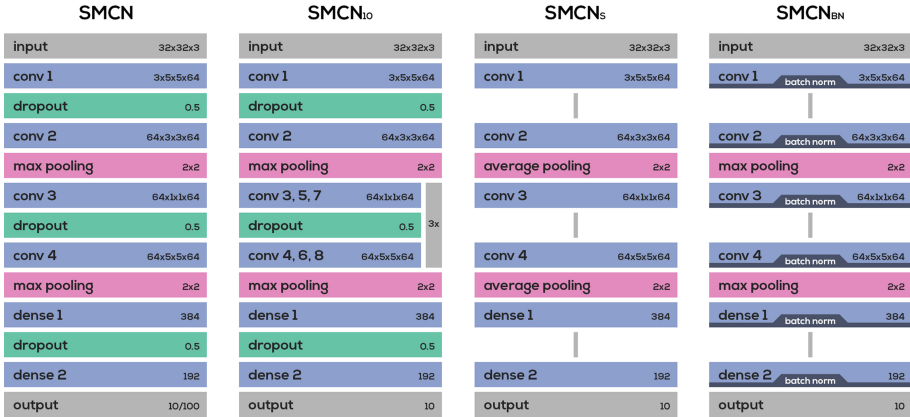
[22] performed a large reinforcement learning-based search for successful activation functions, and found multiple novel and well-performing functions. The most successful, given by  $f(x) = x \cdot \text{sigmoid}(\beta x)$  and named *Swish*, uses the trainable parameter  $\beta$  to control the overall shape of the function. E-Swish [2] ditches this parameter (setting it to 1), and instead scales the whole function by a manually determined parameter between 1 and 2. In addition to these, there are numerous approaches in which the activation function’s overall shape is learned, often using multiple parameters: *Adaptive Piecewise Linear Units (APL)* learn the slope of all piecewise linear elements and the position of the hinges independently for each neuron via backpropagation, while the number of linear pieces is a hyperparameter that is set manually [1]. Similarly, Maxout activations [9] learn a convex piecewise-linear function by returning the maximum of a fixed set of neurons, while the regular network weights determine the shape of the resulting function. [6] use Fourier series basis expansion to approximate non-linear parameterized basis functions, and train one activation function per feature map in a convolutional network. An approach suggested by [8], called the *soft exponential function*, can switch between a large number of different mathematical operations, such as addition, multiplication and exponentiation, by adjusting a trainable parameter. However, to our knowledge, no empirical validation of the approach was offered so far. In an approach similar to ours, [5] suggested blending a set of activation functions on a per-neuron basis, and constraining the blending weights to values between 0 and 1, by gating them with exponential sigmoid functions. Blending activation functions on a per-neuron basis, however, required downscaling of gradient updates as a form of regularization. Most similar to our approach, [20] suggested a learned blending of multiple common activation functions per layer, where the blending weights are constrained to sum up to 1, and showed this approach to be successful over a range of tasks and network configurations. We will provide further details with respect to the similarities between their approach and ours in the appropriate sections.

### 3 Adaptive Scaling and Adaptive Blending Units

In this section, we will introduce ABUs as an extension to the idea of an adaptive scaling of common activation functions, and analyze both ABUs and adaptive scaling with respect to task performance and the mechanics they introduce to the network. The activation functions we used as a baseline throughout this work are the *hyperbolic tangent (tanh)*, *ReLU*, *ELU*, *SELU*, the *identity* and *Swish*. We will reference the adaptively scaled versions of these by adding “ $\alpha$ ” to the function’s name, e.g., “ $\alpha$ ReLU”.

#### 3.1 Methods

Given a deep neural network of  $n$  layers, and an activation function  $f(x)$ , the adaptively scaled version of the activation function is given by  $\alpha_i \cdot f(x)$ , with



**Fig. 1.** SMCN network architectures for CIFAR10/CIFAR100. (**Vanilla**) **SMCN**: Default architecture, 6 hidden layers. **SMCN<sub>10</sub>**: Medium-sized network, 10 hidden layers. **SMCN<sub>5</sub>**: Additional architecture for robustness tests; 6 hidden layers, omitting dropout layers and replacing max pooling with average pooling. **SMCN<sub>BN</sub>**: Additional architecture for robustness tests; 6 hidden layers, batch normalization after each activation, and omitting dropout layers.

$i = 1, \dots, n$ . The *scaling weights*  $\alpha_i$  are initialized at 1 by default, and trained via backpropagation alongside all other network parameters. Swish’s  $\beta$  is initialized as a trainable parameter per layer (i.e.,  $\beta_i$ ) and likewise trained via backpropagation in all cases. ABUs can be viewed as an extension to this approach, in which the shape of the activation function is determined by a blending of multiple common activation functions within the unit. Given a deep neural network of  $n$  layers, and a set of  $m$  activation functions per layer, the ABU for the  $i$ th layer is defined as  $g_i(x) = \sum_j \alpha_{ij} \cdot f_j(x)$  with  $i = 1, \dots, n$  and  $j = 1, \dots, m$ . The *blending weights*  $\alpha_{ij}$  are initialized at  $\frac{1}{m}$  by default, and also trained via backpropagation alongside all other network parameters. With respect to the set of activation functions used in ABUs, we chose tanh, ELU, ReLU, the identity, and Swish in order to allow for high flexibility of the resulting function. However, we did not conduct an exhaustive search over possible sets of activation functions, so other sets may outperform the chosen configuration.

The *CIFAR 10* and *CIFAR 100* datasets [15] served as benchmarks to assess the performance of our approaches. Per-image z-transformation was applied as pre-processing to all images, and 5% of the training set was used as a validation set during training. To evaluate model performance, we applied post-hoc early stopping: The model was saved once every 8 epochs and the validation accuracy was estimated frequently over the course of training. All networks were trained for 60000 steps, after which we smoothed the validation accuracy curve and selected the model save point for which said curve indicated the highest performance. For each network and task specification, we report the mean of

30 runs, as well as the standard error. Training, validation and testing were all performed using mini-batches.

For the networks used in our tests, we created a set of small to mid-sized convolutional networks, called Simple Modular Convolutional Networks (SMCN). In different variations of these, features were added or subtracted to test the robustness of our approaches across different network design choices. The vanilla SMCN consists of four convolutional layers, followed by two dense layers. Max pooling ( $3 \times 3$ , stride 2) is performed after the second and fourth convolutional layer, and dropout [23] with a rate of 0.5 is applied after the first and third convolutional layer, and after the first dense layer. The convolutional layers use zero-padding and stride 1. They feature filters of size  $[5 \times 5 \times 64]$ ,  $[3 \times 3 \times 64]$ ,  $[1 \times 1 \times 64]$ , and  $[5 \times 5 \times 64]$ , and the dense layers consist of 384 and 192 neurons, respectively (see Fig. 1). The network contains no residual connections, and no batch normalization [12] is performed by default. Initial weights are randomly sampled using He initialization [11]. Bias units were initialized at 0.1, except for the first convolutional layer (0.0). The network is trained for 60000 steps on mini-batches of size 256, using the Adam optimizer [13] with a learning rate of  $\eta = 0.001$ . The total number of trainable parameters in the vanilla SMCN is roughly 1.8M. In addition to this, we used the following variations in our tests:  $SMCN_{10}$ , a mid-sized network (10 layers, roughly 2.0M parameters) identical to SMCN, with the exception that all layers between the two max pooling operations are repeated three times.  $SMCN_S$ , a simplified architecture where max pooling was replaced with average pooling, and all dropout layers were removed from the network (thus, the activation functions constitute the only non-linearities in this network).  $SMCN_{BN}$ , in which batch normalization is performed before applying the activation functions. We decided not to use dropout in this architecture, as batch normalization in conjunction with dropout can be problematic [18]. Note that since batch normalization negates the effect of any preceding scaling, adaptive scaling should not make a difference here. Finally, we also tested the vanilla SMCN with a Stochastic Gradient Descent optimizer with Momentum, instead of the Adam optimizer. Here, the networks were again trained for 60000 steps, with the momentum parameter set to 0.9, an initial learning rate of  $\eta = 0.01$ , and a learning schedule linearly decreasing the learning rate per weight update, reaching 0.0004 at the end of training.

### 3.2 Performance

For our performance tests, we chose a vanilla SMCN with Adam optimizer and CIFAR10 as the default setup to compare the various activation functions. All other tested setups are systematically varied versions of this, and differ in only one aspect each, i.e., network architecture, optimizer, or task. On average, adding adaptive scaling yielded improved performance for all activation functions, as evidenced by higher mean ranks of all adaptively scaled activation functions, compared to their fixed counterparts (see Table 1). However, as expected beforehand, batch-normalized networks ( $SMCN_{BN}$ ) were found to be indifferent to adaptive scaling. Interestingly, also in networks trained with the Momentum

**Table 1.** Performance comparison (percentage correct): Common activation functions, adaptive scaling, and ABUs by task, optimizer and network architecture. Table shows mean values of 30 runs plus standard errors per configuration, as well as mean rank across all six configurations. Highest performing activation function per column in bold.

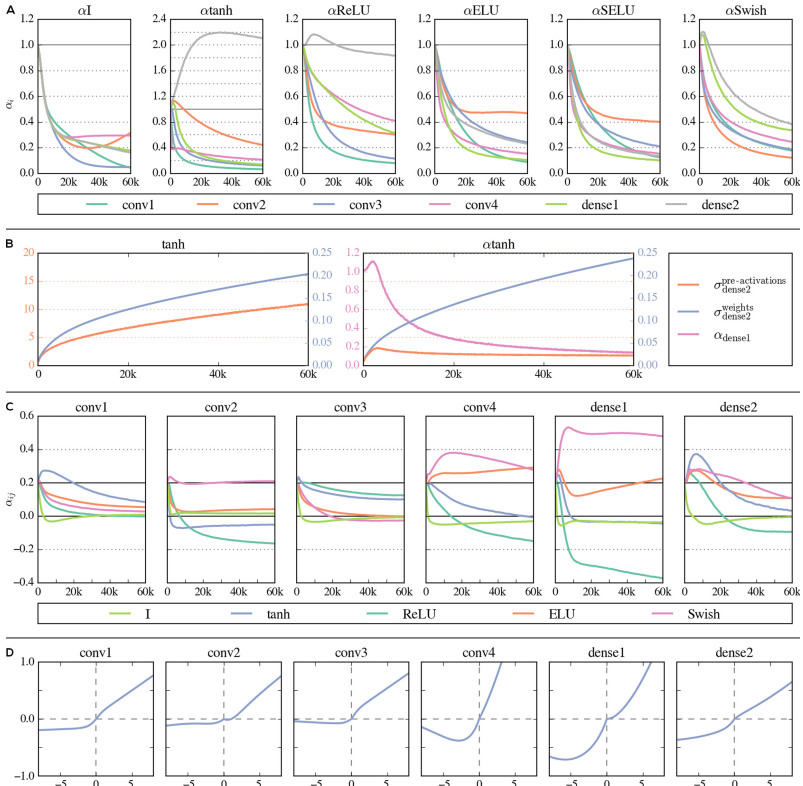
Network Optimizer	SMCN Adam	SMCN <sub>10</sub> Adam	SMCN <sub>S</sub> Adam	SMCN <sub>BN</sub> Adam	SMCN Momentum	SMCN Adam	Mean Rank
Task	CIFAR10	CIFAR10	CIFAR10	CIFAR10	CIFAR10	CIFAR100	
I	75.51 ± 0.11	73.19 ± 0.37	38.87 ± 0.08	71.72 ± 0.09	77.34 ± 0.06	44.11 ± 0.09	12.00
$\alpha$ I	76.52 ± 0.08	77.34 ± 0.18	39.48 ± 0.06	71.34 ± 0.10	76.32 ± 0.06	45.58 ± 0.09	11.50
tanh	75.44 ± 0.06	58.55 ± 4.47	67.19 ± 0.11	75.10 ± 0.07	78.76 ± 0.05	41.02 ± 0.13	12.00
$\alpha$ tanh	79.07 ± 0.07	73.40 ± 3.87	68.82 ± 0.10	75.32 ± 0.10	79.14 ± 0.05	46.85 ± 0.08	9.83
ReLU	79.42 ± 0.17	81.07 ± 0.15	72.79 ± 0.16	81.17 ± 0.06	81.63 ± 0.07	43.66 ± 0.10	8.17
$\alpha$ ReLU	79.23 ± 0.15	82.97 ± 0.12	73.89 ± 0.10	81.12 ± 0.11	81.85 ± 0.07	46.22 ± 0.11	7.17
ELU	81.78 ± 0.06	83.41 ± 0.08	73.33 ± 0.13	80.87 ± 0.06	82.16 ± 0.06	48.59 ± 0.11	5.83
$\alpha$ ELU	82.60 ± 0.06	84.94 ± 0.06	75.03 ± 0.13	80.89 ± 0.06	82.06 ± 0.04	51.03 ± 0.10	3.50
SELU	81.75 ± 0.07	83.29 ± 0.07	71.72 ± 0.14	79.36 ± 0.05	82.48 ± 0.05	48.25 ± 0.08	6.83
$\alpha$ SELU	82.81 ± 0.06	<b>85.04 ± 0.04</b>	73.79 ± 0.15	79.57 ± 0.07	81.99 ± 0.05	51.08 ± 0.08	4.33
Swish	82.07 ± 0.08	83.73 ± 0.07	74.33 ± 0.16	<b>81.77 ± 0.04</b>	82.02 ± 0.06	49.14 ± 0.12	4.33
$\alpha$ Swish	82.27 ± 0.06	84.56 ± 0.05	75.67 ± 0.09	81.61 ± 0.05	82.35 ± 0.05	50.19 ± 0.08	3.17
ABU (ours)	<b>83.12 ± 0.06</b>	84.70 ± 0.06	<b>76.19 ± 0.11</b>	80.63 ± 0.09	<b>83.12 ± 0.06</b>	<b>52.13 ± 0.08</b>	<b>2.33</b>

optimizer, adaptive scaling yielded little to no improvement over the fixed activation functions. Adaptive Blending Units, on the other hand, outperformed all other activation functions in four out of six setups (including the Momentum setup), showing remarkable robustness across architectural choices, and consequently scoring the highest mean rank of all tested activation functions. Since the ability to perform adaptive scaling is an integral part of Adaptive Blending Units, any improvements over adaptively scaled activation functions can likely be attributed to their adaptive shape.

### 3.3 Analysis

But what exactly changes in the networks, when we introduce adaptive scaling or ABUs? In order to provide some insight into the mechanisms introduced by the two approaches, we carried out further analyses based on the default setup, i.e., a vanilla SMCN with Adam optimizer trained on CIFAR10.

Let us first examine how scaling weights behave during training. In our tests, the scaling weights  $\alpha_i$  almost unanimously decreased to values far below 1 (see Fig. 2A). This behavior was highly consistent over repeated runs with random initializations and mini-batch sampling: The mean standard deviation (over repeated runs) of the final scaling weights reached at the end of training is  $\text{mean}(\sigma_{\alpha_i}) = 0.023$ . With respect to how this influences the activations in the network, it is sensible to consider the succeeding layer’s pre-activation statistics, i.e., the distribution of values going into its activation function: The distribution of pre-activation is approximately Gaussian for large layers due to the Central Limit Theorem, and is thus easier to compare between networks with different



**Fig. 2.** **A:**  $\alpha_i$  of adaptively scaled activation functions over the course of training in a vanilla SMCN (mean of 30 runs, 60000 steps). **B:** Effect of adaptive scaling on pre-activation distributions, exemplified by  $\tanh$  &  $\alpha\tanh$ . Scaling weights  $\alpha_i$  (magenta) enforce stable variance of next layer’s pre-activations (orange), compensating for the increased variance of the regular weight matrix  $W_{\text{dense2}}$  (blue). Plots show mean of five runs. **C:** ABU blending weights  $\alpha_{ij}$  over the course of training. **D:** Average activation functions (ABU) by layer at the end of training (axes scaled to improve readability).

activation functions. For many activation functions, the pre-activations are also crucial with respect to the magnitude of the gradients, in that they determine the fraction of inputs reaching saturated regions of the activation function. Our analyses show that the decreasing scaling weights rather precisely counteract an increase in the variance of the following weight matrix over the course of training. This stabilizes the distributions of pre-activation states in the following layers in both mean and variance, thus drastically reducing any covariate shift. We illustrate this by comparing the pre-activation variance of the last layer in SMCN networks, using  $\tanh$  and  $\alpha\tanh$ , in Fig. 2B. Without adaptive scaling, the variance of pre-activations increased throughout training for all layers and all tested activation functions. With adaptive scaling, the standard deviations typically converged to a value between 0.5 and 5 early on in training, and remained stable

at this value for the remainder of the training procedure. At the same time, pre-activation means were kept stable at less than a standard deviation from zero.

We take from this that adaptive scaling acts as a normalization technique, similar to batch normalization [12] or layer normalization [3]. In contrast to these, however, adaptive scaling doesn’t require any explicit calculations of variance or other statistics, or keeping track of running averages in inference, and does not depend on batch or layer size. It also allows the network to optimize the statistics of the neurons’ input distributions. That being said, our analysis does not allow us to infer whether or not the realized distributions actually constitute an optimum for the required computation in a given layer. If an activation function is a homogeneous function of degree 1 (scale-invariant; e.g., ReLU), the network performance would likely not be influenced by the variance of the pre-activation’s distribution, but may still be affected by shifts of the mean, which are also mitigated by adaptive scaling. We consider an in-depth analysis of such self-organizing processes, as well as further exploration of this principle for deep networks highly desirable, but out of scope for this work.

Turning to ABUs, we observe the same normalizing effect on the pre-activation statistics of succeeding layers. As illustrated in Fig. 2C, ABUs realize a layer’s overall downscaling in multiple ways. In the first convolutional layer, for instance, the weights unanimously decrease and mostly converge towards values close to zero. At the end of training, the identity and ReLU have arrived at effectively zero, while the final activation function is mostly a mixture of ELU and tanh. By contrast, the first dense layer achieves the overall downscaling of positive inputs by subtracting ReLU from a mixture of ELU and Swish. In both cases, the resulting function is rather flat, pushing the activations (layer output) closer to zero. These different compositions of blending weights translate into substantially different shapes of the resulting ABU (see Fig. 2D). But while the variation of the ABUs’ shape across layers is substantial, their shape within each layer is remarkably consistent over repeated runs, as indicated by a mean standard deviation of  $\sigma_{\text{mean}}^{\alpha_{ij}} = 0.010$  per layer and blending weight. This consistency, in conjunction with the good performance figures achieved by ABUs, lead to the conclusion that the learned shapes are meaningful with respect to the computations performed in the network.

In summary, while adaptive scaling stabilizes the pre-activation statistics of succeeding layers, the learned shapes of the resulting functions are meaningful, as well. Moreover, both adaptive scaling and an adaptive shape were found to yield improvements in performance for image classification tasks with convolutional networks.

## 4 Normalized Blending Weights

So far, we focused on adaptive scaling as an integral part of ABUs. In order to better understand the effects of shape in ABUs, we conducted an additional experiment, in which we normalized the blending weights of the ABUs in four

**Table 2.** Performance comparison (percentage correct): ABU and the normalized  $ABU_{abs}$ ,  $ABU_{nrm}$ ,  $ABU_{pos}$ , and  $ABU_{soft}$  by task, optimizer and network architecture. Table shows mean values of 30 runs plus standard errors per configuration, as well as mean rank across all six configurations. Highest performing activation function per column in bold.

Network	SMCN	SMCN <sub>10</sub>	SMCN <sub>S</sub>	SMCN <sub>BN</sub>	SMCN	SMCN	Mean Rank
Optimizer	Adam	Adam	Adam	Adam	Momentum	Adam	
Task	CIFAR10	CIFAR10	CIFAR10	CIFAR10	CIFAR10	CIFAR100	
ABU	83.12 ± 0.06	84.70 ± 0.06	<b>76.19 ± 0.11</b>	80.63 ± 0.09	83.12 ± 0.06	<b>52.17 ± 0.08</b>	<b>2.5</b>
$ABU_{nrm}$	82.82 ± 0.07	84.18 ± 0.07	75.99 ± 0.10	81.39 ± 0.07	<b>83.29 ± 0.06</b>	51.88 ± 0.10	3.7
$ABU_{abs}$	<b>83.14 ± 0.08</b>	<b>84.95 ± 0.06</b>	76.07 ± 0.16	81.17 ± 0.08	82.32 ± 0.05	52.16 ± 0.07	2.7
$ABU_{pos}$	82.90 ± 0.05	84.05 ± 0.06	76.10 ± 0.11	81.44 ± 0.07	83.26 ± 0.06	51.90 ± 0.09	3.2
$ABU_{soft}$	82.54 ± 0.07	84.63 ± 0.05	76.18 ± 0.07	<b>81.51 ± 0.06</b>	82.09 ± 0.05	52.10 ± 0.08	3.0

different ways, taking away their ability to scale the layer output by overall increases or decreases of the blending weights.

## 4.1 Methods

The following four methods of normalization for ABUs were used:  $ABU_{nrm}$  denotes the case where a layer’s blending weights are normalized to sum up to 1 ( $\sum_j \alpha_{ij} = 1$ ). The normalization was implemented as part of the graph, dividing the blending weights by their sum, before applying them in the respective ABU. Similarly, in  $ABU_{abs}$ , we divided the raw blending weights by the sum of their absolute values, thus keeping the sum of the absolute values of the blending weights at 1 ( $\sum_j |\alpha_{ij}| = 1$ ). Note that under this constraint, scaling is still possible, albeit not independent of the resulting shape: By having similar activation functions cancel each other out with blending weights on either side of zero, functions can be constructed that return only a fraction of the input, or even zero, for all positive values. We decided to include this form of normalization in the test to provide a more complete account of possible normalizations. In  $ABU_{pos}$ , any negative values are clipped before normalization, such that all blending weights are strictly positive ( $\sum_j \alpha_{ij} = 1; \alpha_{ij} > 0$ ). Finally, in  $ABU_{soft}$ , we realized the same constraint (all-positive and normalized) by applying softmax normalization to the blending weights. With the exception of  $ABU_{abs}$ , none of the normalized versions of ABUs can realize an overall scaling of the resulting functions for positive input values. For the experiments, we used the same network and task configurations as in the previous section.

## 4.2 Performance and Analysis

The results of our performance tests are reported in Table 2. All five versions of ABUs showed remarkably similar performance throughout the tested task and network configurations - the average gap between the best and weakest performing ABU in a given setup is a mere 0.63%. In terms of mean rank, ABU

and  $\text{ABU}_{\text{abs}}$  lead the field, and are thus the two most robust configurations. However, none of the other three versions fell far behind. We again used the default setup (vanilla SMCN, Adam, CIFAR10) for an analysis of the blending weights and their effects on the succeeding pre-activations. We found  $\text{ABU}_{\text{abs}}$ s to behave much like unconstrained ABUs, implementing adaptive scaling, keeping the layer statistics constant over the course of training, thus mitigating covariate shift. Despite the fact that the scaling imposes constraints on the shape of the resulting function (as outlined above),  $\text{ABU}_{\text{abs}}$  performed very similarly to unconstrained ABUs in most settings. By contrast, but very much expectedly,  $\text{ABU}_{\text{norm}}$ ,  $\text{ABU}_{\text{pos}}$ , and  $\text{ABU}_{\text{soft}}$ , were unable to keep the layer statistics at constant levels, and a considerable covariate shift akin to that in fixed activation functions was observed. Interestingly, this appears to have only a minor impact on performance, and they were able to keep up with, or even outperform unconstrained ABUs in some of the tested settings. The good performance of normalized ABUs in our tests is in line with [20], who found very similar or identical units<sup>1</sup> to outperform common activation functions in MNIST, CIFAR and ImageNet tasks, using widely used network architectures, such as AlexNet and ResNet-56. In conclusion, while ABUs generally apply adaptive scaling when possible, the ability to learn the function’s shape by itself already helps to improve network performance beyond the level of the established activation functions they are comprised of.

## 5 Pre-training Scaling and Blending Weights

Finally, we investigated for both adaptive scaling and ABUs, whether or not the adaptiveness of scaling and blending weights by itself is an important factor for the overall performance of the network, and if the performance could possibly be further improved by using pre-trained weights. To this end, we set up an experiment with two main conditions. In both of them, we initialized the networks with the final scaling or blending weights of a preceding run. We then fixed these values after initialization in one condition, while keeping them adaptive in another. In case of ABUs, it is conceivable that the *shape* of the function at the end of training would be ideal, while the *scaling* may be too low at the beginning of a new run. Therefore, we added two more conditions akin to the main condition and only for unconstrained ABUs, in which we normalized the pre-trained blending weights after initialization, thus keeping the learned shape, while resetting the learned scaling. All tests were based on the default setup (vanilla SMCN, Adam, CIFAR10).

The results are shown in Table 3. For  $\alpha\text{tanh}$  and  $\alpha\text{ReLU}$ , initializing the scaling weights at the predominantly low final values of a preceding run helped to

---

<sup>1</sup> With respect to the constraints, the *affine()* units in [20] are equivalent to  $\text{ABU}_{\text{norm}}$ , and their *convex()* units are equivalent to  $\text{ABU}_{\text{pos}}$ . Unfortunately, the authors did not provide details with respect to their implementation, so we cannot say whether or not the implementations are equivalent.



**Table 3.** Performance (percentage correct) after initialization of scaling/blending weights  $\alpha$  at the final values of a preceding run, i.e., after 60000 steps. Additionally for ABUs: Pre-trained weights normalized at initialization to avoid too low initial scaling. Highest performing treatment of scaling and blending weights per activation function (row) indicated in bold.

Network	SMCN	SMCN	SMCN	SMCN	SMCN
Optimizer	Adam	Adam	Adam	Adam	Adam
Task	CIFAR10	CIFAR10	CIFAR10	CIFAR10	CIFAR10
$\alpha$ init	$\frac{1}{m}$	Pre-trained	Pre-trained	Pre-trained (norm.)	Pre-trained (norm.)
$\alpha$ trainable	✓	–	✓	–	✓
$\alpha$ tanh	79.07 $\pm$ 0.07	79.60 $\pm$ 0.07	<b>79.71 <math>\pm</math> 0.06</b>	–	–
$\alpha$ ReLU	79.23 $\pm$ 0.15	79.62 $\pm$ 0.11	<b>79.77 <math>\pm</math> 0.09</b>	–	–
$\alpha$ ELU	<b>82.60 <math>\pm</math> 0.06</b>	79.45 $\pm$ 0.15	81.22 $\pm$ 0.10	–	–
ABU	<b>83.12 <math>\pm</math> 0.06</b>	80.93 $\pm$ 0.15	82.02 $\pm$ 0.15	80.99 $\pm$ 0.12	82.88 $\pm$ 0.06
ABU <sub>nrm</sub>	<b>82.82 <math>\pm</math> 0.07</b>	78.88 $\pm$ 0.08	81.78 $\pm$ 0.14	–	–
ABU <sub>abs</sub>	<b>83.14 <math>\pm</math> 0.08</b>	79.80 $\pm$ 0.24	82.42 $\pm$ 0.21	–	–
ABU <sub>pos</sub>	<b>82.90 <math>\pm</math> 0.05</b>	78.93 $\pm$ 0.07	81.45 $\pm$ 0.16	–	–
ABU <sub>soft</sub>	82.54 $\pm$ 0.07	<b>82.69 <math>\pm</math> 0.05</b>	81.69 $\pm$ 0.08	–	–

improve performance. In both cases, runs with fixed pre-trained  $\alpha_i$  already surpassed the performance of the preceding run, but keeping them adaptive over the course of training led to further improvements. The fact that fixed pre-trained scaling weights yielded an increase in performance suggests that the initial variance of weights in the weight matrices (derived using He initialization), may not have been ideal as initial conditions, despite resulting in pre-activation variances of about 1.  $\alpha$ ELU, by contrast, substantially lost performance after initialization with pre-trained scaling weights, irrespective of whether or not they were fixed or adaptive throughout the run. With the exception of ABU<sub>soft</sub>, the same was the case in all versions of ABUs, where fixed blending weights, in particular, led to sizable drops in performance of up to four percent. ABU<sub>soft</sub>, being the exception to this rule, improved slightly for fixed pre-trained blending weights, but lost performance with adaptive pre-trained blending weights. Normalizing the unrestricted ABU blending weights after initialization with pre-trained values led to improvements over non-normalized pre-trained blending weights, but the default setup with initialization at  $\frac{1}{m}$  still performed best. Overall, we found all but one of the tested activation functions to perform best, when the blending weights were kept adaptive, as opposed to fixed after initialization. These results suggest that in both adaptive scaling and ABUs, much of the gained performance is won by keeping the scaling and/or shape adaptive. Moreover, the fact that this applies also to most normalized versions of ABUs indicates that there may not be any single optimal shape for an activation function in a given layer.

## 6 Limitations

In the following, we briefly highlight two noteworthy limitations of this work. Firstly, the ABUs presented here are based on five distinct activation functions, chosen for their prevalence in literature (e.g., ReLU), their standalone performance (e.g., Swish), and to generate a wide range of achievable shapes (e.g., tanh). We believe that a more principled approach to the choice of activation functions used in ABUs might reveal even higher-performing combinations. Due to limited computing resources, we were not able to perform an exhaustive search for optimal combinations of activation functions. Similarly, an in-depth overview of theoretical considerations concerning optimal blends was not provided here, but should be pursued in future work. Secondly, we have so far only evaluated our approach on supervised learning task from the field of computer vision. Future work on this topic should further include experiments based on other applications of deep learning, such as time-series prediction or reinforcement learning.

## 7 Conclusion and Outlook

In summary, we introduced Adaptive Blending Units (ABUs), and analyzed adaptive scaling for common activation functions. We found robust performance advantages of both approaches over established activation functions in a range of tasks and network architectures. In adaptive scaling, the performance advantages could be traced back to stabilized pre-activation statistics during training, mitigating covariate shift. The same behavior was found for unconstrained ABUs, while normalized ABUs reached similar levels of performance without the ability to significantly scale the layer output. Our results suggest that the adaptiveness of the shape over the course of training may play a major role in this, as well, as opposed to simply converging to some ideal shape.

With respect to adaptive scaling, a logical next step would be to introduce a shifting parameter per layer, to allow the network to further optimize the input distributions to the activation functions, and to move this learned normalization in front of the activation:  $f(\alpha_i \cdot (x + \beta_i))$ . This form of self-organized normalization could be explicitly combined with ABUs, thus detaching the handling of layer statistics from the shape of the activation function. Recurrent networks may be a particularly interesting field of application for self-optimization of layer statistics, as it should, in principle, mitigate some of the issues associated with explicit normalization techniques. Interestingly, adaptive scaling has been discussed for neural populations outside of the field of deep learning and proven helpful in maintaining stable output distributions [17, 24]. Beyond this, an increase in the number of distinct ABUs within a layer may yield further improvements in performance, as well as a systematic search for high-performing sets of activation functions in ABUs.

## References

1. Agostinelli, F., Hoffman, M., Sadowski, P., Baldi, P.: Learning activation functions to improve deep neural networks. arXiv preprint [arXiv:1412.6830](https://arxiv.org/abs/1412.6830) (2014)

2. Alcaide, E.: E-swish: adjusting activations to different network depths. arXiv preprint [arXiv:1801.07145](https://arxiv.org/abs/1801.07145) (2018)
3. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint [arXiv:1607.06450](https://arxiv.org/abs/1607.06450) (2016)
4. Clevert, D.-A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint [arXiv:1511.07289](https://arxiv.org/abs/1511.07289) (2015)
5. Dushkoff, M., Ptucha, R.: Adaptive activation functions for deep networks. *Electron. Imaging* **2016**(19), 1–5 (2016)
6. Eisenach, C., Wang, Z., Liu, H.: Nonparametrically learning activation functions in deep neural nets (2016). <https://openreview.net/pdf?id=H1wgawqxl>, <https://scholar.google.de/scholar?hl=en&assdt=0%2C5&q=https%3A%2F%2Fopenreview.net%2Fpdf%3Fid%3DH1wgawqxl&btnG=>
7. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 315–323 (2011)
8. Godfrey, L.B., Gashler, M.S.: A continuum among logarithmic, linear, and exponential functions, and its potential to improve generalization in neural networks. In: 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), vol. 1, pp. 481–486. IEEE (2015)
9. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. arXiv preprint [arXiv:1302.4389](https://arxiv.org/abs/1302.4389) (2013)
10. Hahnloser, R.H.R., Sarpeshkar, R., Mahowald, M.A., Douglas, R.J., Seung, H.S.: Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* **405**(6789), 947 (2000)
11. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
12. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint [arXiv:1502.03167](https://arxiv.org/abs/1502.03167) (2015)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *CoRR*, abs/1412.6980 (2014). <http://arxiv.org/abs/1412.6980>
14. Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S.: Self-normalizing neural networks. In: Advances in Neural Information Processing Systems, pp. 972–981 (2017)
15. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech Report (2009). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.9220&rep=rep1&type=pdf>, <https://scholar.google.de/scholar?hl=en&assdt=0%2C5&q=Learning+multiple+layers+of+features+from+tiny+images&btnG=>
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
17. Leugering, J., Pipa, G.: A unifying framework of synaptic and intrinsic plasticity in neural populations. *Neural Comput.* **30**(4), 945–986 (2018). <https://scholar.google.de/scholar?hl=en&assdt=0%2C5&q=A+unifying+framework+of+synaptic+and+intrinsic+plasticity+in+neural+populations.+Neural+Comput.&btnG=#d=gscit&u=%2Fscholar%3Fq%3Dinfo%3AXUGb1r4qYAJ%3Ascholar.google.com%2F%26output%3Dcite%26scirp%3D0%26hl%3Den>
18. Li, X., Chen, S., Hu, X., Yang, J.: Understanding the disharmony between dropout and batch normalization by variance shift. arXiv preprint [arXiv:1801.05134](https://arxiv.org/abs/1801.05134) (2018)
19. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the ICML, vol. 30, p. 3 (2013)

20. Manessi, F., Rozza, A.: Learning combinations of activation functions. arXiv preprint [arXiv:1801.09403](https://arxiv.org/abs/1801.09403) (2018)
21. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML 2010), pp. 807–814 (2010)
22. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. CoRR, abs/1710.05941 (2017). <http://arxiv.org/abs/1710.05941>
23. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014). <http://jmlr.org/papers/v15/srivastava14a.html>
24. Turrigiano, G.G., Nelson, S.B.: Homeostatic plasticity in the developing nervous system. *Nat. Rev. Neurosci.* **5**(2), 97 (2004)
25. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint [arXiv:1505.00853](https://arxiv.org/abs/1505.00853) (2015)



# Application of Neural Networks to Characterization of Chemical Sensors

Mahmoud Zaki Iskandarani<sup>(✉)</sup>

Al-Ahliyya Amman University, P.O.BOX: 19328, Amman, Jordan  
m.iskandarani@ammanu.edu.jo

**Abstract.** Prediction of current-voltage characteristics of chemical PbPc sensors with different inter-electrode separation using Neural Networks is carried out successfully. The main purpose of the work is to determine in advance device properties such as current and voltage based on available test data, without the need to build the device. Thus, modification of the design can be carried out based on the optimized and predicted values produced by the Neural Networks model. The produced devices have capabilities to detect small amounts of NO<sub>2</sub>, which is considered a hazardous gas emitted by various vehicles and can cause undesirable pollution. The used Weight Elimination Algorithm (WEA), proved that as the inter-electrode separation increases, the injected current as a function of applied voltage will also increase, due to more available surface area of vacuum sublimed PbPc material. Also, the response showed non-linearity at larger inter-electrode separations due to separation values and increased bulk interaction effect in addition to the surface interaction of charge transfer. The main benefit of Neural Networks model is to predict values resulting from complex mechanisms, which, otherwise hard to evaluate and model.

**Keywords:** Chemical sensors · Neural Networks · Prediction · Intelligent transportation systems · Smart cities

## 1 Introduction

Gas adsorption on the surface of organic materials will have a notable effect on the electrical properties of such materials. The electrical characteristics of an organic material that contains transitional, heavy central atom will suffer large change due to vapor adsorption on the organic material surface, as a result of chemical interaction between the adsorbed gas and the adsorbing organic material.

Chemical vapors adsorbed by Phthalocyanines surface due to interaction with detected chemicals, will evidently affect their electrical conductivity, hence a change in their current-voltage characteristics. The interaction comprises formation of bonds between the adsorbed chemical vapors and the sublimed organic material through charge transfer mechanism. Such charge transfer mechanism occurs both on the surface and in the bulk, through charge injection process. Phthalocyanines are mainly p-type semiconductors and have good stability reacting to chemicals and heat. In addition, Phthalocyanines are easily sublimed, leading to high purity thin films [1, 2]. The physicochemical

properties can be altered by changing the metal ion. When substituting with Pb, the resulting organic devices could have different molecular plan orientation, which affect their overall characteristics.

PbPc regarded as one of the most sensitive and stable among the MPc organic complexes. Its high sensitivity and selectivity is mainly confined to strongly accepting, electrophilic gases such as  $\text{NO}_2$  and general  $\text{NO}_x$ , and has high stability and reproducibility of electrical characteristics. PbPc films which consist of amorphous and/or polycrystalline states can be implemented as carrier transport layers in organic devices.

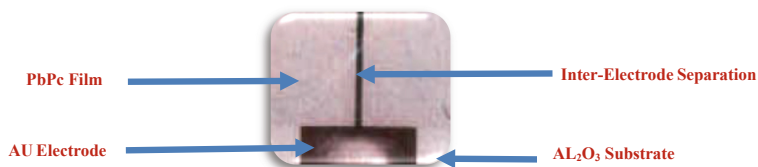
It is important to have low cost, flexible electronic devices based on organic thin film technology, such as Lead-Phthalocyanines to implement chemical and odor detection functionalities. Such applications are supported by the drive to develop and apply the concept of Electronic Noses. Lead-Phthalocyanines complexes have capabilities for applications in different areas of gas and odor detection in addition to other photoelectric applications. Thus, PbPc can be used in the build of an Electronic Nose for detection of gaseous substances, in particular  $\text{NO}_2$ . Such a device produces fingerprints by capturing and analyzing the adsorbed chemical vapor [3–5].

By processing the obtained signal as a result of chemical interaction, the constituent substance(s) would be recognized and classified. Such functionality is important in many environmental, quality and security control systems applications, as in intelligent transportation systems to detect vehicles emission of  $\text{NO}_2$ , and in monitoring of smart cities environment against factory emission, and commercial goods carrying vehicles that runs on diesel fuel. The software processes the obtained signals, and filter out unnecessary information and performs pattern recognition using algorithms such as Back Propagation, Weight Elimination and many others [6–17].

In this work, the current-voltage ( $I$ - $V$ ) characteristics for PbPc chemical sensor devices with various inter-electrode separation is presented. The work is supported by Neural Networks (weight Elimination Algorithm) to optimize tested devices characteristics and to predict current-voltage behavior using other inter-electrode separations for similar design of PbPc chemical sensor devices.

## 2 Materials and Methods

Lead Phthalocyanaine (PbPc) chemical sensors produced through vacuum sublimation on Sapphire substrate and by using Gold electrodes. Each substrate hosted three chemical devices with an overall area of  $1.6 \text{ cm}^2$  as shown in Fig. 1. Each device, is packaged with external contacts to be mounted on the testing board.



**Fig. 1.** PbPc chemical sensor

A testing chamber was designed to obtain the current-voltage characteristics with data acquisition interface to computing facilities as shown in Fig. 2. The testing system performed sensor testing under temperature control with gas and chemical disposing facility. Sensors placed on a plate that is inserted into the testing chamber, thus enabling multi sensor testing. A mass flow control system is used to supply the required NO<sub>2</sub> concentrations with disposal facilities.



Fig. 2. Chemical sensors testing system

The obtained data, are stored and form an input to the neural networks algorithm to predict current-voltage values for other inter-electrode separation and to optimize the obtained data for other values of currents and voltages that were not part of the testing process.

Weight Elimination Algorithm (WEA) is employed as an effective Neural Networks algorithm in order to enable current-voltage prediction and curve optimization using the model shown in Fig. 3, where testing data from two devices with different inter-electrode separation is used for training.

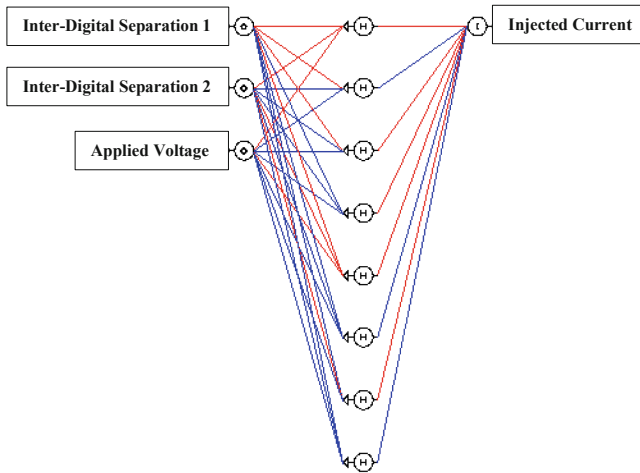


Fig. 3. WEA training and testing model

Weight elimination describes the dynamic changes in neural network convergence through error functions. The overall weight elimination error function is shown in Eq. (1), while Eqs. (2) and (3) describe the terms used for the overall error computation to achieve convergence and optimum prediction.

$$E_{WE} = E_{start} + E_{penalty} \quad (1)$$

Where;

$$E_{start} = \frac{1}{2} \sum_j (d_k - o_j)^2 \quad (2)$$

$$E_{penalty} = \beta \left( \sum_{i,j} \frac{\left(\frac{w_{i,j}}{w_N}\right)^2}{1 + \left(\frac{w_{i,j}}{w_N}\right)^2} \right) \quad (3)$$

Where;

$E_{WE}$ : The combined Start function that includes the initial Error function,  $E_{start}$  and the weight-elimination term  $E_{penalty}$ .

$\beta$ : The weight-reduction factor,

$w_{i,j}$ : Represents the individual weights of the neural network model.

$w_N$ : A scale parameter computed by the WEA.

$d_j$ : The desired Output.

$o_j$ : The actual Output.

### 3 Results

Table 1 shows testing results for two PbPc devices with 5  $\mu\text{m}$  and 15  $\mu\text{m}$  inter-electrode separation. The obtained values are used to train the WEA model in Fig. 3, while Figs. 4 and 5 show the I-V characteristics of each device.

**Table 1.** I-V testing results for two Lead-Phthalocyanine (PbPc).

Applied voltage (mV)	Injected current (pA)	
	Ia 5 $\mu\text{m}$	Ib 15 $\mu\text{m}$
000	0.00	0.00
200	4.90	19.0
250	5.54	21.4
300	6.10	23.3
350	6.72	26.7
400	7.54	29.7
450	7.85	30.9
500	8.47	32.2
550	9.10	33.4
600	9.68	34.1



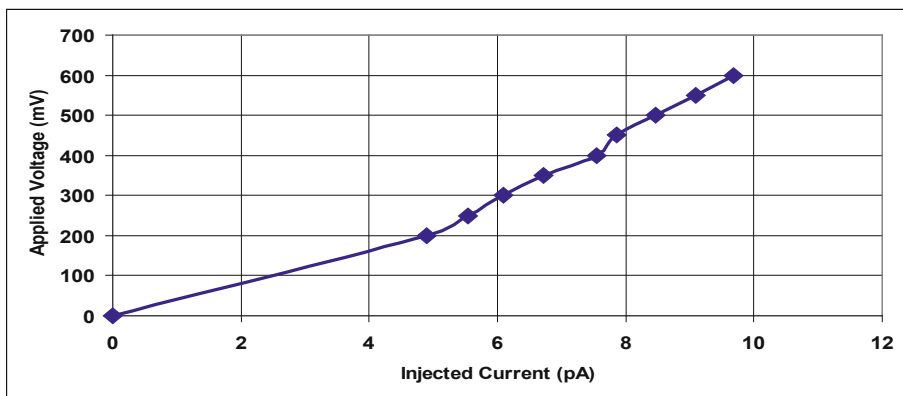


Fig. 4. I-V curve for 5  $\mu\text{m}$  PbPc sensor

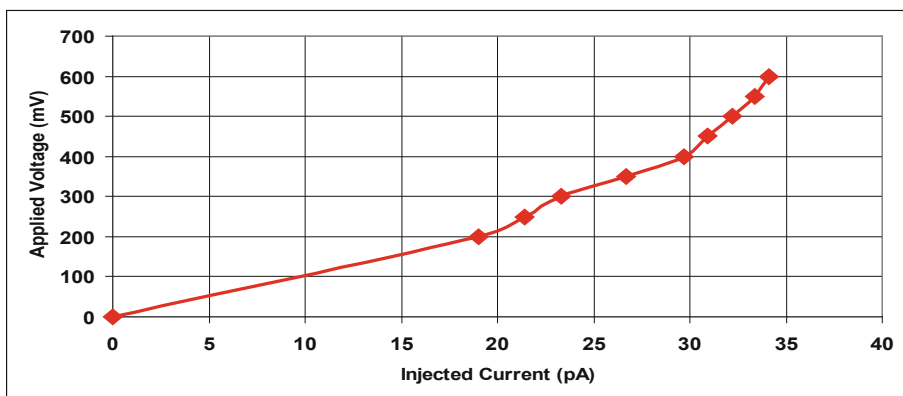


Fig. 5. I-V curve for 15  $\mu\text{m}$  PbPc sensor

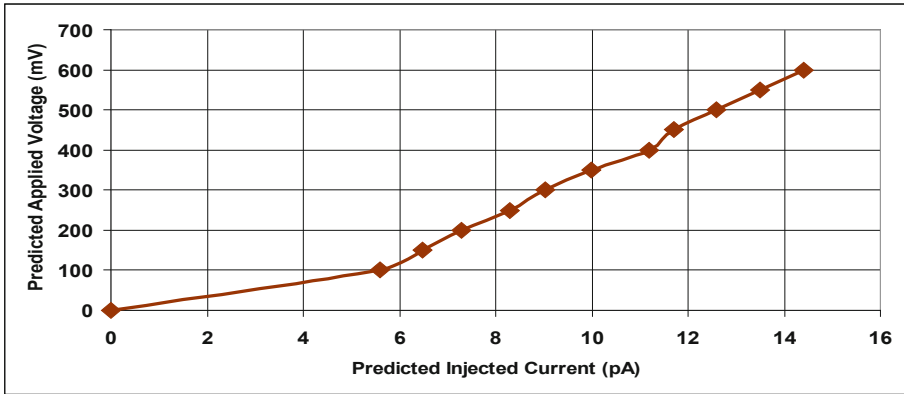
## 4 Discussion and Conclusion

Table 2 shows the testing results using the WEA Neural Networks model shown in Fig. 3, while Fig. 6 shows the I-V characteristics for the predicted device of 10  $\mu\text{m}$  PbPc sensor, with Fig. 7 showing a comparison of three devices response based on prediction and new optimized applied voltage range. Both 5  $\mu\text{m}$  and 15  $\mu\text{m}$  used in the training process, whilst the 10  $\mu\text{m}$  device was predicted. Also the response for 100 mV and 150 mV is used in the prediction process and was not part of the test.

By scaling and examining the previous figures as shown in Figs. 8 and 9, it is clear that for inter-electrode separations of 5  $\mu\text{m}$  and 10  $\mu\text{m}$ , the conductivity process can be regarded as linear, where surface interactions dominates with the presence of some non-linearity due to impurities and electrodes contacts, while, Fig. 10, which presents I-V characteristics of 15  $\mu\text{m}$  inter-electrode separation PbPc sensor, clearly indicates non-linear behavior, which is due to the effect of bulk and surface interactions combined. Figure 11 shows comparison between all three PbPc sensors.

**Table 2.** Predicted and optimized I-V testing results for three PbPc sensors.

Applied voltage (mV)	Injected current (pA)		
	Ia (5 $\mu\text{m}$ )	Ib (15 $\mu\text{m}$ )	Ic (10 $\mu\text{m}$ )
000	0.00	0.00	0.00
100	3.73	9.21	5.60
150	4.31	15.1	6.47
200	4.85	18.9	7.30
250	5.50	21.3	8.30
300	6.02	23.2	9.03
350	6.65	26.5	9.98
400	7.47	29.6	11.2
450	7.77	30.8	11.7
500	8.40	32.0	12.6
550	9.02	33.2	13.5
600	9.60	33.9	14.4



**Fig. 6.** Predicted I-V characteristic for 10  $\mu\text{m}$  PbPc sensor

Figures 12 and 13 show a comparison between the measured values and predicted values of the injected current using Weight Elimination Algorithm. The predicted values show very close approximation to the testing ones, hence, a very good confidence in the obtained values for other inter-electrode separations is achieved.

The used WEA algorithm contributed in three ways:

1. Extended the range of applied voltage to enable testing the devices using other values of voltages without having to subject them to test.

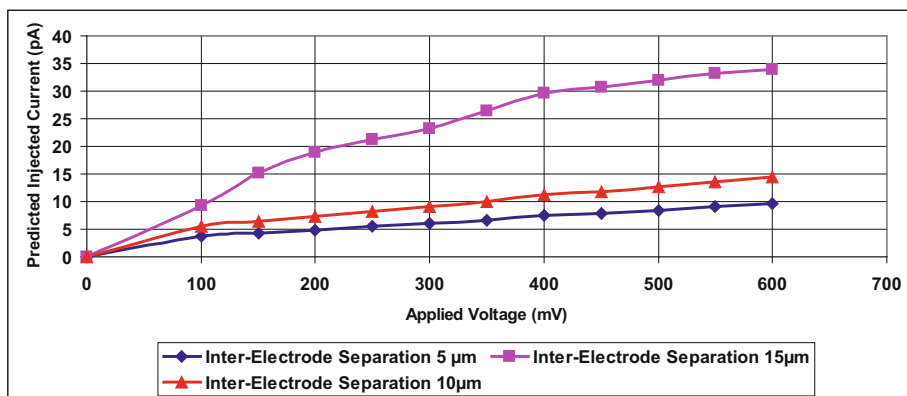


Fig. 7. Comparison of predicted PbPc sensors responses

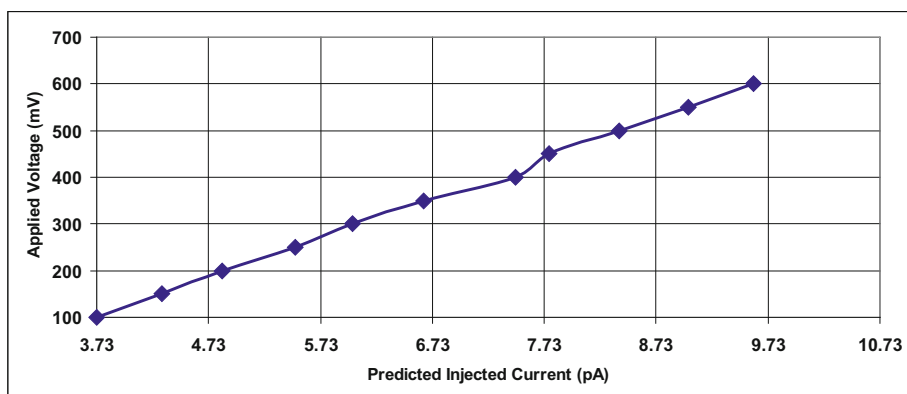


Fig. 8. Predicted and scaled I-V characteristic for 5  $\mu\text{m}$  PbPc sensor

2. Enabled the prediction of I-V characteristics of other similar devices with various inter-electrode separations.
3. Provided a good insight into the I-V behavior of large inter-electrode separations, which in this work shows that above 10  $\mu\text{m}$  separation, the response will display a non-linear characteristics due to bulk charge transfer effect.

The mentioned achievement is very important in terms of response optimization and more so in terms of enabling design changes based on intelligent correlation, prediction, and modeling of complex conduction mechanism known to occur in organic materials and specifically in metal-substituted Phthalocyanines, such as PbPc.

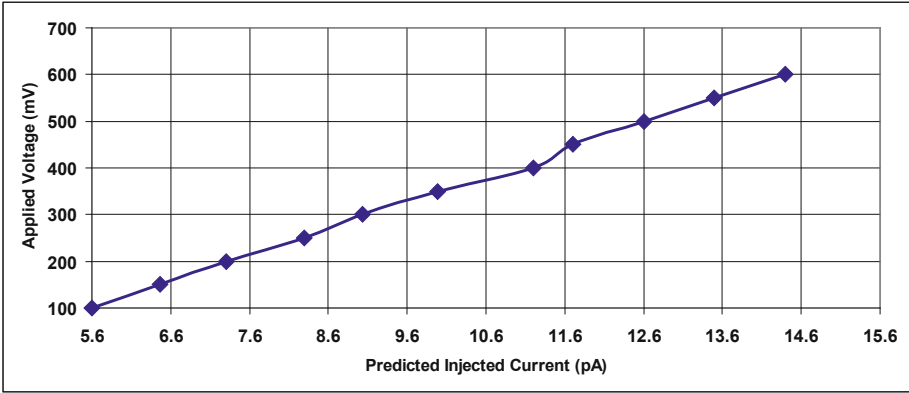


Fig. 9. Predicted and scaled I-V characteristic for 10  $\mu\text{m}$  PbPc sensor

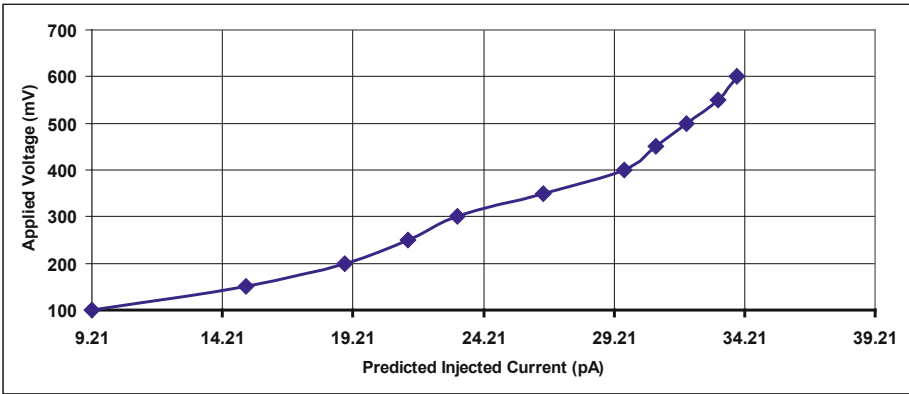


Fig. 10. Predicted and scaled I-V characteristic for 15  $\mu\text{m}$  PbPc sensor

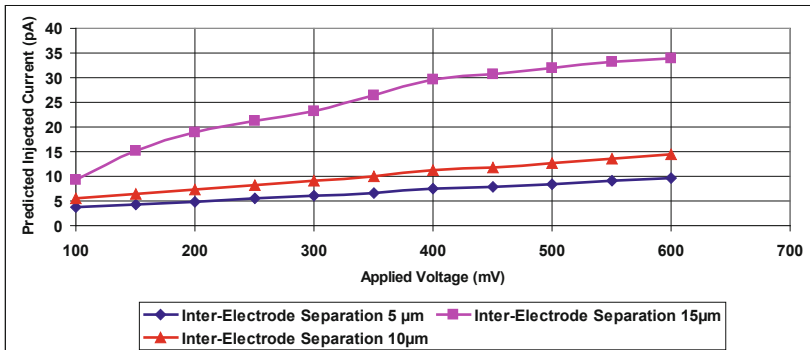


Fig. 11. Comparison of predicted and scaled PbPc sensors response

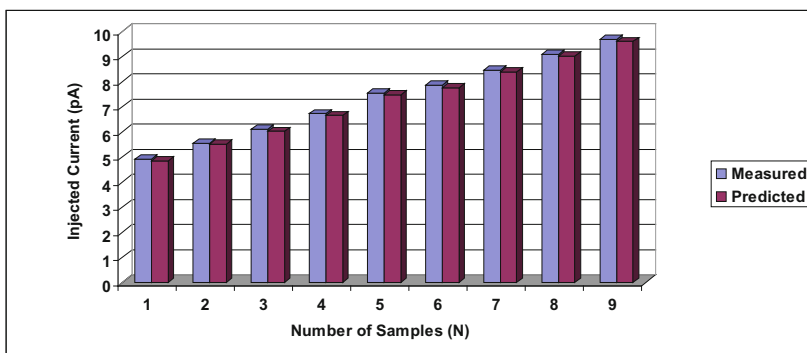


Fig. 12. Comparison between measured and predicted injected current in a 5 μm PbPc sensor

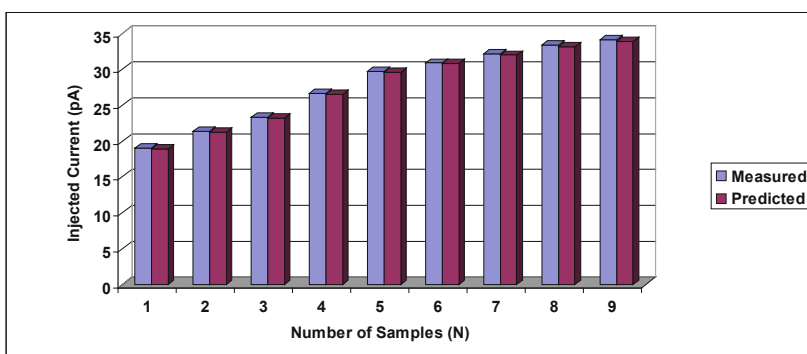


Fig. 13. Comparison between measured and predicted injected current in a 15 μm PbPc sensor

## References

1. Hany, R., Cremona, M., Strassel, K.: Recent advances with optical upconverters made from all-organic and hybrid materials. *Sci. Technol. Adv. Mater.* **20**, 496–510 (2019)
2. Nazemi, H., Joseph, A., Park, J., Emadi, A.: Advanced micro-and nano-gas sensor technology: a review. *Sensors* **19**(1285), 1–23 (2019)
3. Ding, R., Xu, Z., Zheng, T., Huang, F., Peng, Y., Lv, W., Yang, Y., Wang, Y., Xu, S., Sun, L.: Realizing high-responsive superlattice organic photodiodes by C 60 and zinc phthalocyanine. *J. Mater. Sci.* **54**(4), 3187–3195 (2019)
4. Li, Y., Wang, B., Yu, Z., Zhou, X., Kang, D., Wu, Y., He, C., Zhou, X.: The effects of central metals on ammonia sensing of metallophthalocyanines covalently bonded to graphene oxide hybrids. *RSC Adv.* **7**, 34215–34225 (2017)
5. Govardhan, K., Nirmala, A.: Temperature optimized ammonia and ethanol sensing using ce doped tin oxide thin films in a novel flow metric gas sensing chambers. *J. Sens.* **2016**, 1–12 (2016). Article ID 7652450
6. Yan, J., Guo, X., Duan, S., Jia, P., Wang, L., Peng, C., Zhang, S.: Electronic nose feature extraction methods: a review. *Sensors* **15**(11), 27804–27831 (2015)
7. Vishesh, S., Srinath, M., Gubbi, K., Shivu, H., Prashanta, N.: Portable low cost electronic nose for instant and wireless monitoring of emission levels of vehicles using android mobile application. *Int. J. Adv. Res. Comput. Commun. Eng.* **5**(9), 134–140 (2016)

8. Suganya, R., Uthayakumar, R.: Electronic nose for accident prevention and vehicleblack box system **4**(5), 1206–1209 (2015)
9. Guentner, A., Koren, V., Chikkadi, K., Righettoni, R., Pratsinis, S.E.: E-nose sensing of low-ppb formaldehyde in gas mixtures at high relative humidity for breath screening of lung cancer. *ACS Sens.* **1**(5), 528–535 (2016)
10. Sun, Y., Luo, D., Li, H., Zhu, C., Xu, O., Hosseini, H.: Detecting and identifying industrial gases by a method based on olfactory machine at different concentrations. *J. Electric. Comput. Eng.* **2018**, 1–9 (2018). Article ID 1092718
11. Tiele, A., Esfahani, S., Covington, J.: Design and development of a low-cost, portable monitoring device for indoor environment quality. *J. Sens.* **2018**, 1–14 (2018). Article ID 5353816
12. Yan, K., Zhang, D.: Calibration transfer and drift compensation of e-noses via coupled task learning. *Actuators B Chem.* **225**, 288–297 (2016)
13. Ma, Z., Luo, G., Qin, K., Wang, N., Niu, W.: Weighted domain transfer extreme learning machine and its online version for gas sensor drift compensation in e-nose systems. *Wirel. Commun. Mob. Comput.* **2018**, 1–17 (2018). Article ID 2308237
14. Di Gilio, A., Palmisani, J., de Gennaro, G.: An innovative methodological approach for monitoring and chemical characterization of odors around industrial sites. *Adv. Meteorol.* **2018**, 1–8 (2018). Article ID 1567146
15. Wu, Y., Liu, T., Ling, S., Szymanski, J., Zhang, W., Su, S.: Air quality monitoring for vulnerable groups in residential environments using a multiple hazard gas detector. *Sensors* **19**(362), 1–16 (2019)
16. Guerrero-Ibáñez, J., Zeadally, S., Contreras-Castillo, J.: Sensor technologies for intelligent transportation systems. *Sensors* **18**(1212), 1–24 (2018)
17. Iskandarani, M.: Two dimensional electronic nose for vehicular central locking system (E-Nose-V). *Int. J. Adv. Comput. Sci. Appl.* **10**(6), 63–70 (2019)



# Application of Machine Learning in Deception Detection

Owolafe Otasowie<sup>(✉)</sup>

Federal University of Technology, Akure, Nigeria  
oiyare@futa.edu.ng

**Abstract.** The issue of security is a continuous struggle for all. To address this struggle, it is pertinent to reliably detect deception. To reliably detect deception is a knotty task as no ideal technique has been found for the detection. According to literature, past researches focused on single cue, it was observed that combining cues will significantly be a good indicator of deception that using a single cue. Since no single verbal or non-verbal cue is able to detect deception successfully the research proposes to combine verbal and non-verbal cues for the detection. Therefore, this research aims to develop a neurofuzzy model for classifying extracted verbal and nonverbal features as deceptive or truthful. The proposed system extracted desired features from the dataset of Perez-Rosas. The verbal cues include the voice pitch, jitters, pauses, and speechrate. The PRAAT was used in extracting all the verbal cues. The nonverbal features were extracted using the Active Shape Model (ASM) and the classification Model was designed using Neurofuzzy technique. The work was implemented in 2015a MatLab. The developed model was compared with Support Vector Machine (SVM), K-Nearest Neighbour (KNN) and Decision Tree. Neurofuzzy recorded the best performance with the Nonverbal dataset (percentage score of 97.1%), KNN performed well with the Verbal dataset (percentage score of 90.9%) while Decision Tree performed best with the VerbNon dataset (percentage score of 97.6%). From the comparative analysis it was discovered that Neurofuzzy model work well on Nonverbal dataset to detect deception. The result obtained using only verbal cue was 84.3% while that of nonverbal cue was 97.1% but on VerbNon it yielded 92.5% which is far better than the chance level of 50%.

**Keywords:** Neurofuzzy · Verbal cues · Nonverbal cues · SVM · KNN · Decision tree · VerbNon cues

## 1 Introduction

Deception is as old as man and detecting it remains a daunting task [1] as no ideal technique has been found for the detection [2]. Studies revealed that the outcome of detecting even for experienced investigators is 50/50. Studying the nonverbal (psychological) and verbal (speech) cues of deception helps in increasing the success rate of detection. Lying requires the deceiver to keep the fact straight, make the story believable, and be able to withstand scrutiny. In [1], it was stated that when individuals tell the truth,

they often make every effort to ensure that other people understand while liars on the other hand attempt to manage peoples' perceptions. Consequently, people unwittingly signal deception via nonverbal and verbal cues (Cues are those indicators or variables that can be observed and measured and are believed to be indicative of deception). They stated further that no particular nonverbal or verbal cue evinces deception. As the struggle for a perfect method for detecting deception continues, there are disagreements that exist among those studying deception as to how the term should be defined. To resolve this issue an attempt is made to integrate the views of most influential scholars in the field to formulate a comprehensive and clear-cut definition of deception.

As a concept in most fields, deception has been defined in many ways. The authors in [3] defined it as "the communication of altered information so as to change another's perceptions from what the deceiver thought they would be without alteration". This definition specified the acts of lying as information alteration. However, the deceiver could act differently to mislead the receiver, for example, by concealing information. In 1986, [4] defined deception in a general way as "a false communication that tends to benefit the communicator". One flaw in this definition according to [5] was that some attributes of deception was not implicit. The author in [6] defined deception as "one person intends to mislead another, doing so deliberately, without prior notification of this purpose, and without having been explicitly asked to do so by the target". This definition explicitly stated that deception is an intentional action. In 1996 [7] defined deception more precisely and concisely as "a sender's knowingly transmitting messages intended to foster a false belief or conclusion in the receiver". Rather than focusing on the act itself, they judged deception on the basis of the deceiver's motivations in an interpersonal communication context.

The author in [8] defined deception elaborately as the deliberate attempt to hide, formulate, manipulate in any other way, factual or emotional information, by verbal and/or nonverbal means, in order to create or maintain in another or others a belief that the communicator himself or herself considers false. While [9] simply stated that deception is the act of deceiving and [10], a notable scholar in the field of deception defined the concept as "a successful or unsuccessful deliberate attempt, without forewarning, to create in another a belief which the communicator considers to be untrue".

From the definitions above, deception can be seen to include several types of interactions or concealing that serve to change or leave out the complete truth. It can also be deduced that, with deception, purpose is vital as it differentiates between deception and an honest mistake.

The question that remains a subject of controversy and which this research will tend to address is whether deception can reliably be detected through verbal or nonverbal means or a combination of both.

## 2 Related Works

Detecting deception has been an issue in scientific research as no single cue can reliably detect deception [5, 10]. Human investigators perform a little better than chance and as such a reliable means to effectively detect deception becomes paramount.

The authors in [2] studied the behaviour of people in the process of telling the truth and when lying. Their research results show that liars are less informative than truth



tellers, and they tell less convincing tales. The researcher also reported that liars make a more negative impression and are tense. However, behaviours showed no discernible links, or only weak links, to deceit. Cues to deception were more pronounced when people were motivated to succeed, especially when the motivations were identity relevant rather than monetary or material. Cues to deception were also stronger when lies were about transgressions. These cues are verbal and nonverbal. Verbal cues are linguistic patterns exhibited in spoken messages while nonverbal cues are leakages or deformations that occur in the body channels of the deceiver.

The work of [11] examined certain systematically identifiable segments—called CRITICAL SEGMENTS—that bear propositional content directly related to the topics of most interest in the interrogation. They augmented the approach with techniques for adjusting the class imbalance in the data. The results, as much as 23.8% relative improvement over chance, substantially exceed human performance at the task of TRUTH and LIE classification. Further, models generated using these segments employ features consistent with hypotheses in the literature and the expectations of practitioners [12] about cues to deception.

The authors in [13] stated that since some cues (micro expressions) appears in a microseconds, detecting deception by trained and untrained professionals becomes a little better than chance. They stated further that creating or developing an automated tool that will help in flagging these deceptive cues is paramount.

In [14], the authors outlined three basic cues that are associated with deception. They are: the verbal cue, nonverbal and the word cues.

In 2006 [15] focused on how the behaviour of previously unseen persons can be charted using back-propagation neural network. The work was carried out using a simulated theft scenario where 15 participants were asked to either steal or not to steal some money and were later interviewed about the location of the money. A video of each interview was presented to an automatic system, which collected vectors containing nonverbal behavioural data. Each vector represented a participant's nonverbal behaviour related to "deception" or "truth" for a short period of time. These vectors were used for training and testing a back-propagation ANN which was subsequently used for charting the behavioural state of others.

In [16] the authors in their work address the question pertaining to the nature of deception language. The research aimed at the exploration of deceit in Spanish written communication. The work designed an automatic classifier based on Support Vector Machines (SVM) for the identification of deception in an *ad hoc* opinion corpus. In order to test the effectiveness of the LIWC2001 categories in Spanish, the authors drew a comparison with a Bag-of-Words (BoW) model. The results indicate that the classification of the texts was successful. They concluded that the findings were potentially applicable to forensic linguistics and opinion mining, where extensive research on languages other than English is needed.

The authors in [17] developed and implemented a system for automatically identifying deceptive and truthful statements in narratives and transcribed interviews. The research focused only on verbal cues to deception for this preliminary research, without considering prosodic cues. The authors describe a language-based analysis of deception that were constructed and tested using real-life data such as criminal narratives,

police interrogations and legal testimony. The experiment involved two components: a set of deception indicators that were used for tagging a document and an interpreter that associates tag clusters with deception likelihood. The researchers tested the analysis by identifying propositions in the corpus that could be verified as true or false and then comparing the predictions of our model against this corpus of ground truth. The analysis achieved an accuracy of 74.9%.

The authors in [18] examined the effect of lying and telling the truth on cognitive load. The lie detector tried increasing the differences between lying and truth telling by introducing mentally taxing interventions. The authors assumed that more cognitive resources are required when they lie than when they are telling the truth. To support the claim, the authors provided empirical support for the approach. Their result showed that observers can discriminate better between liars and truth tellers when they are subjected to mentally taxing situations.

### 3 System Design

Since no single cue can reliably detect deception, forming a hybrid of verbal and non-verbal cues will help in detecting deception to a reasonably degree. The authors in [19] presented a multimodal deception detection model using real-life occurrences of deceit. The dataset they used in carrying out their research were recordings from public real-life trials and street interviews. The analysis of nonverbal behaviours in deceptive and truthful videos brought insight into the cue that play a major role in deception. They built classifiers relying on individual or combined sets of verbal and nonverbal features and achieve accuracies in the range of 77–82%. Their automatic system outperforms the human detection of deceit by 6–15%. This current research makes use of the dataset of Perez-Rosas.

In extracting the Nonverbal features, four steps are considered:

- (1) face model formation for the purpose of locating the face,
- (2) eye brow,
- (3) blink detection and length measurement, and
- (4) lip movement detection.

The face model is formed by the Active Shape Model (ASM). Active shape model is a different kind of object detection that is not template based. Template based model is for rigid objects like licence plate detection. ASM is a model that does not have a fixed shape but deforms to fit the object in question. In forming the shape model lots of training examples were collected (in this case, different faces) and the correspondence for each of the training examples were formed. Since all the shapes may not be properly aligned, the shapes were translated to be centred at the origin (0, 0). The dimension was reduced using Principal Component Analysis.

After various transformation, any shape  $Z$  can then be approximated using:

$$Z = \mu + Pb \tag{1}$$

where  $b$  is the model parameters,  $P = V_1, V_2 \dots V_k$ . The modes of  $V$  can correspond to shaking, smiling or nodding.

The ASM was used in getting the facial landmarks, after that the features correlating to eyebrow, eyeblink, nose and mouth were extracted and the data points aggregated using Eq. 3 through 6.

After the face model has been localized, the next step is to carryout blink detection. In doing this each eye is represented by 6 coordinated as shown in Fig. 1.

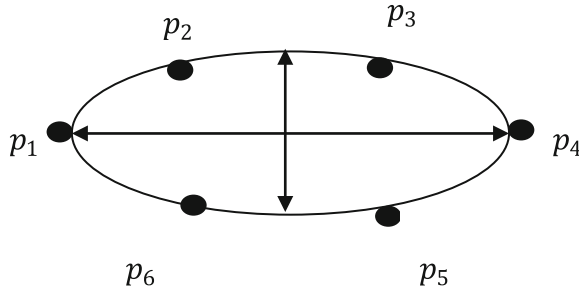


Fig. 1. Eye coordinates

The value between the height and the width of the eye is computed as:

$$E_b = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \tag{2}$$

**a) Data Points Aggregation**

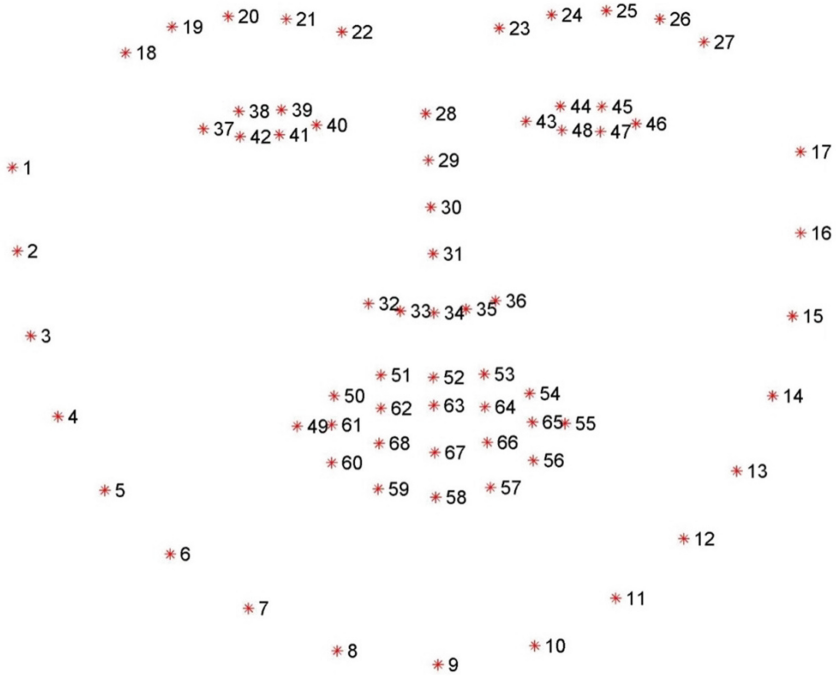
The data points representing each cues/features of interest are aggregated to form a single data point. In Active Shape model, landmarks representing the cues are labelled starting from the jaw outline as shown in Fig. 2. The average for each point representing a landmark is calculated and the value stored. The process is repeated for all landmarks. Equations 3 to 6 show the mathematical representation of the process, which correspond to steps 2 to 4.

Eye blinking ( $E_b$ ) :  
 If  $X \in \{37, \dots, 48\}$  then

$$E_b = \sum_{i=37}^{48} X_i / 12 \tag{3}$$

Where  $X_i$  are the data points representing the eye.

Lip movement ( $L_m$ ) :  
 If  $X \in \{49, \dots, 68\}$  then



**Fig. 2.** 68 facial coordinate points/index

$$L_m = \sum_{i=49}^{68} X_i/20 \tag{4}$$

Eyebrow movement ( $E_m$ ) :  
 If  $X \in \{18, \dots, 27\}$  then

$$E_m = \sum_{i=18}^{27} X_i/10 \tag{5}$$

Nose movement ( $N_m$ ) :  
 If  $X \in \{28, \dots, 36\}$  then

$$L_m = \sum_{i=28}^{36} X_i/9 \tag{6}$$

The values gotten were saved in an excel file and was imported into Matlab workspace for training and testing the neurofuzzy model.

### 3.1 Extracting the Verbal Features

The video clips were downloaded online. About 121 pieces voice data (deceptive and non deceptive) were collected. Useful features from the data collected were automatically extracted using Praat (A program designed to be used in Phonetics to analyse, synthesize and manipulate speech) and the extracted features were saved as a single excel document. The reason for using these cues for analysis resides in the fact that they provide information about voice signal aperiodicity, stability, noise, and frequency levels.

The verbal cues extracted are:

- i. Pauses (average sentence length, average word length, pausality): Pauses are defined as within-speaker silences. The standard duration of normal human pauses is around 200–250 ms (Goldman-Eisler 1968). Anything outside this range is considered questionable. Three linguistic variables (see Table 1) used to represent pauses are: low (200–220 ms), normal (210–240 ms) and high (230–250 ms).
- ii. Jitters: According to the threshold of pathology, a normal speaker’s jitter value should not exceed 1.040% and even much smaller than it. For the purpose of this research, jitter can assume any of the three values: low (90–99), normal (95–104) and high (100–110).
- iii. Speech rate: number of spoken words divided by the length of interview minus latency period. Three linguistic variables used to represent Speechrate are: low (133–160 wps), normal (147–174 wps) and high (166–188 wps).
- iv. Pitch: is the perceived fundamental frequency of voice and it is the rate of vibration of vocal folds. The range for male is 85–196 Hz while that of female is 155–334 Hz. For the purpose of this research, pitch can assume any of the three values: minimum (120–180 Hz), medium (150–210 Hz) and maximum (180–265 Hz).

**Table 1.** Degree of membership for verbal cues

S/N	Linguistic variables	Linguistic terms
1	Pause	Low (200–220), Normal (210–240), High (230–250)
2	Jitter	Low (90–99), Normal (95–104), High (100–110)
3	Speechrate	Low (133–160), Normal (147–174), High (168–188)
4	Pitch	Minimum (120–180), Medium (150–210), Maximum (180–265)

**Pitch Extraction:** All the information of fundamental frequency were stored and represented by PitchTier. PitchTier is one of the types of objects in PRAAT. The object represents a time-stamped pitch contour, that is, it contains number of (time, pitch) points, without voiced/unvoiced information.

In the course of this research, the interval was set as 0.01 s which means pitch value was extracted after every 0.01 s.

Praat uses the autocorrelation method for pitch analysis. The relation (Eq. 7) performs acoustic periodicity detection on the basis of an accurate autocorrelation method.

$$r_x(\tau) \approx r_{x,w}(\tau)/r_w(\tau) \quad (7)$$

Where  $r_x(\tau)$  represents autocorrelation of the original signal,  $r_{x,w}(\tau)$  is the autocorrelation of the windowed signal and  $r_w(\tau)$  is the autocorrelation of the window.

The method calculates the dot product of the original signal and a shifted version. The autocorrelation function  $r(\tau)$  of a signal with time lag  $\tau$  is defined as:

$$r(\tau) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)x(n+\tau) \quad (8)$$

**Jitters Extraction:** Jitter is the perturbation in the vibration of the vocal chords. It is known as Period-to-period fluctuations in fundamental frequency (F0). This causes the variation of the fundamental frequency in different cycles.

$$\bar{\alpha} = \frac{1}{N} \sum_{i=1}^N \alpha_i \quad (9)$$

$\alpha_i$  is any cyclic parameter (amplitude, pitch period, etc.) in the  $i^{th}$  cycle of the waveform,  $N$  is the span of cycles and  $\bar{\alpha}$  is the arithmetic mean.

The fundamental frequency perturbation is defined as the average of the absolute values of all these differences normalized to percentage:

$$jitter = \frac{100}{(N-1)\bar{\alpha}} \sum_{i=2}^N |\alpha_i - \alpha_{i-1}| \quad (10)$$

$\alpha_i$  is the fundamental frequency

Like the pitch extraction, jitter was automatically extracted by Praat. Jitter can assume any of the four different values which are: local, rap, ppq5 and ddp. In this research, the local was used because the range of values corresponded with that of the one given in pathology. This is the average absolute difference between consecutive periods, divided by the average period. The value of 1.040% was given as a threshold for pathology. According to the threshold of pathology, a normal speaker's jitter value should never exceed this threshold and even much smaller than it. The jitter values are all around 100 times smaller than 1. The threshold value for jitter in this research is 1.04%, any value greater than this is considered deceptive.

Pause Extraction: Pause is defined as a temporary stop or interruption in speech.

The pause was extracted using Eq. 11.

$$P_a = T_t - P_t \quad (11)$$

Where  $P_a$  is the total number of Pauses,  $T_t$  is Total length of time taken for the suspect to talk,  $P_t$  is the phonetic time (actual time taken to talk).

**Speechrate Extraction:** Speech rate is the term given to the speed at which one speaks. It is calculated as the number of words or number of syllables spoken in a minute. A *normal* number of words per minute (wpm/spm) can vary hugely.

The speechrate is extracted using Eq. 12.

$$S_r = N_s/T_t \quad (12)$$

Where speechrate is denoted as  $S_r$ , number of syllables as  $N_s$ , and total time taken as  $T_t$ .

### Verbal Features Extraction using Praat

The verbal cue was extracted using Praat software. Since Praat does not work with videos, the audio of the video clip was separated from the pictures so that Praat can work with it. This conversion was done using Aura-video-to-audio converter. To convert the video, the Aura application was launched. The video to convert was added using the “ADD VIDEO” button, the output profile was selected to be.wav (Praat works with.wav files). After adding the video, the “CONVERT” button was clicked and the video was converted to an audio file with a.wav extension and stored in an output folder.

The conversion was done for all the video clips and stored in a separate file. At the end of the conversion a total of 121 audio clips were saved.

When the Praat application is launched, the Praat object and picture comes up. The Praat object window is where most workflows are started. The menu is used to open, create and save files as well as to open the various editors and queries that are needed to work with sound files. When working with sound files, most of the time is spent in the editor. The editor can be accessed by selecting a sound in the object window and choosing the “view and Edit” option.

#### a) Verbal Dataset

The truthful and deceptive dataset of the voice data was extracted separately and saved in an excel file. Table 2 shows an extract of such dataset.

## 3.2 Nonverbal Features Extraction Using ASM

The ASM forms the shape model from various faces. Any other face can be estimated using the landmark of the shape model formed. A landmark signifies discernible marks surrounding most of the images being considered. An example is the location of the left eye brow. Locating landmarks on faces is equivalent to locating facial features, because landmarks mark out facial features.

In the video, there are 474 frames and the length of data in the frame is 68 corresponding to 68 landmarks. After extracting the features, the spaces which represent the region where the face was not correctly identified was padded with 0 or removed.

In this research, the facial features were identified using the index in Table 3 and Fig. 2.

#### a) Nonverbal Dataset

From the raw dataset, the right eyebrow is represented from position 7 through position 12. Same also goes for the left eye, nose and mouth. The nonverbal dataset was then computed from the mapped out facial region using Eq. 2 to 6.

**Table 2.** Verbal dataset

Speech_rate	Pitch	Jitter	Pause	Index
95.95841383	52.54629	0.7999	0	truthful
95.26737176	51.99007	0.765511	0	truthful
96.8306235	53.58682	0.731528	1	truthful
201.2122137	54.97358	0.840025	1	truthful
207.7803527	57.4701	0.880852	1	truthful
210.1547062	60.04052	0.932103	1	truthful
212.0595008	62.09669	0.933631	1	truthful
215.0715231	63.48046	0.955346	1	truthful
210.94644	63.87905	0.956792	1	truthful
209.9535014	63.80526	0.968159	1	truthful
206.1360589	64.03211	0.962456	1	truthful
205.2846898	64.16806	0.9662	1	truthful
206.7106642	64.09926	0.964366	1	truthful
212.7881211	64.27191	0.966944	1	truthful
221.7198483	64.56968	0.950117	1	truthful
137.7344793	49.118	0.838323	1	deceptive
137.1625044	47.59344	0.882926	1	deceptive
136.0443925	46.17882	0.868546	1	deceptive
135.1238952	43.97337	0.870338	1	deceptive
134.2104025	40.9724	0.878724	1	deceptive
133.7743346	43.8647	0.877148	1	deceptive
133.4804245	49.63414	0.873736	1	deceptive
133.4552413	51.50273	0.85624	1	deceptive
133.1051573	50.23301	0.897468	1	deceptive
131.4918257	50.40496	0.871267	1	deceptive
129.8769295	55.91121	0.914085	1	deceptive
128.7771548	59.84788	0.886471	1	deceptive
127.5571773	61.6082	0.855681	1	deceptive
124.8008349	62.11977	0.787959	1	deceptive
122.1898551	62.17991	0.682049	1	deceptive

#### 4 Neurofuzzy Model Implementation (ANFIS)

In MatLab environment, the ANFIS editor is started once the *anfisedit* command is typed at the command line. The Sugeno's inference mechanism is adopted with *Min* and



**Table 3.** Facial feature index

S/N	Feature	Index
1	Mouth	48–68
2	Right eyebrow	17–22
3	Left eyebrow	22–27
4	Right eye	36–42
5	Left eye	42–48
6	Nose	27–35
7	Jaw	0–17

*Max* operators for implication and aggregation operations, respectively. Three linguistic terms are used to describe the input and output variables. The Edit menu of the Fuzzy Inference System makes possible the choice of membership function and fuzzy rules formation.

The model was trained and tested using the verbal, nonverbal and combination of verbal and nonverbal.

#### 4.1 Training with Verbal Cues

The verbal dataset was divided into two sets, one set for training and the other set for testing the neurofuzzy model. 933 dataset was used for training while 760 dataset was used for testing the trained system.

The triangular membership function was adopted. The range of pause value for linguistic terms low is (200, 210, 220), normal is (210, 225, 240) while high is (230, 240, 250), respectively.

The rule editor enables the formation of rules for reasoning process. The total rule set formed is 81; this is evaluated from three (3) membership functions and four (4) attributes of the verbal cues. The rule analysis viewer displays the surface view of the cues. The rule analysis viewer displays the contribution of each cue to the result. The values of the cues can be adjusted at the rule viewer window and the corresponding value of the result will be displayed

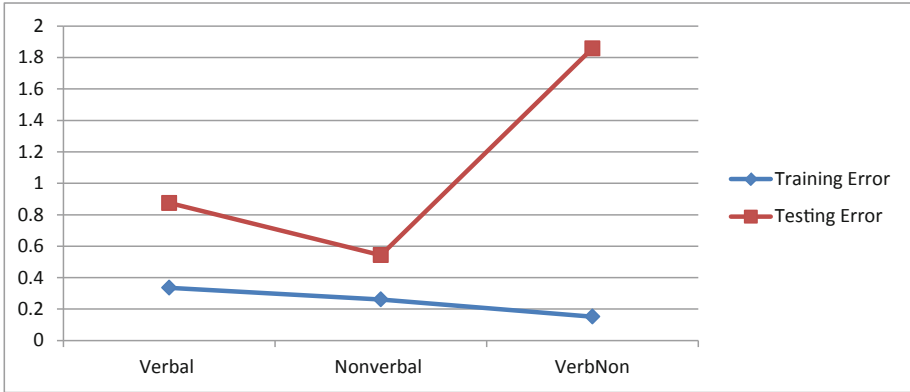
#### 4.2 Result Testing and Evaluation

The model was tested using dataset with known classification. Details of the analysis are shown in the graph shown in Fig. 3. Nonverbal dataset have reduced training error as well as reduce testing error.

Table 4 and Fig. 4 show the extracted confusion matrix for each of the datasets.

#### 4.3 Performance of Different Classifiers on the Verbnon Dataset

The different datasets were passed through different classifiers to ascertain the performance. For the Decision Tree, 847 data were correctly classified as truthful which



**Fig. 3.** Training versus testing error across datasets.

**Table 4.** Confusion matrix for verbal, nonverbal and VerbNon dataset

	Training	Validation	Test	All
Nonverbal (N)	97.1%	97.2%	97.2%	97.1%
Verbal (V)	84.4%	86.6%	81.9%	84.3%
VerbNon	92.7%	92.8%	91.6%	92.5%



**Fig. 4.** Confusion matrix for verbal, nonverbal and VerbNon dataset

corresponds to 97.8% while 26 were wrongly classified as deceptive corresponding to 4.9%. Also, 507 were correctly classified as deceptive representing 95.1% and 19 falsely classified as truthful representing 2.2%.

Likewise for K- Nearest Neighbour classifier, 833 data were correctly classified as truthful which corresponds to 96.2% while 17 were wrongly classified as deceptive corresponding to 3.2%. Also, 516 were correctly classified as deceptive representing 96.8% and 33 falsely classified as truthful representing 3.8%.

Also, for Support Vector Machine, 843 data were correctly classified as truthful which corresponds to 97.3% while 23 were wrongly classified as deceptive corresponding to 4.3%. 510 were correctly classified as deceptive representing 95.7% and 23 falsely classified as truthful representing 2.7%.

Table 5 shows the performance of various classifiers on each of the datasets while Fig. 5, 6 and 7 gives graphical representations of the performance. From the table, it is observed that Decision Tree performs better than the others (97.6%) using a combination of both cues. Using Nonverbal cues, Neurofuzzy performed better than the others with percentage score of 97.1%. KNN perform better with Verbal cues having percentage score of 90.9%.

**Table 5.** Comparative analysis of different classifiers on each dataset

	SVM	Decision tree	KNN	Neurofuzzy
<i>Verbal cues</i>				
Overall accuracy	89.2%	89.9%	90.9%	84.3%
Overall error	10.8%	10.1%	9.1%	15.7%
Total dataset used	1693	1693	1693	1693
<i>Nonverbal cues</i>				
Overall accuracy	91.9%	93.5%	96.4%	97.1%
Overall error	8.1%	6.5%	3.6%	2.9%
Total dataset used	5133	5133	5133	5133
<i>VerbNon cues</i>				
Overall accuracy	97.1%	97.6%	96.9%	92.5%
Overall error	2.9%	2.4%	3.1%	7.5%
Total dataset used	1353	1353	1353	1353

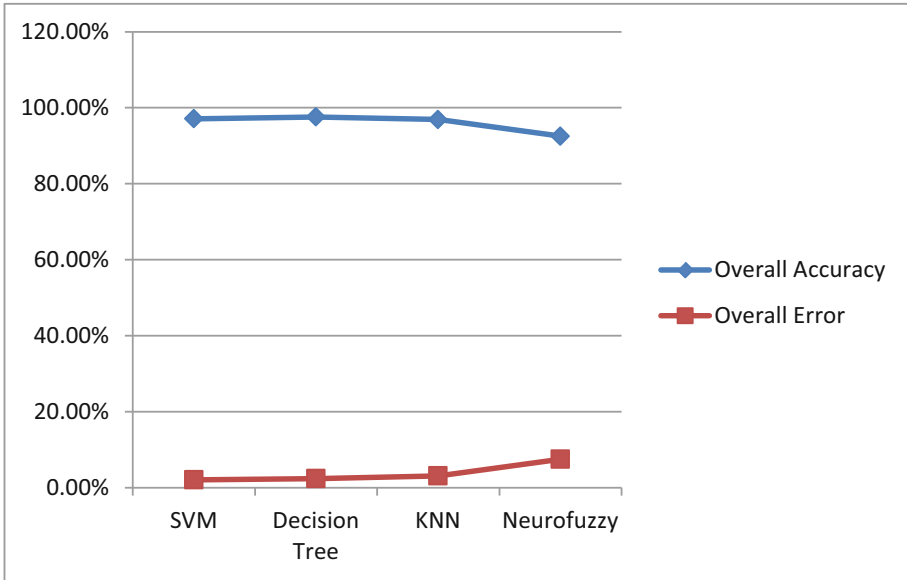


Fig. 5. Performance of classifiers on VerbNon dataset

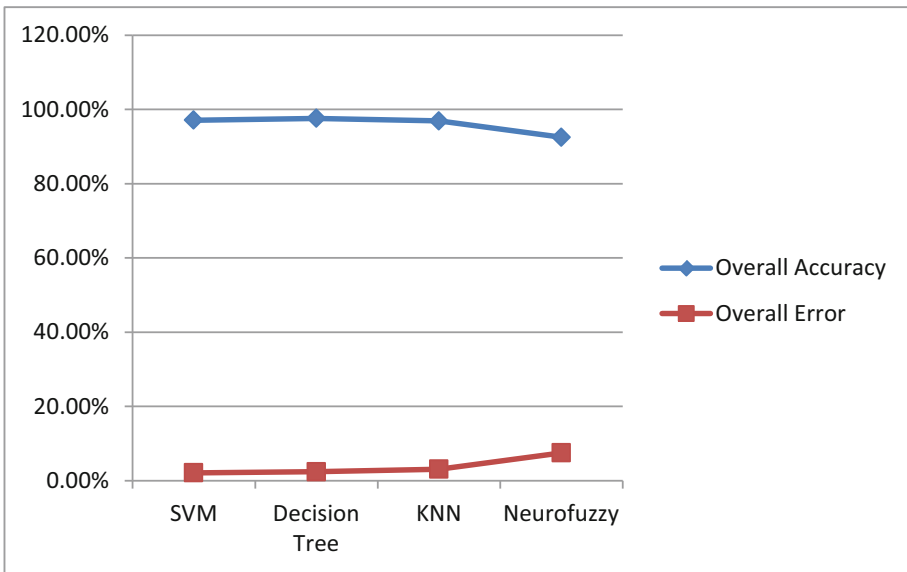
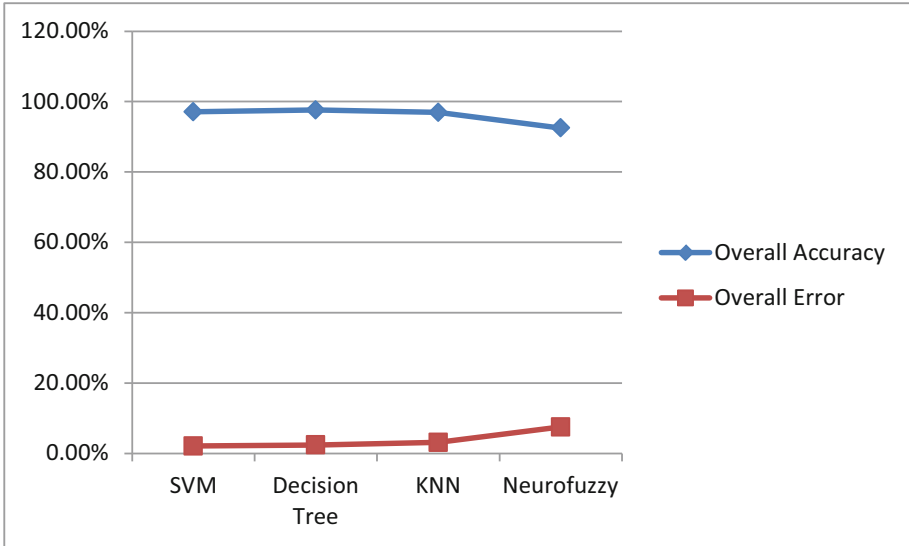


Fig. 6. Performance of classifiers on nonverbal dataset



**Fig. 7.** Performance of classifiers on verbal dataset

## 5 Conclusion

Deception detection is an involved social issue because to successfully deceive the deceiver has to formulate a story that is internally consistent while hiding emotions and true intentions. Facial expressions and voice play a critical role in the identification of deception as shown in this research. Previous research made use of only one cue but this research made use of both verbal and nonverbal cues. The developed system was able to perform better than chance and trained professionals with a result difference of 42.5%.

This work uses verbal, nonverbal cues and a combination of both cues to detect deception. The verbal cues were extracted using Praat while the nonverbal was extracted using Active Shape Model. The classification was done using Neurofuzzy model and the performance was compared with SVM, KNN and Decision Tree. Neurofuzzy recorded the best performance with the Nonverbal dataset, KNN performed best with the Verbal dataset while Decision Tree performed best with the VerbNon dataset.

The proposed system was implemented using Matlab 2015a on window 7 with 2 GB RAM. The extracted data was divided into training data and test data. The neurofuzzy model was trained using the training data while the functionality of the model was ascertained using the test data. At the end of the comparative analysis it was discovered that Neurofuzzy model work well on Nonverbal dataset to detect deception. The result obtained using only verbal cue was 84.3% while that of nonverbal cue was 97.1% but on VerbNon yielded 92.5% which is far better than the chance level of 50%.

## References

1. Navarro, J., Schafer, R.J.: Detecting deception. *FBI Law Enforcement Bull.* **70**, 9 (2001)

2. DePaulo, B.M., Lindsay, J.J., Malone, B.E., Muhlenbruck, L., Charlton, K., Cooper, H.: Cues to deception. *Psychol. Bull.* **129**, 74–112 (2003)
3. Knapp, M.L., Comadena, M.A.: Telling it like it isn't: a review of theory and research on deceptive communications. *Hum. Commun. Res.* **5**, 270–285 (1979)
4. Mitchell, R.W.: A framework for discussing deception. In: Mitchell, R.W., Thomson, N. (eds.) *Deception, Perspectives on Human and Nonhuman Deceit*, pp. 3–40. State University of New York Press, New York (1986)
5. Vrij, A.: *Detecting Lies and Deceit: The Psychology of Lying and its Implications for Professional Practice*. John Wiley, Chichester (2000)
6. Ekman, P.: *Telling Lies: Clues to Deceit in the Market Place, Politics, and Marriage*, 2<sup>nd</sup> ed. Norton, New York (1985/1992)
7. Buller, D.B., Burgoon, J.K.: Interpersonal deception theory. *Commun. Theory* **6**, 203–242 (1996)
8. Jaume, M., Eugenio, G., Carmen, H.: Defining deception. *J. Anales de Psicologia* **20**(1), 147–171 (2004)
9. Baron-Cohen, S., Wheelwright, S., Hill, J., Raste, Y., Plumb, I.: The “reading the mind in the eyes” test revised version: a study with normal adults, and adults with asperger syndrome or high-functioning autism. *J. Child Psychol. Psychiatry* **42**(2), 241–251 (2001)
10. Vrij, A.: *Detecting Lies and Deceit: Pitfalls and Opportunities*. Wiley, Chichester (2008)
11. Enos, F., Benus, S., Cautin, R.L., Graciarena, M., Hirschberg, J., Shriberg, E.: Personality factors in human deception detection: comparing human to machine performance. In: *Proceedings of the 9th International Conference on Spoken Language Processing, ISCA 2006, Pittsburgh, USA* (2006)
12. Reid J., and Associates.: *The Reid Technique of Interviewing and Interrogation*. John E. Reid and Associates, Inc., Chicago (2000)
13. Zhou, L., Twitchell, D.P., Qin, T., Burgoon, J.K., Nunamaker, J.F.: An exploratory study into deception detection in text-based computer-mediated communication. In: *Proceedings of the Thirty-Sixth Annual Hawaii International Conference on System Sciences (HICSS 2003), Big Island, Hawaii* (2003)
14. Gamson, R., Gottesman, J., Milan, Nicholas, Weerasuriya, S.: Cues to Catching Deception in Interviews, p. 2012. START, College Park (2012)
15. Rothwell, J., Bandar, Z., O'Shea, J., McLean, D.: Charting the behavioural state of a person using a backpropagation neural network. *J. Neural Comput. Appl.* **16**, 327–339 (2006). <https://doi.org/10.1007/s00521-006-0055-9>
16. Almela, A., Valencia-García, R., Cantos, P.: Seeing through deception: a computational approach to deceit detection in written communication. In: *Proceedings of the Workshop on Computational Approaches to Deception Detection, Avignon, France*, pp. 15–22 (2012)
17. Bachenko, J., Fitzpatrick, E., Schonwetter, M.: Verification and implementation of language based deception indicators in civil and criminal narratives. In: *Proceedings of the 22nd International Conference on Computational Linguistics, Manchester, U.K., 18–22 August 2008*, pp. 41–48 (2008)
18. Vrij, A., Fisher, R., Mann, S., Leal, S.: A cognitive load approach to lie detection. *J. Invest. Psychol. Offender Profiling* **5**(1–2), 39–43 (2008)
19. Perez-Rosas, V., Abouelenien, M., Mihalcea, R., Xiao, Y., Linton, C.J., Burzo, M.: Verbal and nonverbal clues for real-life deception detection. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015*, pp. 2336–2346. Association for Computational Linguistics (2015)



# A New Approach to Estimate the Discharge Coefficient in Sharp-Crested Rectangular Side Orifices Using Gene Expression Programming

Hossein Bonakdari<sup>1</sup>(✉), Bahram Gharabaghi<sup>2</sup>, Isa Ebtehaj<sup>3</sup>, and Ali Sharifi<sup>3</sup>

<sup>1</sup> Department of Soils and Agri-Food Engineering, Laval University,  
Quebec City, QC G1V0A6, Canada

hossein.bonakdari@fsaa.ulaval.ca

<sup>2</sup> School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada

<sup>3</sup> Environmental Research Center, Razi University, Kermanshah, Iran

**Abstract.** Structures, such as side orifices are used for controlling the flow within a diversion channel or for directing the flow into one. In this study, an equation for estimating discharge coefficient is introduced using “gene expression programming” (GEP). In order to estimate the discharge coefficient, four dimensionless parameters including ratio of depth of flow in main channel to the width of rectangular orifice ( $Y_m/L$ ), Froude number ( $F_r$ ), the ratio of sill height to the width of rectangular orifice ( $W/L$ ) and the ratio of the width of the main channel to the width of the rectangular orifice ( $B/L$ ) are used to present five different models. Therefore, the lacks of effect of each dimensionless parameter on the discharge coefficient predictions are reviewed. The results obtained from the carried out studies indicated that the best model presented in this study estimated the discharge coefficient fairly well with a relative error of 3% against experimental data.

**Keywords:** Discharge coefficient · Gene Expression Programming (GEP) · Soft computing · Side orifice

## 1 Introduction

Side orifices, side sluice gates and side weirs are diversion structures commonly installed on the side of the main channel to divert and control the flow into the diversion channel. This group of hydraulic structures is applied in open channels and can be used in wastewater treatment plants, land drainages, sedimentation tanks, aeration basins, irrigation systems, and flocculation units. The flow within main channels with diversion structures are gradually varied flows with decreasing discharge.

### 1.1 Related Works

Numerous researches and studies have been carried out on diversion structures by various researchers. Ramamurthy et al. [1] were among the first who experimentally investigated

the diversion flow within rectangular side orifices. They presented an equation as a function of the length of the orifice, the width of the main channel and the ratio of the mean velocity in the main channel to the orifice output velocity to calculate discharge coefficient of side orifices in the side of the rectangular channels. Also, different researchers such as Oliveto et al. [2], Ghodsian [3], Kra and Merkley [4], Amaral et al. [5], Lewis et al. [6] have conducted studies on the features of the flow passing through diversion structures. Gill [7] studies the gradually varied flows passing through open channels which have relatively short side rectangular orifice. Swamee et al. [8] calculated the equation of elementary coefficient of discharge of sluice gates as a function of the depth of the flow within the main channel, the openness of the gate and Froude number of the input flow. Ojha and Subbaiah [9] conducted studies relevant to the output flows from side orifices and they obtained discharge coefficient equation as a function of the geometrical features and the crest height of the orifice. Prohaska et al. [10] presented investigations on side orifices on the side of a pipe and the parameters influencing discharge coefficient of the side orifices. Hussain et al. [11] carried out experimental investigations on parameters affecting the volume of the discharge passing through sharp-crested circular side orifices. They obtained an equation as a function of Froude's number and the ratio of the orifice diameter to the width of the main channel in order to calculate discharge coefficient of circular side orifices. Hussain et al. [12] conducted an experimental study on the features of the flow within the main channel which has one single sharp-crested rectangular orifice. They also investigated parameters affecting flows passing through rectangular side orifices. Hussain et al. [12] presented an equation for calculating discharge coefficient of these types of side orifices as a function of Froude number and the ratio of the width of the rectangular side orifice to the width of the main channel.

Recently, the soft computing techniques such as artificial neural network [13–16]; adaptive neuro-fuzzy inference system [17, 18], genetic programming [19]; Minimax Probability Machine Regression [20]; Group Method of Data Handling [21–23] and Genetic algorithm, [24], are used as powerful instruments in modeling, and solving complicated and nonlinear problems. Ebtehaj et al. [25] utilized the Group Method of Data Handling (GMDH) to calculate the Cd. The authors found that the GMDH is a powerful method in the estimation of the discharge coefficient and present different equations to calculate this parameter. Among these artificial intelligence methods, Gene-Expression Programming (GEP) which is an extension to GP is considered popular as an identification system for the purposes of modeling and predicting unknown and complex behavior of different phenomenon. Ebtehaj et al. [26] develop a GEP-based equation with non-dimensional variables to prediction of discharge coefficient in rectangular side weirs. They found that GEP present satisfactory results relative to existing equation. By using GEP-based formulation technique and literature data, Azamathulla [27] predicted scour depth downstream of sills. The presented equation found to be valuable to prediction of scour depth for different bed slopes. Also, the comparison of existing and presented equations shows that the GEP superior than existing equation. Azamathulla [28] presented gene-expression programming as an alternative method to prediction of friction factor of southern Italian rivers. The presented GEP-based model shows satisfactory results in comparison with existing equation. Azamathulla and Ahmad [29] used GEP method to derive a new formulation for the prediction of transverse mixing coefficient in open channel flow. The results show that GEP present acceptable results with compared existing equation for transverse mixing coefficient. Guven and Azamathulla [30]



used GEP and field data to generate new model in prediction of scour downstream of a flip-bucket spillway. The comparison of presented model, convectional genetic programming and regression-based equation showed the better performance of GEP-based formula.

## 1.2 Research Objective

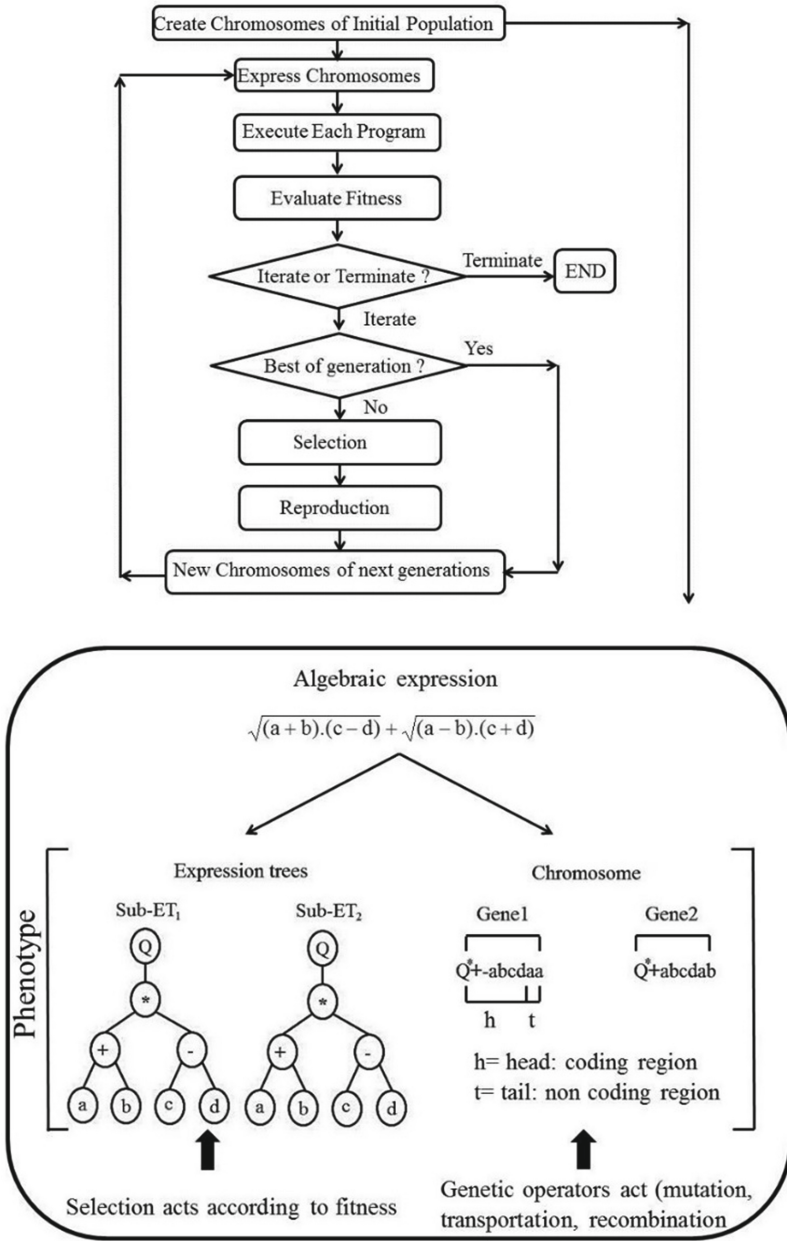
The main objective of the present study is to present an equation for estimating discharge coefficient within flow diversion structures of sharp-crested rectangular side orifices located on the main side of a rectangular channel in subcritical flow conditions through using gene expression programming (GEP). Therefore, in order to present a suitable model, the dimensionless parameters which affect discharge coefficient are determined and then five different models are presented for the purposes of evaluating the effect of each of the dimensionless parameters. Finally, the accuracy of the estimated discharge coefficient is studied by each of these models and the best model is selected.

The rest of the paper contains five sections including “Gene Expression Programming (GEP)”, “Data Collection”, “Methodology”, “Results and Discussion” and “Conclusions”. In the second section, the theoretical definition of the applied technique (GEP) were provided in detail. At the next one, the employed dataset to develop a new model for predicting the discharge coefficient of the side orifice is provided. After that, the applied methodology in this study including five different models and considered statistical indices to evaluate the performance of the developed model at section four. The results of the developed model are provided in section five using quantitative and qualitative tools. Finally, the most finding of the current study is presented in section six.

## 2 Gene Expression Programming (GEP)

Gene expression programming is a developed form of genetic programming [31]. Gene expression programming belongs to evolutionary algorithms family and is closely related to genetic algorithm and genetic programming. It has inherited linear chromosomes with fixed lengths from genetic algorithm and it has inherited tree analysis with varied lengths and shapes from genetic programming [32]. Gene expression programming presents computer programs such as mathematical models, decision trees, multi-sentence structures and logarithmic expressions or different types of models. These models have complex tree structure and training and conforming them according to their sizes, shapes and their combination is very much like that of a living thing while as living organisms, gene expression programming computer programs are also coded in simple linear chromosomes with the same length. Therefore, gene expression programming is a genotype-phenotype system which utilizes a simple genome to preserve and transfer genetic information and is a complex phenotype for discovering the environment and adapting to it.

In comparison to GP in which phenotype and genotype are combined in a simple replicator system, GEP is an evolved genotype-phenotype system in which genotype is usually completely separated from phenotype. Therefore, the evolved genotype/phenotype system in GEP causes superiority with a factor as large as 100 to 60000 times more than the GP system [32]. Figure 1 shows the schematics for GEP modeling process.



**Fig. 1.** General GEP structure

The gene expression programming firstly includes selecting the essential function for creating a model and then the terminal set is selected. In the following stage the present set of data are called upon to estimate the intended parameters and compare them with the real value. Then the chromosomes are produced in order to randomly present the initial population. In the following stage the program is run for the produced population through using the present chromosomes and the suitability of the target function is studied. In case the author reaches the finishing point of the program we will end it, otherwise the target function will be evaluated again using modified genetic operators and the new population. This process will continue to the point where the conditions for stopping the program are provided.

### 3 Data Collection

In order to estimate discharge coefficient of sharp-crested rectangular side orifices the laboratory results presented by Hussain et al. [12] were used in this research. They conducted their experiments in a channel which was 0.6 m depth, 0.5 m width and 9.15 m length. They used sluice gate to regulate the flow depth and they also installed a square orifice to the left side of the channel. The experiment Hussain et al. [12] carried out was subcritical conditions for three square-shaped orifices sizes 0.044, 0.089 and 0.133 m and the crest height of 0.05, 0.1 and 0.15 m. Three to four different discharges can be detected within the main channel for different states (orifices and different crest heights). The author used sluice gate in order to regulate different depths of the flow within the main channel. In the conducted experiments the velocity inside the main channel and the velocity of the flow near the side orifice located on the horizontal sheet passing through the central axis of the side orifice were measured using acoustic Doppler current profiler. The range of the collected data in this study is presented in Table 1.

**Table 1.** Range of Hussain et al. [12] data

Parameters	Unit	Range of data	
		Min	Max
$Q_m$	$m^3/s$	0.028	0.147
$Q$	$m^3/s$	0.001	0.029
$L$	m	0.044	0.1339
$Y_m$	m	0.154	0.59
$W$	m	0.05	0.2
$F_r$	–	0.05	0.48

## 4 Methodology

Based on the conducted researchers related to estimating discharge coefficient in side orifices, it could be generally said that the  $C_d$  parameters are dependent on the independent dimensionless parameters of the ratio of flow depth in the main channel to the rectangular orifice width ( $Y_m/L$ ), Froude number ( $F_r$ ), the ratio of the rectangular orifice width to the sill height ( $W/L$ ) and the ratio of the width of the main channel to the width of the rectangular orifice ( $B/L$ ). Therefore, using the presented dimensionless parameters five different models are presented as follows. Model 1 includes all four presented dimensionless parameters. Models 2 to 5 are presented to study the effect of not using each of these dimensionless parameters on the accuracy of discharge coefficient estimation. In these models the dimensionless parameters affecting discharge coefficient estimation are considered as three different parameters while in model 1 and utilized all presented dimensionless parameters.

$$\text{Model 1. } C_d = f\left(\frac{B}{L}, \frac{W}{L}, \frac{Y_m}{L}, F_r\right)$$

$$\text{Model 2. } C_d = f\left(\frac{B}{L}, \frac{W}{L}, \frac{Y_m}{L}\right)$$

$$\text{Model 3. } C_d = f\left(\frac{B}{L}, \frac{W}{L}, F_r\right)$$

$$\text{Model 4. } C_d = f\left(\frac{B}{L}, \frac{Y_m}{L}, F_r\right)$$

$$\text{Model 5. } C_d = f\left(\frac{W}{L}, \frac{Y_m}{L}, F_r\right)$$

It is necessary, in this study, to use criteria which are capable of estimating each of the models quantitatively in order to investigate the capability to estimate discharge coefficient ( $C_d$ ) through using the presented models. Usually, a model's performance is evaluated through using various statistical indexes known as "goodness of fit" statistics. Therefore, different measuring methods have been presented by Legates and McCabe [33] to measure the accuracy of estimating hydrologic and hydraulic models. However, there are numerous various ways to use global goodness of fit statistics for the purpose of investigating the accuracy of the estimations. Using a sole statistical index cannot be considered as a good enough criterion in evaluating estimation accuracy of a model [34]. Therefore, to study the accuracy of estimation of the presented models, multi-criteria evaluation is used in this research. The mentioned indexes are placed in two groups namely "relative" and "absolute" groups. The relative indexes which are dimensionless indicate the performance of one model in comparison to the others while absolute indexes present the accuracy of the estimation by using the measurement units of the intended parameter.

The statistical indexes used in this study include a dimensionless coefficient criteria called R-Squared ( $R^2$ ), the three relative measuring criteria of Mean Relative Error ( $MRE$ ), Mean Absolute Relative Error ( $MARE$ ) and Mean Squared Relative Error

(*MSRE*) and three absolute measuring criteria of Mean Error (*ME*), Mean Absolute Error (*MAE*) and Root Mean Squared Error (*RMSE*). The Mean Relative Error (*MRE*) index shows the average relative error of the estimated model in comparison to the observed values. This index does not have a high limit and its lowest limit is zero. This means that the more the value of this index in estimating the values versus the considered value approaches to zero, the higher is the model’s validity. The *MRE* index considers the difference between the estimated and the observed values by considering the low and high state of the estimated values in relation to the actual values. Therefore, if the model’s estimated value is an underestimation, *MRE* tends to be positive and if it is an overestimation, the value of this index will be negative. The Mean Absolute Relative Error (*MARE*) expresses the estimated value in relation to the observed value. *MARE* is a non-negative index which has no higher limit. The considered model has the best possible performance when the value of this index is zero. The Mean Square Relative error (*MSRE*) is the second degree of the mean of the squared relative in which the relative error of the estimated versus the observed values is calculated as overestimation or underestimation.

$$R^2 = \left[ \frac{\sum_{i=1}^n (y_i^o - \bar{y}_i^o)(y_i^c - \bar{y}_i^c)}{\sqrt{\sum_{i=1}^n (y_i^o - \bar{y}_i^o)^2 \sum_{i=1}^n (y_i^c - \bar{y}_i^c)^2}} \right]^2 \tag{1}$$

$$MRE = \left( \frac{1}{n} \right) \sum_{i=1}^n \left( \frac{y_i^o - y_i^c}{y_i^o} \right) \tag{2}$$

$$MARE = \left( \frac{1}{n} \right) \sum_{i=1}^n \left( \frac{|y_i^o - y_i^c|}{y_i^o} \right) \tag{3}$$

$$MSRE = \left( \frac{1}{n} \right) \sum_{i=1}^n \left( \frac{y_i^o - y_i^c}{y_i^o} \right)^2 \tag{4}$$

The Mean Error (*ME*) is an index without high or low limit, and when the value of this index equals zero it represents the best performance. Since this index presents the mean difference of the estimated and observed values by taking into account the effect of underestimation and overestimation, it cannot be said that the low value of this index signifies the good accuracy of the model. For this reason, another index called Mean Absolute Error (*MAE*) is applied. *MAE* is a non-negative index, which provides no information about the underestimation or overestimation of the estimates. The results of this index do not consider the effect of underestimation or overestimation of the parameters versus the observed values and evaluates the diversion from estimated values without considering the sign. The Root Mean Squared Error (*RMSE*) is a criterion of mean error, which has no upper limit and has a lowest possible value of zero representing the best estimation by the model.

$$ME = \left( \frac{1}{n} \right) \sum_{i=1}^n (y_i^o - y_i^c) \tag{5}$$

$$MAE = \left(\frac{1}{n}\right) \sum_{i=1}^n (|y_i^o - y_i^c|) \tag{6}$$

$$RMSE = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (y_i^o - y_i^c)^2} \tag{7}$$

where  $y_i^o$  parameter is the value of the parameter observed in laboratory results and  $y_i^c$  parameter is the value of the parameter estimated through using GEP. The indexes presented above present the estimated values as the average of the predicted error and do not present any sort of information on the predicted error distribution of the suggested models. It is obvious that a high correlation coefficient (80–90%) is not always considered as an indication of the high accuracy of a model; on the contrary, this index may lead to showing high accuracy for mediocre models [33, 35]. In addition, *RMSE* index indicates the model’s ability to predict a value away from the mean [36]. Therefore, the presented model must be evaluated using other indexes such as mean absolute relative error (*MARE*) and threshold statistics [37–39].  $TS_x$  index indicates predicted error distribution by each model for  $x\%$  of the predictions. This parameter is determined for various values of average absolute relative error. The value of the  $TS_x$  index for  $X\%$  of the predicted is determined as explained below:

$$TS_x = \frac{Y_x}{n} \times 100 \tag{8}$$

where  $Y_x$  is number of the predicted values of all the data for each value of *MARE* is less than  $x\%$ .

## 5 Results and Discussion

Taking into consideration the parameters affecting discharge coefficient in side orifices that led to presenting five different models, measures have been taken in this section to present different models using the gene expression programming (GEP) based on the laboratory data presented by Hussain et al. [12]. In order to present a model from amongst the existing data, only 80% (137 data) of the data was used to estimate the model and in order to investigate the accuracy of the model when using the data which were not used in model training, 20% (34 data) of the remaining data was used. For each of the presented models (1 through 5) each of which considers different factors affective on estimating discharge coefficient, different equations are presented as follows:

$$C_d = \left( 0.672 + F_r \times \left( \frac{\left( 2.17 \left( \frac{B}{L} \right) \right) - \left( \frac{B}{L} \right)}{\frac{F_r}{\frac{B}{L}}} \right)^{-1} \right) + \left( \frac{\left( \frac{B}{L} + 9.51 \right) \times \left( \frac{\log_2 \left( \frac{B}{L} \right)}{\log_2 \left( \frac{W}{L} \right)} \right)}{9.51 \frac{B}{L} + F_r \times \left( \frac{Y_m}{L} \right)^{10}} \right) + \left( \frac{2F_r + \frac{3.37}{\log_2 F_r}}{\left( \frac{B}{L} \right)^2 + 3.83F_r} \right) \tag{10}$$

$$C_d = \left( \frac{\frac{W}{L} - \frac{B}{L}}{\frac{W}{L} + 5.59} \right) / \left( \frac{W}{L} + 46.21 \right) + \left( 3.72 \left( 12.342 - \frac{W}{L} \right) \right)^{-1} + \left( \frac{\log_2 \left( 1.23 \frac{B}{L} \frac{Y_m}{L} \right)}{\left( \frac{B}{L} \frac{Y_m}{L} \left( \frac{B}{L} + 4.61 \right) \right)} \right) \tag{11}$$

$$C_d = \left( \frac{(-2.33)^{10} - \left(\frac{W}{L} + F_r\right)}{2\frac{W}{L} + 1.5 \times \frac{W}{L}} \right) + \left( -\frac{\frac{W}{L}}{7.82\frac{W}{L}\frac{B}{L}} \right)^{10} + \left( \frac{F_r^2 + F_r}{8.81\left(\frac{W}{L}\right)} \right) \quad (12)$$

$$C_d = \left( \log \left[ \text{Exp} \left( \text{Exp} \left( \left( F_r - 2\frac{B}{L} \right)^{10} \right) \right) \right] \right) + \left( \frac{\log_2 \left( \frac{Y_m}{L} \right)^{0.05}}{\log_2 \left( \frac{Y_m}{L} \right)^{0.25}} \right) + \left( \frac{Y_m}{L} + \frac{\log_2 \left( \frac{B}{L} + 9.9 \right)}{\log_2 \left( \text{Exp}(9.9) \right)} \right) \quad (13)$$

$$C_d = \left( -\frac{0.052\left(\frac{W}{L} - 0.01\right)}{\left(\frac{W}{L} + 0.01\right)\log_2\left(\frac{Y_m}{L}\right)} \right) + (2F_r) + \left( \frac{1.94}{\log_2^{F_r}} + F_r \left( 1 + (F_r) \left( \frac{Y_m}{L} \right) \right) \right) \quad (14)$$

After estimating the discharge coefficient and using the following equation, the discharge for each of the presented models is presented through using the equation presented by Ojha and Subbaiah [9] as follows:

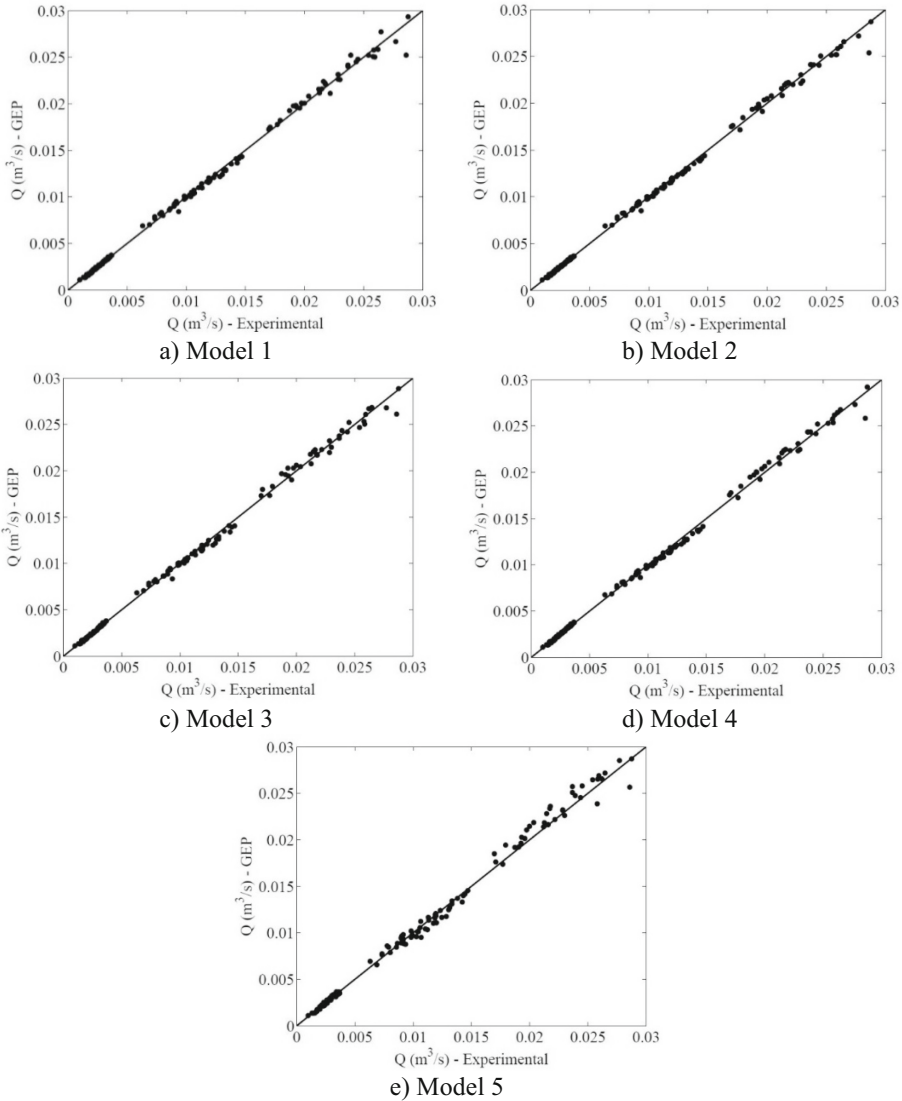
$$Q = C_d L b \sqrt{2gH_0} \quad (15)$$

where  $C_d$  is the discharge coefficient,  $F_r$  Froude number,  $W$  sill height,  $Y_m$  the flow depth in the main channel,  $B$  the width of the main channel,  $L$  the width of the side orifice,  $b$  the height of the side orifice,  $H_0$  water head above the central line of the orifice and  $g$  the gravitational acceleration.

Figure 2(a–e) show the estimated discharge values through using discharge coefficient equations presented for each of the five models against the obtained laboratory results for Train data. Considering the figures presented for different models, it could be concluded that almost all the models provide good results. Model 5 estimates discharges more than 0.02 m<sup>3</sup>/s with less accuracy compared to the four other models, but it should be noted that the differences in the estimated values are not significant in comparison with the real values.

Model 1 which the parameters influencing the estimation of the discharge coefficient needed for calculating the discharge in a dimensionless manner to be the ratio of depth of flow in the main channel to the width of the rectangular orifice ( $Y_m/L$ ), Froude number ( $F_r$ ), the ratio of sill height to the width of the rectangular orifice ( $W/L$ ) and the ratio of the width of the main channel to the width of the rectangular orifice ( $B/L$ ). Table 2 shows that with  $R^2 = 0.997$ , the model presented by using GEP has very well estimated the *value of discharge* coefficient essential for estimating discharge; that is, the largest value of relative error made by this model is approximately 10%. However, considering Fig. 3, it could be seen that 85% of the data have an error less than 5%.

It could be observed in Table 2 that the average relative error of model 1 is approximately 2.7% which indicates the high accuracy of this model in estimating the value of discharge coefficient. *MRE* index which considers the differences in the values estimated and observed by taking into account the estimated value being more or less than the real value, is nearly 0.002 for model 1 which means that the difference average of the estimated values and the real value is little. *MAE* index which presents the difference from the estimated values without considering larger or smaller estimation, is approximately 0.00026 which is a small number and indicates that the estimated value does not significantly differ from the real value in average. Taking into consideration the presented



**Fig. 2.** Comparing discharge for the state in which different coefficients obtained from models 1 through 5 are used with laboratory discharges (Train)

explanations and other indexes presented in Table 2 it can be seen that this model is fairly accurate.

In model 2, in order to estimate the discharge coefficient needed for calculating the flow discharge, the dimensionless parameters of the ratio of the depth of flow in the main channel to the width of the rectangular orifice ( $Y_m/L$ ), the ratio of sill height to the width of the rectangular orifice ( $W/L$ ) and the ratio of the width of the main channel to the width of the rectangular orifice ( $B/L$ ) were used. Taking into consideration the



fact that in comparison with model 1, model 2 does not make use of the Froude number but as it could be observed in Fig. 2(b) and Table 2, the presented estimations by this model with  $R^2 = 0.998$  are fairly accurate. Considering Fig. 3 it could be seen that 90% of the data have a relative error smaller than 5%. In order to investigate the accuracy of model 2, various statistical indexes which are presented in Table 2 are used. This table shows that the presented relative error average by model 2 is approximately 2.5%. Also, considering the fact that as the presented relative indexes (*ME*, *MAE* & *RMSE*) near zero the presented model is more accurate, the high accuracy of model 2 for estimating discharge coefficient is approved.

Model 3 presents the discharge coefficient using dimensionless parameters of Froude number ( $F_r$ ), the ratio of the sill height to the width of the orifice ( $W/L$ ) and the ratio of the width of the main channel to the width of the rectangular orifice ( $B/L$ ). Considering Fig. 2 and the results presented in Table 2 this model is fairly accurate in estimating the discharge coefficient needed for estimating the discharge. The relative error average presented by this model is approximately 2.8%. Also, considering Fig. 3 it could be observed that 85% of the estimated data by this model have been estimated with the relative error less than 5%. Observing the presented explanations, it could be said that model 3 which, except for the ratio of the depth of flow in the main channel to the width of the rectangular orifice parameter ( $Y_m/L$ ), considers all parameters of model 1 in estimating the discharge coefficient, has presented results similar to that of model 1. Therefore, it could be stated that not using  $Y_m/L$  parameter does not have a significant effect on estimating discharge coefficient.

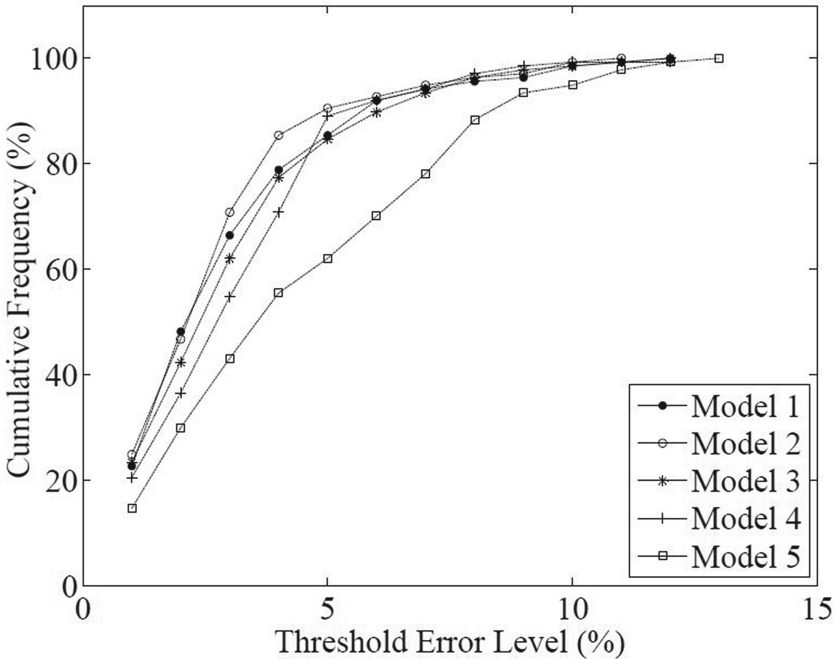
Model 4 considers dimensionless parameters of the ratio of the depth of flow in the main channel to the width of the rectangular orifice ( $Y_m/L$ ), Froude number ( $F_r$ ), and the ratio of the width of the main channel to the width of the rectangular orifice ( $B/L$ ) and model 5 considers dimensionless parameters of the ratio of the depth of flow in the main channel to the width of the rectangular orifice ( $Y_m/L$ ), Froude number ( $F_r$ ) and the ratio of sill height to the width of rectangular orifice ( $W/L$ ) in discharge coefficient. Like models 1, 2 and 3 these two models (4 and 5) estimate the results fairly well as the presented relative error average for these two models are equal to 3% and 4.2%, respectively. As Fig. 3 shows it approximately 90% of the data in model 4 and approximately 65% of the data in model 5 present the results with a less than 5% relative error. Therefore, it could be observed that not using the ratio of sill height to the width of rectangular orifice ( $W/L$ ) parameter, as opposed to model 1, will not have a significant effect on estimating discharge coefficient while not considering the ratio of the width of the main channel to the width of the rectangular orifice ( $B/L$ ) parameter leads to an increase in estimation relative error. As it could be seen in Fig. 3 only 65% of the data have an estimation relative error less than 5% while for the other models this parameter is approximately 85% to 90%.

Therefore, considering the presented explanations it could be stated that presenting models which use dimensionless parameters of the ratio of depth of flow in the main channel ( $W/L$ ) and the ratio of the width of the main channel to the width of rectangular orifice ( $B/L$ ) (model 2) provides the best results in comparison with other models; however, other models present good results as well and models 1, 3, and 4 also present results with a relatively small error in comparison with model 2. Compared to other

states, not using the dimensionless parameter of width of the main channel to the width of rectangular orifice ( $B/L$ ), leads to a decrease in the accuracy of estimation.

**Table 2.** Evaluation of the models proposed by GEP using different validation criteria (Train)

Train	Model 1	Model 2	Model 3	Model 4	Model 5
$R^2$	0.997	<u>0.998</u>	0.997	0.997	0.994
$MRE$	0.002	<u>0.001</u>	0.002	<u>0.001</u>	-0.003
$MARE$	0.027	<u>0.025</u>	0.028	0.030	0.042
$MSRE$	<u>0.001</u>	<u>0.001</u>	<u>0.001</u>	<u>0.001</u>	0.003
$ME$	<u>0.00001</u>	0.00003	0.00003	<u>0.00001</u>	<u>-0.00010</u>
$MAE$	0.00026	<u>0.00024</u>	0.00027	0.00029	0.00043
$RMSE$	0.00045	<u>0.00040</u>	0.00042	0.00043	0.00066

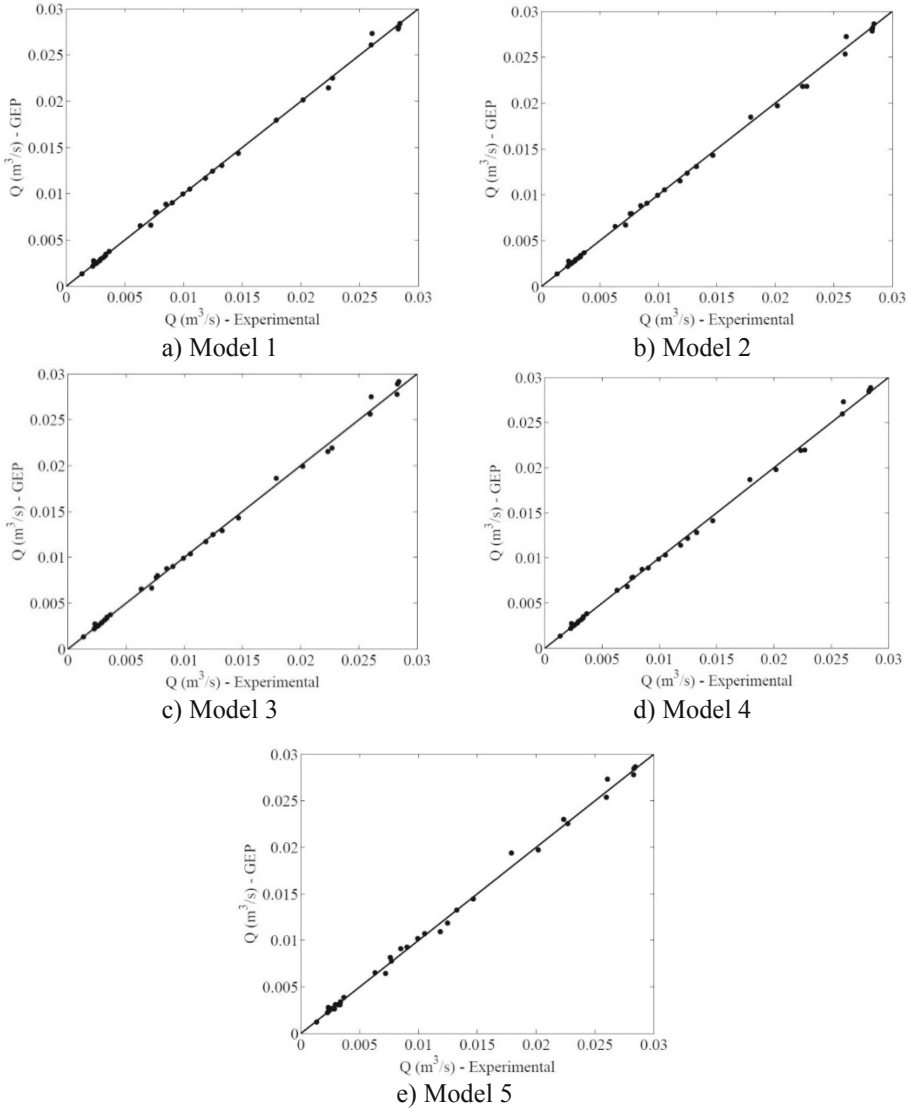


**Fig. 3.** Error distribution of GEP for all models (train)

Figure 4 shows the estimated discharges which were obtained through the presented discharge coefficient equations for each of the 5 models against the laboratory results obtained from the Test data which had no role in estimating the model. Considering the presented figures for different models almost all models present fairly good results. Model 5 estimated the results bigger than the real value in discharges more than  $0.02 \text{ m}^3/\text{s}$

which indicates the low accuracy of this model in estimating in comparison to models 1 through 4, but in case that  $R^2 = 0.997$  and the relative error is approximately 4%, according to Table 3, this model presents fairly good results.

The relative error average presented for models 1 to 4 is approximately 3% and for model 5 is equal to 4.2% (Table 3). Also, it could be seen that the model for all presented indexes presents better result in comparison with the rest of the models, although models



**Fig. 4.** Comparing the discharge when using different coefficients obtained from models 1 to 5 with laboratory discharge (Test)

2, 3, and 4 also present results with accuracy similar to that of model 1. In addition, based on Fig. 5 which presents the distribution of the estimated values by using models 1 to 5 for different relative error percentages, for models 1 to 4 almost 95% of the present data results in a relative error smaller than 6% while for model 5 this value is equal to 74%. The discharge coefficient values estimated through using the 5 models for the test data are shown in Table 4.

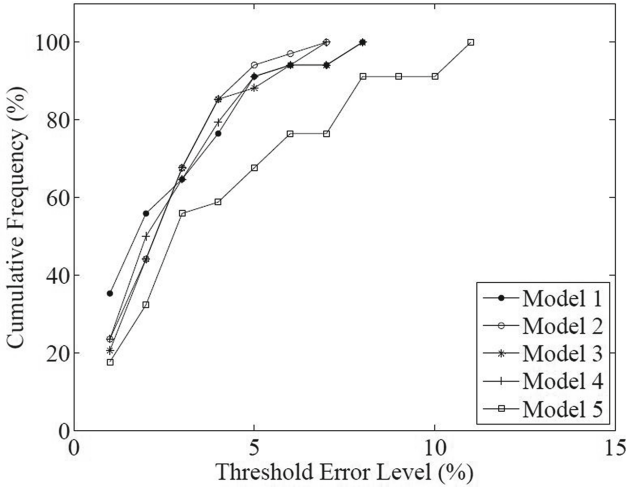


Fig. 5. Error distribution of GEP for all models (Test)

Table 3. Evaluation of the models proposed by GEP using different validation criteria (Test)

Test	Model 1	Model 2	Model 3	Model 4	Model 5
$R^2$	0.998	0.998	0.998	0.998	0.997
$MRE$	-0.004	-0.004	-0.007	-0.006	-0.008
$MARE$	0.027	0.028	0.029	0.028	0.042
$MSRE$	0.002	0.002	0.002	0.002	0.003
$ME$	0.00000	0.00003	-0.00003	-0.00001	-0.00007
$MAE$	0.00021	0.00026	0.00029	0.00026	0.00035
$RMSE$	0.00034	0.00037	0.00043	0.00037	0.00050

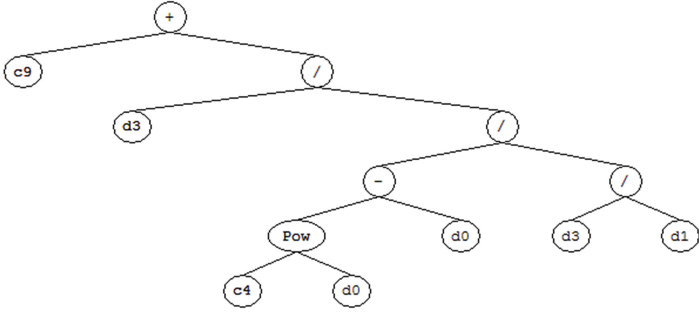
Based on the presented explanations, using the dimensionless parameters of the ratio of the depth of flow in the main channel to the width of rectangular orifice ( $Y_m/L$ ), Froude number, the ratio of sill height to the width of rectangular orifice ( $W/L$ ) and the ratio of the width of the main channel to the width of rectangular orifice ( $B/L$ ) (model 1) to estimate the discharge coefficient provides the best results in comparison with other models. Not using the dimensionless parameter of width of the main channel to the

**Table 4.** The values of estimated discharge coefficient using the proposed models for data unused in the estimation of the models (Test data)

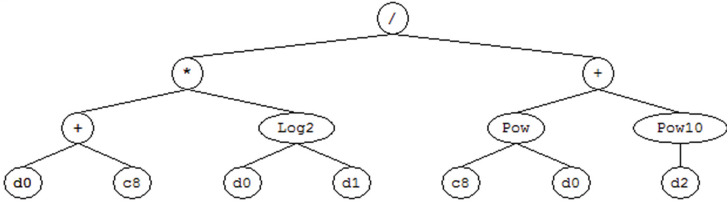
<i>B/L</i>	<i>W/L</i>	<i>Ym/L</i>	<i>Fr</i>	<i>C<sub>d</sub></i> - EXP	<i>C<sub>d</sub></i> - Model 1	<i>C<sub>d</sub></i> - Model 2	<i>C<sub>d</sub></i> - Model 3	<i>C<sub>d</sub></i> - Model 4	<i>C<sub>d</sub></i> - Model 5
11.364	1.136	11.511	0.187	0.647	0.664	0.652	0.662	0.676	0.685
11.364	1.136	9.952	0.132	0.648	0.665	0.655	0.670	0.674	0.653
11.364	1.136	9.177	0.070	0.674	0.667	0.656	0.679	0.673	0.623
11.364	2.273	8.836	0.280	0.664	0.662	0.664	0.657	0.673	0.701
11.364	2.273	7.207	0.114	0.628	0.666	0.667	0.676	0.670	0.632
11.364	2.273	6.791	0.255	0.660	0.663	0.667	0.660	0.669	0.687
11.364	3.409	8.543	0.305	0.679	0.662	0.670	0.664	0.673	0.698
11.364	3.409	5.180	0.419	0.661	0.658	0.675	0.655	0.664	0.608
11.364	3.409	10.611	0.143	0.676	0.665	0.666	0.678	0.675	0.661
11.364	3.409	9.202	0.104	0.689	0.666	0.669	0.681	0.673	0.637
11.364	4.545	9.268	0.286	0.684	0.662	0.672	0.671	0.674	0.703
11.364	4.545	8.309	0.218	0.695	0.664	0.674	0.676	0.672	0.686
11.364	4.545	10.986	0.080	0.531	0.628	0.632	0.627	0.625	0.636
5.618	0.562	5.796	0.182	0.643	0.631	0.628	0.628	0.620	0.634
5.618	0.562	2.824	0.435	0.611	0.638	0.634	0.632	0.628	0.655
5.618	1.124	3.051	0.378	0.608	0.635	0.636	0.625	0.626	0.654
5.618	1.124	3.036	0.298	0.619	0.645	0.639	0.644	0.632	0.625
5.618	1.124	3.828	0.351	0.639	0.642	0.640	0.637	0.635	0.656
5.618	1.124	4.583	0.116	0.658	0.648	0.639	0.650	0.633	0.607
5.618	1.685	4.684	0.266	0.636	0.635	0.637	0.627	0.624	0.648
5.618	1.685	6.037	0.122	0.645	0.635	0.637	0.628	0.624	0.645
5.618	1.685	5.578	0.130	0.646	0.645	0.641	0.647	0.631	0.615
5.618	2.247	4.572	0.113	0.638	0.638	0.642	0.637	0.629	0.656
5.618	2.247	3.692	0.262	0.620	0.642	0.643	0.643	0.631	0.641
5.618	2.247	3.919	0.277	0.635	0.582	0.592	0.587	0.602	0.569
3.759	0.376	3.624	0.251	0.597	0.587	0.588	0.586	0.600	0.587
3.759	0.376	3.191	0.250	0.597	0.600	0.583	0.589	0.597	0.584
3.759	0.376	3.633	0.124	0.596	0.590	0.594	0.609	0.602	0.600
3.759	0.752	2.417	0.378	0.580	0.582	0.599	0.603	0.605	0.629
3.759	0.752	4.002	0.177	0.600	0.599	0.604	0.615	0.609	0.605
3.759	1.128	2.905	0.224	0.625	0.624	0.610	0.617	0.613	0.611
3.759	1.128	3.218	0.264	0.629	0.624	0.606	0.608	0.609	0.626
3.759	1.128	3.218	0.335	0.620	0.595	0.605	0.597	0.608	0.638
3.759	1.504	4.436	0.179	0.585	0.613	0.611	0.617	0.612	0.613

width of rectangular orifice ( $B/L$ ) has the most effect on estimating discharge coefficient in comparison with the other parameters and it leads to significant decrease in estimation accuracy in comparison with other models. The expression tree of this model is presented in Fig. 6 and Table 5. All applied parameters in GEP model are elaborately expressed in Table 6.

Sub-ET 1



Sub-ET 2



Sub-ET 3

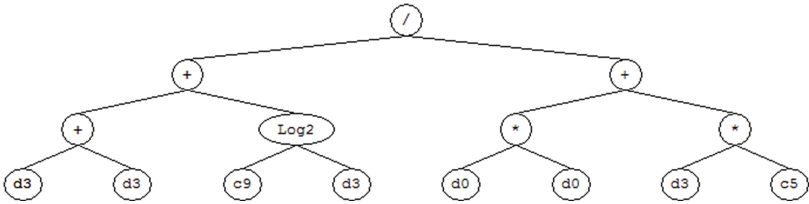


Fig. 6. Expression tree for GEP formulation (Model 1)

**Table 5.** Values of parameters presented in Fig. 6

G1C9	G1C8	G2C8	G2C9	G3C5	d0	d1	d2	d3
0.67	2.17	9.51	10.32	3.83	$B/L$	$W/L$	$Y_m/L$	$F_r$

**Table 6.** Parameters of GEP model

Parameter	Setting
Number of generations	400000
Number of chromosomes	30
Number of genes	3
Mutation rate	0.045
Inversion rate	0.1
One point recombination rate	0.15
Two point recombination rate	0.15
Gene recombination rate	0.2
Gene transportation rate	0.2
Function set	$\times$ , $+$ , $/$ , Pow, Pow10, Log2
Linking function	Addition

Table 7 indicates the results of the current study in comparison with existing regression, computational fluid dynamic (CFD) and AI based techniques. The existing AI-based techniques are as feedforward backpropagation (FFBP), radial basis function (RBF), generalized regression neural network (GRNN), CFD, adaptive neuro fuzzy inference systems (ANFIS) and their hybrids with genetic algorithm (GA), hybrid of particle swarm optimization with GA (PSOGA).

**Table 7.** Comparison of the developed GEP based model versus existing ones

Reference	Model	$MARE$ (%)	$RMSE$ (%)	R
Eghbalzadeh et al. [40]	FFBP	1.418	1.21	0.9369
	RBF	1.291	1.19	0.9418
	GRNN	1.479	1.36	0.9213
	Regression	4.006	3.24	0.443
Azimi et al. [41]	CFD	11.717	9.3	0.551
	ANFIS	9.19	0.8	0.95
	ANFIS-GA	2.44	0.2	0.996
Azimi et al. [42]	ANFIS-PSOGA	1.8	1.67	0.856
Current study	GEP	2.7	0.034	0.999

Although the ANFIS-PSOGA method has the lowest relative error, its difference with the GEP is less than 1%. In addition, the performance of the GEP method over other methods shows that the highest correlation coefficient and the lowest RMSE are related to this method. It should be noted that the GEP method, by providing a definite relation, resolves the problem of existing methods that did not provide explicit relation at practical tasks.

## 6 Conclusions

Gene expression programming (GEP) was used in this study to present an equation for estimating discharge coefficient in sharp-crested rectangular side orifice flow diversion structure located on the side of a rectangular channel subcritical flow conditions. The factors affecting discharge coefficient were presented through introducing four dimensionless parameters namely the ratio of the depth of flow in the main channel to the width of rectangular orifice ( $Y_m/L$ ), Froude number ( $F_r$ ), ratio of sill height to the width of rectangular orifice ( $W/L$ ) and the ratio of the width of the main channel to the width of rectangular orifice ( $B/L$ ). Five different models were presented that use these parameters in order to analyze the sensitivity of each of the presented dimensionless parameters. The results of the investigations indicate that not using the ratio of the width of the main channel to the width of rectangular orifice ( $B/L$ ) dimensionless parameter affects the accuracy of estimating discharge coefficient as it leads to a 2% increase in the average of the relative error. The investigations indicated that discharge coefficient can be estimated well by using models 1 to 4. However, in case we want to choose a model from amongst the presented models, model 1 is suggested for estimating discharge coefficient. Model 1 considers  $F_r$ ,  $Y_m/L$ ,  $B/L$  and  $W/L$  parameters to be affective on discharge coefficient estimation. Considering the given explanations, GEP is suggested to be used as an efficient method in estimating discharge coefficient in rectangular side orifice. For future work it is recommended to apply new developed group method of data handling techniques (GMDH) with generalized structure and compared with the results of the current study in term of accuracy and complexity.

## References

1. Ramamurthy, A.S., Tim, U.S., Sarraf, S.: Rectangular lateral orifices in open channels. *J. Environ. Eng.* **112**(2), 292–300 (1986)
2. Oliveto, G., Biggiero, V., Hager, W.H.: Bottom outlet for sewers. *J. Irrig. Drainage Eng.* **123**(4), 246–252 (1997)
3. Ghodsian, M.: Flow through side sluice gate. *J. Irrig. Drainage Eng.* **129**(6), 458–463 (2003)
4. Kra, E.Y., Merkley, G.P.: Mathematical modeling of open-channel velocity profiles for float method calibration. *Agric. Water Manag.* **70**(3), 229–244 (2004)
5. Amaral, L.G., Righes, A.A., Filho, P.S.S., Costa, R.D.: Automatic regulator for channel flow control on flooded rice. *Agric. Water Manag.* **75**(3), 184–193 (2005)
6. Lewis, J.W., Wright, S.J., Pribak, M., Sherrill, J.: Bottom slot discharge outlet for combined sewer diversion structure. *J. Hydraul. Eng.* **137**(2), 248–253 (2010)
7. Gill, M.A.: Flow through side slots. *J. Environ. Eng.* **113**(5), 1047–1057 (1987)



8. Swamee, P.K., Pathak, S.K., Ali, M.S.: Weir orifice units for uniform flow distribution. *ASCE J. Irrig. Drainage Eng.* **119**(6), 1026–1035 (1993)
9. Ojha, C.S.P., Subbaiah, D.: Analysis of flow through lateral slot. *J. Irrig. Drainage Eng.* **123**(5), 402–405 (1997)
10. Prohaska, P.D., Khan, A.A., Kaye, N.B.: Investigation of flow through orifices in riser pipes. *J. Irrig. Drainage Eng.* **136**(5), 340–347 (2010)
11. Hussain, A., Ahmad, Z., Asawa, G.L.: Discharge characteristics of sharp-crested circular side orifices in open channels. *Flow Meas. Instrum.* **21**(3), 418–424 (2010)
12. Hussain, A., Ahmad, Z., Asawa, G.L.: Flow through sharp-crested rectangular side orifices under free flow condition in open channels. *Agric. Water Manag.* **98**(10), 1536–1544 (2011)
13. Ebtehaj, I., Bonakdari, H.: Evaluation of sediment transport in sewer using artificial neural network. *Eng. Appl. Comput. Fluid Mech.* **7**(3), 382–392 (2013)
14. Ebtehaj, I., Bonakdari, H., Zaji, A.H.: A new hybrid decision tree method based on two artificial neural networks for predicting sediment transport in clean pipes. *Alexandria Eng. J.* **57**(3), 1783–1795 (2018)
15. Ebtehaj, I., Bonakdari, H., Zaji, A.H.: An expert system with radial basis function neural network based on decision trees for predicting sediment transport in sewers. *Water Sci. Technol.* **74**(1), 176–183 (2016)
16. Ebtehaj, I., Bonakdari, H., Zaji, A.H., Bong, C.H.J., Ab Ghani, A.: Design of a new hybrid artificial neural network method based on decision trees for calculating the Froude number in rigid rectangular channels. *J. Hydrol. Hydromechanics* **64**(3), 252–260 (2016)
17. Ebtehaj, I., Bonakdari, H.: Performance evaluation of adaptive neural fuzzy inference system for sediment transport in sewers. *Water Resour. Manage* **28**(13), 4765–4779 (2014)
18. Khoshbin, F., Bonakdari, H., Ashraf Talesh, S.H., Ebtehaj, I., Zaji, A.H., Azimi, H.: Adaptive neuro-fuzzy inference system multi-objective optimization using the genetic algorithm/singular value decomposition method for modelling the discharge coefficient in rectangular sharp-crested side weirs. *Eng. Optim.* **48**(6), 933–948 (2016)
19. Sharifipour, M., Bonakdari, H., Zaji, A.H.: Comparison of genetic programming and radial basis function neural network for open-channel junction velocity field prediction. *Neural Comput. Appl.* **30**(3), 855–864 (2016)
20. Bonakdari, H., Ebtehaj, I., Samui, P., Gharabaghi, B.: Lake water-level fluctuations forecasting using minimax probability machine regression, relevance vector machine, gaussian process regression, and extreme learning machine. *Water Resour. Manage* **33**(11), 3965–3984 (2019)
21. Shaghghi, S., Bonakdari, H., Gholami, A., Ebtehaj, I., Zeinolabedini, M.: Comparative analysis of GMDH neural network based on genetic algorithm and particle swarm optimization in stable channel design. *Appl. Math. Comput.* **313**, 271–286 (2017)
22. Bonakdari, H., Ebtehaj, I., Gharabaghi, B., Vafaieifard, M., Akhbari, A.: Calculating the energy consumption of electrocoagulation using a generalized structure group method of data handling integrated with a genetic algorithm and singular value decomposition. *Clean Technol. Environ. Policy* **21**(2), 379–393 (2018)
23. Ebtehaj, I., Bonakdari, H., Zaji, A.H., Azimi, H., Khoshbin, F.: GMDH-type neural network approach for modeling the discharge coefficient of rectangular sharp-crested side weirs. *Eng. Sci. Technol. Int. J.* **18**(4), 746–757 (2015)
24. Ebtehaj, I., Bonakdari, H.: Comparison of genetic algorithm and imperialist competitive algorithms in predicting bed load transport in clean pipe. *Water Sci. Technol.* **70**(10), 1695–1701 (2014)
25. Ebtehaj, I., Bonakdari, H., Khoshbin, F., Azimi, H.: Pareto genetic design of group method of data handling type neural network for prediction discharge coefficient in rectangular side orifices. *Flow Meas. Instrum.* **41**, 67–74 (2015)

26. Ebtehaj, I., Bonakdari, H., Zaji, A.H., Azimi, H., Sharifi, A.: Gene expression programming to predict the discharge coefficient in rectangular side weirs. *Appl. Soft Comput.* **35**, 618–628 (2015)
27. Azamathulla, H.M.: Gene expression programming for prediction of scour depth downstream of sills. *J. Hydrol.* **460**, 156–159 (2012)
28. Azamathulla, H.M.: Gene-expression programming to predict friction factor for Southern Italian rivers. *Neural Comput. Appl.* **23**(5), 1421–1426 (2013)
29. Azamathulla, H.M., Ahmad, Z.: Gene-expression programming for transverse mixing coefficient. *J. Hydrol.* **434**, 142–148 (2012)
30. Guven, A., Azamathulla, H.M.: Gene-expression programming for flip-bucket spillway scour. *Water Sci. Technol.* **65**(11), 1982–1987 (2012)
31. Koza, J.R.: *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, vol. 1. MIT Press, Cambridge (1992)
32. Ferreira, C.: Gene expression programming: a new adaptive algorithm for solving problems. *Complex Syst.* **13**(2), 87–129 (2001)
33. Legates, D.R., McCabe Jr., G.J.: Evaluating the use of “goodness-of-fit” measures in hydrologic and hydroclimatic model validation. *Water Resour. Res.* **35**(1), 233–241 (1999)
34. Sudheer, K.P., Jain, S.K.: Radial basis function neural network for modeling rating curves. *J. Hydrol. Eng.* **8**(3), 161–164 (2003)
35. Garrick, M., Cunnane, C., Nash, J.E.: A criterion of efficiency for rainfall-runoff models. *J. Hydrol.* **36**(3–4), 375–381 (1978)
36. Hsu, K.L., Gupta, H.V., Sorooshian, S.: Artificial neural network modeling of the rainfall-runoff process. *Water Resour. Res.* **31**(10), 2517–2530 (1995)
37. Jain, A., Varshney, A.K., Joshi, U.C.: Short-term water demand forecast modelling at IIT Kanpur using artificial neural networks. *Water Resour. Manag.* **15**(5), 299–321 (2001)
38. Jain, A., Ormsbee, L.E.: Short-term water demand forecast modeling techniques—conventional methods versus AI. *J. Am. Water Works Assoc.* **94**(7), 64–72 (2002)
39. Rajurkar, M.P., Kothiyari, U.C., Chaube, U.C.: Modeling of the daily rainfall-runoff relationship with artificial neural network. *J. Hydrol.* **285**(1–4), 96–113 (2004)
40. Eghbalzadeh, A., Javan, M., Hayati, M., Amini, A.: Discharge prediction of circular and rectangular side orifices using artificial neural networks. *KSCE J. Civ. Eng.* **20**(2), 990–996 (2016)
41. Azimi, H., Shabanlou, S., Ebtehaj, I., Bonakdari, H., Kardar, S.: Combination of computational fluid dynamics, adaptive neuro-fuzzy inference system, and genetic algorithm for predicting discharge coefficient of rectangular side orifices. *J. Irrig. Drainage Eng.* **143**(7), 04017015 (2017)
42. Azimi, A.H., Rajabi, A., Shabanlu, S.: Optimized ANFIS-Genetic algorithm-particle swarm optimization model for estimation of side orifices discharge coefficient. *J. Numer. Methods Civ. Eng.* **2**(4), 27–38 (2018)



# DiaTTroD: A Logical Agent Diagnostic Test for Tropical Diseases

Sandra Mae W. Famador<sup>1</sup>(✉) and Tardi Tjahjadi<sup>2</sup>

<sup>1</sup> Department of Computer Science, College of Science,  
University of Philippines Cebu, Cebu, Philippines  
upcebusmf@gmail.com

<sup>2</sup> School of Engineering, University of Warwick, Coventry, UK

**Abstract.** Medical diagnosis is one of the critical areas in medicine. Diagnosing tropical diseases can be confusing if their signs and symptoms are similar. This paper presents a formal logic for constructing a diagnostic test capable of guiding patient examination. A logical agent based on morphological data is used to aid diagnostic procedures and decision-making. To ensure that the appropriate diagnostic procedure is undertaken, the signs and symptoms of the patient are examined first before deciding which exact laboratory examination is needed by the patient. The logical agent perceives the signs, symptoms, medical history, and environment of the patient. Its actuation includes request for laboratory examination. A test kit result can be used, if available, to further confirm the diagnosis. The decision is not based on statistical inference but on logical analysis of the perceived data. Since not all signs and symptoms are present at a certain point in time, using this logical agent will aid the user in diagnosing the patient. A developed test case is presented and result is shown. Test results show 100% accuracy for diseases present in the knowledge base. Also, this paper shows the importance of using morphology in correctly diagnosing a disease. Digital image processing, if completely embedded in this logical agent, will guide the agent in correctly identifying the disease.

**Keywords:** Logical agent · Decision support system · Morphology · Clinical inference · Resolution · Tropical diseases

## 1 Introduction

Rising mortality rate caused by tropical diseases in tropical countries is an alarming fact. There are also increasing reports on deaths caused by tropical diseases in non-tropical countries. At present, about one sixth of the world's population is infected with these deadly diseases [1].

Massive efforts have been made to prevent and control tropical diseases. In 2012, the World Health Organization published a document on accelerating work to overcome the global impact of neglected tropical diseases (NTDs). The ultimate destination of this roadmap is the elimination of NTDs or reduction in their impact to levels at which they are no longer considered public-health problems [2].

Studies range from diagnostic procedure to drug design; and vaccines have also been developed to help fight the problem. However, due to mutation of bacterial/virus, environmental problem, poor nutrition, lifestyle, and several other factors that affect health condition, the diagnosis as well as appropriate treatment become more and more complex. These factors encourage researchers worldwide to broaden their studies in the field.

Medical diagnosis is one of the critical areas in medicine. If the patient is wrongly diagnosed, this will lead to incorrect treatment which can cause mishap or even death. Complications may also arise. Within healthcare, artificial intelligence (AI) is becoming a major constituent of many applications, including drug discovery, remote patient monitoring, medical diagnostics and imaging, risk management, wearables, virtual assistants, and hospital management. The employment of AI in medical diagnosis has aided in providing a better way of treating the patient. Medical fields that rely on imaging data, including radiology, pathology, dermatology, and ophthalmology, have already begun to benefit from the implementation of AI methods [3].

A decision support system (DSS) provides varied contributions to greatly improve diagnostic procedures. In radiology, the burgeoning research and technology that add complexity to medical procedures has diminished the ability of radiologists to consider available data in their clinical judgment. Further, it has increased the tendency of physicians confronted with very complex situations to make decisions based on heuristics rather than careful consideration of every possible alternative and its probability. These are some of the reasons why a DSS is an advantage [4]. Decision making in this paper refers to the correct identification of disease.

In response to the universal call to help manage this global problem, a DSS is designed to accurately assess the conditions of the patient. This paper presents a DSS, Diagnostic Test for Tropical Diseases (DiaTTroD), for diagnosing known tropical diseases in the Philippines specifically designed to aid medical practitioners and health workers, especially in scenarios where there are death of experts, or none at all. This is limited to eighteen diseases, the discussion of DiaTTroD, and the morphology of each variant. Testing is limited to theoretical testing, but broad enough to cover all possible occurrence of signs and symptoms of a patient. As of the writing of this paper, users of DiaTTroD are expected to have proper training in diagnosing diseases and should have knowledge in the trends of tropical diseases. This is to ensure that the area where the patient is diagnosed or history of the patient is considered.

The rest of the paper will describe how DiaTTroD is engineered. Section 2 presents the related work. Section 3 presents the knowledge engineering and implementation of the proposed DiaTTroD. Section 4 presents the experimental results and discussion. Finally, Sect. 5 concludes the paper.

## 2 Related Work

Diagnosis is the process of identifying the nature of an illness by an examination of its symptoms. Typically the diagnosis is performed by experts in the field such as the experimental pathologists who spend most of their time investigating the causes and mechanisms of the disease [5]. AI in Medicine is not a new idea. Several studies

have been made in the past employing computer aided diagnosis for different areas of medicine. MYCIN [6], developed for identifying bacteria and recommending appropriate treatment, is one of the early AI systems developed. The development of computer aided diagnostic systems account for the great potential of AI's contribution to improved diagnosis.

DSS is an AI tool designed to aid in decision making. The combined strength of human and computers have complementary advantage, and has the capability to surpass the abilities of either alone. Human can reason inductively, recognize patterns, apply multiple strategies to solve a problem, and adapt to unexpected events while computers can store large amount of information, can recall data accurately, perform complex calculations and execute repetitive actions reliably without tiring [4]. These advantages propelled the design of a diagnostic tool for tropical diseases.

Some of the previous works on DSS include the following. Johansson et al. [7] in 2018 created a DSS for patients with severe infection conditions. The system is used by pre-hospital emergency nurses to diagnose three medical conditions, namely, severe respiratory infection, severe central nervous system infection, and sepsis. The authors pointed out that the three diseases need qualified personnel from the emergency medical services and the emergency department to assess the patient. A decision support tool to identify patients with severe infectious diseases, and a validation process were developed. Three ambulance companies and a large city environment were considered. One problem identified in the study was the possibility of incorrectly filling up the electronic patient care system forms. Data collection were performed with actual patients and statistical analyses were performed using the Kruskal-Wallis test for non-parametric comparison of the median values of the groups which showed 94% accuracy.

Kumar and Anima [8] in 2017 used data mining methods and techniques for clinical DSS. The study used clinical records and mined them to aid medical professionals in their diagnosis. The study evaluated several techniques both involving knowledge based (KB) systems and non-KB systems. For KB systems, fuzzy logic rules, production rules, evidence, and Bayesian network were used. A rule-based system captures the knowledge of a domain expert and converts them into production rules. Fuzzy logic resolves vagueness in making a decision while the Bayesian network is used to compute the presence of a possible disease. For non-KB systems, artificial neural network is used for training data to help process incomplete data. Genetic algorithm is used to derive information from patient data. Statistical method is used for data collection, and a hybrid system is a combination of two or more approaches. Part of the conclusion of the study in [8] is that a DSS using KB will yield a high accuracy if the KB system is properly defined. Cabrera and Edye [9] in 2010 proposed a clinical DSS for acute bacterial meningitis which comprises an integration of a rule based expert system and case-based reasoning. For case-based reasoning, the implementation is based on an existing knowledge base of previous cases and re-utilizes past experience of solved cases to come up with a solution. Combining it with rule based expert system, whenever a new case is entered, a query is built from the newly entered case. Then the system tries to retrieve three most similar cases using nearest neighbour method. After retrieval, the new case is compared with the retrieved cases and a solution is drawn. The study in [9] presented excellent results for

precision and robustness with partial information and learning capacity. A 97% accuracy was achieved.

Olabiyisi et al. [10] in 2011 proposed a DSS for tropical diseases. Here, the symptoms of the patient is keyed into the DSS. A fuzzy rule base was created based on linguistic tags. In this study, data were gathered using questionnaires from various experts in tropical diseases in Nigeria. Included in the data sampling was the way a doctor carries out the diagnosis, and the symptom grouping. Weights were evaluated from the data gathered using pair wise comparison matrix for the application of the generalized fuzzy soft sets. There were six symptom intensities each with corresponding range for the fuzzy linguistic function, and these are: no sign, very mild, mild, moderate, severe, and very severe. The generalized fuzzy soft set designed was used to generate diagnosis or suspected disease.

Uzoka et al. [11], developed a clinical DSS for diagnosing malaria diagnosis in 2011. The DSS utilized fuzzy system and analytic hierarchy process (AHP). AHP uses priorities which are derived from eigenvalues of the pairwise comparison matrix of a set of elements expressed on ratio scales. Three levels are considered in the diagnostic criteria of malaria: Level 1, the goal which is malaria diagnosis; Level 2, the criteria; and Level 3, the variables. Four fuzzy values are considered for the symptoms and the linguistic labels are: mild, moderate, severe, and very severe. The study shows that fuzzy logic is slightly better than AHP. The AHP together with the medical expert achieved 67% exact diagnosis while the fuzzy system 80% diagnosis. An exact matching of the AHP and fuzzy system achieved 76.67%. In 2011, Djam, et al. [12] used fuzzy system for Tuberculosis DSS. Similar to the work of [11], the linguistic variables are mild, moderate, severe, and very severe, but the fuzzy values are different. Their experiment showed 61% possibility of patient having tuberculosis. Their study claimed a quick and efficient way of diagnosing tuberculosis. Another fuzzy system was developed by Sharma et al. [13] in 2013 to diagnose malaria and dengue fever with an accuracy of 91.3%. The study generated more than 200 rules based on information acquired from experts, books, and the internet. Symptoms as inputs to the system were fuzzified in the KB and the inference engine defuzzified in the model.

This paper focuses on the development of a rule-based DSS for diagnosing tropical diseases. One difficulty in diagnosing some of the diseases included in this study is the fact that some diseases have very similar symptoms in its early stage. As a result, health practitioners find it difficult to diagnose the correct disease. In most cases, a laboratory work is prescribed to accurately diagnose the disease. However, laboratory procedure is tedious and takes some time before the health practitioner can obtain a result. Also, some laboratory procedures can be confusing, which may result to inaccurate diagnosis. To help solve the problem, a DSS for laboratory analysis is included in this study. Unlike the several papers reviewed, this study focuses on the development of a rule-based system which focused on morphology.

One significant use of this study is to guide the health practitioners to correctly diagnose the disease by identifying the correct laboratory requirement and the correct morphology of the disease. This study uses the theoretical description of the disease and its cause as a basis for its diagnosis. It also proves that a logical agent can increase its scope by updating the KB of the DSS. Diseases, both simple and complex, can be

incorporated through appropriate update of the KB. Theoretical implementation using rule-based system may serve as a basis for statistical inference studies that use actual data. Statistical inference that will not yield 100 percent accuracy can be used with theoretical implementation for comparison.

### 3 DiaTTroD: Diagnostic Test for Tropical Diseases

One of the major difficulties in handling tropical diseases is the accuracy in identifying the correct virus, bacteria, fungus, helminth, or other pathogens present in the infected patient. This paper describes a method used in an attempt to create a framework for building a DSS for tropical diseases. The study involves several tropical diseases with almost similar signs, symptoms namely: amoebiasis, capillariasis, dengue fever, diphtheria, filariasis, giardiasis, helminthiasis, hepatitis A-C, leptospirosis, malaria, meningococemia, paragonimiasis, rabies, rota viruses, schistosomiasis, tetanus, tuberculosis, and typhoid fever.

DiaTTroD is a DSS designed to diagnose tropical diseases. A logical agent is used in decision making where the agent perceives the signs, symptoms, medical history, and environment of the patient. Its actuation includes request for laboratory examinations and the results will be included in the decision making. This study uses general rules established by the medical experts in the field that are published in medical books and journals. The first version of DiaTTroD also used the same general rules but an inclusion of medical data was also considered resulting in several misdiagnosis. General rules in identifying a disease may lead to a broader possibility of infection. On the other hand, specific rules to diagnose the disease may also lead to several misdiagnosis. This is due to the absence of a specific sign or symptom, or the patient may not have experienced it. The design of DiaTTroD is disease dependent, and not data dependent. The several errors that were encountered in the first version of DiaTTroD are the main reason why this version is created. The difference is, the second version creates a theoretical basis for creating the KB and is highly dependent on the morphology of the infection.

Unlike most diagnostic systems using statistical inferences, this study uses logical analysis and clinical inference in designing the logical agent. Conclusions of the logical analysis follow from premises making the inferential process explicit. Clinical inference makes use of clinical knowledge to draw conclusions out of the observed data [1]. One advantage of employing a logical agent is in enabling the system to guide the patient in identifying the sign and symptom experienced. This helps eliminate misdiagnosis in case the patient forgets to mention it to the attending practitioner. In cases where the patient identifies a certain symptom even if he is unsure of its existence, the system is capable of guiding the patient and the practitioner so that it can decide rationally.

#### 3.1 Sensation and Perception

In a broader sense, sensation is usually thought to be simple, basic experiences elicited by simple stimuli whose perception is usually thought to be more complicated experiences elicited by complex, often meaningful, stimuli. It is often said to be the result of a higher-order cognitive process than sensation, being the result of integration [14]. This study considers both sensation and perception as an input to the logical agent.

The diagnostic procedure encompasses three major steps, namely, history recording of symptoms, physical examination or sign and laboratory tests. Signs, symptoms and laboratory abnormalities of a disease manifest in a patient. Symptoms are evidenced of a disease perceived by the patient and recorded as history while signs are physical observations made by the health practitioner who examines the patient and are recorded as physical examination. Laboratory abnormalities in the broad sense refer to observations made by tests or special procedures recorded as laboratory findings [5].

In this study, the system takes in signs and symptoms manifested in the infected individual and are treated as a perception of the logical agent. A first order predicate logic is then performed to the sensed data. First order logic tends to fail in medical diagnosis due to laziness, theoretical ignorance, and practical ignorance [15]. This paper addresses the causes for failure by including all the possible signs and symptoms that a patient can encounter if infected with the considered diseases. The DSS takes in laboratory findings as a perceived data that may include chemical and physiological abnormalities. It is not limited to handling morphology but is also capable of analyzing kit results. Existing kits may be capable of early detection of disease, but it might not be accurate at all times.

### 3.2 Knowledge Engineering of DiaTTroD

DiaTTroD starts with identifying its task, i.e., its functionality is designed as a diagnostic agent. Its main purpose is to correctly identify the infection of the patient. With this in mind, the infections should be identified and described properly. Several infections included in this study have almost similar signs and symptoms. To correctly identify or specify the infection, the morphology of each cause of infection is included.

#### Signs and Symptoms

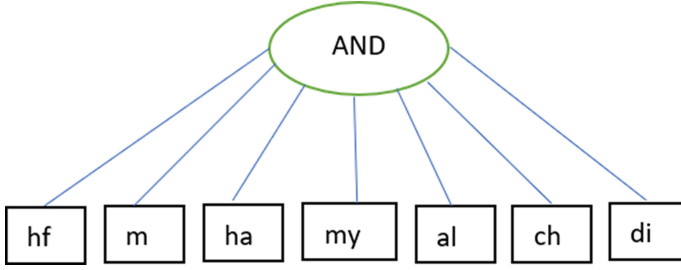
Looking closely at several possible signs and symptoms can lead to failure in designing a logical agent due to too much work. However, in this study, DiaTTroD tries to include in its KB all possible signs and symptoms identified. With morphology, the possibility of misdiagnosis is slim. As an example, consider a patient infected with malaria. Looking closely at the possible signs and symptoms, DiaTTroD may sense one or more of the following: high fever, headache, myalgia, malaise, loss of appetite, chills and diarrhea. In some cases, several other symptoms can be included. For a patient infected with typhoid fever, DiaTTroD may also sense some symptoms in malaria. This will eventually lead to confusion even if the history of patient and the environment were included. In this case, laboratory test will be considered for narrowing down the possible diagnoses.

Most patients infected with malaria will experience the following: high fever (hf), malaise (m), headache (ha), myalgia (my), loss of appetite (al), chills (ch) and diarrhea (di). These symptoms will be considered as a perception of the agent and will lead to the patient being considered as a candidate for malaria infection. Figure 1 shows an AND-OR graph of its signs and symptoms.

Denoting the disease as  $SSx$ , logically, the figure is expressed as

$$\forall(x)\{[Patient(x) \wedge SSx(x, high\ fever) \wedge SSx(x, malaise) \wedge SSx(x, headache) \wedge SSx(x, myalgia) \wedge SSx(x, appetite\ loss) \wedge SSx(x, chills) \wedge SSx(x, diarrhea)] \rightarrow Infection(x, malaria)\}.$$





**Fig. 1.** AND-OR graph of the signs and symptoms of Malaria.

The Conditional Elimination of the Logical Equivalence Law states that

$$((P \wedge Q) \rightarrow R) = \neg P \vee \neg Q \vee R.$$

Reasoning by resolution converts the predicates to clause form, i.e.

$$\begin{aligned} &\neg Patient(x) \vee \neg SSx(x, high\ fever) \vee \neg SSx(x, malaise) \vee \neg SSx(x, headache) \neg \\ &SSx(x, myalgia) \vee \neg SSx(x, appetite\_loss) \neg SSx(x, chills) \\ &\vee \neg SSx(x, diarrhea) \vee Infection(x, malaria). \end{aligned}$$

Applying resolution by refutation will result in the following:

$$\begin{aligned} &\neg Patient(x) \vee \neg SSx(x, high\ fever) \vee \neg SSx(x, malaise) \vee \neg SSx(x, headache) \neg \\ &SSx(x, myalgia) \vee \neg SSx(x, appetiteloss) \neg SSx(x, chills) \vee \neg SSx(x, diarrhea) \\ &\vee Infection(x, malaria) \neg Patient(x)\{p/x\} \\ &Patient(p) \neg SSx(x, high\ fever) \vee \neg SSx(x, malaise) \vee \neg SSx(x, headache) \neg \\ &SSx(x, myalgia) \vee \neg SSx(x, appetiteloss) \neg SSx(x, chills) \vee \neg SSx(x, diarrhea) \\ &\vee Infection(x, malaria) \neg SSx(p, high\_fever)\{p/x\} \\ &Symptom(p, high\ fever) \neg SSx(x, malaise) \vee \neg SSx(x, headache) \neg \\ &SSx(x, myalgia) \vee \neg SSx(x, appetite\ loss) \neg SSx(x, chills) \vee \neg SSx(x, diarrhea) \\ &\vee Infection(x, malaria) \neg SSx(p, malaise)\{p/x\} \\ &Symptom(p, malaise) \neg SSx(x, headache) \neg SSx(x, myalgia) \vee \neg SSx(x, appetite\ loss) \neg \\ &SSx(x, chills) \vee \neg SSx(x, diarrhea) \vee Infection(x, malaria) \\ &\neg SSx(p, headache)\{p/x\} \\ &Symptom(p, headache) \neg SSx(x, myalgia) \vee \neg SSx(x, appetite\ loss) \neg SSx(x, chills) \vee \\ &\neg SSx(x, diarrhea) \vee Infection(x, malaria) \neg \\ &SSx(p, myalgia)\{p/x\} \\ &Symptom(p, myalgia) \neg SSx(x, appetite\ loss) \neg SSx(x, chills) \vee \neg SSx(x, diarrhea) \vee \\ &Infection(x, malaria) \neg SSx(p, appetite\ loss)\{p/x\} \end{aligned}$$

$$\text{Symptom}(p, \text{appetite loss}) \neg \text{SSx}(x, \text{chills}) \vee \neg \text{SSx}(x, \text{diarrhea}) \vee \\ \text{Infection}(x, \text{malaria}) \neg \text{SSx}(p, \text{chills}) \{p/x\}$$

$$\text{Symptom}(p, \text{chills}) \neg \text{SSx}(x, \text{diarrhea}) \vee \text{Infection}(x, \text{malaria}) \neg \text{SSx}(p, \text{diarrhea}) \\ \{p/x\}$$

$$\text{Symptom}(p, \text{diarrhea}) \text{Infection}(x, \text{malaria}) \text{Infection}(p, \text{malaria}) \{p/x\} \neg \\ \text{Infection}(p, \text{malaria}) \\ \text{NIL}$$

This follows that the original goal is consistent. Some patients may encounter some or more of the following: hypotension, splenomegaly, cough, anemia, arthralgia, vomiting, nausea, tachycardia, lethargy or anorexia. Each of the above-mentioned sign or symptom is “ANDed” to the other common signs and symptoms of a patient infected with malaria, i.e.

$$\forall(x) \{ [\text{Patient}(x) \wedge \text{SSx}(x, \text{high fever}) \wedge \text{SSx}(x, \text{malaise}) \wedge \text{SSx}(x, \text{headache}) \\ \wedge \text{SSx}(x, \text{myalgia}) \wedge \text{SSx}(x, \text{appetite loss}) \wedge \text{SSx}(x, \text{chills}) \wedge \text{SSx}(x, \text{diarrhea}) \\ \wedge \text{SSx}(x, \text{hypotension})] \rightarrow \text{Infection}(x, \text{malaria}) \}.$$

Not all patients will experience the same signs and symptoms but the “ANDed” signs and symptoms are the common ones. Patients that do not have all of the “ANDed” signs and symptoms may have other infection. Figure 1 shows the AND\_OR Graph of Malaria.

A patient infected with typhoid fever will experience some, if not all, symptoms experienced by a patient infected with malaria. With typhoid fever the common symptoms, include high fever, malaise, headache, myalgia, loss of appetite, and diarrhoea. These symptoms are almost the same as that of malaria. Logically, it is expressed as

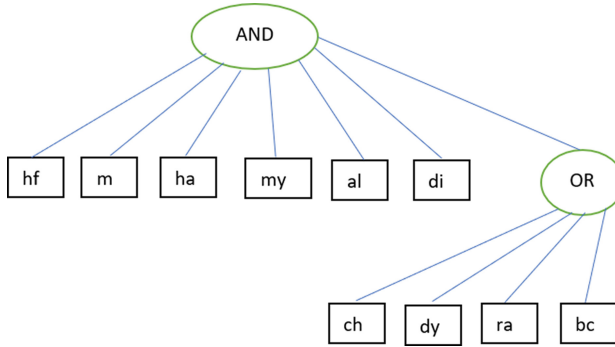
$$\forall(x) \{ [\text{Patient}(x) \wedge \text{SSx}(x, \text{high fever}) \wedge \text{SSx}(x, \text{malaise}) \wedge \text{SSx}(x, \text{headache}) \wedge \\ \text{SSx}(x, \text{myalgia}) \wedge \text{SSx}(x, \text{appetite loss}) \wedge \text{SSx}(x, \text{diarrhea})] \\ \rightarrow \text{Infection}(x, \text{typhoid fever}) \}.$$

Similarly, some patients may also experience one or more of the following: chills, dehydration (dy), rashes (ra), and bradycardia (bc). Each of these additional sign or symptom is also “ANDed” to the common signs and symptoms. Logically it is expressed as:

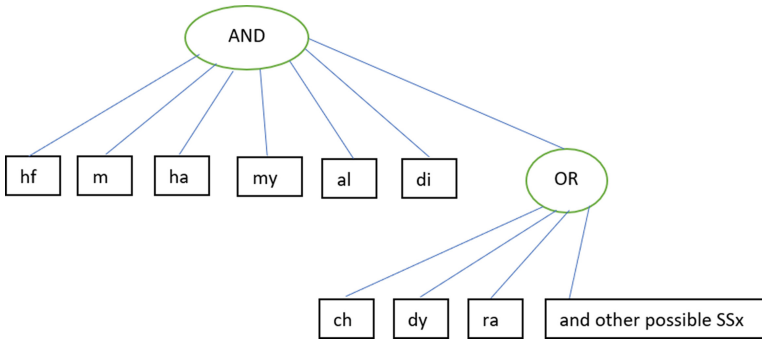
$$\forall(x) \{ [\text{Patient}(x) \wedge \text{SSx}(x, \text{high fever}) \wedge \text{SSx}(x, \text{malaise}) \wedge \text{SSx}(x, \text{headache}) \wedge \\ \text{SSx}(x, \text{myalgia}) \wedge \text{SSx}(x, \text{appetite loss}) \wedge \text{SSx}(x, \text{diarrhea}) \\ \wedge \text{SSx}(x, \text{chills})] \rightarrow \text{Infection}(x, \text{typhoid fever}) \}.$$

Figure 2 shows the AND-OR graph of typhoid fever.

Looking closely at the AND-OR graph of typhoid fever and malaria, a difference of a single node, the node chills, is observed. Combination of symptoms is not limited to the combinations presented above. To solve the problem, DiaTTroD looks at the common symptoms before it will suggest laboratory test. Figure 3 shows the AND-OR graph of diseases with similar signs and symptoms:



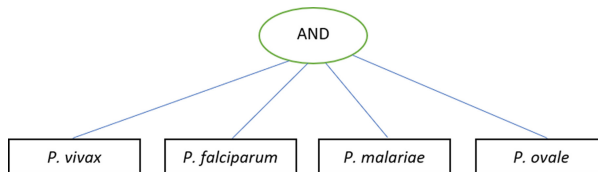
**Fig. 2.** AND-OR graph of the signs and symptoms of typhoid fever.



**Fig. 3.** AND-OR graph of common signs and symptoms of malaria and typhoid fever.

### Laboratory Tests

Abnormalities in the laboratory result can lead to accurate diagnosis. DiaTTroD considers four types of Malaria, namely, Plasmodium Vivax (*P. vivax*), Plasmodium Falciparum (*P. falciparum*), Plasmodium Malariae (*P. malariae*) and Plasmodium Ovale (*P. ovale*). Figure 4 shows the OR graph of malaria.



**Fig. 4.** OR graph of malaria.

For *P. falciparum* malaria, the following can be identified:

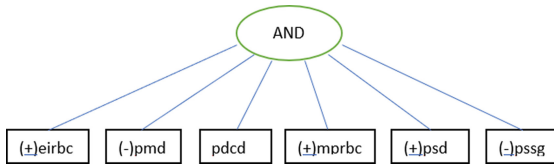
For thin blood smear

(+) enlarged infected RBC (eirbc)

- (-) presence of Maurer dots (pmd)
- (rare) parasites with double chromatin dots (pdcd)
- (+) presence of Shuffner dots (psd)
- (rare) multiple parasites per RBC (mprbc)
- (-) Parasites with sausage shaped gametocytes (pssg)

*P. vivax* malaria can be easily identified if morphology is considered. An AND graph for the morphology of *P. vivax* malaria is presented in Fig. 5. Logically, it is expressed as

$$\forall(x) \{ [Patient(x) \wedge Eirbc(x, positive) \wedge Pmd(x, negative) \wedge Pdcd(x, rare) \wedge Psd(x, positive) \wedge Mprbc(x, rare) \wedge Pssg(x, negative)] \rightarrow Result(x, p\ vivax\ malaria) \}.$$



**Fig. 5.** AND graph of *P. vivax* malaria.

For *P. malariae* malaria, the following can be identified:

Thin blood smear

- (-) enlarged infected RBC (eirbc)
- (+) presence of Maurer dots (pmd)
- (-) parasites with double chromatin dots (pdcd)
- (-) presence of Shuffner dots (psd)
- (-) multiple parasites per RBC (mprbc)
- (-) Parasites with sausage shaped gametocytes (pssg)

Logically, it is expressed as

$$\forall(x) \{ [Patient(x) \wedge Eirbc(x, negative) \wedge Pmd(x, positive) \wedge Pdcd(x, negative) \wedge Psd(x, negative) \wedge Mprbc(x, negative) \wedge Pssg(x, negative)] \rightarrow Result(x, p\ malariae\_malaria) \}.$$

Figure 6 illustrates the AND graph of *P. malariae* malaria.

In cases where test kits are available, e.g. for typhoid fever, Typhidot test can be used to verify if the patient is really infected with the disease. In cases where Typhidot test fails, the system will recommend blood culture. This is true if test for malaria is negative. Logically, Typhidot is expressed as

$$\forall(x) \{ [Patient(x) \wedge IgM(x, positive) \wedge IgG(x, positive)] \rightarrow Result(x, typhoid) \}.$$

Figure 7 illustrates the Typhidot test AND graph.

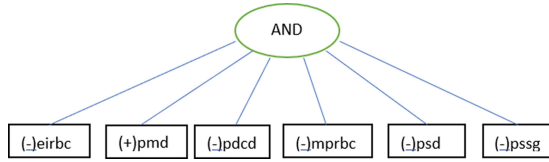


Fig. 6. AND graph of *P. malariae* malaria.

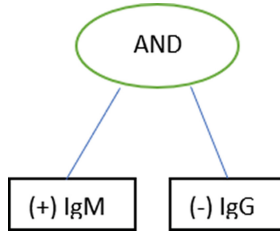


Fig. 7. AND graph of typhoid fever.

### 3.3 Implementation

In its implementation, DiaTTroD considers an algorithm for diagnosing a patient using signs and symptoms only, and another algorithm for the laboratory examinations. Algorithms 1 and 2, i.e., as shown in Fig. 8 and Fig. 9, respectively, implement DiaTTroD.

```

FUNCTION diattrodSSx (percept) returns diagnosis and lab requirement
Inputs: signs, symptoms, history, environment
Static: KB
INFORM(KB, S, Sx, Environment, History)
if S and Sx == associated S and Sx of KB then Infection-X
else
if S and Sx and history and environment == associated S and Sx of KB then Infection-X else
if S and Sx and (history or environment) == associated S and Sx of KB then Infection-X else
infection not in KB
if INFORM(KB,S,Sx,Environment, History) == TRUE
then
suggest specific lab analysis else infection not in KB.
  
```

Fig. 8. Algorithm 1: signs and symptoms.

There are two ways to diagnose a patient in DiaTTroD. A user can either use a guided diagnostic system wherein the agent asks for several signs and symptoms. Each answer of the user is considered as a sensation of the agent. The agent will decide the next question for the user to answer. This enables the user to complete the exam before the agent decides. If the user is uncertain of an answer, the agent will still consider whatever the answer is. The agent has tags for important data to consider before it decides. In the case where the tags are not met, the agent will suggest a laboratory exam. The second test displays all signs and symptoms for the user to check if a certain sign or

```

Function diattrodLab (percept) returns diagnosis
Inputs: laboratory test results
if (INFORM(KB, lab analysis result)) == TRUE
return infection
else
infection not in KB
    
```

**Fig. 9.** Algorithm 2: laboratory examination.

symptom is present. The logical agent will decide after the user has checked all present signs and symptoms. This is the architecture used for the laboratory exam. There are no uncertainties considered because the laboratory results are discrete. A disease is defined by its own morphological structure and is unique.

### 4 Testing and Results

This study uses theoretical testing to test the proposed logical agent. Several possible cases and combinations of signs and symptoms were considered. To correctly implement the diagnosis, some symptoms or signs are quantified, such as fever, where its degree is specified numerically. In such cases, correct input of sign or symptom is expected by the logical agent.

**Table 1.** Case 1 test cases.

SSx	Value
High fever	T
Appetite loss	T
Abdominal pain	T
Malaise	T
Feeling weak	T
Pain	T
Rash	T
Wbc_reveal_leukopenia	F
Congested_face	T
Rose_spot_in_trunk_and_abdomen	F
Right_iliac_fossa_tenderness	F

Correct and complete definition of a disease in the KB and complete input of signs or symptoms yield 100% accuracy. However, in cases where a certain disease is not correctly defined, less than 80% accuracy is met. Thus, the accuracy depends on how the KB is defined and how signs and symptoms are queried. In cases where test kits are

used, and the test kits fail, DiaTTroD will also fail if there is no culture present in the KB. To prevent failure in uncertain cases, the user must refute the results of test kits, and use the culture method for laboratory examination.

Table 1 shows the developed test cases for Case 1, indicating when patient has True(T) or False(F) value for the SSx. This will result in an Unknown Infection. When WBC\_reveal\_leukopenia (WBCrl) gets the value of false, this reconsiders another disease that is included in the knowledge of DiaTTroD. If (WBCrl) should be present in dengue fever, for example, then the system will eliminate dengue fever, and will proceed to the next disease. It then asks for the value of Congested\_face. Sensing that the value is true, it considers the next SSx present. Sensing again that the rose\_spot\_in\_trunk\_and\_abdomen (rsta) is false, it asks another question. The logical agent will ask another question related to the disease because rsta must not be necessarily present as an SSx to identify the next disease. The logical agent then asks for the presence of right\_iliac\_fossa\_tenderness (rift). For example, a patient must have either rsta or rift to declare that the infection is typhoid fever. Since both are absent in Case 1, and these SSx are no longer significant in other infections included in the KB, then the logical agent will give an output of Unknown Infection. Meaning, the patient may be infected with another disease that is not part of DiaTTroD. Several test cases were repeated including those that are possibly impossible.

Table 2, a summary of some test cases, shows that 100% is achieved if the KB is hit or KB is satisfied since this is a logical agent. 80% result does not mean that the accuracy of the system is only 80%, rather, this means that only 80% of the SSx were considered. In this case, the Logical agent will guide the user what to look for because the agent will try to traverse all necessary common SSx before it will decide an Unknown infection. If all SSx are considered, then a 100% result can be obtained. Common type of test here refers to common SSx of both dengue and typhoid fever, for example. Prognosis is the actual input of all SSx.

Table 3 shows the test run of the program in SWI Prolog (AMD 64, Multi-threaded, version 8.02) using an Intel® Core™ i7-7500 CPU @ 2.70 GHz processor with 12.0 GB RAM running on Windows 10. This machine is used to test run the code to get the average time of the diagnosis. The machine chosen to test the run time is considered as a not so fast and not slow platform. The output in time will resemble an average run time of DiaTTroD. Here, the test run assumes that all data are readily available. The logical agent expects the user to answer all questions to get a logical answer. Looking at Cases 4 and 6, both have 15 questions, and both have dengue fever infection, but the time required to run the logical agent differs. Here, the time in seconds include the speed of the user in using the system which means, slow users require more time, thus, the time values are usually not the same. However, as Table 3 shows time is not a constraint in running DiaTTroD. All cases have test runs in less than 60 s. This shows that using the system is a good tool to guide the user in decision making. If the user is not sure what SSx to look for from the patient and what laboratory procedure to prescribe, the user can run the logical agent, and the system will guide the user. Since not all SSx are present at a certain point in time, the user can take the history of the patient, and run DiaTTroD again whenever a new SSx is observed from the patient. This saves time and ensures that all data from the patient are recorded. Using DiaTTroD will help create a patient

**Table 2.** Summary of test cases.

Case number	Type of test	Infection	Result (%)
1	Common	Typhoid_dengue	100
	Prognosis	Unknown	100
2	Common	Typhoid_dengue	100
	Prognosis	Unknown	100
3	Common	Typhoid_dengue	100
	Prognosis	Unknown	100
4	Common	Typhoid_dengue	100
	Prognosis	Dengue fever	100
	Lab	Dengue fever	100
5	Common	Typhoid_dengue	100
	Prognosis	Typhoid fever	100
	Lab	Typhoid fever	100
6	Common	Typhoid_dengue	100
	Prognosis	Dengue fever	100
	Lab	Dengue fever	100
7	Common	Unknown	80/100
	Prognosis	Unknown	100
8	Common	Unknown	80/100
	Prognosis	Unknown	100
9	Lab	Giardiasis	100
10	Lab	<i>S. mansoni</i>	100

**Table 3.** Test run of DiaTTroD.

Case number	Number of questions	Time (seconds)	Infection
1	6	14	Typhoid_dengue
4	15	34	Dengue fever
5	9	18	Typhoid fever
6	15	36	Dengue fever

database in the future. This can then be used to create an automated learning or to mine data to improve the KB.

The present study involves image processing of the laboratory findings. This is where morphology is considered. A disease is defined by its infection. Infection is characterized



by an infective agent. For example, there are four kinds of *Schistosoma* considered in this study, these are: *Schistosoma japonicum* (*S. japonicum*), *Schistosoma haematobium* (*S. haematobium*), *Schistosoma intercalatum* (*S. intercalatum*), and *Schistosoma mekongi* (*S. mansoni*). They have very similar SSx and thus, it is very difficult to identify which *Schistosoma* infects a patient. Although in some infections, with those that are identified that have specific sources, it is good to confirm what *Schistosoma* is present in the infection. In this case, a laboratory examination is required. To identify what specific *Schistosoma* is present, morphology is considered. Stool, urine or blood samples can be used to check a parasite egg. Stool is used for *S. mansoni* and *S. japonicum*, and urine for *S. haematobium* eggs [16]. These are morphologically examined under a microscope. For example, a *S. mansoni* has the following characteristics: colour is yellow, size is 140 by 66 micrometers, and shape is elongated with prominent lateral spine near posterior end. For *S. japonicum*, the following are the characteristics: size is 90 by 70  $\mu\text{m}$ , shape is oval, and colour is yellow. For *S. haematobium*: colour is yellow, size is 143 by 69  $\mu\text{m}$ , and shape is elongated with rounded anterior end. For *S. intercalatum*: colour is yellow, size is 175 by 60  $\mu\text{m}$ , shape is elongated with tapered anterior end and terminal spine. These are just a few of their differences in morphology but these are unique to each infectious agent. In this study, an infection is defined by these differences. Each difference is described by an algorithm to digitally analyze it.

Under the microscope, the morphology of the egg is examined. In the case of Schistosomiasis, the colour, the shape, and the size are some of the data automatically extracted using image processing. This ensures that an accurate laboratory finding will be fed to the system. In cases where the accuracy is small, another blood sample from the patient will be considered. In some cases, the infectious agent will not manifest right away. This is the advantage of using DiaTTroD because the system will guide the user what to look for to help diagnose the infection.

#### 4.1 Design of DiaTTroD

The use of morphology provides a good input for the logical agent. This simplifies the knowledge engineering. Although most of the work reviewed in Sect. 2 used signs and symptoms to diagnose the patient, it is always good to confirm the infection by considering a laboratory examination. The results of the laboratory examination will provide an exact diagnosis of the infection. The use of a DSS in diagnosing tropical diseases will increase the accuracy of diagnosis as a patient must not be diagnosed based on how the disease is defined. While human intuition has a tendency to miss the diagnosis and experts may sometimes fail, a logical agent will not if properly defined since morphology is static. The definition of an infection will always be the same. If there are changes, a new infection is defined. The advantage of being an expert is the experience. However, that experience can also lead to missed diagnosis whenever mutation of virus or complication is encountered. The expert has the tendency to rely on experience but a logical agent will always stick to what it was taught, and since DiaTTroD is a logical agent that is based on morphology it will always have a logical output. In case of mutation, the agent will return a FALSE value since it is not the right disease. On the other hand, if there is complication, DiaTTroD will return TRUE if the disease is still present in the patient. In the worst case, e.g., where a patient has both

malaria and dengue fever, then if it is already TRUE for malaria, then the user can run the agent again and refutes some of those that will lead to repeat diagnosis of malaria so that the user can test if the patient has dengue fever.

The results above show that a logical agent that is not designed to learn can still produce a very good result. If automatic learning is used, there is a tendency to increase the search space of the knowledge base and will slow down diagnosis. In DSS such as DiaTTroD, time is an important element because some of the patient cases that will be diagnosed are acute. Dengue fever can be very fatal if left unattended. The most important thing to consider is that the user is guided on what to look for in a patient, i.e., only necessary information from the patient is elicited. If the test fails for the diseases included in the KB, then the search space for some other diseases outside the scope of the KB will decrease. In complicated cases where some of the infections are present in the KB, the user can repeatedly run the agent and test for other diseases present in the KB.

**Table 4.** Comparative results of diagnostic systems

Author	Year	Diseases	Data	Method	Input	Accuracy
[7]	2018	severe respiratory infection, severe central nervous system infection, and sepsis	performed with actual patients in pre-hospital environment	statistical analyses using Kruskal-Wallis test	symptoms	94%
[8]	2017	disease records in clinic	clinical records	data mining, KB and non-KB	survey	
[9]	2010	acute bacterial meningitis	patient records	rule based expert system and case-based reasoning	symptoms	97% excellent results for precision & robustness
[10]	2011	malaria, typhoid, tuberculosis, sexually transmitted	patient records	fuzzy inference	symptoms	
[11]	2011	Malaria	patient records	fuzzy system and analytic hierarchy process (AHP)	symptoms fuzzy better than AHP	80%
[12]	2011	tuberculosis	patient records	fuzzy system	signs and symptoms	61% (possibility)
[13]	2013	malaria and dengue	patient records	fuzzy system	symptoms	91.30%

(continued)

**Table 4.** (continued)

Author	Year	Diseases	Data	Method	Input	Accuracy
DiaTTroD	2019	amoebiasis, capillariasis, dengue fever, diphtheria, filariasis, giardiasis, helminthiasis, hepatitis A-C, leptospirosis, malaria, meningococemia, paragonimiasis, rabies, rota viruses, schistosomiasis, tetanus, tuberculosis, and typhoid fever	morphology of diseases	logical agent	signs and symptoms; laboratory	100%

### Signs and Symptoms

The test results for DiaTTroD show that 100% accuracy is achieved if the signs and symptoms of a disease are correctly defined in its KB. Common SSx must be thoroughly investigated so that the agent will have a good sensation of the condition of the patient which will result to the correct actuation. If these were not correctly engineered, DiaTTroD will fail, as in the case of diarrhea and constipation. In some cases, a patient infected with disease, X, may suffer constipation, but in other cases, patient infected with the same disease, X, will suffer diarrhea. If for example, diarrhea is ANDed with the common SSx, then in cases where this is False, DiaTTroD will fail.

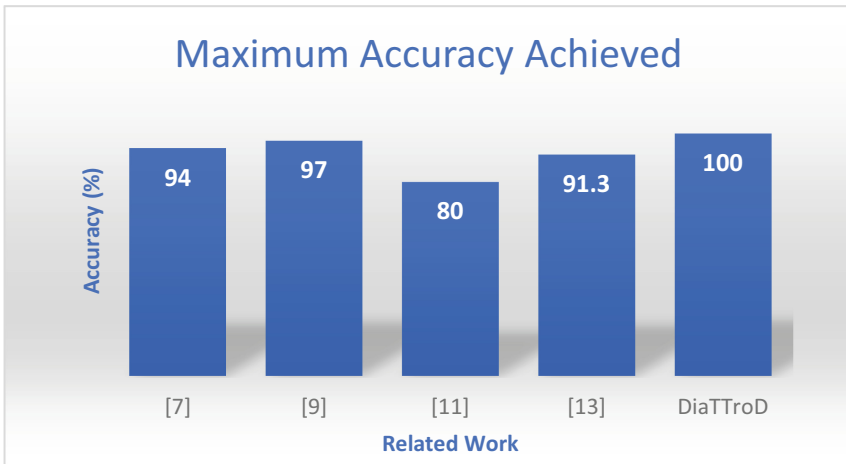
### Laboratory

The use of logical agent in the design of DiaTTroD crafted the KB of the morphology of the pathogen. The results of laboratory of DiaTTroD are expected to clarify what infection is present in the patient. Using the morphology of a specific disease will lead to correct diagnosis. Since a disease is defined by its infection, and an infection is defined by the morphology of the pathogen, proper engineering of the knowledge included in the KB of DiaTTroD will lead to exact diagnosis. In this way, the patient can be treated properly if the correct diagnosis is made.

## 4.2 Comparison of DiaTTroD with Other Work

Table 4 compares the performances of DiaTTroD and several studies reviewed in Sect. 2. In [12], the number shows possibility of patient infected with tuberculosis.

A graphical comparison of the maximum accuracy attained by each study is shown in Fig. 10.



**Fig. 10.** Graphical comparison of maximum accuracy achieved.

## 5 Conclusion and Future Work

This paper presents DiaTTroD for accurately diagnosing a patient infected with diseases considered in its KB. DiaTTroD comprises two algorithms that address the possible failures of a logical agent design for diagnostic medicine, by considering all known possibilities that can happen to a patient. The inclusion of morphology further improves the accuracy of its diagnosis. Addition of diseases can be easily achieved by adding knowledge associated with each infection in its KB.

In cases where the sign or symptom is contradictory, which is possible in some places or certain environment, the KB can be altered and not the algorithms. In certain parts of the world, a patient experiencing a disease, X, may experience diarrhea, but in other parts of the world, patients also suffering from X, may experience constipation. In such a case, only a qualified practitioner or an expert in the field can modify the KB.

The proposed logical agent can be used not just for medical diagnosis but also for other areas where a DSS is desired. For as long as the design of the KB is accurate and complete, this study can be used as a reference. The present work includes the use of image processing technique to correctly analyze the images in the laboratory and development of persuasive technology to correctly extract signs and symptoms of the patient.

Future work includes addition of infections and the use of several other algorithms to improve the processing speed and the accuracy of diagnosis in failed cases, if any. A real time and multimodal diagnostic examination is also considered. Finally, complex cases such as complications of diseases will be investigated.

## References

1. Neglected Tropical Diseases Quick Facts. <https://www.niaid.nih.gov/research/neglected-tropical-diseases-quick-facts>. Accessed 20 Sept 2019
2. World Health Organization. Accelerating work to overcome the global impact of neglected tropical diseases: a roadmap for Implementation (2012). [https://www.who.int/neglected\\_diseases/NTD\\_RoadMap\\_2012\\_Fullversion.pdf](https://www.who.int/neglected_diseases/NTD_RoadMap_2012_Fullversion.pdf). Accessed 20 Sept 2019
3. Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L.H., Alerts, H.J.W.: Artificial intelligence in radiology. *Nat. Rev. Cancer* **18**(8), 500–510 (2018)
4. Burnside, E.S., Kahn, C.E.: Artificial intelligence helps provide decision support in radiology. *Diagn. Imag.* (2004). <https://www.diagnosticimaging.com/articles/artificial-intelligence-helps-provide-decision-support-radiology>. Accessed 20 Sept 2019
5. Kent, T.H., Hart, M.N.: *Introduction to Human Disease*. Appleton-Century-Crofts, Connecticut (1987)
6. Van Melle, W.: MYCIN: a knowledge-based consultation program for infectious disease diagnosis. *Int. J. Man-Mach. Stud.* **10**(3), 313–322 (1978)
7. Johansson, N., Spindler, C., Valik, J., Vicente, V.: Developing a decision support system for patients with severe infection conditions in pre-hospital care. *Int. J. Infect. Dis.* **72**, 40–48 (2018)
8. Kumar, B.S., Anima, P.: Data mining methods and techniques for clinical decision support systems. *J. Netw. Commun. Emerg. Tech. (JNCET)* **7**(8), 29–33 (2017)
9. Cabrera, M.M., Edey, E.O.: Integration of rule based expert systems and case based reasoning in an acute bacterial meningitis clinical decision support system. *Int. J. Inf. Sci. Inf. Secur. (IJCSIS)* **7**(2), 112–118 (2010)
10. Olabiyisi, S.O., Omidlora, E.O., Olaniyan, M.O., Derikoma, O.: A decision support system model for diagnosing tropical diseases using fuzzy logic. *African J. Comput. ICT* **4**(2), 1–6 (2011)
11. Uzoka, F.-M., Osuji, J., Obot, O.: Clinical decision support system (DSS) in the diagnosis of malaria: a case comparison of two soft computing methodologies. *Exp. Syst. Appl.* **38**, 1537–1553 (2011)
12. Djam, X.Y., Kimbi, Y.H.: A decision support system for tuberculosis diagnosis. *Pac. J. Sci. Technol.* **12**(2), 410–425 (2011)
13. Sharma, P., Singh, D.B.V., Bandil, M.K., Mishra, N.: Decision support system for malaria and dengue disease diagnosis (DSSMD). *Int. J. Inf. Comput. Technol.* **3**(7), 633–640 (2013). International Research Publication House. ISSN 0974-2239
14. Goldstein, B.E.: *Sensation and Perception*, 2nd edn. Wadsworth Publishing Company, Belmont (1984)
15. Russel, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River (2003)
16. Diagnosis. Parasites- Schistosomiasis. Centers for Disease Control and Prevention. <https://www.cdc.gov/parasites/schistosomiasis/diagnosis.html>. Accessed 20 Sept 2019



# A Weighted Combination Method of Multiple K-Nearest Neighbor Classifiers for EEG-Based Cognitive Task Classification

Abduljalil Mohamed<sup>1</sup>(✉), Amer Mohamed<sup>2</sup>, and Yasir Mustafa<sup>1</sup>

<sup>1</sup> Computer Information Systems and Computer Science Department,  
Ahmed Bin Mohamed Military College, Doha, Qatar  
{ajmaoham, yasir}@abmmc.edu.qa

<sup>2</sup> Computer Science and Engineering Department, College of Engineering,  
Qatar University, Doha, Qatar  
amrm@qu.edu.qa

**Abstract.** The effectiveness of wireless electroencephalograph (EEG) sensor-based medical or Brain Computer Interface applications largely depends on how to classify EEG signals as accurately as possible. Empirical studies show that EEG channels respond differently to different cognitive tasks. Thus, to effectively classify cognitive tasks, the channel cognitive sensitivity should be taken into account during the classification process. In this paper, we propose a weighted combination of multiple  $k$ -nearest neighbor approach for cognitive task classification. Each EEG channel is assigned a weight that reflects its sensitivity to the cognitive task space. First, for each channel, a  $k$ -nearest neighbor algorithm is performed and an output is produced. To combine all the channel outputs, a modified aggregation method is utilized such that the weights assigned to the channels are accommodated. Experimental work shows that the proposed technique achieved 96.7% classification accuracy utilizing all the available channels, and 96.4% and 92.7% classification accuracies utilizing only 70% and 60% of the available channels, respectively, for subject 1; and 99.4% classification accuracy utilizing all the available channels, and 98.9% and 97.6% classification accuracies utilizing only 70% and 60% of the available channels, respectively, for subject 2.

**Keywords:** Cognitive task classification · K-Nearest neighbor · EEG signal classification

## 1 Introduction

The functional magnetic resonance imaging (fMRI) data has been used as a main source of information for identifying cognitive tasks [1–4]. However, electroencephalograph (EEG) sensors are cheaper [1], and more feasible for applications such as Brain Computer Interface (BCI) epilepsy treatment [2, 5]. Any prospective algorithm should meet two basic requirements to be efficiently applicable, namely acceptable classification accuracy and energy conservation. For wired EEG sensors the second requirement may not be a

critical issue. Different classifier models with different learning and reasoning strategies have been utilized for the purpose of cognitive task classification and brain disease analysis such as artificial neural networks [6–11], support vector machines [12, 13], and fuzzy logic [14].

Most of these approaches utilize all EEG channels. To the best of our knowledge, none of the existing techniques consider the sensitivity of EEG channels to different cognitive tasks. To increase the performance of the proposed approach in terms of classification accuracy, EEG channels are assigned weights that appropriately reflect their discriminant capabilities regarding cognitive tasks. When compared with other well-known techniques, experimental work shows that the proposed approach outperforms all of them. Even when the number of the participating channels are reduced, for better channel utilization, the proposed approach still yield a very acceptable classification accuracy compared with other techniques.

The rest of the paper is organized as follows: Sect. 2 explains the k-nearest neighbor technique. The proposed approach is detailed in Sect. 3. Section 4 reports the results obtained by the new approach and final remarks are summarized in Sect. 5.

## 2 K-Nearest Neighbor-Based Classifier Model

The  $k$ -nearest neighbor technique is adopted in this work as the classifier model for the individual EEG channels. The nearest neighbor is considered a simple and intuitive technique. According to the  $k$ - $nn$  algorithm, a new channel reading is assigned the cognitive task of the channel readings in the training set that is closest to the new channel reading. The similarity property is based on distance measures. Formally, the nearest neighbor strategy can be described as follows. Let

$$D = \{(x_i, y_i), i = 1, \dots, n_D\} \quad (1)$$

be the training set of the channel reading data, where  $y_i \in \{1, \dots, T\}$  denotes cognitive task membership and the input vector  $x'_i = (x_{i1}, \dots, x_{ip})$  represents the readings of an EEG sensor of  $p$  channels. The nearest neighbor is determined by a distance function  $d(., .)$ . For a new channel reading  $(x, y)$  the nearest neighbor  $(x_{(1)}, y_{(1)})$  within the training set is determined by:

$$d(x, x_{(1)}) = \min_i(d(x, x_i)) \quad (2)$$

and  $y_{(1)}$  the cognitive task of the nearest neighbor is selected as a prediction for  $y$ . One typical distance function is the Euclidean distance:

$$d(x_i, x_j) = \left( \sum_{t=1}^p (x_{it} - x_{jt})^2 \right)^{\frac{1}{2}} \quad (3)$$

For the  $k$ -nearest neighbor,  $k$  refers to the  $k$  nearest neighbors. The decision, then, is determined based on the cognitive task label that belongs to the majority of the neighbors. Let  $k_r$  denote the number of channel readings from the group of the nearest neighbors, that belong to cognitive task  $r$ :

$$\sum_{r=1}^c k_r = k \quad (4)$$

The cognitive class  $k_l$  assigned to a new channel reading is decided as follows:

$$k_l = \max_r(k_r) \tag{5}$$

### 3 A Weighted Combination Approach of Multiple K-NN Classifiers

The majority of the EEG-based brain computer applications utilize all the EEG channels in order to identify cognitive tasks (i.e. all the channels have the same weight). However, EEG channels may vary in their discriminant capabilities regarding certain cognitive tasks. Empirical studies show that EEG channels respond differently to different cognitive tasks, that is, a given channel can be more accurate in identifying a particular cognitive task than other channels. In order to exploit this important information, each channel is given an appropriate weight for each cognitive task. The task weight signifies the channel’s classification accuracy of this cognitive task. The channel weights are then incorporated in the classification process. The architecture of the proposed approach is shown in Fig. 1.

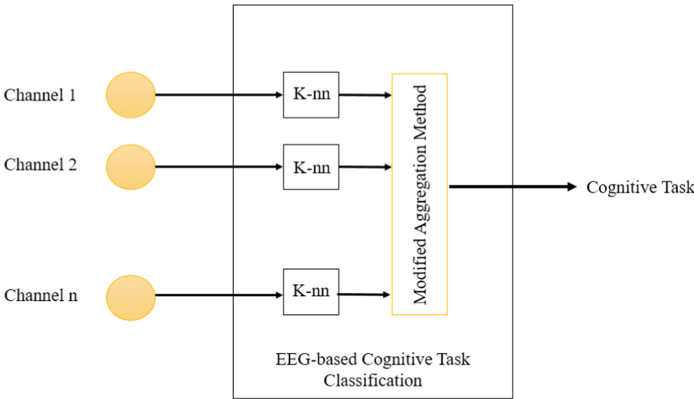


Fig. 1. A general architecture of the proposed approach.

#### 3.1 Channel Weight

As stated in the previous section, EEG channels have different discriminant power, and as such they should not be treated with the same significance during the task classification. Formally, this simple fact can be described as follows. Let us assume that  $G$  is an EEG sensor and has  $p$  channels:

$$G = \{c_1, c_2, \dots, c_p\} \tag{6}$$

where  $c_i$  is the  $i^{th}$  channel in  $G$  ( $i = 1, \dots, p$ ), and  $G'$  is the set of the current channel readings of  $G$ :

$$G' = \{r_{c_1}, r_{c_2}, \dots, r_{c_p}\} \tag{7}$$



where  $r_{c_i}$  is the current reading of the channel  $c_i$  in  $G$ . Let us further assume that the cognitive task space is represented as follows:

$$T = \{t_1, t_2, \dots, t_Q\} \quad (8)$$

Thus, the channel weight  $w_{c_i}$  for the above task space takes the following form:

$$w_{c_i} = \left\{ \frac{w_{c_i}(t_1)}{t_1}, \frac{w_{c_i}(t_2)}{t_2}, \dots, \frac{w_{c_i}(t_Q)}{t_Q} \right\} \quad (9)$$

where  $w_{c_i}(t_j)$  is the channel  $c_i$ 's weight for the cognitive task  $t_j$ , and is a positive value between 0 and 1.

### 3.2 Weighted Combination Method

The  $k$ -nearest neighbor technique, as described in the previous section, is used as a classifier model for each channel. Let  $f_{c_i}$  refer to the  $k$ -nearest neighbor applied to the channel  $c_i$ . The expected output for the channel  $c_i$  takes the following form:

$$f_{c_i} = \left\{ \frac{f_{c_i}(t_1)}{t_1}, \frac{f_{c_i}(t_2)}{t_2}, \dots, \frac{f_{c_i}(t_Q)}{t_Q} \right\} \quad (10)$$

If  $t_j$  is identified as the correct cognitive task by the channel  $c_i$  given its current reading  $r_{c_i}$ , then the channel's output is:

$$f_{c_i}(r_{c_i}) = \left\{ \frac{0}{t_1}, \dots, \frac{1}{t_j}, \dots, \frac{0}{t_Q} \right\} \quad (11)$$

The value of 1, given to the cognitive task  $t_j$ , means that the channel is absolutely certain regarding its outcome.

To be more realistic, this value should be modified with the channel's weight during the aggregation of the outputs of the channels. Given the current channel reading of the EEG sensor  $G$ , the combination method  $T(G')$  is described as follows:

$$T(G') = \left\{ \frac{T(t_1)}{t_1}, \frac{T(t_2)}{t_2}, \dots, \frac{T(t_Q)}{t_Q} \right\} \quad (12)$$

where  $T(t_j)$  is defined as follows:

$$T(t_j) = \frac{\sum_{i=1}^p f_{c_i}(t_j)w_{c_i}(t_j)}{u} \quad (13)$$

for  $j = 1, \dots, Q$ , and  $u$  is a normalizing factor and defined as follows:

$$u = \sum_{q=1}^Q \sum_{i=1}^p f_{c_i}(t_q)w_{c_i}(t_q) \quad (14)$$

Hence, the current channel readings are predicted into the cognitive task  $l$  with:

$$T(t_l) = \max_j(T(t_j)) \quad (15)$$

### 3.3 Principal Component Analysis-Based Channel Selection

For wireless sensors, depending on the given application, minimizing energy consumption is actually a major concern. Moreover, it has been shown that wireless transceivers consume more power on average than processors [2]. Thus, most of the energy-aware algorithms reported in the literature address this issue at the communication level [5–8]. This objective can be achieved by reducing the number of channels participating in identifying cognitive tasks. However, randomly reducing channels may inadvertently reduce the classification accuracy of the system as informative channels can be at risk of being eliminated. Viewing channel selection as a feature selection (FS) problem, many FS techniques can be utilized. On top of these techniques comes the principal component analysis which is widely known for its capability of selecting the best informative features. In this work, the PCA is utilized for the EEG channels as depicted in Fig. 2. As shown in the figure, first the PCA is applied to the EEG sensor with  $p$  channels. The PCA selects  $m$  channels from the EEG sensor (i.e.,  $m \ll p$ ), which in turn are fed to the cognitive classification system.

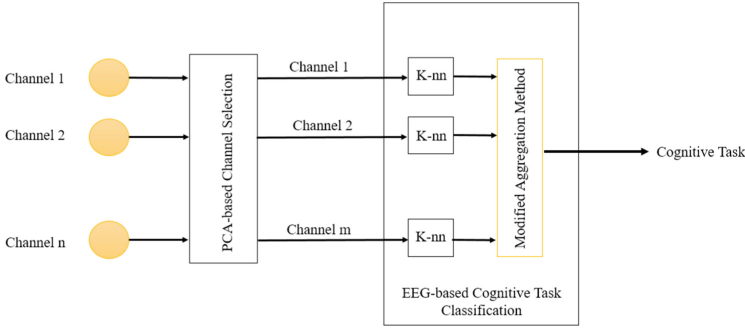


Fig. 2. A PCA-based channel selection.

## 4 Performance Evaluation

To evaluate the proposed approach, extensive experimental work has been conducted. Emotiv EPOC EEG headset is used to collect brain signals. The EEG sensors are connected to well-defined parts of the human scalp. The data is initially collected from two subjects. The EEG signals are sampled at 128 Hz. In this study, three mental tasks, namely sending an email ( $t_1$ ), dialing a phone number ( $t_2$ ), and initiating a Skype session ( $t_3$ ), are identified. Thus, the cognitive task space contains three elements:

$$T = \{t_1, t_2, t_3\} \quad (16)$$

### 4.1 Channel Weights for Each Subject

The quality of the discriminant power of each channel is examined using the  $k$ -NN classifier model. The  $k$  parameter is set to 4. The overall classification accuracy of the sensor channels for the two subjects are reported in Table 1 and Table 2, respectively. Notably, no single channel yields a satisfactory performance level in terms of overall classification accuracy (less than 45% accuracy). However, it can be noted from the table that EEG channels respond differently to different mental tasks. The tables are used to calculate channel weights for each subject as follows.

For a given subject, let us assume the classification accuracy for a given EEG channel  $c_i$  is given by:

$$c_i = \{a_{t_1}, a_{t_2}, a_{t_3}\} \tag{17}$$

where  $a_{t_1}$ ,  $a_{t_2}$ , and  $a_{t_3}$  are the classification accuracy for the three cognitive tasks  $t_1$ ,  $t_2$ , and  $t_3$  as given in (16). Now, the weight of channel  $w_{c_i(t_j)}$ , for the cognitive task  $t_j$  can be calculated as follows:

$$w_{c_i(t_j)} = \frac{a_{t_j}}{a_{t_1} + a_{t_2} + a_{t_3}} \tag{18}$$

for  $j = 1, 2$ , and  $3$ .

For example, the weights of channel 4 (i.e.,  $c_4$ ), for subject 1, for the three cognitive tasks can be calculated as follows:

$$c_4 = \{59\%, 37\%, 25\%\}$$

**Table 1.** EEG channels classification accuracy to cognitive tasks for subject 1.

EEG channel number	Cognitive tasks			Overall accuracy
	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	
4	59%	37%	25%	40.2%
5	57%	34%	19%	36.6%
6	68%	39%	19%	41.7%
7	66%	33%	32%	43%
8	65%	40%	28%	44%
9	69%	21%	21%	36.8%
10	67%	39%	29%	44.6%
11	75%	26%	20%	40%
12	69%	29%	19%	38.8%
13	65%	34%	27%	41.6%
14	59%	64%	36%	53.3%
15	65%	35%	26%	42%
16	51%	46%	31%	42.8%
17	56%	49%	25%	43.2%

**Table 2.** EEG channels classification accuracy to cognitive tasks for subject 2.

EEG channel number	Cognitive tasks			Overall accuracy
	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	
4	50%	47%	36%	44.1%
5	65%	37%	20%	40.8%
6	64%	45%	17%	42.1%
7	70%	48%	14%	44%
8	50%	40%	43%	44.2%
9	62%	42%	20%	41.4%
10	70%	41%	7%	39.5%
11	72%	35%	2%	36.3%
12	69%	47%	16%	44.4%
13	52%	52%	38%	47.2%
14	56%	39%	16%	37.1%
15	64%	39%	16%	39.7%
16	48%	42%	36%	41.8%
17	47%	45%	37%	42.9%

The classification accuracy (as reposted in Table 1). Using (18), the channel weights are then computed:

$$w_{c_4} = \left\{ \frac{0.50}{t_1}, \frac{0.30}{t_2}, \frac{0.20}{t_3} \right\}$$

The remaining weights for the other channels are calculated in the same way.

The performance of the proposed weighted combination method of multiple  $k$ -nearest neighbor classifiers (WCMMKNN) approach is measured by two important criteria: the receiver operating characteristics (ROC) curves and the confusion matrices. In ROC, the true positive rates (sensitivity) are plotted against the false positive rates (1-specificity) for different cut-off points. For a specific cognitive task, the closer its ROC curve is to the left upper corner of the graph, the higher its classification accuracy is. In the confusion matrix plot, the rows correspond to the predicted class (output class), and the columns show the true class (target class). The proposed approach is compared with the four well-known classifiers, namely the Quadratic Analysis (QA) classifier, Decision Tree (DT) classifier, Artificial Neural Network (ANN) classifier, and the Support Vector Machine (SVM) classifier. The confusion matrix and the ROC curves are shown in Fig. 3, 4, 5, 6 and 7 for the models DT, QA, SVM, ANN, and the proposed weighted combination method of multiple  $k$ -nearest neighbor classifiers (WCMMKNN) approach, for subject 1, and in Fig. 8, 9, 10, 11, and 12, for subject 2, respectively.

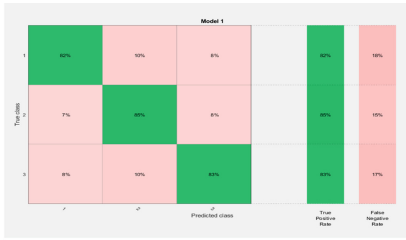
**Table 3.** Classification accuracy for the QA, DT, SVM, ANN, and WCMMKNN models for subject 1.

Classifier model	Cognitive tasks			Overall accuracy
	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	
QA	64%	99%	60%	74.5%
DT	82%	85%	83%	83.2%
SVM	90%	97%	93%	93.5%
ANN	85.1%	84.8%	84.3%	84.8%
WCMMKNN	98%	99%	96%	97.6%

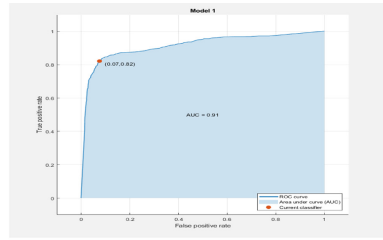
**Table 4.** Classification accuracy for the QA, DT, SVM, ANN, and WCMMKNN models for subject 2.

Classifier model	Cognitive tasks			Overall accuracy
	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	
QA	67%	72%	95%	77.6%
DT	86%	93%	91%	89.9%
SVM	97%	97%	98%	97.5%
ANN	87.6%	91%	82.6%	86.8%
WCMMKNN	99%	>99%	99%	99.4%

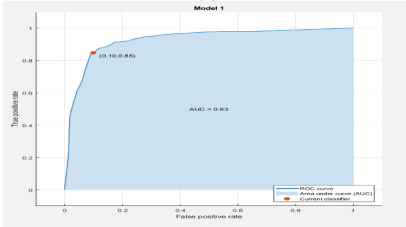
It should be noted from these figures that the proposed WCMMKNN model outperforms all other models. It yields 97.6% classification for subject 1, and 99.4% classification for subject 2. Moreover, the Quadratic Analysis (QA) yields the worst performance at 74.5% detection accuracy for subject 1 and 77.6% classification accuracy for subject 2. The performance of all classifier models are summarized in Table 3 subject 1, in Table 4 for subject 2. The proposed approach still performs better than the other models when reducing the number of channels to 10 and 8, and yield 96.4%, 92.7%, for subject 1 as shown in Table 5 and Table 6, respectively; and 98.9%, 97.6%, for subject 2 as shown in Tables 7 and Table 8, respectively.



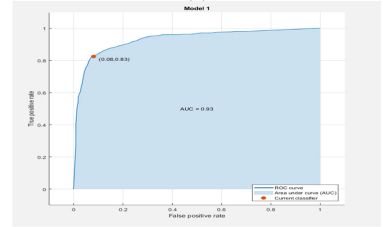
(a)



(b)

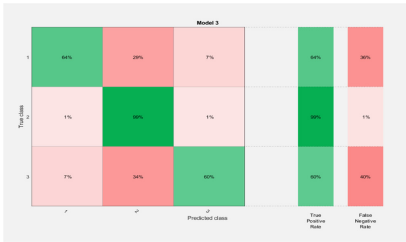


(c)

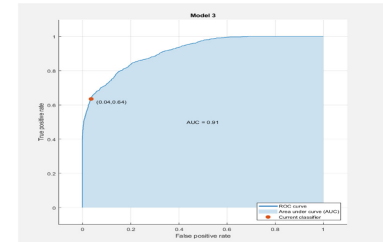


(d)

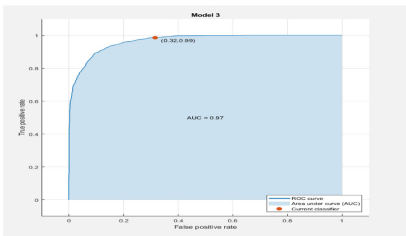
**Fig. 3.** The confusion matrix (a) of the DT classifier and its ROCs (b, c, and d) for tasks  $t_1$ ,  $t_2$ , and  $t_3$ , respectively, for subject 1.



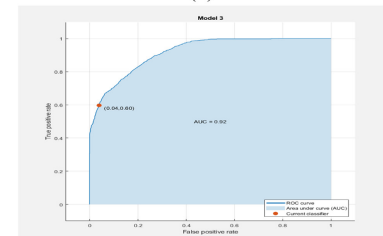
(a)



(b)



(c)



(d)

**Fig. 4.** The confusion matrix (a) of the QA classifier and its ROCs (b, c, and d) for tasks  $t_1$ ,  $t_2$ , and  $t_3$ , respectively, for subject 1.

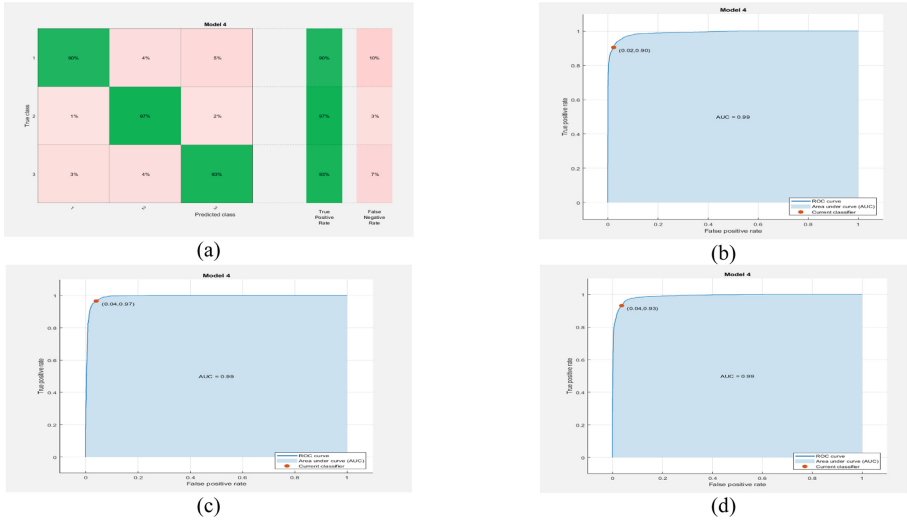


Fig. 5. The confusion matrix (a) of the SVM classifier and its ROCs (b, c, and d) for tasks  $t_1$ ,  $t_2$ , and  $t_3$ , respectively, for subject 1.

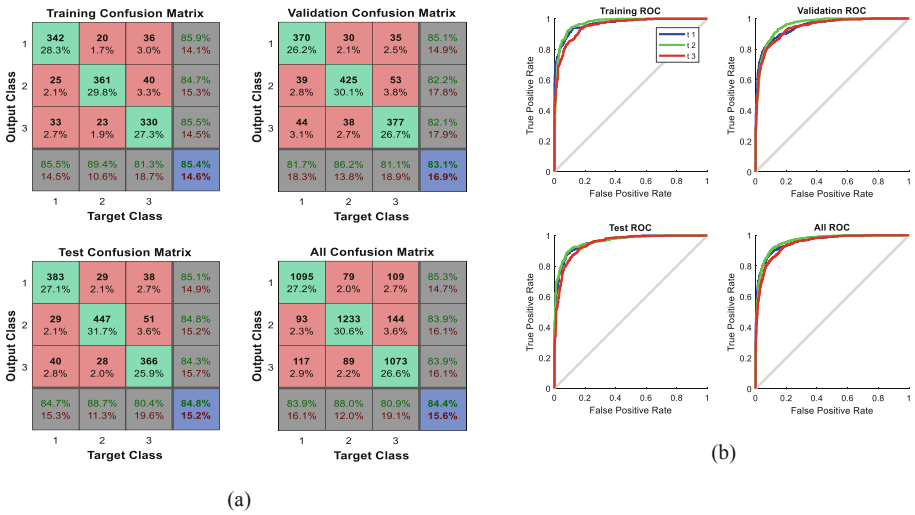
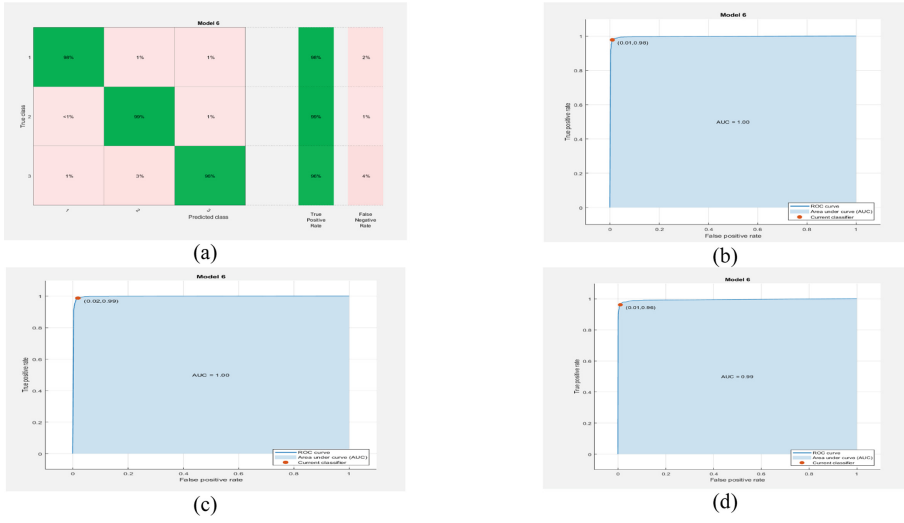
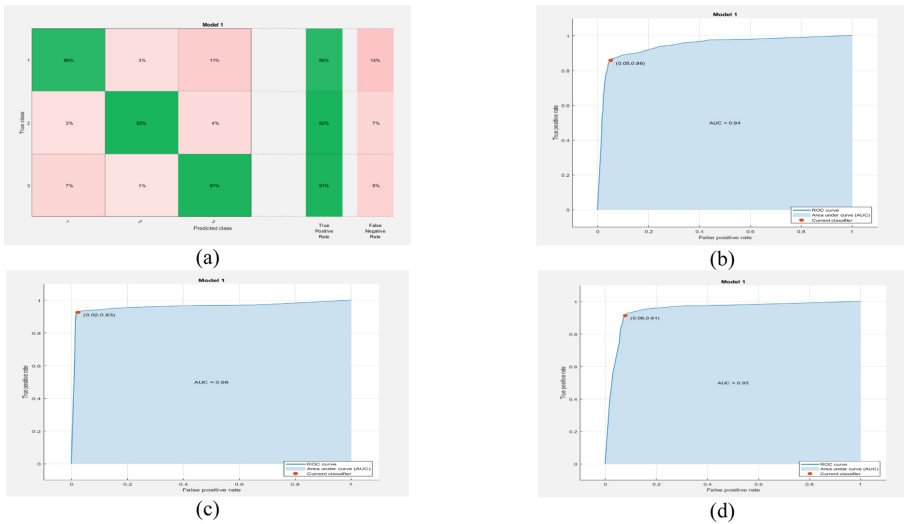


Fig. 6. The confusion matrix (a) of the ANN classifier and its ROCs (b, c, and d) for tasks  $t_1$ ,  $t_2$ , and  $t_3$ , respectively, for subject 1.

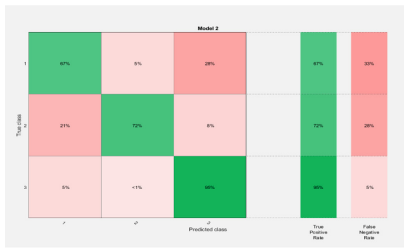


**Fig. 7.** The confusion matrix (a) of the proposed WCMKNN approach and its ROCs (b, c, and d) for tasks  $t_1$ ,  $t_2$ , and  $t_3$ , respectively, for subject 1.

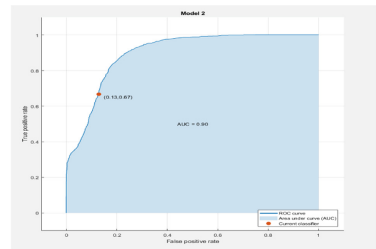


**Fig. 8.** The confusion matrix (a) of the DT classifier and its ROCs (b, c, and d) for tasks  $t_1$ ,  $t_2$ , and  $t_3$ , respectively, for subject 2.

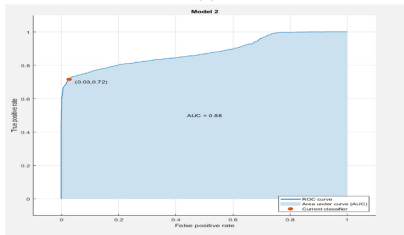




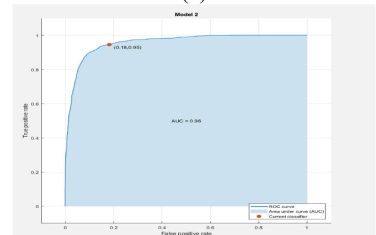
(a)



(b)

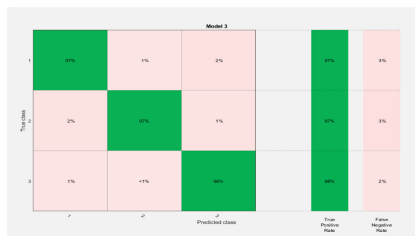


(c)

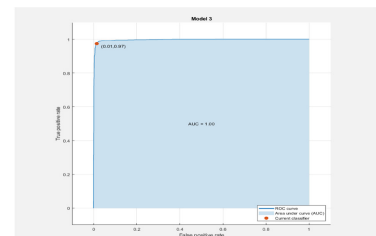


(d)

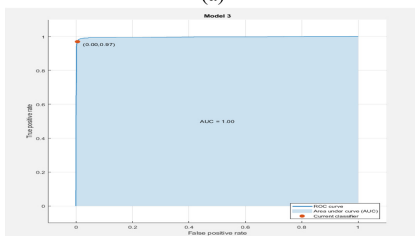
**Fig. 9.** The confusion matrix (a) of the QA classifier and its ROCs (b, c, and d) for tasks  $t_1$ ,  $t_2$ , and  $t_3$ , respectively, for subject 2.



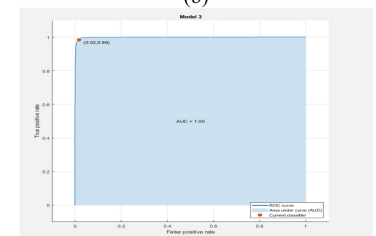
(a)



(b)



(c)

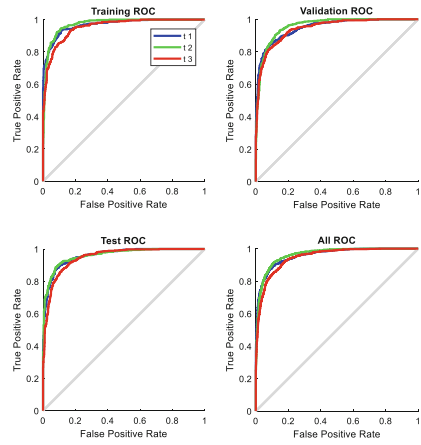


(d)

**Fig. 10.** The confusion matrix (a) of the SVM classifier and its ROCs (b, c, and d) for tasks  $t_1$ ,  $t_2$ , and  $t_3$ , respectively, for subject 2.

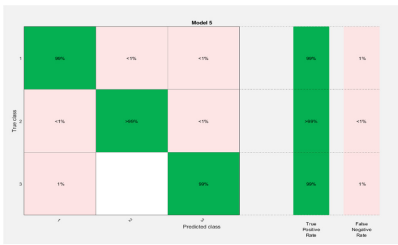


(a)

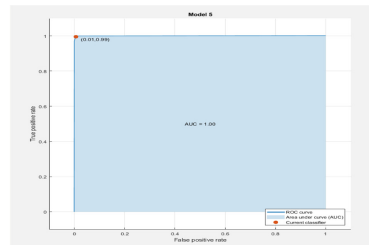


(b)

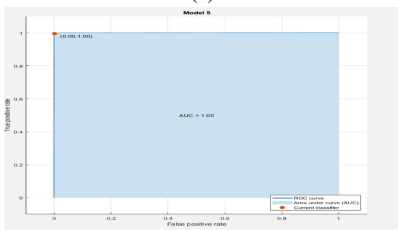
**Fig. 11.** The confusion matrix (a) of the ANN classifier and its ROCs (b, c, and d) for tasks  $t_1$ ,  $t_2$ , and  $t_3$ , respectively, for subject 2.



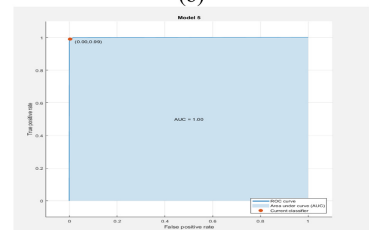
(a)



(b)



(c)



(d)

**Fig. 12.** The confusion matrix (a) of the proposed WCMMKNN approach and its ROCs (b, c, and d) for tasks  $t_1$ ,  $t_2$ , and  $t_3$ , respectively, for subject 2.

**Table 5.** Classification accuracy for the QA, DT, SVM, ANN, and WCMMKNN models for subject 1, utilizing 10 channels.

Classifier model	Cognitive tasks			Overall accuracy
	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	
QA	50%	99%	53%	67.9%
DT	78%	82%	80%	80.1%
SVM	84%	93%	91%	89.4%
ANN	82.1%	78%	72.2%	77.2%
WCMMKNN	97%	98%	94%	96.4%

**Table 6.** Classification accuracy for the QA, DT, SVM, ANN, and WCMMKNN models for subject 1, utilizing 8 channels.

Classifier model	Cognitive tasks			Overall accuracy
	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	
QA	44%	98%	40%	61.4%
DT	76%	80%	80%	78.6%
SVM	75%	87%	89%	83.7%
ANN	77.5%	68.6%	64.2%	68.4%
WCMMKNN	92%	95%	91%	92.7%

**Table 7.** Classification accuracy for the QA, DT, SVM, ANN, and WCMMKNN models for subject 2, utilizing 10 channels.

Classifier model	Cognitive tasks			Overall accuracy
	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	
QA	56%	70%	90%	73.1%
DT	84%	93%	90%	89.1%
SVM	96%	95%	99%	96.6%
ANN	72.8%	87.5%	80%	79.9%
WCMMKNN	99%	99%	<99%	98.9%

**Table 8.** Classification accuracy for the QA, DT, SVM, ANN, and WCMMKNN models for subject 2, utilizing 8 channels.

Classifier model	Cognitive tasks			Overall accuracy
	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	
QA	50%	64%	93%	68.9%
DT	85%	93%	89%	89.2%
SVM	93%	92%	96%	93.7%
ANN	54.4%	86.6%	59.2%	65.6%
WCMMKNN	97%	98%	98%	97.6%

## 5 Conclusion

The cognitive task classification is an important component of many EEG-based applications such as Brain Computer Interface and epilepsy treatment. In this work, we proposed a weighted combination of multiple  $k$ -nearest neighbor approach. The new approach is evaluated using two important criteria: the receiver operating characteristics (ROC) curves and the confusion matrices. Extensive experimental work has been carried out on two subjects and the performance of the proposed approach along with four other well-known techniques are reported. The new approach outperforms all of them with classification accuracy at 97.6%, utilizing all the available channels, and 96.4% and 92.7% classification accuracies utilizing only 70% and 60% of the available channels, for subject 1, respectively; and classification accuracy at 99.4%, utilizing all the available channels, and 98.9% and 97.6% classification accuracies utilizing only 70% and 60% of the available channels, for subject 2, respectively. For future work, sophisticated methods of feature extraction and selection will be incorporated in the proposed technique to improve the classification accuracy.

**Acknowledgments.** The authors would like to thank the NPRP 09-310-1-058 from the Qatar National Research Fund (a member of Qatar Foundation) for providing the experimental data.

## References

1. Zanzotto, F.M., Croce, D.: Comparing EEG/ERP-like and fMRI-like techniques for reading machine thoughts. In: International Conference on Brain Informatics, pp. 133–144. Springer, Heidelberg, 2010 August
2. Bashashati, A., Fatourehchi, M., Ward, R.K., Birch, G.E.: A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals. *Neural Eng.* **4**(2), R32 (2007)
3. Naselaris, T., Kay, K.N., Nishimoto, S., Gallant, J.L.: Encoding and decoding in fMRI. *Neuroimage* **56**(2), 400–410 (2011)
4. Mitchell, T.M., Hutchinson, R., Niculescu, R.S., Pereira, F., Wang, X., Just, M., Newman, S.: Learning to decode cognitive states from brain images. *Mach. Learn.* **57**(1–2), 145–175 (2004)

5. Mohamed, A., Shaban, K.B., Mohamed, A.: Evidence theory-based approach for epileptic seizure detection using EEG signals. In: 21th IEEE International Conference on Data Mining Workshops, pp. 79–85. IEEE (2012)
6. Sivachitra, M., Vijayachitra, S.: Planning and relaxed state EEG signal classification using complex valued neural classifier for brain computer interface. In: 2015 International Conference on Cognitive Computing and Information Processing (CCIP), pp. 1–4. IEEE, 2015 March
7. Zarjam, P., Epps, J., Lovell, N.H.: Beyond subjective self-rating: EEG signal classification of cognitive workload. *IEEE Trans. Auton. Mental Dev.* **7**(4), 301–310 (2015)
8. Mazumder, A., Rakshit, A., Tibarewala, D.N.: A back-propagation through time based recurrent neural network approach for classification of cognitive eeg states. In: 2015 IEEE International Conference on Engineering and Technology (ICETECH), pp. 1–5. IEEE, March 2015
9. Sreeshakthy, M., Preethi, J.: Classification of emotion from EEG using hybrid radial basis function networks with elitist PSO. In: 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), pp. 1–4. IEEE, January 2015
10. Dobarjeh, M.G., Wang, G.Y., Kasabov, N.K., Kydd, R., Russell, B.: A spiking neural network methodology and system for learning and comparative analysis of EEG data from healthy versus addiction treated versus addiction not treated subjects. *IEEE Trans. Biomed. Eng.* **63**(9), 1830–1841 (2015)
11. Tomasiello, S.: A granular functional network classifier for brain diseases analysis. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pp. 1–7 (2019)
12. Wang, S., Gwizdka, J., Chaovalitwongse, W.A.: Using wireless EEG signals to assess memory workload in the  $n$ -back task. *IEEE Trans. Hum. Mach. Syst.* **46**(3), 424–435 (2015)
13. Zhang, J., Yin, Z., Wang, R.: Pattern classification of instantaneous cognitive task-load through GMM clustering, laplacian eigenmap, and ensemble SVMs. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **14**(4), 947–965 (2016)
14. De, A., Konar, A., Samanta, A., Biswas, S., Ralescu, A.L., Nagar, A.K.: Cognitive load classification in learning tasks from hemodynamic responses using type-2 fuzzy sets. In: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–6. IEEE, July 2017



# Detection and Localization of Breast Tumor in 2D Using Microwave Imaging

Abdelfettah Miraoui<sup>1,2(✉)</sup>, Lotfi Merad Sidi<sup>2,3</sup>, and Mohamed Meriah<sup>2</sup>

<sup>1</sup> Mascara Faculty of Technology Telecommunications, University Mustapha Stambouli, Mascara, Algeria

[af.miraoui@yahoo.fr](mailto:af.miraoui@yahoo.fr)

<sup>2</sup> Faculty of Technology, Telecommunications Laboratory, University Abou Bekr Belkaid-Tlemcen, Tlemcen, Algeria

<sup>3</sup> Department of Physics, Preparatory School in Science and Technology, Bel-Horizon, Tlemcen, Algeria

**Abstract.** In recent years, microwave imaging has gained a remarkable significance due to its numerous applications in different domains like civil engineering, meteorology; detection and localization of people through the walls and medicine (detection and localization of breast cancer). Through this research, we suggest a novel technique in order to detect and localize the breast tumor in 2D. The recommended method relies on a technique called artificial neural networks (ANN). Using the EM simulator CST, a sphere-shaped tumor was created and settled at arbitrary locations in a breast model. The equipment used was bow-tie antennas for the transmission and reception of Ultra-Wide Band (UWB) signals at 4 GHz. The simulation results are extremely satisfying in terms of detecting and localizing the breast tumor.

**Keywords:** UWB antennas · Artificial Neural Network (ANN) · Breast cancer

## 1 Introduction

Recently, breast cancer is affecting many women but the early recognition will help in fast and efficient treatment. X-ray mammography is the most efficient technique used for imaging method for the clinical detection of the occult breast cancer. Although the noticeable progress in enhancing mammographic techniques for detecting and characterizing breast lesions, mammography showed a report of high false-negative rates [1] and high false-positive rates [2]. These difficulties are related to the intrinsic contrast between normal and malignant tissues at X-ray frequencies. In the soft tissues like human breast, it is difficult for X-ray to scan and show the breast anomalies at an early stage; as well as the variation in density between normal and malignant breast tissues is not significantly detectable.

With the new medical progress, microwave imaging is a modern technology which has potential applications in the domain of diagnostic medicine [4, 5]. This technology

works on the molecular interactions (dielectric) rather than atomic (density) based on the microwave radiation with the target compared to X-ray imaging.

Many studies rely on microwave as a powerful electromagnetic tool to recover the physical and electrical properties of objects. For the application of detection and localization of breast cancer, microwave imaging has been extremely efficient in achieving the desired results which are detecting and localizing the tumor at a very early stage. The difference between the dielectric properties of breast tissue shows the creation of multiple scattering waves in these tissues which presents a nonlinear inverse scattering problem. In this work, we use a novel technique based on neural networks which was recently introduced in the detection and localization of objects using microwave imaging without solving this problem [5]. The proposed technique has been already used, however, the database used is not large enough and the precision of localization is not better [6].

In our study, we have improved the results by increasing the precision of detection and localization of tumors. For this latter, we used a larger database the one used in [6] with other learning algorithm. A spherical tumor was created and put at random locations in a breast model using EM simulator. With the use of bow-tie antennas that have transmitted and received Ultra-Wide Band (UWB) signals of 4 GHz, the simulation results depict that the ANN presents more precision in the detection and localization of tumor.

## 2 Breast Model for Data Collection

In the literature, different dimensions of the breast model have been used. The choice of model depends on the intended application, in our application, we used a model given by “S. A. AlShehri and S. Khatun” Fig. 1 and Table 1 [6].

**Table 1.** Model parts sizes

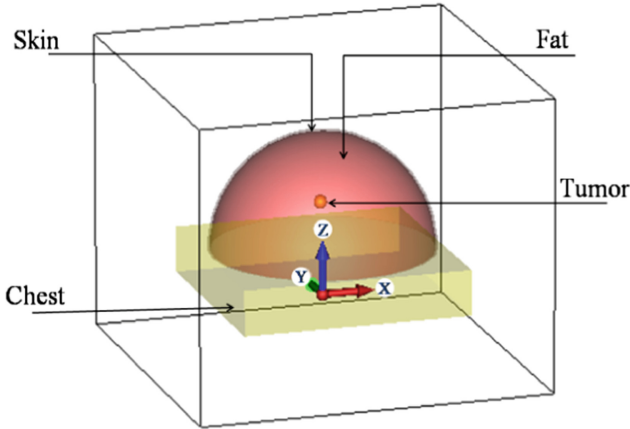
Model part	Size (cm)
Breast diameter	10
Breast height	6
Skin thickness	0.2
Chest thickness	2

Table 2 show the dielectric properties used in model part. Where  $\sigma$  is the tissue conductivity in (Siemens/meter) and  $\epsilon_r$  is the relative permittivity [7].

In the literature [8, 9] the tumor radius size ranges from 0.2 cm to about 1.5 cm or more. In this case, we took a tumor with a radius of 0.25 cm is close to its minimum size.

**Table 2.** Dielectric properties

	Conductivity $\sigma$ (S/M)	Permittivity $\epsilon_r$
Skin	1.49	37.9
Fat	0.14	5.14
Chest	1.85	53.5
Tumor	1.20	50

**Fig. 1.** View of the model in CST

### 3 Detection and Localization in Two Dimension (2D)

The construction of the neural network is done through an iterative process on samples of a previously built database [10]. This database contains a set of data (input/output) obtained by simulation using the software “CST” Fig. 2. For this one, we proceeded as follows:

- Dispose a pair of transmitter – receiver antenna at opposite sides of breast model.
- Dispose a tumor at any location in the model.
- Transmit a Gaussian pulse of a plane wave in the direction of the x-axis.
- Receive the signal on the opposite side.
- Change tumor location and repeating the steps (c–d).

This process of data generation was performed for 481 different locations by moving the tumor along the axis ‘x’ and ‘y’. Also, breast model without tumor tissue was used 2 times to obtain signals propagated through the breast tissue. As a result, two groups of signals received were formed as follows:

A set of 432 signals (431 with tumor and 1 without tumors) were used for ANN learning.



A set of 50 signals (49 with tumor and 1 without tumors) were used for the test phase of the ANN.

The signals received by the receiver contain a number of samples that can be from 4500 to 7200 samples Fig. 2. To reduce the number of samples and to fix the sampling interval of the signal, we used a Cubic Hermite Interpolating Polynomial to generate a polynomial  $P(x_i)$  while keeping the same pace of the signal [11]. The number of samples obtained after interpolation with a step of 0,01 in the segment (0,3 ns and 3 ns) is 271 samples.

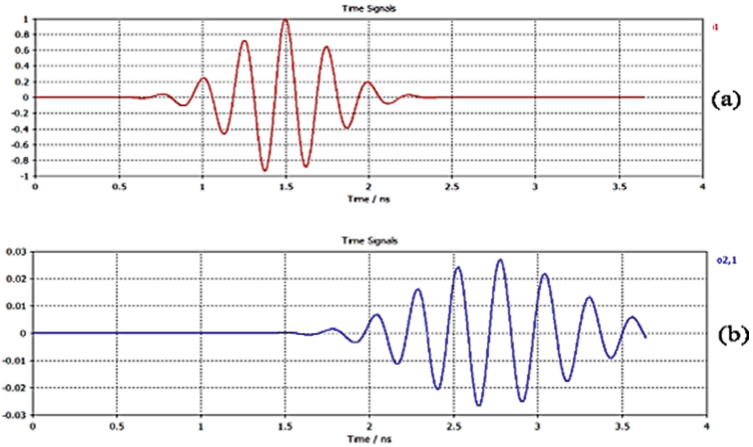


Fig. 2. (a) Transmitted signal, (b) received signal

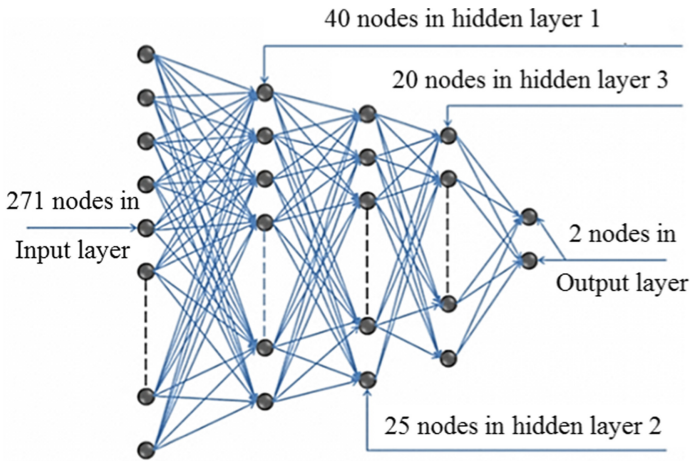


Fig. 3. ANN model

There are no theoretical results, or even empirical rules that allow dimensioning a neural network based problem to resolve. The design of a multilayer network is experimentally selecting the difficulty usually arises when choosing the number of hidden layer and the number of the nodes in each hidden layer. After several tests, a multilayer network was selected with the topology presented in Fig. 3. Therefore, the input layer consists of 271 neurons and the output layer consists in to two neurons representing the position ‘X’ and ‘Y’ of the tumor in the breast model.

The parameters of the topology and learning are summarized in the Table 3 and Fig. 3. The number of samples obtained after interpolation with a step of 0.01 in the segment (0,3 ns and 3 ns) is 271. The input layer contains as many neurons as the number of elements of the input vector.

**Table 3.** ANN parameters

Parameters of ANN	Values
Training function	Traingdm
Number of nodes in input layer	271
Number of nodes in hidden layer 1	40
Number of nodes in hidden layer 2	20
Number of nodes in hidden layer 3	12
Number of nodes in output layer	2
Activation function	Sigmoid
Number of iterations	1 000 000

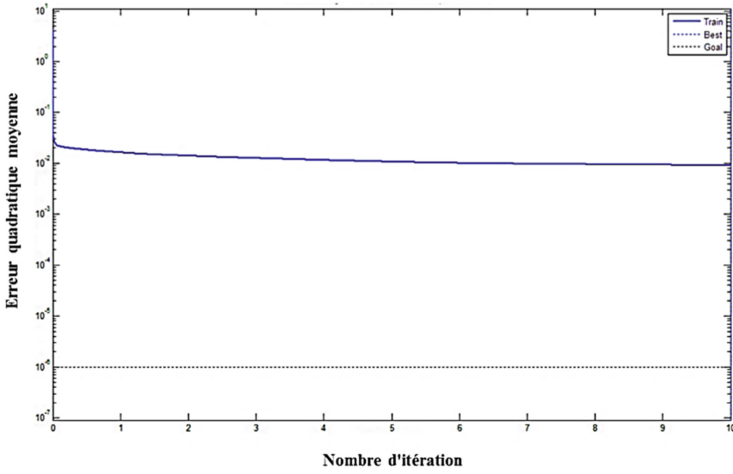
When the learning phase is finished for the learning algorithm Fig. 4, we tested the performance of ANN using group 2 Table 4.

Table 4 shows the results of detection and localization in two dimensions where Gradient descent with momentum algorithm is used for the learning phase. The learning phase lasted 36 h with an error 0,00905.

## 4 Results

Figure 5, 6, 7, 8, 9 and 10 show examples of images for some signals of group (2). We display that for images (5), (6), (7), and (8) the ANN output provides a position of tumor confused with real position of tumor in “CST”, which shows good tumor localization. However, in the image (9) and (10), ANN output gives apposition far to from al tumor position of this latter in “CST” where we have a wrong tumor localization.

Based on Table 4, we note that for input signals without the presence of a tumor, the output of ANN is negative and for input signals which the presence of tumor, the output of ANN is positive this means that the detection rate is almost 100%. We also note, for input signals of ANN in the presence of a tumor at different positions, the output of ANN is very similar to the actual position in “CST” except for 16 cases where the output ANN is relatively far from there al position of the tumor in “CST” Table 4, where the localization rate is the range of 67%.



**Fig. 4.** Performance learning phase for traingdm algorithms

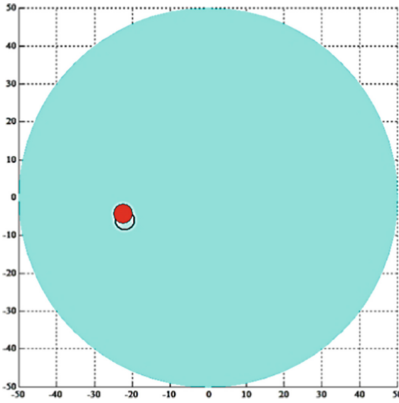
**Table 4.** Actual tumor position and output of ANN (2-D)

Actual tumor location (cm/10)		ANN output (cm/10)	
(X)	(Y)	(X)	(Y)
-1	-1	-1.0116	-0.9924
0.2600	0.4600	0.3077	0.4527
0.2800	0.5600	0.2584	0.4794
0.2800	0.4400	0.2760	0.4576
0.3000	0.5600	0.2685	0.4743
0.3000	0.4600	0.3736	0.4274
0.3200	0.5600	0.2929	0.5165
0.3200	0.4000	0.3486	0.5359
0.3400	0.5400	0.3499	0.4916
0.3400	0.4400	0.3339	0.4656
0.3600	0.5400	0.3762	0.4840
0.3600	0.3800	0.3746	0.5273
0.3800	0.6400	0.4389	0.5202
0.3800	0.5400	0.3773	0.5312
0.3800	0.4600	0.3841	0.5067
0.4000	0.6400	0.4945	0.5078
0.4000	0.5400	0.4062	0.4907

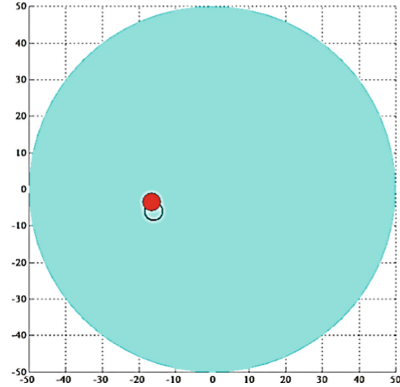
(continued)

**Table 4.** (continued)

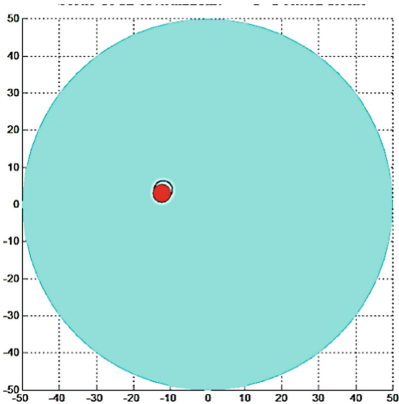
Actual tumor location (cm/10)		ANN output (cm/10)	
(X)	(Y)	(X)	(Y)
0.4000	0.4200	0.3865	0.5524
0.4200	0.6000	0.4080	0.5093
0.4200	0.4600	0.4095	0.4807
0.4400	0.4600	0.3985	0.5275
0.4600	0.7200	0.4019	0.5272
0.4600	0.5600	0.4466	0.5018
0.4600	0.4000	0.4497	0.5240
0.4800	0.5800	0.4703	0.4912
0.4800	0.4400	0.4871	0.4751
0.5000	0.5600	0.5149	0.4977
0.5000	0.4200	0.5111	0.4896
0.5200	0.6000	0.4848	0.4839
0.5200	0.4600	0.4980	0.5246
0.5400	0.6200	0.4920	0.4962
0.5400	0.4000	0.5512	0.4744
0.5600	0.5400	0.5530	0.4990
0.5600	0.2800	0.5725	0.4690
0.5800	0.6600	0.5981	0.5016
0.5800	0.5600	0.5882	0.4896
0.5800	0.4000	0.5617	0.5095
0.6000	0.5400	0.5617	0.5129
0.6000	0.4200	0.5776	0.4744
0.6200	0.5800	0.5974	0.5247
0.6200	0.4400	0.5446	0.4733
0.6400	0.5400	0.6307	0.4660
0.6400	0.4400	0.6578	0.5198
0.6600	0.5800	0.6561	0.5318
0.6600	0.4400	0.6534	0.4805
0.6800	0.5400	0.6833	0.4955
0.6800	0.4000	0.6848	0.5309
0.7000	0.4600	0.6794	0.5020
0.7200	0.5400	0.7413	0.5733
0.7400	0.5400	0.6341	0.5610



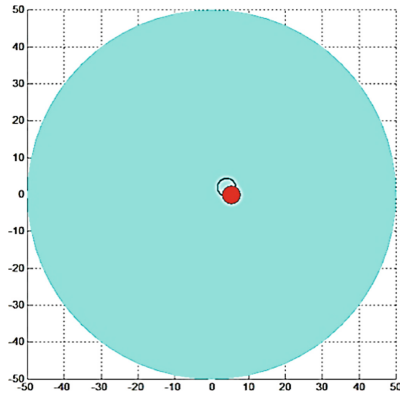
**Fig. 5.** Position of tumor at  $x = 22$  mm;  $y = -6$  mm ( $-50 \leq X \leq 50$  and  $-50 \leq Y \leq 50$ )



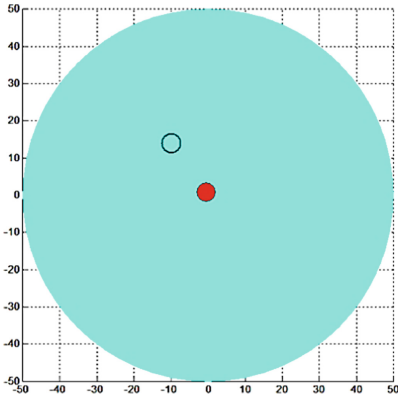
**Fig. 6.** Position of tumor at  $x = -6$  mm;  $y = -6$  mm ( $-50 \leq X \leq 50$  and  $-50 \leq Y \leq 50$ )



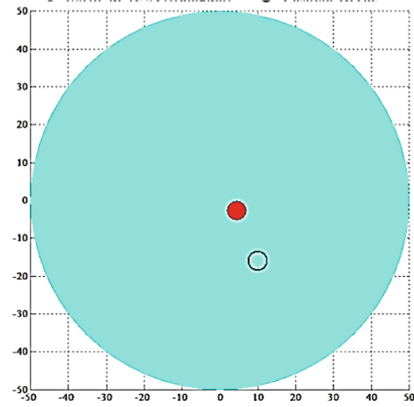
**Fig. 7.** Position of tumor at  $x = -2$  mm;  $y = 4$  mm ( $-50 \leq X \leq 50$  and  $-50 \leq Y \leq 50$ )



**Fig. 8.** Position of tumor at  $x = 4$  mm;  $y = 2$  mm ( $-50 \leq X \leq 50$  and  $-50 \leq Y \leq 50$ )



**Fig. 9.** Position of tumor at  $x = 10$  mm;  $y = 14$  mm ( $-50 \leq X \leq 50$  and  $-50 \leq Y \leq 50$ )



**Fig. 10.** Position of tumor at  $x = -10$  mm;  $y = -16$  mm ( $-50 \leq X \leq 50$  and  $-50 \leq Y \leq 50$ )

## 5 Conclusion

In this work, the Artificial Neural Network was used for detection and localization the tumors in one and two dimensional based on the dielectric properties of human mammary tissues. The received signals were used to construct a database for the learning phase of ANN model. These simulation results are very satisfactory in terms of detection and localization. In future work, we will study, the detection and localization of tumors in three dimensions.

## References

1. Elmore, J.G., Barton, M.B., Mocer, V.M., Polk, S., Arena, P.J., Fletcher, S.W.: Ten year risk of false positive screening mammography and clinical breast examinations. *New England J. Med.* **338**, 1089–1096 (1998)
2. Fear, E.C., Hagness, S.C., Meany, P.M., Okoniewski, M., Stuchly, A.: Enhancing breast tumor detection with near field imaging. *IEEE Microw. Mag.* **3**, 48–56 (2002)
3. Fear, E.C., Li, X., Hagness, S.C., Stuchly, M.A.: Confocal microwave imaging for breast cancer detection: localization of tumors in three dimensions. *IEEE Trans. Biomed. Eng.* **49**, 812–821 (2002)
4. Chaudhary, S.S., Mishra, R.K.A., Swarup, A., Thomas, J.M.: Dielectric properties of normal and malignant human breast tissues at radio wave and microwave frequencies. *Indian J. Biochem. Biophys.* **21**, 76–79 (1981)
5. Lazebnik, M.A.: A large-scale study of the ultra-wideband microwave dielectric properties of normal, benign and malignant breast tissues obtained from cancer surgeries. *Phys. Med. Biol.* **52**, 6093–6115 (2007)
6. Alshetri, S.A., Khatum, S., Jantan, A.B.: UWB imaging for breast cancer detection using neural network. *Progr. Electromag. Res. C* **7**, 79–93 (2009)
7. Miraoui, A., Merad, L., Meriah, S.M., Hassain, N., Benahmed, M., Bousahla, M., Taleb, A., Ahmed, A., Belarbi, B.: Microwave imaging for breast cancer detection using artificial neural network. In: *International Congress on Telecommunication and Application*, University of AMIRA Bejaia, Algeria, AM-P03 (2012)

8. Miyakawa, M., Ishida, T., Wantanabe, M.: Imaging capability of an early stage breast tumor by CP-MCT. In: Proceedings of the 26th Annual International Conference of the IEEE EMBS, San Francisco, CA, USA, vol. 1, pp. 1427–1430 (2004)
9. Fear, E.C., Stuchly, M.A.: Microwave detection of breast cancer. *IEEE Trans. Microw. Theory Tech.* **48**, 1854–1863 (2000)
10. Wang, M., Yang, S., Wu, S., Luo, F.: ARBFNN approach for DoA estimation of ultra-wideband antenna array. *Neurocomputing* **71**, 631–640 (2008)
11. Salmon, S.: *Analyse Numérique*, Université Louis Pasteur, L2 Mathématiques (2005–2006)



# Regression Analysis of Brain Biomechanics Under Uniaxial Deformation

O. Abuomar<sup>1</sup>(✉), F. Patterson<sup>2,3</sup>, and R. K. Prabhu<sup>2,3</sup>

<sup>1</sup> Department of Computer and Mathematical Sciences, Lewis University, Romeoville, IL, USA  
oabuomar@lewisu.edu

<sup>2</sup> Department of Agricultural and Biological Engineering, Mississippi State University, Starkville, MS, USA

<sup>3</sup> Center for Advanced Vehicular Systems, Mississippi State University, Starkville, MS, USA

**Abstract.** Traumatic brain injury is one of the most prevalent health conditions in the United States. However, despite its significance and frequency there is not that much understanding of the mechanism that controls the brain response during injurious loading. Because brain testing conditions are different between several assessment methods, this is considered as a confounding problem as brain biomechanics cannot be analyzed and understood completely. Multivariate linear regression has been applied in this article as a statistical method to expound the correlations between brain biomechanical response and in vitro brain testing conditions under uniaxial deformation. Neighborhood component analysis has been used to extract ten relevant continuous parameters, namely, age, strain rate, diameter, thickness, length, width, height, storage temperature, testing temperature, and post-mortem preservation time, five different categorical parameters, namely, stress condition, species, specimen location, brain matter composition, and geometry. In addition, multivariate regression model has been estimated with the storage, loss, and complex moduli as the responses. Intercept, strain rate, gray brain matter, and white brain matter have been discovered to be the most consistently significant parameters across the three response variables.

**Keywords:** Multivariate linear regression · Uniaxial deformation · Neighborhood component analysis · Traumatic brain injury

## 1 Introduction

Traumatic brain injury (TBI) is a highly prevalent health condition in the US. In 2013, the number of emergency room visits were 2.5 million [1] and there were 56,000 people died from TBI and another 280,000 were hospitalized [1]. TBI can lead to a lifelong disability, with 5.3 million people currently live with a TBI-related disability in the US [1]. The most possible causes of TBI are motor vehicle accidents, falls, and blunt force trauma. In 2010, TBI's cost on the US economy was around \$76.5 billion, most of that cost resulting from hospitalization and fatalities [1].

In order to prevent and treat TBI effectively, the brain's mechanical response to traumatic loads must be determined. However, there is a high degree of variability of



all in vitro biomechanical studies that attempted to establish such response. It has been hypothesized that this variability may be due to brain specimens' intrinsic properties as well as due to various testing conditions [2]. For example, the brain becomes stiffer with age [3, 4] and its mechanical response is sensitive to both temperature [5, 6] and post-mortem preservation time [6, 7].

Although there are studies in literature investigating the influence of these parameters, characterizing brain biomechanical response under different conditions was rarely studied before. Prior work utilized an unsupervised learning approach in order to determine the effect of different testing conditions on the brain under high strain rate compression, discovering that age, strain rate, and brain matter composition have the highest impact on the brain's mechanical response [8]. In this study, a data analysis method is utilized where multivariate linear regression (MLR) has been applied to brain data in order to estimate the relationship between different testing conditions and brain specimen properties and the dynamic brain's response under uniaxial deformation conditions.

## 2 Materials and Methods

### 2.1 Data Source

Data was collected from a variety of sources in literature [2, 4, 5, 9–14]. Data samples were collected or calculated using a plot digitizer which was developed by Ankit Rohatgi [15], or by using other properties. The storage (E1), loss (E2), and complex (E3) moduli were considered to be the dependent variables that represent three different responses.

### 2.2 Neighborhood Component Analysis

Prior to regression, as a feature selection method neighborhood component analysis (NCA) was utilized to determine which predictor features were most relevant to the dataset.

Briefly, let the training set be for  $n$  observations be,

$$T = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, n\}, \quad (1)$$

Where  $\mathbf{x}_i$  is the  $p$ -dimensional feature vector and  $\mathbf{y}_i \in \mathbb{R}$  is the continuous response variable [16]. The most relevant features can be determined by minimizing the objective function:

$$f(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^n l_i + \lambda \sum_{r=1}^p w_r^2. \quad (2)$$

The predictor variables selected using NCA are shown in Table 1.

### 2.3 The General Multivariate Linear Regression Model

Multivariate linear regression (MLR) can be used to fit a linear combination of predictor variables to a multivariate response vector and to predict any other future outputs from the fitted model [17]. A response cannot be usually interpreted by a single predictor

variable and so the predictions from such a model may be inaccurate. Thus, a more complex model will be more helpful to predict new responses.

A general form of multivariate linear model is:

$$\mathbf{Y}_{n \times d} = \mathbf{X}_{n \times (p+1)} \mathbf{B}_{(p+1) \times d} + \mathbf{E}_{n \times d} \quad (3)$$

or it can be represented using the expanded matrix form as:

$$\begin{bmatrix} y_{11} & y_{12} & \dots & y_{1d} \\ y_{21} & y_{22} & \dots & y_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \dots & y_{nd} \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_{01} & \beta_{02} & \dots & \beta_{0d} \\ \beta_{11} & \beta_{12} & \dots & \beta_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{p1} & \beta_{p2} & \dots & \beta_{pd} \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \dots & \varepsilon_{1d} \\ \varepsilon_{21} & \varepsilon_{22} & \dots & \varepsilon_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{n1} & \varepsilon_{n2} & \dots & \varepsilon_{nd} \end{bmatrix}, \quad (4)$$

where  $n$  is the number of data points,  $p$  is the number of parameters, and  $d$  is the number of dimensions. Covariance-weighted least squares can be used to estimate the regression coefficients, such that the solution is the vector  $\mathbf{b}$  which minimizes the following:

$$\sum_{i=1}^n (\mathbf{y}_i - \mathbf{X}_i \mathbf{b})' \mathbf{C}_0 (\mathbf{y}_i - \mathbf{X}_i \mathbf{b}), \quad (5)$$

The standard errors of the regression coefficients were calculated and the statistical significance of the regression coefficients was determined using the t-test.

### 3 Results

#### 3.1 Features Selected from Neighborhood Component Analysis

NCA reduced the dimensionality of the uniaxial deformation data from thirty variables in the original dataset to ten. The most relevant features selected based on feature weight were strain rate, white brain matter, gray brain matter, thickness, post-mortem preservation time, storage temperature, age, and specimen length.

As indicated in Table 1, these features have the required variability. However, the categorical parameters (i.e., stress condition, species, specimen location, brain matter composition, and geometry) don't have as much variability trends as in the other features utilized in this study.

#### 3.2 Estimation of Uniaxial Deformation Regression Model

Following dimensionality reduction by NCA, a multivariate linear model was utilized for each dataset. For the uniaxial data, as shown in Table 2, E1 contained only one

**Table 1.** Uniaxial deformation feature inputs for dimensionality reduction by neighborhood component analysis

	Feature	Range	Mean	Standard deviation
<i>Continuous</i>	Age (mo)	6–867	175.7	320.9
	Log[Strain rate] ( $s^{-1}$ )	$6.4 \cdot 10^{-6}$ –3,000	469.4	932.9
	Diameter (mm)	3–30	10.0	7.0
	Thickness (mm)	1.7–14.4	4.4	3.9
	Length (mm)	5–50	14.1	11.2
	Width (mm)	5–25.4	11.3	5.2
	Height (mm)	3.5–9	5.1	0.97
	Storage temperature ( $^{\circ}C$ )	0–37	14.8	15.6
	Testing temperature ( $^{\circ}C$ )	21–37	26.6	6.2
	Post-mortem preservation time (h)	0–96	23.6	37.6
<i>Categorical</i>	Stress condition	Tension Compression		
	Species	Porcine Bovine Caprine Human Canine Ovine		
	Specimen location	Sylvian fissure Corona radiata Corpus Callosum Motor Strip Cerebral cortex Frontal lobe Spinal cord Thalamus		
	Brain matter composition	White Gray Mixed		
	Geometry	Prism Cylinder		

statistically significant feature at a 95% confidence interval which is strain rate. However, strain rate was not significant for E2 (the P-value was 0.151), but was significant for E3 (P-value was  $< 0.001$ ). Gray brain matter and white brain matter were both significant for E3 (the P-value  $< 0.01$  for both), but not E1 or E2 as the P-values were 0.456 and 0.392 (for E1), and 0.998 and 0.258 (for E2). The only parameter significant for

E2 was the intercept (P-value < 0.001). The other parameter estimations for uniaxial deformation are given in Table 2 as well. However, as shown in Table 3, after using the Analysis of Variance (ANOVA) method [16, 17], the “F test” for a regression relation was statistically significant for all three response variables and the adjusted coefficients of multiple determination were 0.855, 0.849, and 0.784. The P-value for all of the three responses were statistically significant (<0.001) indicating that the selected responses are important for this dataset. In addition, the mean absolute percentage error (MAPE) for E1, E2, and E3 were 58.99%, 39.700%, and 50.239% respectively (Table 4).

**Table 2.** Estimates of parameters for the uniaxial deformation regression model

Feature	E1			E2			E3		
	Estimate	Standard error	P	Estimate	Standard error	P	Estimate	Standard error	P
Intercept	<b>9.11·10<sup>-1</sup></b>	<b>3.17·10<sup>-1</sup></b>	<b>&lt;0.01</b>	<b>7.24·10<sup>-1</sup></b>	<b>3.17·10<sup>-1</sup></b>	<b>&lt;0.001</b>	<b>8.77·10<sup>-1</sup></b>	<b>3.17·10<sup>-1</sup></b>	<b>&lt;0.001</b>
Age	-5.64·10 <sup>-4</sup>	2.51·10 <sup>-3</sup>	0.589	-2.51·10 <sup>-3</sup>	2.51·10 <sup>-3</sup>	0.526	-3.78·10 <sup>-3</sup>	2.51·10 <sup>-3</sup>	0.504
Strain rate	<b>3.62·10<sup>-1</sup></b>	<b>5.57·10<sup>-3</sup></b>	<b>&lt;0.001</b>	<b>3.49·10<sup>-1</sup></b>	<b>5.57·10<sup>-3</sup></b>	<b>0.151</b>	<b>3.15·10<sup>-1</sup></b>	<b>5.57·10<sup>-3</sup></b>	<b>&lt;0.001</b>
Thickness	-3.16·10 <sup>-2</sup>	3.60·10 <sup>-3</sup>	0.809	-9.25·10 <sup>-3</sup>	3.60·10 <sup>-2</sup>	0.833	-2.64·10 <sup>-2</sup>	3.60·10 <sup>-2</sup>	0.682
Length	-1.26·10 <sup>-3</sup>	1.29·10 <sup>-2</sup>	0.539	-3.17·10 <sup>-3</sup>	1.29·10 <sup>-2</sup>	0.504	-6.35·10 <sup>-3</sup>	1.29·10 <sup>-2</sup>	0.507
Gray brain matter	3.76·10 <sup>-3</sup>	3.37·10 <sup>-1</sup>	0.456	-1.62·10 <sup>-1</sup>	3.37·10 <sup>-1</sup>	0.998	<b>5.34·10<sup>-2</sup></b>	<b>3.37·10<sup>-1</sup></b>	<b>&lt;0.01</b>
White brain matter	9.64·10 <sup>-2</sup>	3.52·10 <sup>-1</sup>	0.392	8.41·10 <sup>-3</sup>	3.52·10 <sup>-1</sup>	0.258	<b>1.68·10<sup>-1</sup></b>	<b>3.52·10<sup>-1</sup></b>	<b>&lt; 0.01</b>
Storage temperature	9.31·10 <sup>-3</sup>	9.53·10 <sup>-3</sup>	0.166	1.97·10 <sup>-2</sup>	9.53·10 <sup>-2</sup>	0.478	1.68·10 <sup>-2</sup>	9.53·10 <sup>-3</sup>	0.480
Post-mortem preservation time	2.23·10 <sup>-3</sup>	2.10·10 <sup>-2</sup>	0.458	2.13·10 <sup>-2</sup>	2.10·10 <sup>-2</sup>	0.157	3.09·10 <sup>-2</sup>	2.10·10 <sup>-2</sup>	0.072

**Table 3.** ANOVA table for the uniaxial deformation regression model

	Source variation	Sum of squares	df	Mean squares	F	P	R <sup>2</sup>	R <sup>2</sup> (adj.)
<b>E1</b>	Regression	73.484	8	91.185	84.391	<0.001	0.865	0.855
	Residuals	11.429	105	0.109				
	Total	84.913	113					
<b>E2</b>	Regression	84.279	8	10.535	80.625	<0.001	0.860	0.849
	Residuals	13.820	105	0.131				
	Total	97.999	113					
<b>E3</b>	Regression	73.548	8	F	52.432	<0.001	0.800	0.785
	Residuals	18.411	105	0.175				
	Total	91.959	113					

**Table 4.** Mean absolute percentage error for the uniaxial deformation regression model

Response variable	Uniaxial deformation		
	E1	E2	E3
MAPE (%)	0.590	0.397	0.502

## 4 Discussion

Both neighborhood component analysis and multivariate linear regression have been applied to uniaxial deformation dataset. NCA allowed for the selection of the most relevant features in predicting stress and uniaxial response in the specimens of brain tissue. MLR was used to fit a response to the selected features and predict new outputs based on new inputs' samples.

In order to highlight the efficiency of the proposed model, a prior published article utilizes MLR and other sets of data analysis techniques [18], that is, MLR was applied to brain data to estimate associations between testing conditions and brain specimen properties with the dynamic response of the brain under *shear loading conditions* where four parameters had the most significance in the biomechanical response: the location of the specimen in the brain stem or thalamus, white matter composition, and post-mortem preservation time [18]. The proposed model, however, provides an initial step forward in order to correlate the conditions in which the specimen is tested with the mechanical response of the brain under *uniaxial deformation*. Three parameters stood out as having the most significance in this mechanical response (excluding the intercept): strain rate, white matter composition, and gray matter composition.

Strain rate has been previously considered to be among the most significant parameters affecting the mechanical response of the brain [8].

There has been an extensive discussion in literature on the differences in brain matter material properties between white and gray matter compositions and between their locations in the brain, although there is variation in the literature on what the differences are [19–21]. In addition, getting consistent specimens of white or gray matter compositions and distinguishing between them is quite difficult.

One limitation is that there are not sufficient variables explicitly considered in the reported physical experiments. Due to the amount of missing information, many sources of data cannot be included. Furthermore, some parameters lack the required variability. For instance, post-mortem preservation time values are 3 h, 4 h, 36 h, and 37 h, but there is not enough data in between those four values to determine a true correlation between post-mortem preservation time and the underlying brain biomechanical response. In addition, the uncertainty in the storage, loss, and complex moduli responses was not taken into consideration.

Future work will include refining the model by including more data samples from other loading conditions, such as compression and tension. In addition, a more comprehensive database for brain mechanical data will be processed and built in order to conduct other set of analyses.

## References

1. Faul, M., Xu, L., Wald, M.M., Coronado, V.G.: Traumatic brain injury in the united states: emergency department visits, hospitalizations and deaths. In: Centers for Disease Control and Prevention, National Center for Injury Prevention and Control, Atlanta, GA, pp. 2002–2006 (2010)
2. Nicolle, S., Lounis, M., Willinger, R.: Shear properties of brain tissue over a frequency range relevant for automotive impact situations: new experimental results. *Stapp Car Crash J.* **48**, 239–258 (2004)
3. Arani, A., et al.: Measuring the effects of aging and sex on regional brain stiffness with MR elastography in healthy older adults. *NeuroImage J.* **111**, 59–64 (2015)
4. Chatelin, S., Vappou, J., Roth, S., Raul, J., Willinger, R.: Towards child versus adult brain mechanical properties. *J. Mech. Behav. Biomed. Mater.* **6**, 166–173 (2012)
5. Hrapko, M., Van Dommelen, J.A.W., Peters, G.W.M., Wismans, J.S.: The influence of test conditions on characterization of the mechanical properties of brain tissue. *J. Biomech. Eng.* **130**, 031003 (2008)
6. Zhang, J., et al.: Effects of tissue preservation temperature on high strain-rate material properties of brain. *J. Biomech.* **44**, 391–396 (2011)
7. Garo, A., Hrapko, M., Van Dommelen, J., Peters, G.W.M.: Towards a reliable characterisation of the mechanical behaviour of brain tissue: the effects of post-mortem time and sample preparation. *Biorheol. J.* **44**, 51–58 (2007)
8. Crawford, F., Abuomar, O., Jones, M., King, R., Prabhu, R.: Data mining the effects of testing conditions on brain biomechanical properties. In: Proceedings of the 2017 International Conference on Data Mining, Las Vegas, NV, USA (2017)
9. Brands, D.W., Bovendeerd, P.H., Peters, G.W., Wismans, J.S.: The large shear strain dynamic behaviour of in-vitro porcine brain tissue and a silicone gel model material. *Stapp Car Crash J.* **44**, 249–260 (2000)
10. Forte, A.E., Gentleman, S.M., Dini, D.: On the characterization of the heterogeneous mechanical response of human brain tissue. *Biomech. Model. Mechanobiol.* **16**(3), 907–920 (2016)
11. Hrapko, M., Van Dommelen, J.A.W., Peters, G.W.M., Wismans, J.S.: The mechanical behaviour of brain tissue: large strain response and constitutive modelling. *Biorheol. J.* **43**(5), 623–636 (2006)
12. Thibault, K.L., Margulies, S.S.: Material properties of the developing porcine brain. In: Proceedings of the 1996 International IRCOBI Conference on the Biomechanics of Impact, Dublin, Ireland, pp. 75–85 (1996)
13. Thibault, K.L., Margulies, S.S.: Age-dependent material properties of the porcine cerebrum: effect on pediatric inertial head injury criteria. *J. Biomech.* **31**, 1119–1126 (1998)
14. Vappou, J., Breton, E., Choquet, P., Goetz, C., Willinger, R., Constantinesco, A.: Magnetic resonance elastography compared with rotational rheometry for in vitro brain tissue viscoelasticity measurement. *J. Magn. Reson. Mater. Phys. Biol. Med.* **20**, 273–278 (2007)
15. Rohatgi, A.: WebPlotDigitizer (2016). <http://arohatgi.info/WebPlotDigitizer/app/>
16. Yang, W., Wang, K., Zuo, W.: Neighborhood component feature selection for high-dimensional data. *J. Comput.* **7**(1), 161–168 (2012)
17. Beck, N., Katz, J.N.: What to do (and not to do) with time-series-cross-section data in comparative politics. *Am. Polit. Sci. Rev.* **89**(3), 634–647 (1995)
18. Crawford, F., Fisher, J., Abuomar, O., Prabhu, R.: A multivariate linear regression analysis of in vitro testing conditions and brain biomechanical response under shear loads. In: Proceedings of the 14th International Conference on Data Science (ICDATA 2018), Las Vegas, USA, (2018)

19. Bilston, L.E., Liu, Z., Phan-Thien, N.: Linear viscoelastic properties of bovine brain tissue in shear. *Biorheol. J.* **34**, 377–385 (1997)
20. Ozawa, H., Matsumoto, T., Ohashi, T., Sato, M., Kokubun, S.: Comparison of spinal cord gray matter and white matter softness: measurement by pipette aspiration method. *J. Neurosurg.* **95**(2), 221–224 (2001)
21. Van Dommelen, J.A.W., Van Der Sande, T.P.J., Hrapko, M., Peters, G.W.M.: Mechanical properties of brain tissue by indentation: Interregional variation. *J. Mech. Behav. Biomed. Mater.* **3**, 158–166 (2010)



# Exudate-Based Classification for Detection of Severity of Diabetic Macula Edema

Nandana Prabhu<sup>1,2(✉)</sup>, Deepak Bhoir<sup>2</sup>, Nita Shanbhag<sup>3</sup>, and Uma Rao<sup>4</sup>

<sup>1</sup> K. J. Somaiya College of Engineering, Vidyavihar, Mumbai 400077, Maharashtra, India  
nandanaprabhu@somaiya.edu

<sup>2</sup> Fr. Conceicao Rodrigues College of Engineering, Bandra, Mumbai 400050, Maharashtra, India  
bhoir@fragnel.edu.in

<sup>3</sup> D. Y. Patil University and School of Medicine, Navi Mumbai 400706, Maharashtra, India  
nita.eyegmail.com

<sup>4</sup> Shah and Anchor Kutchi Engineering College, Mumbai 400088, Maharashtra, India  
uma.rao@sakec.ac.in

**Abstract.** Macula Edema is observed in many patients having diabetes for more than ten years. It is more so in patients who have fluctuating sugar level or uncontrolled diabetes. In the case of Macula Edema, in spite of being the commonest cause, the patient realizes the issue, only when there is deterioration of vision. Experts use surrogates such as exudates near to fovea in fundus photographs for detection of Macula Edema through clinical examination. The severity is based on the proximity of the exudates to the fovea. In the present scenario with the rising rate of diabetes, an automated technique can act as an aid for the quick detection of the disease and also adds value to healthcare. This paper proposes a morphological method for extraction of exudates. A novel approach is proposed for locating the macula irrespective of the position of optic disc. The overall accuracy obtained for classification is 94.74%. The balanced accuracy obtained for classification of Normal, Non Clinically Significant Macula Edema and Clinically Significant Macula Edema is 97.92%, 92.42% and 96.77%, respectively.

**Keywords:** Diabetic Retinopathy · Macula Edema · Exudates · Balanced accuracy

## 1 Introduction

Diabetes is a metabolic disorder in which blood sugar levels in the body get elevated. It is either due to the failure of the pancreas to produce sufficient insulin or due to the inability of the body cells to properly utilize the insulin produced by the pancreas. Untreated diabetes in the long run affects the functioning of various parts of the body such as eyes, heart, kidney and nervous system. Different abnormalities caused in the eye due to prolonged diabetes are Diabetic Retinopathy (DR), Macula Edema, Cataract and rarely Glaucoma. DR affects the blood vessels in the retina. It occurs when loss of pericytes of capillaries leave the endothelial versus pericyte ratio to be altered. This makes the capillary wall weak. Blood flow will pre-empt it to cause outpouching to



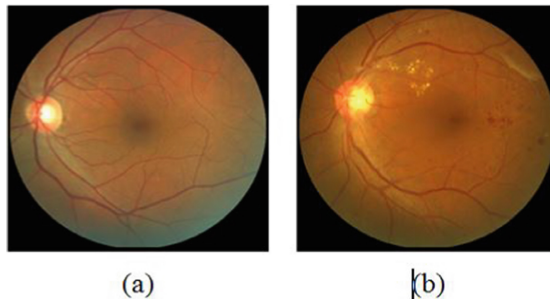
form microaneurysms. Stagnation of blood flow in these microaneurysms will cause leakage of fluid which is revealed as exudates. This becomes foremost identifying factor to classify DR. These abnormalities affect the vision. It is estimated that one third of people affected with diabetes end up with getting DR [1].

The macula is located at the center of the retina. It is responsible for our perception of colours and fine details. The fluid from blood vessels can leak into macula as well. The leakage causes the macula to swell, resulting in Macula Edema. Arrangement of the fibres at macula is radial (Henle's layer). Fluid imbibition will cause retinal cells to be separated causing visual distortion. Accumulation of exudates will block vision. In a study conducted by Roglic et al. on Diabetes in a limited number of countries and extrapolating the data to WHO member states it is estimated that number of people affected by Diabetes would rise to 366 million by the end of year 2030 [2].

Macula Edema is painless and hence patients do not complain of this disease in the beginning. Appearance of exudates indicates Macula Edema. The severity of the disease is judged by the proximity of these exudates to the macula. Stage 1 is mild wherein the symptoms are far away from macula. It is Non Clinically Significant Macula Edema (NCSME). In Stage 2 hard exudates are seen within 500  $\mu\text{m}$  of the macula center with adjacent retina thickening. Stage 2 is severe and is known as Clinically Significant Macula Edema (CSME). It leads to loss of vision.

DR experts use the predefined standard method for determining the Macula Edema by visually estimating the distance of exudates with respect to fovea. This paper proposes to use morphological operations on fundus image for locating the exudates and estimating distance of exudates from fovea in order to predict the severity.

Figure 1 shows fundus images. Figure 1(a) shows the Fundus image of a healthy eye. The anatomic structure of a healthy eye consists of optic disc, blood vessels and macula. Figure 1(b) shows the fundus image of retina affected with Macula Edema.



**Fig. 1.** (a) Fundus image of a normal retina (b) Fundus image of retina affected with Macula Edema

## 2 Related Work

Researchers have proposed different methods for detecting the optic disc(OD), locating the macula and extracting exudates. An overview of several such methods developed over a span of a decade from 2008 to 2018 has been presented in this literature review.

The preprocessing technique used by Soparak et al. is median filtering for noise removal and Contrast Limited Adaptive Histogram Equalization (CLAHE) for contrast enhancement on a HSI color model of the fundus image. Following the preprocessing, OD and blood vessels are eliminated and exudates are extracted using morphological operations [3].

Reza et al. have used average filtering, adaptive histogram equalization and thresholding for preprocessing on the extracted green channel of the fundus image. OD and Blood vessels are eliminated using morphological opening. Extended maxima operator, minima imposition, and watershed transformation are further used prior to the evaluation [4].

A two level strategy for classification of fundus images in to normal and Macula Edema affected images is used by Sai Deepak et al. [5]. First level of classification separating normal and Macula Edema affected cases is performed based on certain global features. Rotation symmetry is employed in the second level for further grading the severity of Macula Edema.

In a different preprocessing approach, Zaidi et al. have applied Gabor filter bank for enhancing the exudates [6]. Morphological closing operation is done for removal of blood vessels and OD detection is done through Hough transform. Naïve Bayes classification is used for detection of disease.

Jaya T. et al. have used a similar approach of morphological operations and Hough transform for removal of OD [7]. However, color and texture features are used as representatives of exudates in order to extract them. Five major texture features used are average grey level, wave, spot, ripple, and edges. Opponent color space is used for better perception of color. Classification is done using Fuzzy Support Machine. A region based approach for detecting the severity grades of DR is reported by Dutta et al. Detection of exudates is done with morphological operations, thresholding and logical operation [8].

Wavelet decomposition and automated lesion segmentation is adopted by Giancardo et al. for extracting the features of exudates [9]. They have performed training using Hamilton Eye Institute Macular Edema Dataset (HEI-MED) and performed testing using MESSIDOR database in the stage of classification.

One more different approach for exudate extraction is reported by Ramya et al. [10]. Though the pre-processing, OD and Macula detection are done using traditional methods, exudate extraction is done by motion pattern estimation. The fundus image is transformed to an intermediate representation and rotated over different angles for extracting the features. The classification is performed using Principal Component Analysis and Gaussian data description. A Comparison of performance of the two methods is presented.

Several Neural Network based classification techniques are also reported in the literature. Franklin et al. have used a multi perceptron neural network to classify each pixel of the fundus image as belonging to an exudate and a non-exudate region [11]. The pre-processing technique reported is converting the RGB image to Lab color space, adjusting the L channel and converting it back to RGB. Non uniform illumination is addressed through contrast enhancement. The inputs for the multi perceptron neural network have fifteen features characterizing the exudates in terms of color, size, shape and edge strength.

Tjandrasa et al. on the other hand, resized the image, extracted the green channel followed by grey scaling. Contrast enhancement is done using morphology [12]. The exudates are extracted using morphological reconstruction. OD removal is performed by determining the maximum intensity pixel. Four features such as area, perimeter, number of centroids and standard deviation are extracted and used as inputs to Support Vector Machine classifier.

A few researchers have used Convolutional Neural Networks (CNN) as an extension to neural network processing. Tan et al. have used CNN for segmentation of fovea, OD and vasculature of retina [13]. In a seven layered architecture (0 to 6), layers 5 and 6 are fully connected. Details of neighbours of each pixel are given as input to network. A patch or the sub image of size  $33 \times 33$  is adopted after experimenting.

A deep learning algorithm with CNN to detect the lesions at image level is used by Quellec et al. [14]. A ten layered CNN with raw pixel intensities as inputs to extract exudates is also reported [15]. Mo et al. have used a two stage approach for detection of Macula Edema. A fully connected residual convolution network is used to extract the exudates in the first stage. This is followed by a second stage of classification using regions of maximum probability for the presence of exudates [16].

Another approach to view retinal images to detect abnormalities is through Optical Coherence Tomography (OCT). It helps to get cross sectional view of the retina. Researchers have used OCT databases for automatic detection of retinal abnormalities. Srinivasan et al. have used block matching and 3D filtering for denoising OCT images. This preprocessing is followed by retinal curvature flattening and image cropping. They have used a multiscale histogram of oriented gradient descriptors as feature vectors and performed classification using support vector machine [17].

Retinal abnormalities such as Serous Retinal Detachment (SRD), Diffuse Retinal Thickening (DRT) and Cystoid Macula Edema (CME) have been detected using OCT images [18].

### 3 Methodology

In this paper, an automated system for exudate detection and classification is proposed. The input images are taken from a random selection of images obtained from tertiary Reference Centre of Medical College. The camera used for capturing the images is Zeiss visucam, and the field of view is  $50^\circ$ . Another set of images are obtained from retina speciality centre. A total of 76 images of different categories belonging to normal, mild NPDR, moderate NPDR, severe NPDR and PDR and belonging to different grades of Macula Edema are used. The images contained the different lesions such as exudates, hemorrhages, microaneurysms, and cotton wools. The size of images is  $2124 \times 2056$  with 24 bits per pixel. They are in JPEG format. The second is public dataset, DIABRETO with 130 images [19]. The size of the images is  $1150 \times 1152$ . All the images are graded by an expert ophthalmologist. This grading is used for evaluation purpose.

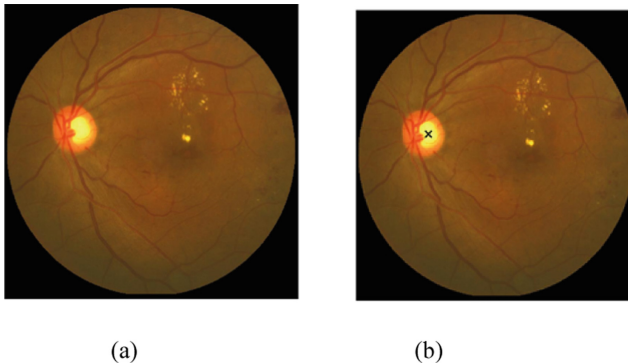
The method consists of preprocessing, OD detection, locating the center of the macula, exudate extraction and calculation of distance between fovea and the exudates.

### 3.1 Preprocessing

Preprocessing is necessary to make the image suitable for effectively extracting the exudates. Fundus images acquired through different cameras provide images of different dimensions, resolution and file formats. Resizing the images to  $512 \times 512$  pixels is done in order to maintain the uniformity. Image cropping is done using circle mask technique to obtain an image containing only the region of interest. These two operations minimize the computational cost.

### 3.2 Optic Disc Detection

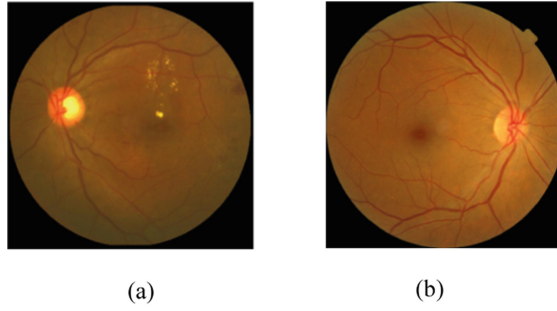
OD is seen as a bright yellowish region in the fundus image. Its intensity is same as that of exudates. Hence, detection, localization and elimination of OD without removing the exudates is quite challenging. We propose a method of template matching followed by cross correlation for detecting the OD. Figure 2 shows detection of centre of optic disc. Fig. 2(a) shows the original fundus image and Fig. 2(b) shows the fundus image with OD centre marked.



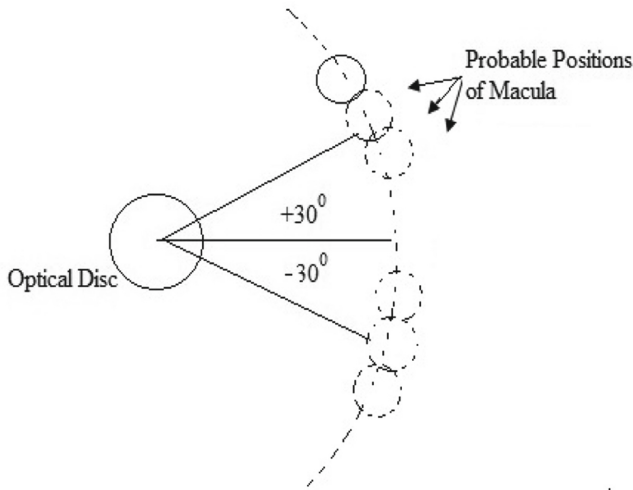
**Fig. 2.** (a) Original image (b) Optic Disc centre detected

### 3.3 Macula Detection

In a fundus image, OD may be located either towards the left or to the right as shown in Fig. 3(a) and (b). From the centre of OD, Macula is approximately at a distance of two and a half times the diameter of OD. In some images, the macula is not horizontally aligned with the OD. Hence to identify the correct location of macula, all the pixels lying on an arc with a radius of two and a half times the diameter of OD and subtending an angle of  $\pm 30^\circ$  are scanned as shown in Fig. 4. Circular regions with size of macula at each pixel on arc are obtained. The one with minimum intensity is considered as macula.



**Fig. 3.** (a) Image with OD at the left side (b) Image with OD at the right side



**Fig. 4.** Diagram showing arc with probable locations of Macula

### 3.4 Exudate Extraction

It is observed that the best contrast among the pixel values is exhibited by the green channel of an RGB image. Hence further processing is done on the green channel. This also reduces the computational cost as we need to handle only one channel. The top hat and bottom-hat transforms are applied to the green channel extracted image.

Subtraction is performed between the two images obtained and only those pixels beyond a particular threshold are retained thereby resulting in image  $I_E$  with exudates extracted. The thresholding at a higher level is important to retain all the pixels genuinely corresponding to exudates. However, some of the bright spots near to blood vessels emerging from the OD still remain as artefacts. In order to handle these artefacts, we need the blood vessel extracted image [8]. To get the blood vessel extracted image  $I_{BV}$ , the extracted green channel is complemented and subjected to adaptive histogram equalization to improve the contrast. Morphological opening, median filtering and binarization as used by Patwari et al. [20] are performed. This process highlights

the blood vessels in the image. Further logical OR operation is performed between the two images,  $I_{BV}$  and  $I_E$ . This gives image  $I_c$ , which has exudates, blood vessels and artefacts. Morphological dilation operation is performed on this image to combine all the bright spots near the blood vessels forming one single component. This largest single component is eliminated. The resulting image, is logically ANDed with dilated  $I_{BV}$  and then subtracted from  $I_s$  thereby retaining only the exudates. Figure 5 shows the steps for extracting exudates.

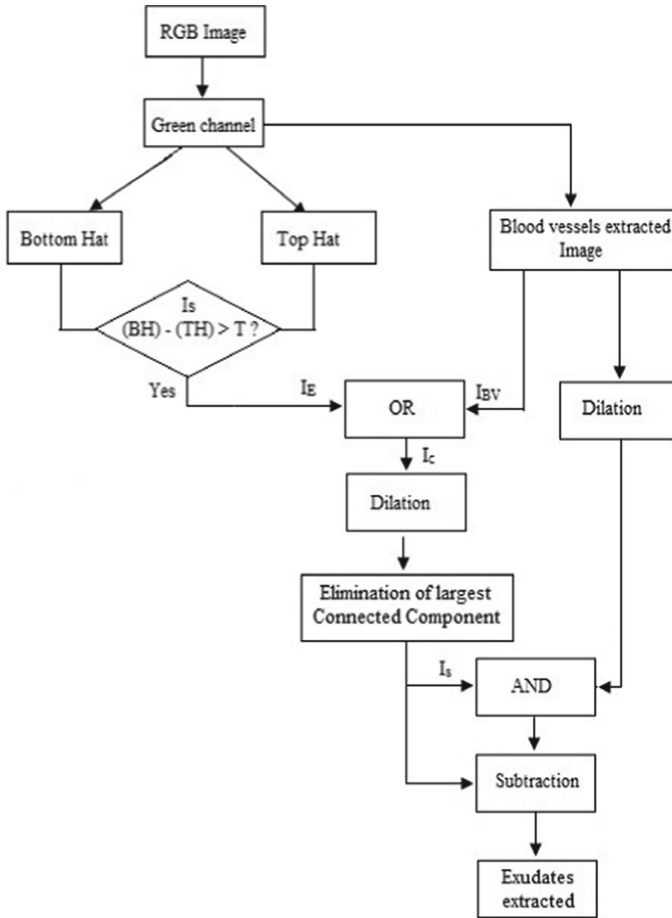
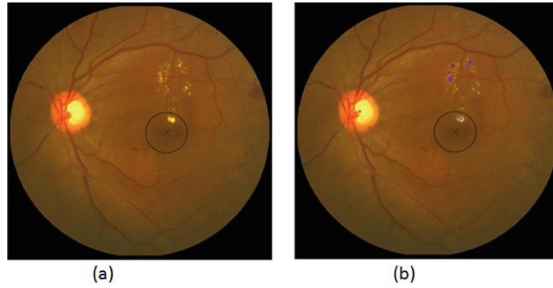


Fig. 5. Steps for extracting exudates

Figure 6 shows the final stages of detection. Figure 6(a) shows Image with macula centre detected and Fig. 6(b), Image with exudates extracted.

Once the exudates are extracted, the Euclidean distance from centre of macula to each of exudates is calculated. The minimum distance is used to determine the severity of macula Edema. If the distance is less than one OD diameter, the severity is considered



**Fig. 6.** (a) Image with Macula centre detected (b) Image with Exudates extracted

as CSME. If it is greater than one OD diameter and less than two OD diameters it is considered as Non CSME.

Performance of the proposed method is evaluated based on accuracy, sensitivity, specificity, and balanced accuracy for each class. Classification is a challenging task when the data set is unbalanced. Confusion matrix is used to determine the performance of the classifier. Balanced accuracy is more suitable for unbalanced data set. It computes the average of the proportion corrects of each class individually.

## 4 Results

The exudates are extracted using the steps mentioned in Sect. 3. The simulation is performed using Matlab2016b and data is analyzed using the R tool. An overall accuracy of 94.74% is achieved for classification. Table 1 shows the classification of Macula Edema based on severity for private data base.

**Table 1.** Results of classification based on severity for private database

Grade	Class	Number of images	Number of correctly detected images	Balanced accuracy	Sensitivity	Specificity
Normal	0	28	28	97.92	100.00	95.83
Non CSME	1	17	15	92.42	88.24	96.61
CSME	2	31	29	96.77	93.55	100.00

Table 2 shows the classification of Macula Edema based on severity for public database. Table 3 shows comparison of the results from the proposed method with similar studies made earlier. The accuracy is improved to a great deal as compared to earlier published work.

**Table 2.** Results of classification based on severity for DIABRETO database

Grade	Class	Number of images	Number of correctly detected images	Balanced accuracy	Sensitivity	Specificity
Normal	0	94	93	93.91	98.94	88.90
Non CSME	1	18	16	92.66	88.89	96.43
CSME	2	18	13	86.11	72.22	100.00

**Table 3.** Performance comparison with other existing methods

Author	Accuracy	Normal	Non CSME	CSME
Deepak et al. [5]	–	–	81	100
Zaidi et al. [6]	94.1	–	–	–
Ramya et al. [10]	92	89.83	94.73	95.45
Senger et al. [21]	80 to 90	80	85	90 to 98
Proposed method	<b>94.74</b>	<b>97.92</b>	<b>92.42</b>	<b>96.77</b>

## 5 Conclusion

Macula Edema is the complication developed with prolonged diabetes. In this work, an automated method for detecting the presence and severity of Macula Edema is proposed. In the preprocessing stage, the input image is resized and cropped to reduce the computational cost. The morphological operations are performed to extract the exudates. In this process, the thresholding needs to be carefully chosen during binarization. Utmost care is taken to locate the macula irrespective of its different orientation with respect to OD. The balanced accuracy obtained for the three classes namely, Normal, Non CSME and CSME are 97.92%, 92.42%, 96.77%, respectively. The reasons for misclassification are very small or pale exudates not getting recognised by the algorithm used. Secondly, at times, if there are dark regions in the image, they may be misclassified as corresponding to macula and distance criteria will not work in these cases.

The implication of these results in screening at distant location through an application will facilitate in identifying and classifying Macula Edema. This in turn will help in better management and treatment of blindness.



## References

1. Lee, R., Wang, T.Y., Sabanayagam, C.: Epidemiology of diabetic retinopathy, diabetic macula edema and related vision loss. In: *Eye vision*, pp. 1–25 (2015)
2. Wild, S., Roglic, G., Green, A., Sicree, R., King, H.: Global prevalence of diabetes: estimates for the year 2000 and projections for 2030. *Diabetes Care* **27**, 1047–1053 (2004). [PMID: 15111519]
3. Sopharak, A., Uyyanonvara, B., Barman, S., Williamson, T.H.: Automatic detection of diabetic retinopathy exudates from non-dilated retinal images using mathematical morphology methods. *Comput. Med. Imaging Graph.* **32**, 720–727 (2008). <https://doi.org/10.1016/j.compmedig.2008.08.009>
4. Reza, A.W., Eswaran, C., Hati, S.: Automatic tracing of optic disc and exudates from color fundus images fixed and variable thresholds. *J. Med. Syst.* **33**, 73–80 (2009). <https://doi.org/10.1007/s10916-008-9166-4>
5. Sai Deepak, K., Siyaswamy, J.: Automatic assessment of macular edema from color retinal images. *IEEE Trans. Med. Imaging* **31**(3), 766–776 (2012). <https://doi.org/10.1109/TMI.2011.2178856>
6. Zaidi, Z.Y., Akram, M.U., Tariq, A.: Retinal Image Analysis for Diagnosis of Macular Edema using Digital Fundus Images. In: *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies* (2013)
7. Jaya, T., Dheeba, J., Singh, N.A.: Detection of hard exudates in colour fundus images using fuzzy support vector machine-based expert system. *J. Digital Imaging* **28**, 761–768 (2015). <https://doi.org/10.1007/s10278-015-9793-5>
8. Dutta, M.K., Ganguly, S., Srivastava, K., Ganguly, S., Parthasarathi, M., Burget, R., Masek, J.: An efficient grading algorithm for non-proliferative diabetic retinopathy using region based detection. In: *IEEE 38th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 743–747 (2015)
9. Giancardo, L., Meriaudeau, F., Karnowski, T.P., Yaqinli, G., Tobin Jr., S.W., Chaum, E.: Exudate based diabetic macula edema detection in fundus images using publicly available datasets. *Med. Image Anal.* **16**(1), 216–226 (2012). <https://doi.org/10.1016/j.media.2011.07.004>
10. Ramya, M., Vijayprasad, S.: An effective analysis of macular edema severity for diabetic retinopathy. *IJIRSET* **3**(3), 739–746 (2014). <http://www.ijirset.com>
11. Franklin, S.W., Rajan, S.E.: Diagnosis of Diabetic Retinopathy by employing image processing technique to detect exudates in retinal images. *Inst. Eng. Technol. Image Process.* **8**(10), 601–609 (2014). <https://doi.org/10.1049/iet-ipr.2013.0565>
12. Tjandrasa, H., Putra, R.E., Wijaya, A.Y., Arieshanti, I.: Classification of non-proliferative diabetic retinopathy based on hard exudates using soft margin SVM. In: *IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia*, pp. 376–380 (2013)
13. Tan, J.H., Acharya, U.R., Bhandary, S.V., Chua, K.C., Sivaprasad, S.: Segmentation of optic disc, fovea and retinal vasculature using a single convolutional neural network. *J. Comput. Sci.* **20**, 70–79 (2017). <http://dx.doi.org/10.1016/j.jocs.2017.02.006>
14. Quelled, G., Charrière, K., Boudi, Y., Cochener, B., Lamard, M.: Deep image mining for diabetic retinopathy screening. *Med. Image Anal.* **39**, 178–193 (2017). <https://doi.org/10.1016/j.media.2017.04.012>
15. Prentašić, P., Lončarić, S.: Detection of exudates in fundus photographs using deep neural networks and anatomical landmark detection fusion. *Comput. Methods Programs Biomed.* **137**, 281–292 (2016). <http://dx.doi.org/10.1016/j.cmpb.2016.09.018>

16. Mo, J., Zhang, L., Feng, Y.: Exudate-based diabetic macular edema recognition in retinal images using cascaded deep residual networks. *Neurocomputing* **290**, 161–171 (2018). <https://doi.org/10.1016/j.neucom.2018.02.035>
17. Srinivasan, P.P., Kim, L.A., Mettu, P.S., Cousins, S.W., Comer, G.M., Izzat, J.A., Farsiu, S.: Fully automated detection of diabetic macular edema and dry age-related macular degeneration from optical coherence tomography images. In: *Biomed. Optic. Express*. (2014). <https://doi.org/10.1364/boe.5.003568>
18. Samagaio, G., Estévez, A., Moura, J., Novo, J., Fernández, M.I., Ortega, M.: Automatic macular edema identification and characterization using OCT images. *Comput. Methods Programs Biomed.* **63**, 47–63 (2018). <https://doi.org/10.1016/j.cmpb.2018.05.033>
19. Kauppi, T., Kalesnykiene, V., Kamarainen, J.K., Lensu, L., Sorri, I., Uusitalo, H., Kälviäinen, H., Pietilä, J.: DIARETDB0 evaluation database and methodology for diabetic retinopathy algorithms. Technical report, Finland (2006). [http://www.it.lut.fi/project/imageret/diaretdb0/doc/diaretdb0\\_techreport\\_v\\_1\\_1.pdf](http://www.it.lut.fi/project/imageret/diaretdb0/doc/diaretdb0_techreport_v_1_1.pdf)
20. Patwari, M.B., Manza, Dr. R.R., Rajput, Y.M., Saswade, M., Deshpande, N.K.: Automatic detection of retinal venous beading and tortuosity by using image processing techniques. *IJCA* (2014). ISBN: 973-93-80880-06-7
21. Senger, N., Dutta, M.K., Burget, R., Povoda, L.: Detection of diabetic macula edema in retina images using a region based method. In: *IEEE TSP*, pp. 412–415 (2015)



# Analysis and Detection of Brain Tumor Using U-Net-Based Deep Learning

Vibhu Garg, Madhur Bansal, A. Sanjana, and Mayank Dave<sup>(✉)</sup>

Department of Computer Engineering, National Institute of Technology, Kurukshetra,  
Kurukshetra, India

g.vibhu05@gmail.com, bansalmad210@gmail.com,  
sanjanaannamaneni@gmail.com, mdave@nitkkr.ac.in

**Abstract.** Brain tumor could be a life threatening disease and the survival rate of such disease is low. It is generally the abnormal growth of cells inside the brain. Early and accurate detection of the brain tumor is very difficult. The manual segmentation of the brain tumor extent from 3D MRI (Magnetic Resonance Imaging) volumes is a time consuming process and depends a lot on the operator's experience. The automatic tumor segmentation has the potential to decrease lag time between diagnosis tests and the treatment for the same. Hence, there is a high demand of time and memory efficient, and reliable computer algorithms to do this accurately and quickly. In this paper, we first highlight limitations of the image processing based solutions and subsequently present a novel deep learning based technique. The proposed technique relies on U-Net based Deep Convolutional Networks for the automatic detection and analysis of brain tumors.

**Keywords:** Brain tumor · Segmentation · Multimodal MRI · Thresholding · Deep neural networks

## 1 Introduction

Multimodal MRI (Magnetic Resonance Imaging) is a popular technology used for the detection of brain tumor by observing the soft tissues. The MRI images are better in terms of quality as compared to other non-invasive imaging techniques such as X-Ray or Computed Tomography. Brain tumors could be life threatening and thus the accurate and timely detection of brain tumors is important. The brain tumors are classified into two groups – benign and malignant tumors. The malignant ones consist of fast growing cancerous tissues as compared to the benign tumors. The MRI images consist of weighted images or segments: T1-weighted, T2-weighted, Flair-weighted (Fluid Attenuated Inversion Recovery) and T1c. Obtaining these images is a difficult problem because the manual segmentation is a time-consuming process and the accuracy depends a lot on the experience of the operator. The process of MRI scan is also quite exhaustive and needs a lot of efforts at the operator's part. Any mistake(s) at the operator's part will lead to chaos due to incorrect diagnosis. The conclusions also may vary from one operator to another operator. Hence, the efficient and reliable computer algorithms are

required to solve this problem. The possible solutions of automatic brain tumor segmentation are image processing based and deep learning based. The solutions strictly based on image processing techniques like Thresholding Based Segmentation [1] are insufficient in brain tumor detection. The automatic segmentation and further classification of tumor into various other types from the multimodal MRI scans remains a popular area of research in the field of medical science. The proposed model in this paper extracts features using Convolutional Neural Networks (CNN) technique and then tries to learn the characteristics of tumors through extensive training on a high-quality dataset. This model is used to detect the brain tumor after performing segmentation on the given MRI scan of a patient.

## 2 Literature Survey

In the field of biomedical imaging, the segmentation of tumor from the MRI scans of a human brain has become an important area of research for detecting tumors [2]. The automatic brain tumor segmentation is not easy due to high variation of brain tumors in size, shape, location, etc. The variations in different sub-regions of tumors and various types of brain tumors are visualized by using multimodal-MRI data. An early study for segmentation of brain tumor from MRI scans is made in [3]. The authors propose a rule-based expert system for tumor prediction based on an unsupervised clustering algorithm for segmenting the image. The multimodal imaging techniques enable examination or visualization of more than one tissue at a time. The existing techniques for segmentation of brain tumor from multimodal MRI images can be broadly classified into four categories which are threshold based segmentation, edge based segmentation, region based segmentation and clustering based segmentation. In [4] a survey on detection of brain tumor from MRI images is presented. In [5] the authors present a detailed survey of MRI-based brain tumor segmentation techniques and also mention some open tools and databases that may be used for making studies for brain tumor segmentation. Some important image processing based segmentation techniques are as follows [1]:

### 2.1 Thresholding Technique

The tumor affected region is of high intensity pixels as compared to the healthy region of the brain, which are of low intensity pixels. In this segmentation technique, the intensity is considered as a major parameter and so this method is suitable for images with different intensities of pixels. The technique classifies the tumor based on gray-level. Using this method, the image is partitioned directly into different regions i.e. healthy region and infected region based on the appropriate threshold value. There are two types of thresholding techniques – global and local.

### 2.2 Region-Based Image Segmentation

This method divides an image into regions that are similar on the basis of a set of a particular criterion. The region-based segmentation is good for high contrast images. However, for low contrast images it does not provide efficient results. In this method,

the intensity of same image is grouped into one region using 4 or 8 pixel-connected neighborhood. If the intensity belongs to the same seed, it belongs to one region and similarly, the process is repeated.

### 2.3 Edge-Based Segmentation

This method divides an image based on sudden changes in the intensity of pixels near the edges in the image. The result is a binary image with edges of the objects being detected.

### 2.4 Clustering-Based Segmentation

In the case of clustering based segmentation techniques, an image is divided into a number of clusters based on the value of membership functions allotted to each pixel in the image.

The medical image analysis and segmentation problems present several unique challenges. The patient data in medical imaging tends to be extremely heterogeneous. Also, the available data-sets for training purpose are extremely less and not easily available and inconsistent as well. Although the CNN based algorithms are prone to overfitting, still they offer some advantages. The main advantage of using CNN in deep learning based techniques is that the convolutional layers in CNN have fewer weights to train than dense fully connected layers. This makes CNN easier to train with reducing the possibility of overfitting. CNN has become popular in the field of medical image analysis. In [6] CNN-based method for segmentation of brain tumors in MRI images is proposed. The CNN used by the authors is built over convolutional layers with  $3 \times 3$  kernels to allow designing deeper architectures. The scheme is shown to perform better than other approaches.

## 3 Proposed Methodology

In this paper, we have first applied image processing based segmentation for brain tumor detection using two different techniques. The first technique uses image difference and the second technique uses thresholding based segmentation. We have next applied deep learning based solution for brain tumor detection.

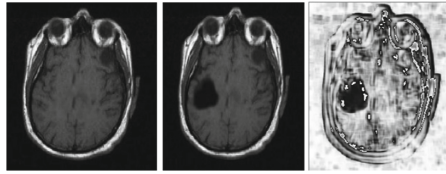
### 3.1 Image Difference Approach

This technique is used to analyze the difference between the MRI images of an unhealthy brain and a healthy brain of the same person. But, the limitation of this approach is that only abnormalities could be detected and it does not actually signify that a tumor has been detected. This method only gives a possibility that the abnormality could be a tumor. In order to compute the difference between two images the Structural Similarity Index Measurement (SSIM) metric [7] is used.

The algorithmic steps for performing image difference is as follows:

Step1: Get 1<sup>st</sup> image i.e. MRI of healthy brain as 'img1'  
 Step2: Get 2<sup>nd</sup> image i.e. MRI of unhealthy brain as 'img2'  
 Step3: gray1 = rgb\_to\_gray (img1)  
 Step4: gray2 = rgb\_to\_gray (img2)  
 Step5: (score, diff) = compare\_ssim (gray1, gray2)  
 Step6: print(score) //Structural Similarity Index  
 Step7: show\_image(diff)

Figure 1 shows an example, how image difference techniques works. The abnormality has been segmented out as shown in the third resultant image.

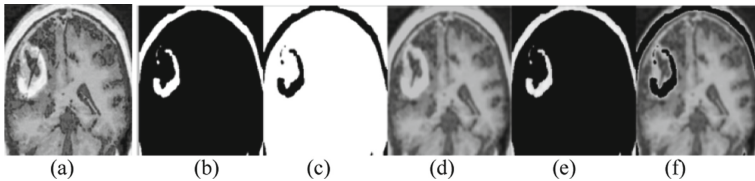


**Fig. 1.** Healthy Brain Image (Left-Most), Brain with Tumor (Middle), Image Difference (Right-Most) (with SSIM: 0.6290011764396584)

### 3.2 Threshold Based Segmentation

It is the most basic image segmentation technique where the thresholding is applied on the image to segment out the desired (*tumor*) region. Thus to segment the brain tumor from non-tumor region, thresholding is applied on the given MRI input image of the brain with various threshold values based on pixel intensity. Table 1 shows various thresholding functions. Figure 2 shows the list of 5 different threshold functions [8] that were applied.

The algorithmic steps for performing Thresholding Based Segmentation are as follows:



**Fig. 2.** a) Input image of Brain b) Thresh\_Binary c) Thresh\_Binary\_Inv d) Thresh\_Trunc e) Thresh\_Tozero f) Thresh\_Tozero\_Inv

**Table 1.** Various thresholding functions

Function	Definition
THRESH_BINARY	$dst(x, y) = \begin{cases} maxval & \text{if } src(x, y) > thresh \\ 0 & \text{otherwise} \end{cases}$
THRESH_BINARY_INV	$dst(x, y) = \begin{cases} 0 & \text{if } src(x, y) > thresh \\ maxval & \text{otherwise} \end{cases}$
THRESH_TRUNC	$dst(x, y) = \begin{cases} thresh & \text{if } src(x, y) > thresh \\ src(x, y) & \text{otherwise} \end{cases}$
THRESH_TOZERO	$dst(x, y) = \begin{cases} src(x, y) & \text{if } src(x, y) > thresh \\ 0 & \text{otherwise} \end{cases}$
THRESH_TOZERO_INV	$dst(x, y) = \begin{cases} 0 & \text{if } src(x, y) > thresh \\ src(x, y) & \text{otherwise} \end{cases}$

```

Step1: Get MRI image of Brain as 'imgA'
Step2: Get threshold value as 'thresh' (b/w 0 to 255)
Step3: grayA = rgb_to_gray(imgA)
Step4: for method in ("THRESH_BINARY",
                    "THRESH_BINARY_INV", "THRESH_TRUNC",
                    "THRESH_TOZERO",
                    "THRESH_TOZERO_INV") do
Step5:     result = apply_threshold(grayA, thresh, method)
Step6:     show_image(result)

```

The limitation of thresholding approach is that we cannot generalize a specific threshold value for all brain images of different patients. The result will be varying for different images on the same threshold value. The factors like noise level, brightness, and contrast will affect the threshold value for a given image. Hence, the operator has to set different threshold values and discover out the most specific threshold based upon the observations of large number of results.

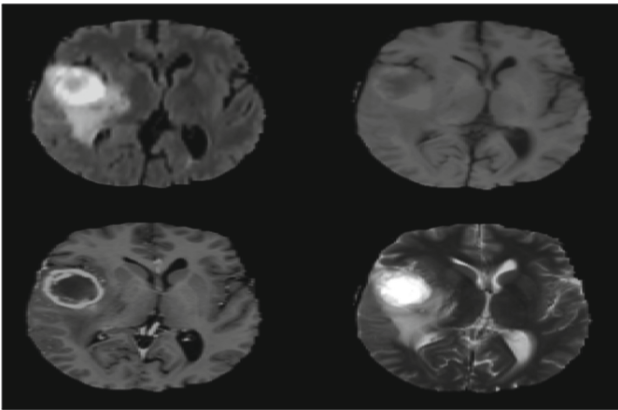
### 3.3 Convolutional Neural Networks Based Segmentation

To overcome the limitations of image processing based methods, in this work CNN based technique is used to segment out the brain tumor. A high quality dataset of brain MRIs is used, which is trained using U-Net based convolutional neural network model [9]. This model is able to learn shapes of the tumor regions and their characteristics through the feature extraction process and hence, segments out the tumor region from the non-tumor region.

### 3.3.1 Dataset for Training

The dataset used in this work is taken from an International Challenge organized by MICCAI, the Multimodal Brain Tumor Segmentation BRATS'17 challenge. The dataset is not available publicly. It has been taken after taking permission from Section of Biomedical Image Analysis, Centre for Biomedical Image Computing and Analytics, Department of Radiology, Perelman School of Medicine, University of Pennsylvania [10–13]. It consists of Brain MRI Scan of 210 HGG (High Grade Glioma) and 75 LGG (Low Grade Glioma) patients. For each patient, 3D MRI Scan of the brain is available on 4 different pulse sequences named as: T1-weighted, T2-weighted, T1ce and FLAIR in form of 4 medical image files.

A 3D MRI Scan consists of 155 2D brain slices, hence a total of 620(= 155 × 4) 2D images of the brain is to be analyzed per patient. Figure 3 shows the MRI scan of a slice of the brain on four different pulse sequences. Figure 4 shows the complete MRI scan of a patient that consists of total 620 images of brain.



**Fig. 3.** Flair (top), T1 (top-right), T1C (bottom-left) and T2 (bottom right) pulse sequences

### 3.3.2 Training Process

#### *Preparing and Validating Dataset*

The dataset was validated first by checking the complete dataset such that there should be 210 HGG and 75 LGG patients and for each patient, there must be 5 medical imaging files: 4 for four different pulse sequences 3D MRI and 1 for the ground truth segmentation of the brain tumor. Figure 5 shows 3-D representation MRI scan of a patient. Figure 6 shows the joint representation of four different pulse sequences shown in Fig. 5. The preparation of dataset includes following two steps:

Step1: Splitting dataset into train/dev/test suites in the ratio of 0.6:0.2:0.2.

Step2: Converting 3D MRI Scan into 2D images for each patient. Figure 7 shows the 2-D representation of 155 slices of the patient's MRI scan.



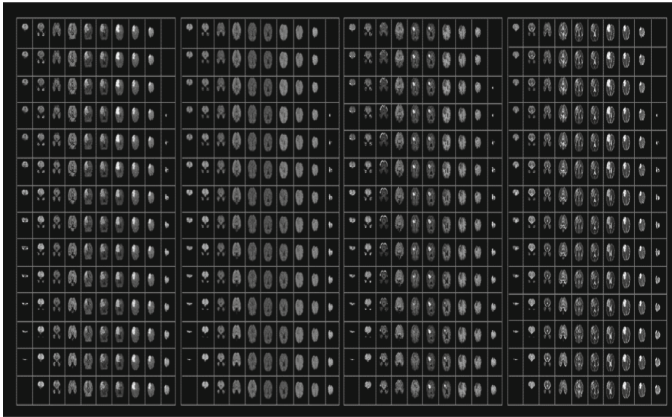


Fig. 4. 2-D Representation of Complete MRI scan of one patient which consists of 620 images.

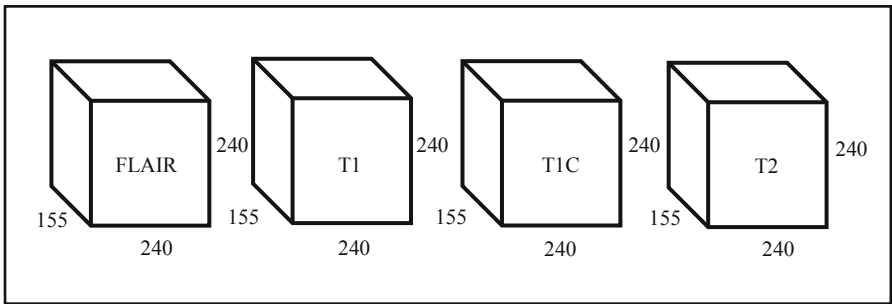


Fig. 5. 3-D representation of a patient's MRI scan

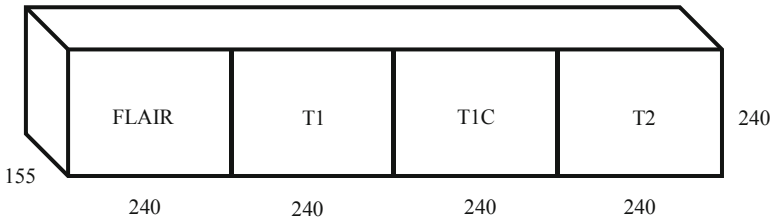
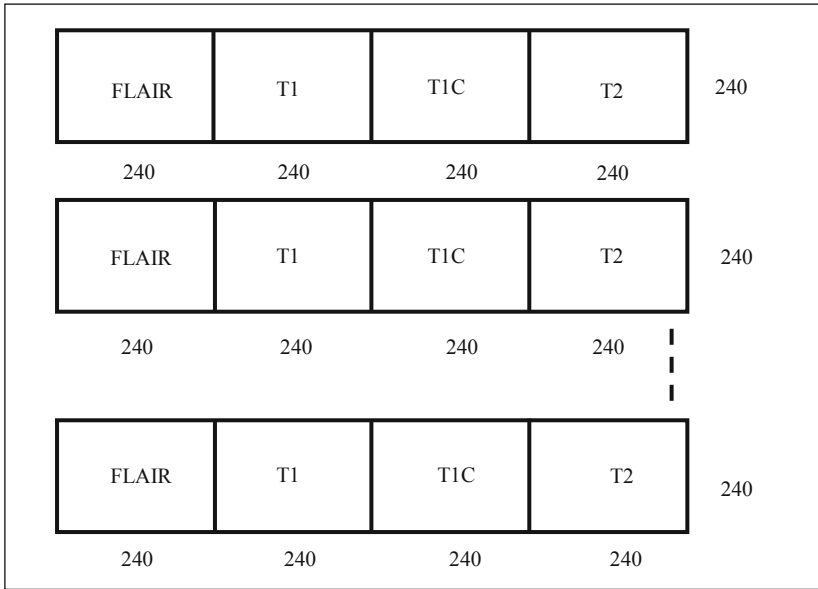


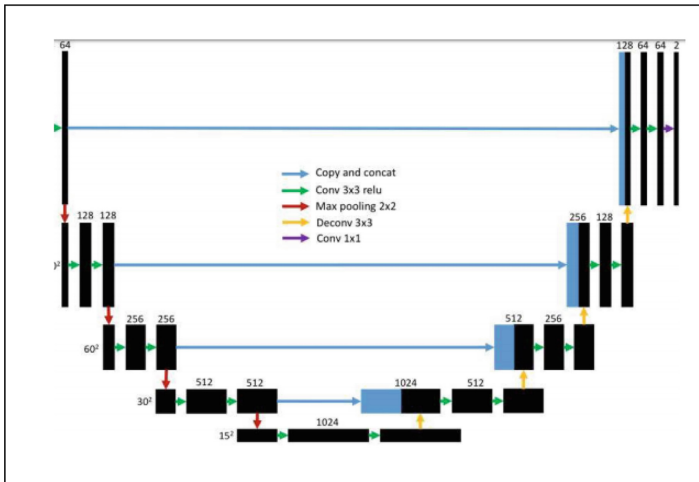
Fig. 6. Joint representation of four different pulse sequences given in Fig. 5.

**Convolution Neural Network Model**

The U-Net Architecture based Model used in this work could be represented as given in Fig. 8. The structure consists of down-sampling and up-sampling. The down-sampling path has 5 convolutional blocks. Every block has two convolutional layers with a filter size of  $3 \times 3$ , stride of 1 in both directions and rectifier activation, which increase the number of feature maps from 1 to 1024.



**Fig. 7.** 2-D representation of 155 slices of patient’s MRI scan.



**Fig. 8.** U-net architecture

For the down-sampling, max pooling with stride  $2 \times 2$  is applied to the end of every blocks except the last block, so the size of feature maps decrease from  $240 \times 240$  to  $15 \times 15$ . In the up-sampling path, every block starts with a de-convolutional layer with filter size of  $3 \times 3$  and stride of  $2 \times 2$ , which doubles the size of feature maps in both directions but decreases the number of feature maps by two, so the size of feature maps increases from  $15 \times 15$  to  $240 \times 240$ . In every up-sampling block, two convolutional

layers reduce the number of feature maps of concatenation of de-convolutional feature maps and the feature maps from encoding path.

Different from the original U-Net architecture, zero padding is used to keep the output dimension for all the convolutional layers of both down-sampling and up-sampling path. Finally, a  $1 \times 1$  convolutional layer is used to reduce the number of feature maps to two that reflect the foreground and background segmentation respectively. No fully connected layer is invoked in the network.

The dataset has been prepared in form of  $X_{train}$ ,  $X_{test}$ ,  $Y_{train}$  and  $Y_{test}$ . The algorithmic steps of the training process are as follows:

---

**Algorithm 1:** Training Algorithm using U-Net based Architecture

**Input:**  $X_{train}$ (pre-processed data-set for training),  $X_{test}$ (pre-processed data-set for testing),  $Y_{train}$ (ground truth segmentation for training),  $Y_{test}$ (ground truth segmentation for testing)

**Output:** Trained CNN Model.

---

1.  $batch\_size \leftarrow 10$
  2.  $learning\_rate \leftarrow 0.0001$
  3.  $no\_of\_epoches \leftarrow 75$
  4.  $no\_of\_batches \leftarrow |X_{train}| / batch\_size$
  5. **for**  $i \leftarrow 1$  to  $no\_of\_epoches$
  6.     **do for**  $j \leftarrow 1$  to  $no\_of\_batches$
  7.         **do**  $DATA\_AUGMENTATION(j_{batch})$  // Apply data\_augmentation on the  $j^{th}$  batch
  8.          $Y'_{train} \leftarrow U\_NET.fit(j_{batch})$              // Feed  $j^{th}$  batch in U\_NET model
  9.          $train\_loss \leftarrow 1 - DICE\_COEFFICIENT(Y_{train}, Y'_{train})$
  10.          $ADAM\_OPTIMIZER(U\_NET, train\_loss)$
  11.          $Y'_{test} \leftarrow U\_NET.fit(X_{test})$
  12.          $test\_loss \leftarrow 1 - DICE\_COEFFICIENT(Y_{test}, Y'_{test})$
  13.          $print(test\_loss)$
  14.          $SAVE\_MODEL(U\_NET)$
- 

Once the training has been completed, the saved model is able to segment the tumor from given MRI scan of the patient. The algorithmic steps for the brain tumor segmentation are:

---

**Algorithm 2:** Brain Tumor Segmentation using U-NET based trained model (Algorithm 1)

**Input:** U-NET(model), X(patient's MRI scan containing four different pulse sequences), Y(corresponding ground truth segmentation)

**Output:** Segmented Tumor Region

---

```

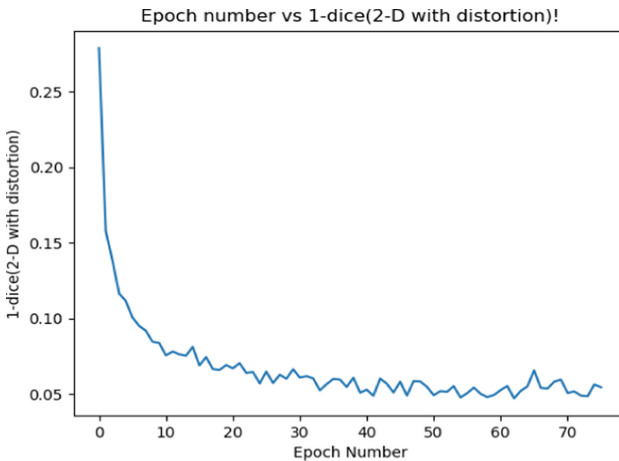
1. total_loss ← 0
2. for i ← 0 to 154           // 155 slices of brain
3.   do Y' ← U_NET.fit(X[i])
4.     loss ← 1 - DICE_COEFFICIENT(Y, Y')
5.     total_loss ← total_loss + loss
6.     SAVE_IMAGE(Y')           // Saving segmented tumor image
7. avg_loss ← total_loss / 155
8. print(avg_loss)

```

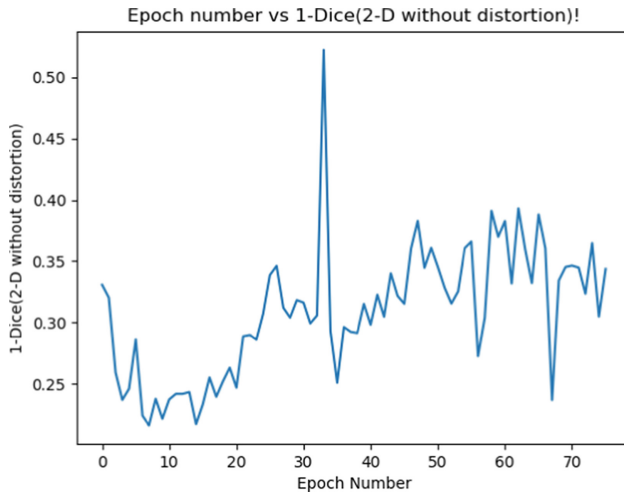
---

## 4 Experimental Results

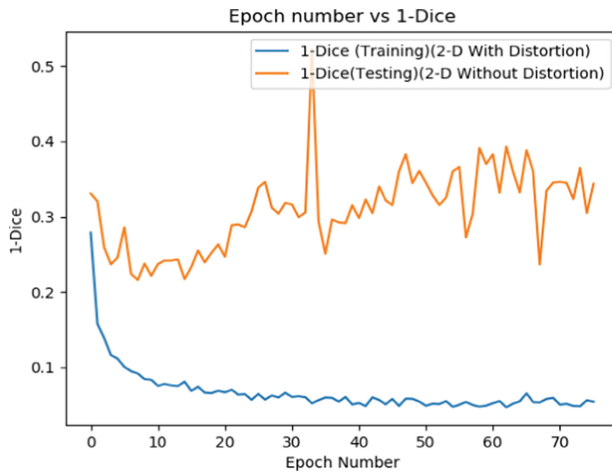
For analyzing the results, the loss function method based on soft dice coefficient [14] is used for comparing the similarity of two batches of data. The coefficient is between 0 and 1, where 1 means a total match. (Loss = 1 – soft dice coefficient). Figure 9 shows the curve between epoch number and 1-Dice (Loss) in the training process. Initially, the loss is high and as the training processes, it approaches zero, which results in better prediction and hence, better and more accurate segmented tumor region. Figure 10 shows the curve between epoch number and 1-Dice in the testing process. Figure 11 shows the comparison between testing and training loss. In Fig. 12, the initial four images are four different MRI pulse sequences, the fifth image is its ground truth segmentation and the final image actually segments the tumor region with the help of trained model.



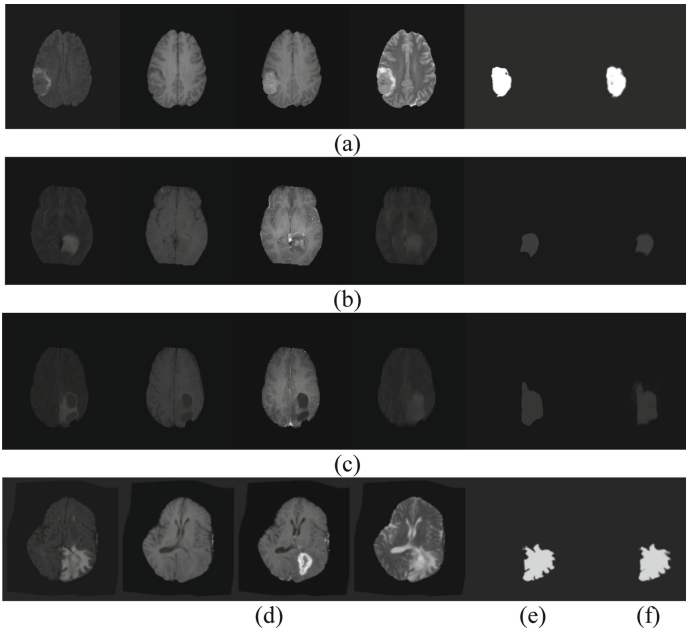
**Fig. 9.** Training loss curve



**Fig. 10.** Testing loss curve



**Fig. 11.** Comparison of training and testing loss curves



**Fig. 12.** (a) Flair (b) T1 (c) T1ce (d) T2 (e) Ground Truth Segmentation (f) Model Output (Here, the initial four images are four different MRI pulse sequences, the fifth image is its ground truth segmentation and the final image actually segments the tumor region with the help of trained model)

## 5 Conclusions

The existing image processing techniques may be used for the brain tumor detection, however, these techniques lack in accurate and reliable detection of tumor. Thus, more reliable and accurate solution is required. The automation of brain tumor detection is needed to make it independent of MRI operator's experience. Further, this will enable timely delivery of patient's diagnosis and so, decrease lag time between diagnosis tests and the treatment for the same. The solution based on CNN consists of a model, which is regressively trained and learns all the features of tumors to detect tumor accurately. This trained model is then used to segment the patient's brain MRI scan. This solution is realistic, efficient, more accurate solution over other proposed solutions. In future, we may focus on dividing "all-tumors" region into tumor stages i.e. Advancing, Edema, etc.

## References

1. Kapoor, L., Thakur, S.: A survey on brain tumor detection using image processing techniques. In: 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, Noida, pp. 582–585 (2017)
2. Shanthakumar, P., Ganeshkumar, P.: Performance analysis of classifier for brain tumor detection and diagnosis. *Comput. Electr. Eng.* **45**(7), 302–311 (2015)

3. Clark, M.C., Hall, L.O., Goldgof, D.B., Velthuizen, R., Reed Murtagh, F., Silbiger, M.S.: Automatic tumor segmentation using knowledge-based techniques. *IEEE Trans. Med. Imaging* **17**(2), 187–201 (1998)
4. Aswathy, S.U., Deva Dhas, G.G., Kumar, S.S.: A survey on detection of brain tumor from MRI brain images. In: 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, pp. 871–877 (2014)
5. Liu, J., Li, M., Wang, J., Wu, F., Liu, T., Pan, Y.: A survey of MRI-based brain tumor segmentation methods. *Tsinghua Sci. Technol.* **19**(6), 578–595 (2014)
6. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans. Med. Imaging* **35**(5), 1240–1251 (2016)
7. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
8. “OpenCV: Image Thresholding”, Docs.opencv.org. (2019). [https://docs.opencv.org/3.4/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html). Accessed 20 May 2019
9. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
10. Menze, B., Jakab, A., Bauer, S., et al.: The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS) (2018)
11. Menze, B.H., et al.: The Multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Med. Imaging* **34**(10), 1993–2024 (2015)
12. Anon (2018). <https://www.med.upenn.edu/sbia/brats2017/data.html>. Accessed Mar 2019
13. Surgery, S.: SMIR - SICAS Medical Image Repository. [online] Smir.ch (2018). <https://www.smir.ch/>. Accessed Jan 2019
14. En.wikipedia.org. Sørensen–Dice coefficient (2018). [https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice\\_coefficient](https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient)



# Implementation of Deep Neural Networks in Facial Emotion Perception in Patients Suffering from Depressive Disorder: Promising Tool in the Diagnostic Process and Treatment Evaluation

Krzysztof Michalik<sup>1(✉)</sup> and Katarzyna Kucharska<sup>2</sup>

<sup>1</sup> Department of Artificial Intelligence, College of Computer Science and Communication, University of Economics in Katowice, 1 Maja 50, 40-287 Katowice, Poland  
krzysztof.michalik@ue.katowice.pl

<sup>2</sup> Institute of Psychology, Cardinal Stefan Wyszyński University, 1/3 Woycicki, 01-938 Warsaw, Poland  
k.kucharska@uksw.edu.pl

**Abstract.** According to World Health Organization, depression is a common illness worldwide, with more than 300 million sufferers. This article describes relatively new research that is giving Deep Neural Networks (DNN) and Expert System-based hybrid solutions, skills of recognizing human affect and its intensity in standardized manner with more precision and objectivity than human eye. At present diagnostic process of depression relies mostly on diagnostic and statistical manual (DSM-5) and international statistical classification of mental disorders (ICD-10) alongside other standardized clinical measures conducted by clinicians. Implementation of DNN in recognition facial affect in depression appears a promising diagnostic tool, in conjunction with above mentions classifications and clinical measures, *via* improving early detection of depressive symptoms or facilitating evaluation of treatment efficacy in depressive disorder. The article particularly aims at automatic analysis of facial affect in depressed individuals, highlighting applications together with challenges to their implementation in medicine.

**Keywords:** Depression · Facial emotion recognition · Deep neural networks · Hybrid systems

## 1 Introduction

Results of a large study conducted in 27 EU member states [1, 51] show that each year 164.8 million inhabitants (38.2%) suffer from psychiatric disorders [50]. According to World Health Organization, depression is a common illness worldwide, with more than 300 million people affected. In reference to National Survey on Drug Use and Health data from 2017, 17.3 million adults in the United States, equals 7.1% of all adults in the



country, have experienced a major depressive episode in the past year. Long-lasting depression with severe depressive episodes remains serious health condition which may lead to premature death due to suicide, social isolation or somatic co-morbidities. Around 800 000 people die due to successful suicidal attempt every year and suicide is the second leading cause of sudden death in young population between 15 and 29-year-olds.

Could emotional intelligence demonstrated by ‘machines’ help in prevention suicidal deaths *via* improving early detection of depressive symptoms or facilitating evaluation of treatment efficacy in depressive disorder?

Implementation of DNN in recognition of facial affect in depression appears a promising diagnostic tool, in conjunction with DSM-5 and ICD-10 diagnostic criteria [28] and standardized diagnostic clinical measures, in the diagnostic process and treatment evaluation [20, 21, 26]. This article describes relatively new research that is giving DNN and expert system-based (ES) hybrid solutions, skills of recognizing human affect and its intensity in more standardized manner and being more precise and undoubtedly more objective than human eye (see e.g. [47]).

The aim of our paper is to present updated review of literature on using DNN and hybrid solutions based on expert systems (ES) in diagnosis and treatment evaluation in depression. Authors attempt to explore up-to-date state of artificial intelligence (AI) practical experiments throughout the project they currently conduct. The article is particularly aiming at automatic analysis of facial affect in depressed individuals. The current paper presents several examples illustrating innovative forms of ‘machine’ emotional intelligence, highlighting applications together with challenges to their implementation in medicine.

## 2 Facial Emotion Perception in Depression

Over the last few decades there have been numerous studies examining the perception of human affect in normal and pathological populations.

Perception of facial emotion is thought to be a complex cognitive ability which relies on the integrity of a select set of more basic neurocognitive processes such as visual scanning, working memory, and vigilance which may be asymmetrically distributed across the cerebral hemispheres [31].

There is a substantial body of research evidencing impaired facial affect recognition in depressive disorder [7, 22, 23, 34, 37, 45]. Such deficits may offer an explanation for the decreased psychosocial functioning and even social isolation at the worst stage of depressive phase [30]. Several studies have indicated that deficits in recognizing facial affect reflect a negative bias in facial perception, where happy facial expressions are interpreted as neutral, and neutral faces are perceived as sad mimic expressions. Subjects with depressive disorder show longer response time in happy facial expressions than healthy controls [46]. As far as perception of negative facial emotions is concerned, these processes remain considerably impaired as well. People with depression experience negative information as more stressful and more negative than healthy controls on a subjective and physiological level. This intensified negative emotion perception may further impact daily social functioning and emotional

competence [49]. Moreover, the way that depressed patients perceive facial affect corresponds with the course of their disorder, including its chronicity and symptoms severity. Patients with depression who perceive high level of negative emotions when viewing faces are mainly characterized by greater severity of depression, its chronic course, symptoms persistence, and poor clinical prognosis [2, 24]. These results stay in line with study results of Zwick and Wolkenstein [52] who compared facial emotion recognition using the Amsterdam Dynamic Facial Expression Set in two patients groups in acute and remitted stage of illness and healthy controls. Furthermore, the activity of Zygomaticus Major and Corrugator supercilii were recorded. Patients in acute depression presented with impaired perception of happy faces compared to healthy subjects and found more difficulty in perceiving happiness, anger and fear than healthy controls. Remitted patients only show mild impairments in the recognition of emotional expressions: happiness, anger and fear than healthy controls. Emotion perception deficits in remitted stage of illness may be a consequence of disrupted connectivity within the salience and emotion network, including the amygdala, subgenual anterior cingulate cortex (sgACC), and insula [29]. Patients with depression presented with perceptual bias towards unpleasant versus pleasant facial expressions and the hypersensitivity to angry facial signals might influence the interaction behaviors between depressed patients and others [38].

### 3 Evaluation of Emotions in Psychiatry: Facing Methodological Challenges and Shortcomings in Current Traditional Approach

At present, widely available, comprehensive measuring have allowed researchers to explore further current understanding of the dynamic and morphological differences between voluntary and involuntary expressions and what is more, the relationship between what people show on their faces and what they say they feel in depression.

**Facial Action Coding System (FACS)** is a comprehensive, anatomically based system of taxonomizing all human facial movements by their appearance on the face. It breaks down facial expressions into individual components of muscle movement as one muscle contracting called Action Units (AUs).

FACS was originally developed by a Swedish anatomist [27] and was later adopted and published in 1978 by Paul Ekman and Wallace V. Friesen [8–12]. Significant update to FACS was published by Joseph C. Hager in 2002 [12]. Due to subjectivity and time consumption issues, FACS has been set up as a computed automated system that detects facial mimicry in videos, extracts the geometrical features of the faces, and subsequently produces temporal profiles of each facial movement [25]. The use of FACS has been proposed for use in the analysis of depression [44]. FACS coders must meet certain standards of reliability before they are certified in identifying individual facial action units when they are active. Specific combinations of facial action units correspond to emotion-specific facial expressions, where each expression is created by contracting a set of facial muscles. Unfortunately, the one-to-one correspondence between individual facial actions and facial expressions is weakened because different

combinations of action units can produce similar looking expressions [48]. Another version of FACS is called “Emotion FACS” or **EMFACS** (Emotional Facial Action Coding System) [17] - unpublished manual, University of California, California] and FACS/AID (Facial Action Coding System Affect Interpretation Dictionary).

EMFACS was designed for FACS coders who may selectively apply the coding criteria to use EMFACS procedures with no additional training needed. They identify the presence or absence of the putative facial expression. Considering the fact of lesser precision of FACS, EMFACS appears even less reliable [18] and potentially more prone to bias.

Another drawback of EMFACS stems from its difficulty to get intercoder agreement on its coding as the coders need to agree on two: 1) whether to code an event and 2) how to code those events that they have chosen to code. Freitas-Magalhães set up in 2018 the pioneer F-M Facial Action Coding System 3.0 (F-M FACS 3.0) [14–16]. It presents 5,000 segments in 4 K, using 3D technology and automatic and real-time recognition (FaceReader 7.1). The F-M FACS 3.0 features 8 pioneering action units (AUs), 22 pioneering tongue movements (TMs), and a pioneering Gross Behavior GB49 (Crying) [14–16]. The latest version F-M NeuroFACS 3.0 was created in 2019 by Dr. Freitas-Magalhães.

## 4 Deep Neural Networks Support of Facial Emotions in the Context of Psychiatry

### 4.1 Recognition of Facial Emotions Using DNN *MoodAnalyzer*

For many years, the authors have been working on systems to support psychiatric diagnostics in the field of depression. Initially, the research mainly focused on ES technology, including those developed by the authors: the PC-Shell hybrid expert system shell [40–42] with a full integration in the meaning of e.g. publication [39] and its application in the diagnosis of affective disease in the form of a system built using a PC-Shell tool, called Salomon [32, 33]. As part of this research current, the authors also took into account the possibilities of fuzzy logic. The hybrid nature of the PC-Shell system results from the fact that as early as 1990 its first version combined both the technology of ES and artificial neural networks (ANN) at a deep level of data structures and a specially designed knowledge representation language (i.e. communication between different AI models not via files), which slows down and hinders knowledge processing. Then developed and functioning ANN Neuronix allowed the use of up to three hidden layers, so in a sense - in the light of some modern definitions - it could be considered a DNN.

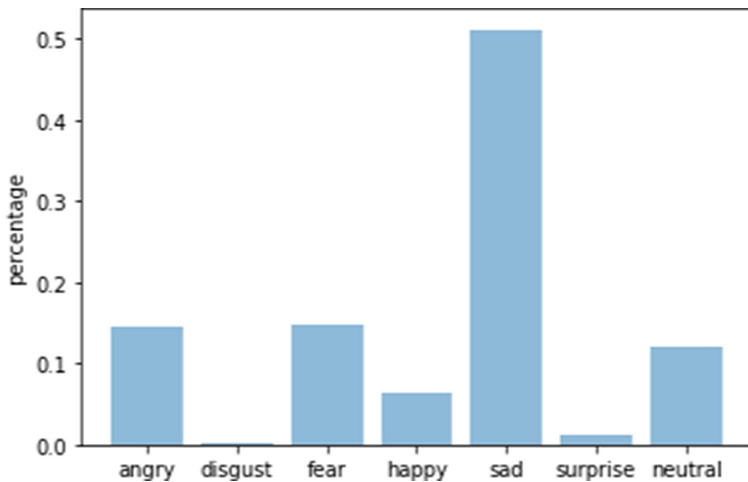
Establishment of the first neural networks from the so-called deep learning [20, 21, 26], called here more precisely as DNN after 2010 and the first successes of neural networks, including CNN – Convolutional Neural Networks (kind of DNN) [6] encouraged authors to enrich existing software with a component based on this approach. In the case of learning emotions based on photos or video fragments, the main issue is to get access to correct and representative case databases (evidence based medicine). Currently there are many databases containing pictures of faces, available

for learning DNN, but many of them are only used to recognize or identify objects, e.g. in photos, which seems to be much easier task for DNN (in the case of face recognition) than recognition of emotions. That is why the authors have chosen the dataset FER2013 [13] database as the training set. An additional problem related to dataset concerns a serious issue related to professional ethics, including consent to the processing of personal data. The mentioned data consists of 48x48 pixel grayscale images of faces. The training set consists of 28,709 examples and test set consists of 3,589 examples. The classification of emotions in this dataset is identical to the mentioned Ekman and Friesen method. This means that, as in that method, six classes of emotions were distinguished: 0 = Angry, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise, 6 = Neutral.

For the analysis of emotions, the DNN model of *MoodAnalyser* has been built using the Tensorflow library API (more e.g.: [19]), Keras (see e.g.: [6]) and the Python language (Python used for DNN see e.g.: [3]). The achieved results of emotion evaluation and empirically determined (hyper) parameters and the topology of the DNN gave satisfactory results (e.g. the “accuracy” parameter, “lost function” etc.), although experiments and modeling of both (hyper) parameters and DNN topology are still items for further development. However, the results already achieved are better than the subjective assessment based on the aforementioned imprecise method of Ekman. The *MoodAnalyser* system architecture used is rather typical for CNN neural networks. For example, the method of the “transfer learning” method is being also used for current experiments. Figure 1 shows a male photo underwent the analysis of emotional processing Fig. 2 shows an example of the analysis of emotional processing from Fig. 1 by *MoodAnalyser* using some (hyper) parameters.



**Fig. 1.** Picture on input to prediction by DNN system *MoodAnalyser*



**Fig. 2.** Example of emotion evaluation based on the photo from Fig. 1 by the *MoodAnalyzer* system

#### 4.2 Hybrid and More Holistic Approach to Depression Analysis

One of the most important components of described project is the developed ES system called *Salomon*, which uses fuzzy logic to diagnose depression (affective illness). Face emotions recognition is important part of this more general diagnostic problem. Some aspects of this work have been described in more detailed way, among others in publications: [35, 36, 43]. For this purpose, a hybrid PC-Shell system was used, which is part of the proprietary Sphinx artificial intelligence package [40–42]. On its basis, an application called *Salomon* [32, 33] supporting the diagnosis of depression, including its severity, was created.

A hybrid system can be understood as a system combining various IT methods or techniques, in particular in the field of AI. Already in 1991 a hybrid system was designed and implemented connecting the ES called PC-Shell and the *Neuronix* neural network simulator, with full integration of different AI technologies as well as multimedia and built-in imperative language compiler. This solution is original and the said strong integration results from the combination of both classes of systems at the level of data structure rather than exchange of knowledge via files. One of the assumptions was the use of ES **blackboard architecture** [4, 5] with heterogeneous knowledge sources and the possibility of linking ES with many applications of automatically generated domain-specific ANN applications. Another important assumption from the point of view of the flexibility of created practical applications was the possibility of two-way knowledge exchange, i.e. both from ES to ANN and vice versa from ANN to ES. A solution to this problem was applied at the level of the language of knowledge representation, which greatly facilitates the work of a knowledge engineer. ANN domain applications defined earlier are generated automatically and dynamically during the operation of the entire system based on the model definition of individual neural networks, including e.g. topology and selected parameters. The following general

description of the flow of knowledge with the ES knowledge source and the source of knowledge in the form of ANN applications. The bi-directional flow of knowledge between ANN and ES sources in the sense of blackboard architecture [4, 5]:

- a) Scheme of flow of knowledge from ANN to ES

```

knowledge base hybrid_application_outline
  sources
    type kb
    file "c:\\agents\\diagnosis.zw"
  forecast1:
    type neural_net
    file "c:\\agents\\prediction1.prj"
    ...
  end;
...
end;

control
  record NeuralNet INP[Size1];
  record NeuralNet OUT[Size2];
  char Problem_to_be_solved;
  ...
  initNetwork ( nnSourcei );
  runNetwork ( nnSourcei, INP, OUT );
  delNetwork ( nnSourcei );
  ...
  addFact( Object1, Attribute1, OUT.value1 );
  ...
  addFact( ObjectN, AttributeN, OUT.valueSize2 );
  solve( esSourcej, Problem_to_be_solved );
  ...
end;

```

- b) Scheme of flow of knowledge from ES to ANN

```

control
  record NeuralNet INP[Size1];
  record NeuralNet OUT[Size2];
  char Problem_to_be_solved, M[Size3];
  int M, N;
  ...
  solve( esSourcei, Problem_to_be_solved);
  saveSolution( M, N );
  ... // place for instructions assigning ES solution to the ANN input
  runNetwork( nnSourcej, INP, OUT );
  ...
end;

```

The hybrid architecture described enables easier implementation of a holistic approach to support diagnosis and therapy evaluation of depression using various AI technologies - here is shown a model of cooperation between ANN and ES at the level of the language of knowledge representation and strong integration at the level of internal data structures in the computer system. A more detailed discussion of the details and the ES module is beyond the scope and purpose of this work.

## 5 Summary

There are a few strengths and limitations of this work. As far as shortcomings are concerned, the article omits some important data, such as the “accuracy” and “lost function” parameters achieved at the time of writing. This is due to the continuation of work and ongoing experiments with the addition of the “transfer learning” method and some others, which theoretically can increase the value of “accuracy” of prediction by up to 7–8%.

Undoubtedly, the strength of this project is its hybrid nature combining, among others DNN, ES, multimedia technologies, as well as a description of some pieces of knowledge using the imperative language built into the system described, similar in form to C and Pascal. Implementation of AI in facial emotion perception in patients suffering from depressive disorder can vastly improve the diagnostic process with risk assessment included and help out to measure therapeutic outcome.

In addition, the use of heuristic knowledge might be crucial for the next stage of research on the suicide prediction as a consequence of AI based suicide risk assessments. It is to be described the *Salomon* ES that allows determining the severity of depression according to an established classification. It seems that the hybrid approach creates the possibility of comprehensive substantive analysis of the patient’s condition. At the same time, the ES component ensures transparency of the solution method (e.g. in the form of How? explanations), unlike the ANN components, which, in the language of cybernetics, still meet the “black box” model. Response protocols are needed on how to properly handle high risk cases that are flagged by AI technology, and what to do if AI risk assessments differ from clinical opinion.

One of the serious challenges may be the phenomenon of hiding/masking emotions by people suffering from depression. However, this clinical dilemma remains equally challenging for both psychiatrists as well as for AI methods (hybrid and ES solutions).

To sum up, artificial intelligence technology, especially DNN/ANN and ES provides a great opportunity for further progress in crucial aspects of psychiatric care, including both the diagnosis and treatment of patients suffering from depression, and suicide screening or suicide risk assessment. One of the main goals of our research is to use the created AI systems, beside both diagnosis and treatment of depression sufferers, to predict and prevent suicides via suicide screening/suicide risk assessments. Very important aspects of using AI for these purposes is the ability to communicate with patients through today’s mobile devices and monitor their emotions, including faces by an automatic AI system, free from human subjectivity. Further researches in medicine on how AI technology fits into diagnosis and treatment of depression is strongly needed.

## References

1. American Psychiatric Association: Diagnostic and Statistical Manual of Mental Disorders, 5th edn. American Psychiatric Publishing, Arlington (2013)
2. Bouhuys, A.L., Geerts, E., Gordijn, M.C.: Depressed patients' perceptions of facial emotions in depressed and remitted states are associated with relapse: a longitudinal study. *J. Nerv. Mental Disease*. **187**, 595–602 (1999)
3. Chollet F.: *Deep Learning with Python*. Manning Publications, Shelter Island (2018)
4. Craig, I.: *Formal Specification of Advanced AI Architectures*. Ellis Horwood Series in Artificial Intelligence. Ellis Horwood, Chichester (1991)
5. Craig, I.: *Blackboard Systems*. Ablex Publishing Corporation, Norwood (1995)
6. Dadchich, A.: *Practical Computer Vision, Extract Insightful Information From Images Using Tensorflow, Keras and OpenCV*. Birmingham – Bumbai, Pack (2018)
7. Derntl, B., Seidel, E.M., Kryspin-Exner, I., Hasmann, A., Dobmeier, M.: Facial emotion recognition in patients with bipolar I and bipolar II disorder. *Br. J. Clin. Psychol.* **48**, 363–375 (2009)
8. Ekman, P., Friesen, W.V., Phoebe, E.: *Emotion in the Human Face*. Pergamon, New York (1972)
9. Ekman, P., Friesen, W.V., Tomkins, S.S.: Facial affect scoring technique (FAST): a first validity study. *Semiotica* **3**(1), 37–58 (1972)
10. Ekman, P., Friesen, W.V.: *Facial Action Coding System*. Consulting Psychologist Press, Palo Alto (1978)
11. Ekman, P., Friesen, W.V.: *Rationale and reliability for EMFACS Coders*. Unpublished (1982)
12. Ekman, P., Friesen, W.V., Hager, J.C.: *Facial Action Coding System: The Manual on CD ROM. A Human Face*, Salt Lake City (2002)
13. FER13 dataset. <https://www.kaggle.com/deadskull7/fer2013>. Accessed 30 Oct 2019
14. Freitas-magalhães, A.: *Facial Action Coding System 3.0: Manual of Scientific Codification of the Human Face*. FEELab Science Books, Porto (2018)
15. Freitas-magalhães, A.: Scientific measurement of the human face: F-M FACS 3.0 - pioneer and revolutionary. In: Freitas-Magalhães, A. (ed.) *Emotional Expression: The Brain and the Face*, vol. 10, pp. 21–94. FEELab Science Books, Porto (2018)
16. Freitas-magalhães, A.: *NeuroFACS 3.0: The Neuroscience of Face*. FEELab Science Books, Porto (2019)
17. Friesen, W., Ekman, P.: *EMFACS-7: Emotional Facial Action Coding System*. Unpublished Manual, University of California, California (1983)
18. *Facial Action Coding System Affect Interpretation Dictionary (FACSAID)*: Archived from the original on 20 May 2011. Accessed 23 Feb 2011
19. Geron, A.: *Hands-On Machine Learning with Scikit-Learn and Tensorflow*. O'Reilly, Cambridge (2017)
20. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, Cambridge (2016)
21. Graupe, D.: *Deep Learning Neural Networks, Design and Case Studies*. World Scientific, New Jersey (2016)
22. Gray, J., Venn, H., Montagne, B., Murray, L., Burt, M., Frigerio, E., et al.: Bipolar patients show mood congruent biases in sensitivity to facial expressions of emotion when exhibiting depressed symptoms, but not when exhibiting manic symptoms. *Cognit Neuropsychiatry*. **11**, 505–520 (2006)
23. Gur, R.C., Erwin, R.J., Gur, R.E., Zwil, A.S., Heimberg, C., Kraemer, H.C.: Facial emotion discrimination: II. Behavioral findings in depression. *Psychiatry Res*. **42**, 241–251 (1992)



24. Hale, W.W.: 3rd judgment of facial expressions and depression persistence. *Psychiatry Res.* **80**, 265–274 (1998)
25. Hamm, J., Kohler, C.G., Gur, R.C., Verma, R.: Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders. *J. Neurosci. Methods* **200**(2), 237–256 (2011)
26. Heaton, J.: *Artificial Intelligence for Humans, vol. 3: Deep Learning and Neural Networks*. Heaton Research, St. Louis (2015)
27. Hjortsjö, C.-H.: *Man's Face and Mimic Language*. Studentlitteratur, Lund (1969)
28. ICD-10: international statistical classification of diseases and related health problems: tenth revision, 2nd ed. World Health Organization (2004)
29. Jenkins, L.M., Stange, J.P., Barba, A., et al.: Integrated cross-network connectivity of amygdala, insula, and subgenual cingulate associated with facial emotion perception in healthy controls and remitted major depressive disorder. *Cogn. Affect Behav. Neurosci.* **17**(6), 1242–1254 (2017)
30. Judd, L.L., Akiskal, H.S., Schettler, et al.: Psychosocial disability in the course of bipolar I and II disorders: a prospective, comparative, longitudinal study. *Arch. Gen. Psychiatry* **62**(12), 1322–1330 (2005)
31. Kee, K.S., Kern, R.S., Green, M.F.: Perception of emotion and neurocognitive functioning in schizophrenia: what's the link? *Psychiatry Res.* **81**(1), 57–65 (1998)
32. Kielan, K.: The use of the Salomon computer expert system in the diagnosis of depression. *Eur. Psychiatry* **12**(Suppl. 2) (1997)
33. Kielan, K., Kwiatkowska, M., Kucharska, K., Michalik, K., Węgrzyn-Wolska, K.: PHOENIX - AI technology in neuroscience evidence base medicine. In: Goluchowski, J., Fraczkiewicz-Wronka, A. (eds.) *Knowledge Technologies in Public Management*. Publisher University of Economics, Katowice (2008)
34. Kohler, C.G., Hoffman, L.J., Eastman, L.B., Healey, K., Moberg, P.J.: Facial emotion perception in depression and bipolar disorder: a quantitative review. *Psychiatry Res.* **188**(3), 303–309 (2011)
35. Kwiatkowska, M., Kielan, K., Michalik, K.: A fuzzy-semiotic framework for modelling imprecision in the assessment of depression. In: *Proceedings of 2009 International Fuzzy Systems Association WORLD CONGRESS (IFSA)*, 20–24 June, Lisbon, Portugal, pp. 1717–1722, July 2009
36. Kwiatkowska, M., Kielan, K., Michalik, K.: Computational representation of medical concepts: a semiotic and fuzzy logic approach. In: Seizing, R., Sanz, V. (eds.) *Soft Computing in Humanities and Social Sciences*. Springer, Heidelberg (2012)
37. Leppanen, J.M., Milders, M., Bell, J.S., Terriere, E., Hietanen, J.K.: Depression biases the recognition of emotionally neutral faces. *Psychiatry Res.* **128**, 123–133 (2004)
38. Liu, W.H., Huang, J., Wang, L.Z., Gong, Q.Y., Chan, R.C.: Facial perception bias in patients with major depression. *Psychiatry Res.* **197**(3), 217–220 (2012)
39. Medsker, L.R.: *Hybrid Neural Network and Expert Systems*. Kluwer Academic Publishers, Boston (1994)
40. Michalik, K.: Financial analysis using a hybrid expert system. In: *Proceedings of the Workshop "AI in Finance and Business" ECAI 1994*, Amsterdam, pp. 109–114, August 1994
41. Michalik, K.: Intelligent system for financial analysis. In: *Proceedings of the SPICIS 1994 International Conference on Intelligent Systems*, 14–17 November, Singapore, pp. B129–B134 (1994)
42. Michalik, K.: Selected aspect of multi-level hybrid environment for decision support. *J. Artif. Intell. Stud.* **2**(24) (2004). (Special Edition)

43. Michalik, K., Kwiatkowska, M., Kielan, K.: Application of knowledge-engineering methods in medical knowledge management. In: Seising, R., Tabacchi, M.E. (eds.) *Fuzziness and Medicine: Philosophical Reflections and Application Systems in Health Care*. Springer, Heidelberg (2013)
44. Reed, L.I., Sayette, M.A., Cohn, J.F.: Impact of depression on response to comedy: a dynamic facial coding analysis. *J. Abnorm. Psychol.* **116**(4), 804–809 (2007)
45. Schaefer, K.L., Baumann, J., Brendan, A.R., et al.: Perception of facial emotion in adults with bipolar or unipolar depression and controls. *J. Psychiatr. Res.* **44**(16), 1229–1235 (2010)
46. Suslow, T., Dannlowski, U., Lalee-Mentzel, J., Donges, U.S., Arolt, V., et al.: Spatial processing of facial emotion in patients with unipolar depression: a longitudinal study. *J. Affect. Disord.* **83**, 59–63 (2004)
47. Yoon, K.L., Joormann, J., Gotlib, I.H.: Judging the intensity of facial expressions of emotion: depression related biases in the processing of positive affect. *J. Abnorm. Psychol.* **118**, 223–228 (2009)
48. Tian, Y.-L., Takeo, K., Cohn, J.F.: Recognizing action units for facial expression analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(2), 1–19 (2001)
49. Wenzler, S., Hagen, M., Tarvainen, M.P., Hilke, M., Ghirmai, N., Huthmacher, A.C., Trettin, M., van Dick, R., Reif, A., Oertel-Knöchel, V.: Intensified emotion perception in depression: differences in physiological arousal and subjective perceptions. *Psychiatry Res.* **253**, 303–310 (2017)
50. WHO Homepage. <https://www.who.int/news-room/fact-sheets/detail/depression>. Accessed 11 Oct 2019
51. Witchen, H.U., et al.: The size and burden of mental disorders and other disorders of the brain in Europe (2010). PubMed US National Library of Medicine, National Institutes of Health. <https://www.ncbi.nlm.nih.gov/pubmed>
52. Zwick, J.C., Wolkenstein, L.: Facial emotion recognition, theory of mind and the role of facial mimicry in depression. *J. Affect. Disord.* **1**(210), 90–99 (2017)



# Invisibility and Fidelity Vector Map Watermarking Based on Linear Cellular Automata Transform

Saleh Al-Ardhi<sup>(✉)</sup>, Vijey Thayanathan<sup>(✉)</sup>, and Abdullah Basuhail<sup>(✉)</sup>

Faculty of Computing and Information Technology (FCIT), King Abdulaziz University,  
Jeddah, Saudi Arabia

s\_ardhi@hotmail.com, {vthayanathan, abasuhail}@kau.edu.sa

**Abstract.** Invisibility and fidelity influence 2D vector maps, especially the way vector data is used after applying various watermarking techniques to obtain a veiling of the digital vector map's information via distortion control. This study proposes a linear cellular automata technique to safeguard vector maps for copyright protection purposes. Performance evaluation of the proposed system indicates higher invisibility and fidelity compared to previous frequency techniques. Additionally, the proposed technique indicates that, in digital watermarking, it is possible to use several frequency domains.

**Keywords:** Copyright protection · Vector map · Invisibility · Fidelity · Linear cellular automata transform

## 1 Introduction

With the growing use of geospatial data in recent decades, paired with advances in computer hardware such as geographic data collection instruments, a large number of paper maps have been digitised, and devices such as Geographic Positioning Systems (GPS) have been designed to leverage satellites to retrieve spatial positioning data. To give an example, Geographic Information Systems (GIS) have enabled users to move away from hardcopy printing or analogue data to vector maps, which serve as realisations and standard representations [1].

Map stakeholders in digital watermarking can resolve the question of who owns a particular digital map, as well as the map's validity. Digital watermarking can safeguard against data alteration or data extraction, and vector maps can be used in both spatial and transformation spaces. Robust digital watermarking systems can be applied for copyright protection purposes [2]. The main transform algorithms include Discrete Wavelet Transform (DWT) [3], Discrete Cosine Transform (DCT) [4], and the Fast Fourier Transform (FFT) [5]. However, robustness and invisibility are undermined by the ease of use associated with the spatial domain for watermarking. Therefore, rather than the spatial domain, digital watermarking techniques can address the transformation domain in order to achieve a higher performance in terms of copyright protection [6–8].

This study focuses on the issue of protecting vector map copyright. In order to achieve this, the linear cellular automata transform (LCAT) algorithm is proposed, which is an advanced technique for vector map watermarking. The cellular automata transform (CAT) algorithm was proposed in [9] for multimedia watermarking, but embedded media and their use in vector maps have yet been extensively researched. The LCAT algorithm is associated with a range of benefits, including fidelity, insertion, and invisibility [10].

The performance evaluation tools used to assess the LCAT algorithm included normalised correlation (NC) computation, quality evaluation based on invisibility with root mean square error (RMSE) computations, and the fidelity with the longest distance. Based on the evaluation results, the technique produced acceptable values for the invisibility and RMSE tests. Additionally, the NC and distance values were satisfactory, and the technique had a high level of resistance to geometric attacks.

## 2 Methods

### 2.1 Linear Cellular Automata

A cellular automaton can be described as a grid of cells, where every cell has a finite number of states, and the grid can have any finite number of dimensions. Linear cellular automata (LCA) can be expressed as in Eq. (1). Every cell inside the grid has a finite number of states, and all together they form a lattice structure [11].

$$(C^{t+1})^T = M_n \cdot (C^t)^T \pmod{2} \tag{1}$$

where  $M_n$  refers to the local transition matrix. Given that  $n = 5k$ , the local transition matrix can be expressed as follows:

$$M_n = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & \dots & \dots & \dots & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & \dots & \dots & \dots & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & \dots & \dots & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & \dots & \dots & \dots & 0 & 0 & 0 \\ & & & \dots & & & & & & & \\ & & & \dots & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & 1 & 1 & 1 \end{pmatrix}.$$

Suppose that the transition matrix of a cellular automation ( $An$ ) is denoted by  $M_n$ . Since  $An$  is of the  $n^{th}$  order penta-diagonal matrix, the non-zero coefficients will be 1. The transpose of a linear matrix that consists of an interchange of random binary numbers can be represented by  $(C^t)^T$ , which is defined in the following way:

$$(C^t)^T = M_n^{-1} \cdot (C^{t+1})^T \pmod{2} \tag{2}$$

The inverse of the cellular automaton, given  $n = 5k$ , the transition matrix can be expressed as:

$$M_n^{-1} = \begin{pmatrix} M_5^{-1} B & B & \dots & B \\ B^T & M_5^{-1} B & \ddots & \vdots \\ B^T & A^T & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & M_5^{-1} B \\ B^T & \dots & B^T & B^T & M_5^{-1} \end{pmatrix},$$

where

$$M_5^{-1} \begin{vmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{vmatrix} \pmod{2}, B = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

In the above expressions,  $|M_n| \pmod{2}$  refers to the transition matrix that begins with five integers, which can be defined as follows:

$$|M_n| \pmod{2} = \begin{cases} 1, & \text{if } n = 5k \text{ or } n = 5k + 1, \text{ with } k \in \mathbb{N} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

## 2.2 Linear Cellular Automata Transform

Linear cellular automata transform (LCAT) involves the transformation of the coordinates of the vertices. It is applied to the coefficient of the transformation result from the data of the vector map. An overview of the transformation of map data to the linear cellular automata (LCA) space is shown in Fig. 1. Once this transformation is complete, the vector map is transformed to the frequency domain, as shown in Eq. (4).

$$T(M) = \sum_{n=0}^{N-1} M_n \cdot v_{x1} \pmod{2} \quad (4)$$

where,  $T(M)$  refers to the domain transformation value of the host map,  $M_n$  denotes the LCA transition matrix,  $v_{x1}$  is the digital media value of the host map, and  $N$  represents the number of vertices altered by the transformation.

The method of the transformation in LCAT by the  $v_{x1}$  coordinate is given by Eq. (5). It includes the encrypted watermark part.

$$v''_{x1} = v'_{x1} + \alpha W \quad (5)$$

where  $\alpha$  represents the embedding parameter and  $W$  refers to the watermark.

Variations of the vector map are directly proportional to the embedding parameter ( $\alpha$ ). At the same time, the resistance of the watermark ( $W$ ) increases. The equation

applies acceptable changes to the vector map, including a large resistance value and 3-part  $\alpha$  values.

The inverse of LCAT's can be expressed as follows:

$$iT(M) = \sum_{n=0}^{N-1} M_n^{-1} \cdot v''_{x1} \pmod 2 \tag{6}$$

where  $iT(M)$  refers to the inverse transformation value of the host map and  $v''_{x1}$  is the digital media value of the transformation of the host map.

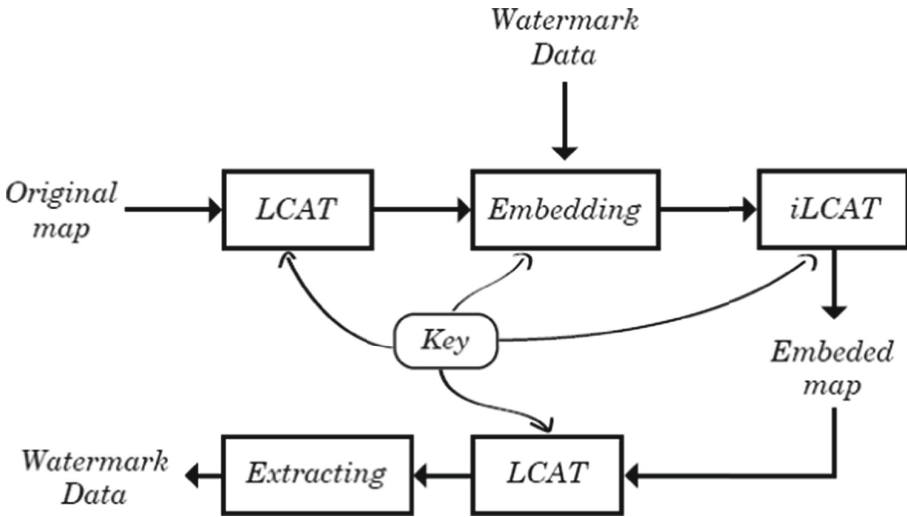


Fig. 1. Linear cellular automata transform (LCAT).

### 2.3 Watermark Embedding Phase

As previously noted, the proposed technique is operated in the frequency space. An overview of the embedding model of the watermark is demonstrated in Fig. 2. To facilitate encryption, a public key that consists of three parts (vector map, LCA transition matrix size  $(Mn)$ , and watermark for event scrambling) is used. A private key is used to perform decryption.

The algorithm is as follows:

1. Choose a pair of reference vertices,  $v_{f1}$  and  $v_{f2}$ , in the range  $(1 \leq v_{f1}, v_{f2} \leq n)$ . This serves as the vector map of  $M$  for security assurance.
2. Determine the number of vertices in the map file ( $M$ ) according to the transformed length ( $N$ ) to the frequency domain (do not include references).
3. Convert the coordinates of the vertices to the LCAT.

4. Using Eq. (5), encrypt the coefficients of  $W^*$ . This generates the data sequence  $W^* = \{w_i^* | w_i^* \in \{0, 1\}, i = 0, 1, \dots, l - 1\}$ .
5. Place  $W^*$  into the last two consecutive digits, which lowers the influence on precision, and assume that a double floating-point 16-digit coordinate value exists in a decimal fractional version. The value to be embedded falls in the range of 0 to 99 and does not correlate with  $w_i^*$ . Under the assumption that  $D$ , an integer, consists of the two digits, then:

$$W^* = \left\{ \begin{array}{l} \text{if } w_i^* \text{ is } 0 \text{ then } D \leq 50 \text{ and saved at the positions;} \\ w_i^* = 1, \text{ otherwise} \end{array} \right\} \quad (7)$$

6. After the watermark has been placed, use LCAT's inverse to restore the initial shape file of the frequency domain vector map.

### 2.4 Watermark Extraction Phase

Comparable stages are involved in the watermark insertion and watermark extraction processes, but the reverse order is adopted. In the extraction process, the stages used are the outcomes of the insertion phase, namely,  $v_{f1}$  and  $v_{f2}$  (reference vertices), a fixed size LCAT matrix ( $M_n$ ), and the watermarked vector map. These elements are illustrated in Fig. 3.

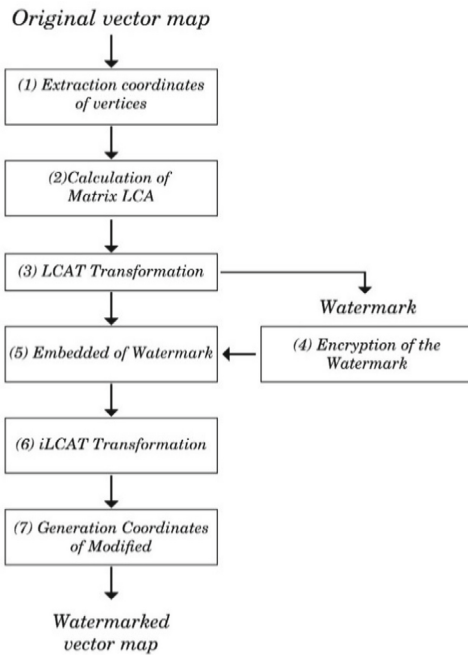


Fig. 2. Embedding process

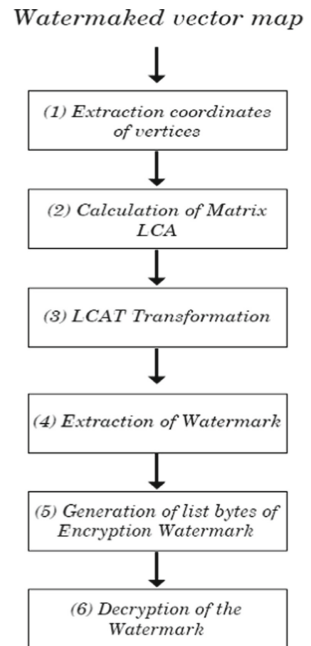


Fig. 3. Extraction process

The stages involved are the following:

1. Choose  $v_{f1}$  and  $v_{f2}$  ( $1 \leq v_{f1}, v_{f2} \leq n$ ), under the control of the private key  $k$ . These serve as the reference vertices for the vector map of  $M$ .
2. Identify the number of vertices in the map file ( $M$ ) and the length ( $N$ ) that will subsequently be transformed to the frequency space that have no references.
3. Transform the coordinates of each feature to LCAT.
4. Use the following equation to extract the watermark location and bits:

$$W^* = \left\{ \begin{array}{l} \text{if } D \leq 50 \text{ then } w_i^* \text{ is } 0 \\ w_i^* = 1, \text{ otherwise} \end{array} \right\} \quad (8)$$

5. Extract the original embedded watermark sequence ( $W$ ) by taking the inverse of the watermark pattern using the private key  $k$ .
6. Reconstruct the watermark pattern.

### 3 Results and Discussions

#### 3.1 Experimentation

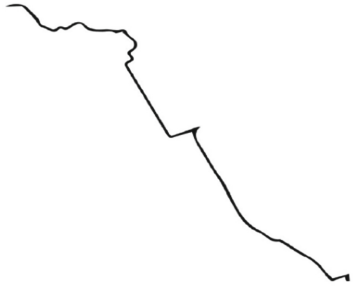
To evaluate the LCAT algorithm, a pair of *shapefile* maps were employed. The two vector maps consisted of the file type ESRI standard [12], which shows a number of maps covering Riyadh city in the Kingdom of Saudi Arabia. The maps are thus of various forms, including polyline, point, and polygon. A bitmap image was used for the copyright marker. In terms of the hardware used for the experiments, a machine with the Windows 10 Professional operating system with a 2.3 GHz processor and 16 GB memory were employed, as well as QGIS version 3.0. The Python programming language was also used.

The parts linked to every transform coordinate,  $M_n = 30$ ,  $\alpha$  was observed in the least-significant-bit (LSB) and  $T = 1$  for iterative embedding. The first test focused on the invisibility of the proposed approach. Vector 4 illustrates the watermarked vector maps using the method described previously. These generated the watermarked types are illustrated in Fig. 4 and 5.

NC computation was used for the performance evaluation, and it was also applied to investigate aspects of similarity between the original watermarks prior to and following the extraction (the values ranged from 0 to 1). The high-quality nature of watermarking approach is reflected in the elevated NC value, which is suggestive of a robust correlation. Equation 2 defines the NC between the initial value ( $w$ ) and the extraction watermarks ( $w_i^*$ ).

$$NC = \frac{\sum_{i=0}^M w_i X w_i^*}{\sqrt{\sum_{i=0}^M (w_i)^2} \sqrt{\sum_{i=0}^M (w_i^*)^2}} \quad (9)$$





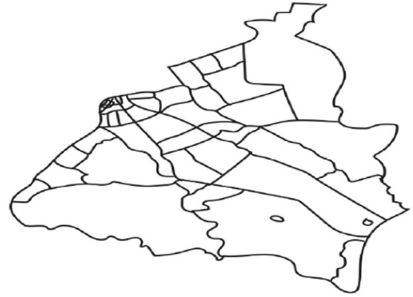
King Abdullah street



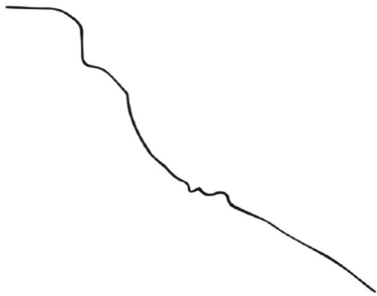
Spot height map of Riyadh City



Al-Safarat District



Al-'Olayya District

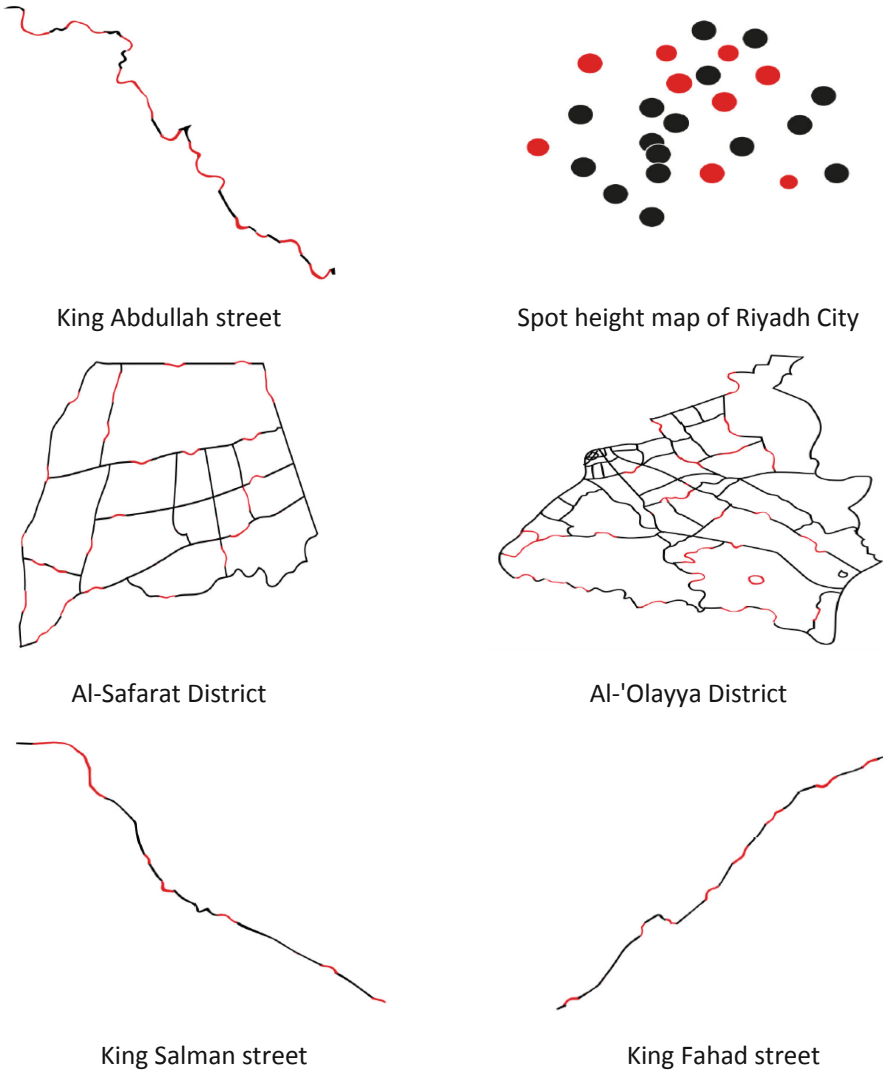


King Salman street



King Fahad street

**Fig. 4.** 2D vector map testing maps.















**Fig. 5.** Watermarked 2D vector maps for the testing maps in previous figure.

Table 1 provides a summary of the similarity test based on the NC value. Based on these results, it is reasonable to conclude that the extracted and original watermarks are closely comparable (reflected in the fact that the NC value is around 1). Furthermore, the content of the extracted and original watermarks was the same, along with the length.

The proposed algorithm protected the copyright using a watermark without impairing quality. This is because the watermarks that are extracted again from vector map files do not need to go through dimensions or content changes.

**Table 1.** Similarity test results

Used Map	Type	Features/Vertices	Embedded Data	Extracted Data	NC
Spot height map of Riyadh City	Point	30/30			0.998028
King Fahad street	Polyline	5/300			0.998109
King Abdullah street	Polyline	3/180			0.998114
King Salman street	Polyline	10/60			0.998067
Al-'Olayya District	Polygon	14/140			0.998085
Al-Safarat District	Polygon	40/400			0.998008

### 3.2 Invisibility Testing

The two parameters used in the invisibility evaluation (as a reference analysis to compute the RMSE) are given in Table 2. In turn, this decides on the alteration between the results of the interpolated watermark and the start of the map file. Equation 10 shows the equation used to create the RMSE:

$$\text{RMSE} = \left( \frac{1}{N_V} \sum_{i=0}^{N_V} |v_i - v_i^{*'}| \right) \quad (10)$$

where  $N$  refers to the number of vertex map vectors,  $v_{x1}$  denotes equivalent x coordinates in the initial vector map, and  $v_{x1}^{*//}$  denotes the equivalent x coordinates in the restored vector map.

**Table 2.** RMSE results (T = 1)

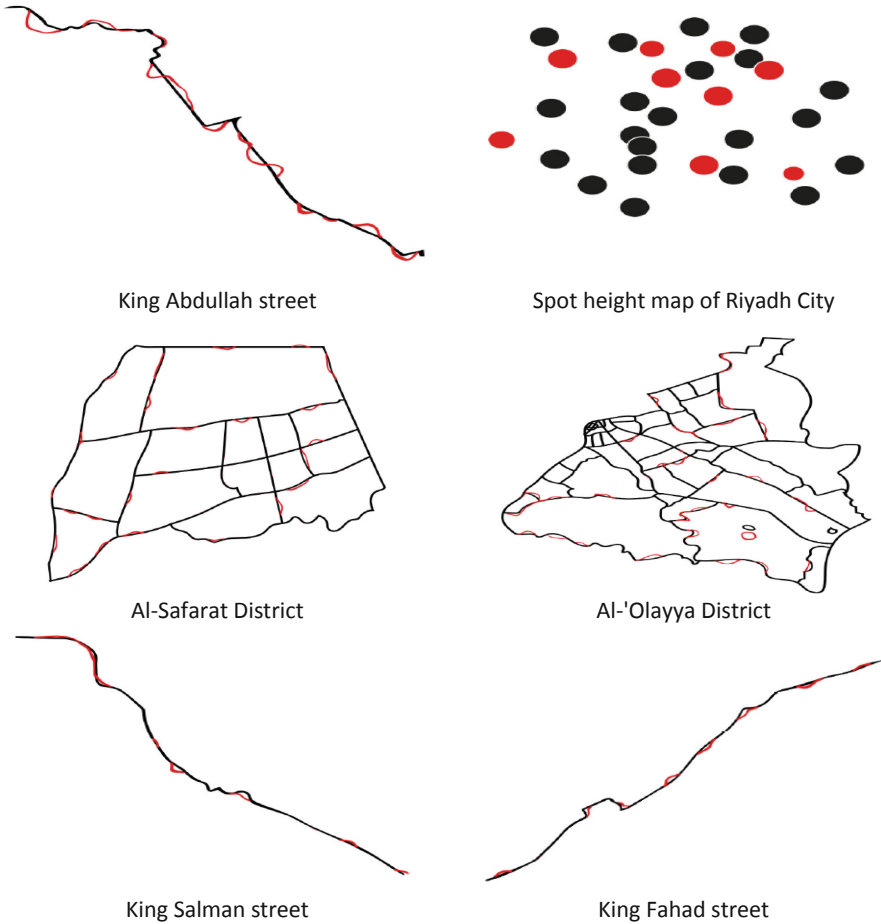
Used map	DC transform	DW transform	FF transform	LCA transform
Spot height map of Riyadh City	$3.6582 \times 10^{-5}$	$6.2875 \times 10^{-6}$	$8.8872 \times 10^{-7}$	$5.258 \times 10^{-10}$
King Fahad street	$4.6382 \times 10^{-4}$	$5.3245 \times 10^{-5}$	$8.1342 \times 10^{-6}$	$8.9877 \times 10^{-8}$
King Abdullah street	$2.6382 \times 10^{-4}$	$3.1361 \times 10^{-5}$	$4.5912 \times 10^{-6}$	$8.2319 \times 10^{-9}$
King Salman street	$1.6382 \times 10^{-4}$	$4.3129 \times 10^{-5}$	$7.2231 \times 10^{-7}$	$2.5612 \times 10^{-10}$
Al-'Olayya District	$6.6382 \times 10^{-4}$	$3.7326 \times 10^{-5}$	$6.5231 \times 10^{-6}$	$8.1723 \times 10^{-9}$
Al-Safarat District	$3.6382 \times 10^{-4}$	$5.9213 \times 10^{-5}$	$2.3111 \times 10^{-6}$	$1.2510 \times 10^{-9}$

### 3.3 Fidelity Evaluation

Digital watermarking cannot be identified using the naked eye, which improves system reliability. Additionally, the approach does not lead to any significant deterioration in the media file, and in terms of the RMSE, it is possible to access the furthest changes. Watermarking into the vector files is responsible for position change, which is reflected in the furthest distance. A measurement that compares the original vector map file, the coordinated vertex, and the watermarked vector map file can be applied to obtain a position change. QGIS was used in this study to express the further distance in terms of meters. Based on the test results, the longest notable shift in the data analysis amounted to 60 cm (Table 3). It is reasonable to conclude that the furthest distance value still achieves satisfactory accuracy in the vector map (Fig. 6).

Processing of the results on the watermark insertion map was identified as the factor leading to the failure of the extraction of the watermark. Furthermore, the distortion value was influenced by using the limiting factor in the phase of the watermark insertion. The identification of the watermark value took place based on the use of a bit matrix size of  $M_n \leq 30$ , which permitted extraction. In addition, the limiting factor is affected by the variation amplitudes of  $M_n \geq 35$  bits.

The watermark was retained in several techniques in certain tests, where despite a low level of robustness, both watermarks, the extracted and the embedded were similar with normalized-correlation value of 1. Regarding each LCAT value, changes on the value of the sequence complex on the vector mapping involved an LCAT computation being spread to ensure that the LCAT value stayed within the extraction range. Changes to the coordinate value directly influenced the inserted watermark bit value, where insertion occurred on the spaced-out domain. This was identified as a precursor to different results. Watermark conservation can be achieved more effectively by the transform domain



**Fig. 6.** Overlaying of the original and watermarked maps

**Table 3.** Fidelity test results

Map used	$\tau$ (m)	DC transform	DW transform	FF transform	Farthest distance (m) LCAT
Spot height map of Riyadh City	10	8.50	7.50	5	2.90
King Fahad street	10	8	7	5	2.5
King Abdullah street	30	15	11	7	3.20
King Salman street	78	20	15	10	7
Al-'Olayya District	78	22	14	11.60	6.58
Al-Safarat District	100	45	37	30	9.23

computation spanning the LCAT rather than the spatial technique. The degree to which the method is robust is dependent on the frequency domain algorithm (FDA), related programming methods, the data storage length, the extraction limit, and the quality of the asymmetric algorithm key.

## 4 Conclusion

The common challenges associated with existing watermarking schemes in vector maps relate to the issues of invisibility and fidelity. Additionally, the original cover is necessary for the scheme when extracting the watermark, given the “non-blind” nature of the process. In this paper, a novel frequency watermarking technique for 2D vector maps was proposed. In the LCAT domain, the proposed watermarking technique was used for embedding into the vector map. Invisibility evaluation indicated that similarity in the fidelity stages in the watermarked map are maintained. The root-mean-square-error value was consistent at approximately 0, and the distance from the original vector map remained within a maximum value of 10%.

## References

1. Chang, K.T.: Introduction to Geographic Information Systems. McGraw-Hill (2012)
2. Abubahia, A., Cocea, M.: Advancements in GIS map copyright protection schemes - a critical review. *Multimedia Tools Appl.* **76**(10), 12205–12231 (2016). <https://doi.org/10.1007/s11042-016-3441-z>
3. Ling, Y., Lin, C.F., Zhang, Z.Y.: A zero-watermarking algorithm for digital map based on dwt domain. In: He, X., Hua, E., Lin, Y., Liu, X. (eds.) *Computer, Informatics, Cybernetics and Applications*. LNEE 107, pp. 513–521. Springer, Dordrecht (2012)
4. Wu, J., Liu, Q., Wang, J., Gao, L.: A robust watermarking algorithm for 2D cad engineering graphics based on DCT and chaos system. In: Tan, Y., Shi, Y., Mo, H. (eds.) *Advances in Swarm Intelligence*. LNCS, vol. 7929, pp. 215–223. Springer, Berlin (2013)
5. Neyman, S.N., Pradnyana, I.N.P., Sitohang, B.: A new copyright protection for vector map using FFT based watermarking. *TELKOMNIKA Telecommun. Comput. Electron Control* **12**(2), 367 (2014)
6. AL-Ardhi, S., Thayanathan, V., Basuhail, A.: Copyright protection and content authentication based on linear cellular automata watermarking for 2D vector maps. In: Arai, K., Kapoor, S. (eds.) *Advances in Computer Vision CVC 2019. Advances in Intelligent Systems and Computing*, vol. 943. Springer, Cham (2020)
7. AL-ardhi, S., Thayanathan, V., Basuhail, A.: Fragile watermarking based on linear cellular automata using manhattan distances for 2D vector map. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **10**(6) (2019)
8. AL-ardhi, S., Thayanathan, V., Basuhail, A.: A watermarking system architecture using the cellular automata transform for 2D vector map. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **10**(6) (2019)
9. Wang, N., Zhang, H., Men, C.: A high capacity reversible data hiding method for 2D vector maps based on virtual coordinates. *Comput. Aided Des.* **47**, 108–117 (2014)
10. Wang, C., Peng, Z., Peng, Y., Yu, L.: Watermarking 2D vector maps on spatial topology domain. In: *International Conference on Multimedia Information Networking and Security*, pp. 71–74 (2009)

11. Martin del Ray, A., Rodriguez Sanchez, G.: Reversibility of linear cellular automata. *Appl. Math. Comput.* **217**(21), 8360–8366 (2011)
12. ESRI shapefile technical description. Technical report, 1 July



# Implementing Variable Power Transmission Patterns for Authentication Purposes

Hosam Alamlah<sup>(✉)</sup>, Ali Abdullah S. Alqahtani<sup>(✉)</sup>, and Dalia Alamlah<sup>(✉)</sup>

Louisiana Tech University, Ruston, LA 71270, USA  
hosam.amleh@gmail.com, alqahtani.aasa@gmail.com,  
dalia.alamlah@yahoo.com

**Abstract.** The last decade has witnessed an increase in adopting wireless systems. A wireless system enables two devices to utilize radio frequencies to communicate wirelessly. Moreover, in wireless systems, authentication can be performed wirelessly through transferring authentication information over-the-air. Alternatively, antennas in most devices enabled to communicate wirelessly today can control the power level of transmission to improve the wireless system's performance. In the present study, we examine the feasibility of varying the power level of transmission of a device's antenna for authentication purposes. This can be used in applications, such as obstructing relay attacks on wireless authentication systems. Furthermore, a prototype is built and tested utilizing Wi-Fi enabled systems.

**Keywords:** Access control · Wireless authentication · Variable power · Relay attack

## 1 Introduction

Wireless authentication has been out there for a while, mainly used for purposes like connecting a device to a wireless networks. For example, connecting to a Wi-Fi network. Recently, there has been a huge growth in the number of wireless systems deployed in different areas in our lives automating many manual functions for our convenience. A contemporary example of these wireless systems is passive keyless entry systems (PKES). These systems are utilized in the locks of some of the modern cars and in smart homes. Passive keyless entry is an access control method that responds in an automatic fashion when the key holder is in the vicinity of the lock granting physical access on approach. Such systems perform authentication wirelessly. Generally, authentication that is performed wirelessly incorporates a device attempting access to resources which is known as the prover, and a party that verifies the prover's credentials and makes access decisions. This entity is known as the verifier. The hardware of the prover and the verifier incorporate radio frequency transceivers. These transceivers are used to give the ability to the prover and the verifier to detect each other, communicate, and to transfer the authentication information.

Wireless authentication is user friendly. For instance, it is more convenient for users to simply approach a locked door and it unlocks automatically. However, employing



wireless authentication comes with the risk of relay attacks. In this attack, the communication between the prover and the verifier is begun by the attacker. This attacker then simply relays messages between the prover and the verifier without changing these messages or even reading them. Generally, the relay attack device captures communications then simply retransmits them. As can be seen in Fig. 1, in usual circumstances, when the prover and the verifier are not in range, they are not able to communicate nor exchange authentication information. Therefore, no authentication attempts occur. However, when an attacker places a relay device to repeat the wireless signals of the prover to reach the verifier and vice versa. It will appear to the prover and to the verifier that they are in range. Thus, they will be able to communicate and exchange authentication information leading to successful authentication. Doing this grants the attacker illegitimate access to resources without the prover requesting them. For example, unlocking a wireless lock without the key holder approving it.

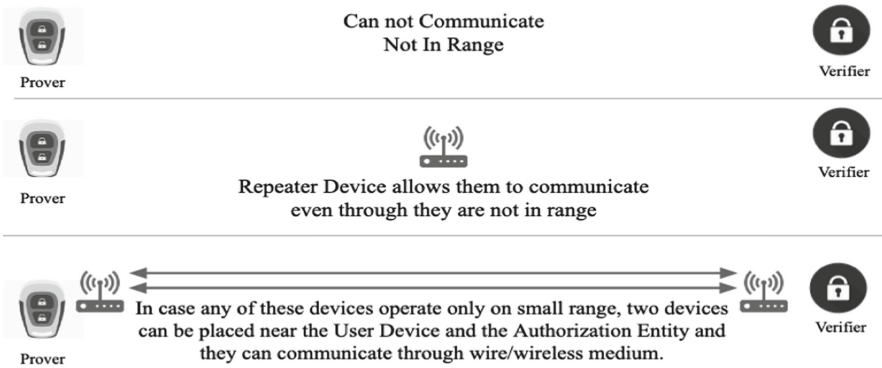


Fig. 1. The relay attack.

In the present study, we propose using changeable levels of the power of transmission to be utilized in the authentication process. This is done by utilizing the antenna’s ability of changing the power level of transmission and the ability of the receiver devices of measuring the power levels of the received signals. The control of the power level of transmission is used in many of the wireless communication protocols as in Bluetooth, Wi-Fi, LTE, RFID, and NFC to assure effective transmission of data and to reduce interference. For example, a Node-B in the Universal Mobile Telecommunications Service (UMTS) mobile can execute power control up to 1500 times in a second [6]. In Radio transceivers, power control can be performed by adjusting bias voltages for the power amplifier in the transceiver circuit. This paper shows that using changeable power levels in transmission as an authentication factor can, in some cases, obstruct relay attacks on wireless authentication systems. After covering the related work in the next section, the system model is presented in Sect. 3, then, tested in Sect. 4.

## 2 Related Work

As stated earlier, antennas transmission power control is essentially utilized in wireless communications systems to assure good system performance (e.g., effective data rate, low interference). For instance, power control is used in UMTS cellular networks to compensate for the changes due to pathloss and Rayleigh fading [8]. Now, received signal strength (RSS) is utilized in some systems for proximity-based authentication [9]. However, changeable power levels at transmission are not utilized in authentication applications. We believe it is feasible to use changeable power levels at transmission in authentication applications, which can be employed as a simplistic and straightforward method to prevent some of the relay attacks in wireless systems. Today, there are a few countermeasures proposed to defend against relay attacks. The distance bounding protocol for example. This protocol enables the verifier device to establish a distance bound where provers requests for access are approved [1]. The distance bound is determined by the round-trip time that the signal takes to travel back and forth between the verifier and the prover. It was proposed for the distance bounding protocol to combat relay attacks in contact smart cards [3] and RFID [5]. Nevertheless, the distance bounding protocol is complicated and requires highly precise time measurements, and has limitations especially in non-line-of-sight settings due to multipath delays. The authors [7] utilized jamming signals to counter relay attacks in NFC systems. In their system, the prover transmits a jamming signal at the beginning of the NFC communication. However, jamming can impacts other legit radio frequency applications. The authors in [4] proposed countering relay attacks by analyzing physical layer characteristics, which relies on detecting relaying attackers by measuring the noise statistic variations at the receiver. However, their proposed method needs calibration. Moreover, it may fail to work in case of an external factor producing noise at the same frequency. The authors in [2] proposed using ambient sound information to counter relay attack on cars, which is done by having both the car and the key record sound fingerprint then compare the recordings to make sure they are proximate. However, their proposed system requires multiple audio recorders. In the next section, we present the proposed system model that utilizes changeable power levels at transmission for authentication purposes.

## 3 System Model

In the proposed system model, the prover device transmits a pre-shared secret key at varying power levels, at transmission, following a predetermined pattern. The transmission pattern is determined by a pre-shared equation. This equation is pre-shared between the prover and verifier. The pre-shared equation uses the value of the time and yields the power levels which will be used when sending the pre-shared secret key to the verifier. Consequently, the prover transmits the pre-shared secret key using the same power transmission levels the verifier will use in authenticating the prover. Assuming that the prover device can transmit keys at  $n$  different power levels and the power pattern consists of  $k$  different power levels. Then the number of possible power level's combinations is  $\frac{n!}{(n-k)!}$ . As previously discussed, the used combination varies with time according to the pre-shared equation. In the proposed system,  $P_{max}$  is transmitted first, then fractions of  $P_{max}$  are transmitted as follows:

At  $t_1$  the prover transmits  $P_1 = P_{max}$   
 At  $t_2$  the prover transmits  $P_2 = a_1 P_{max}$   
 At  $t_3$  the prover transmits  $P_3 = a_2 P_{max}$   
 At  $t_4$  the prover transmits  $P_4 = a_3 P_{max}$   
 At  $t_k$  the prover transmits  $P_k = a_k P_{max}$   
 Where  $a_1, a_2, \dots, a_k < 1$

The verifier receives the pre-shared secret key at the different power levels from  $t_1$  to  $t_k$ . Therefore, it measures  $k$  different received power levels. Since  $P_1 > P_2, P_3, \dots, P_k$ ,  $P_{Received}(P_1)$  is going to be the instance in which the highest received power measured, assuming the distance between the verifier and receiver did not change during the period from  $t_1$  to  $t_k$ . Generally, when an electric signal propagates through a path it loses a part of its power. This loss is known as the pathloss. Pathloss is a result of several phenomena (e.g. free-space loss, and others). Pathloss can be influenced by several factors such as the terrain, medium, and the distance between the transmitter and the receiver. Hence, the received power at the verifier can be presented as follows:

$$P_{Received} = P_{sent} - pathloss \quad (1)$$

Using (1) for the cases when  $P_1$  and  $P_2$  are transmitted:

$$P_{Received}(P_1) = P_{max} - pathloss \quad (2)$$

$$P_{Received}(P_2) = a_1.P_{max} - pathloss \quad (3)$$

Then, using (2) and (3), the verifier solves for  $P_{max}$  and  $pathloss$  as the following:

$$P_{max} = \frac{1}{1 - a_1} \times [P_{Received}(P_1) - P_{Received}(P_2)] \quad (4)$$

$$pathloss = \frac{1}{1 - a_1} \times [a_1.P_{Received}(P_1) - P_{Received}(P_2)] \quad (5)$$

Now, when  $P_3, P_4, \dots, P_k$  are transmitted, the verifier measures  $P_{Received}(P_3), P_{Received}(P_4), \dots, P_{Received}(P_k)$  which also can be calculated from the following equations:

$$\begin{aligned}
 P_{Received}(P_3) &= a_2.P_{max} - pathloss \\
 P_{Received}(P_4) &= a_3.P_{max} - pathloss \\
 &\vdots \\
 P_{Received}(P_k) &= a_{k-1}.P_{max} - pathloss
 \end{aligned} \quad (6)$$

The verifier now has the values of  $P_{max}$  and  $pathloss$  which were calculated from (4) and (5) and the values of  $P_{Received}(P_3), P_{Received}(P_4), \dots, P_{Received}(P_k)$ , which are measured by the verifier. These values are used by the verifier to verify if (6) checks. If it does, power authentication is successful. Otherwise, power authentication fails. It is possible to verify only one instance of (6). However, more instances can be verified to increase the security and the robustness of the system. After power authentication is completed, the pre-shared secret key which was transmitted by the prover is verified by the verifier, and then the access decision is made.

As previously discussed, relay attack devices capture and repeat the prover signal. If changeable power levels at transmission are used, it becomes difficult for the attacker device to reproduce the same power levels transmitted by the prover. Since, the transmission levels vary with time according the pre-shared transmission patterns equation. This equation is known only by the prover and the verifier. As shown in Fig. 2, in the case of a relay attack, the signal is received from a repeater device not from the prover's device, (6) verification fails resulting in power authentication failure.

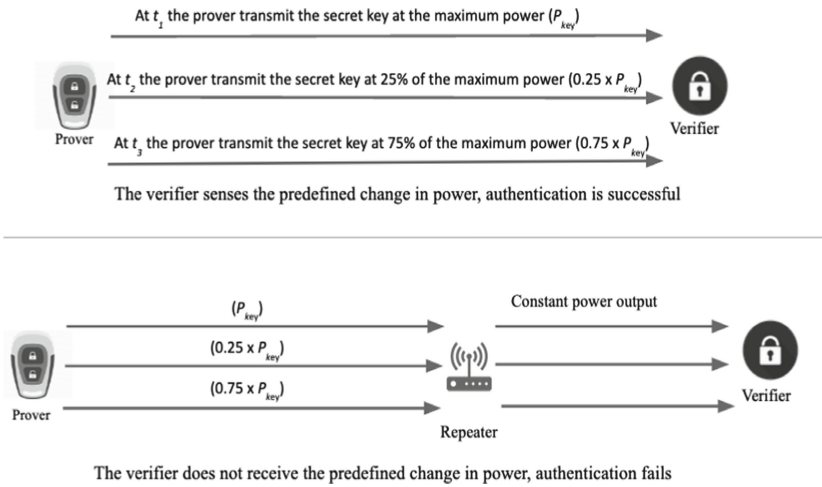


Fig. 2. System model against relay attacks

## 4 Experiment

To test the system model proposed in the present study, IEEE 802.11 devices were used for the experiment. An IEEE 802.11 access point that has the ability to manipulate the power level of transmission via software was employed to perform the prover's device function. Thus, it was configured to send a pre-shared secret key via broadcasts. This key was transmitted at varying power levels. A Raspberry Pi was employed to be the verifier device to simulate the scenario shown in Fig. 2. As can be seen from the figure, the pre-shared secret key was sent at varying transmission power levels (three in this case). An IEEE 802.11 repeater was utilized as the attacker device, and was programmed to repeat the signals coming from the prover to the verifier. Figure 3 shows the authentication software installed on the Raspberry Pi. Figure 3(a) demonstrates how authentication was completed successfully when the key is sent directly from the prover device. Alternatively, Fig. 3(b) demonstrates how authentication was not successful when the key is retransmitted by the repeater device.

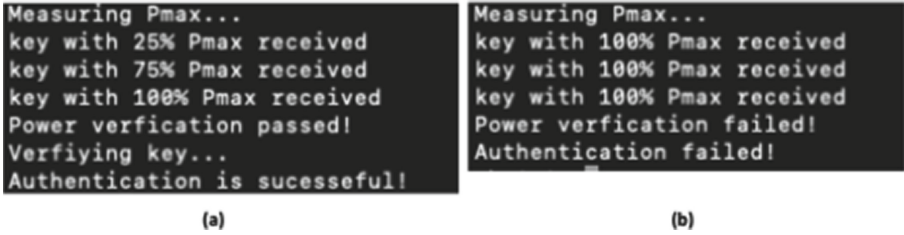


Fig. 3. Experiment results

## 5 Conclusion

In the present study, utilizing changeable power transmission levels for wireless authentication was purposed. This paper presented a system model that utilizes changeable power transmission patterns for authentication. Then, successfully implemented and tested prototype. Most of the devices sold in the market nowadays are able of transmitting at varying power levels; therefore, the presented system is a direct and simplistic way to counter relay attacks. Future research would include implementing other methodologies that use changeable power transmission for authentication purposes.

## References

1. Brands, S., Chaum, D.: Distance-bounding protocol. In: Advances in Cryptology-EUROCRYPT 93. LNCS, vol. 765, January 2010
2. Cho, W., Seo, M., Hoon, M., Lee, D.: Sound-proximity: 2-factor authentication against relay attack on passive keyless entry and start system. *J. Adv. Transp.* **2018** (2018). Article ID 1935974
3. Drimer, S., Murdoch, S.J.: Keep your enemies close: distance bounding against smartcard relay attacks. In: Proceedings of the USENIX Security (2007)
4. Hamida, S., Thevenon, P., Pierrot, J., Savry, O., Castelluccia, C.: Detecting relay attacks in RFID systems using physical layer characteristics. In: Wireless and Mobile Networking Conference (WMNC) (2013)
5. Hancke, G., Kuhn, M.: An RFID distance bounding protocol. In: IEEE/Create Net Secure Communication (2005)
6. Markus, L.: Evaluation and modeling of power control information in a 3G cellular mobile network. Techn. Univ., Dipl.-Arb., Wiens (2009)
7. Oh, S., Doo, T., Ko, T., Kwak, J., Hong, M.: Countermeasure of NFC relay attack with jamming. In: 12th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT) (2015)
8. Wei Tan, C.: Optimal power control in Rayleigh-fading heterogeneous networks. In: 2011 Proceedings IEEE INFOCOM (2011)
9. Zhang, J., Wang, Z., Yang, Z., Zhang, Q.: Proximity based IoT device authentication. In: Proceedings IEEE Conference Computer Communication (INFOCOM), pp. 1–9, May 2017



# SADDLE: Secure Aerial Data Delivery with Lightweight Encryption

Anthony Demeri<sup>1</sup>, William Diehl<sup>1(✉)</sup>, and Ahmad Salman<sup>2</sup>

<sup>1</sup> Virginia Tech, Blacksburg, VA 24061, USA  
{demeri,wdiehl}@vt.edu

<sup>2</sup> James Madison University, Harrisonburg, VA 22807, USA  
salmanaa@jmu.edu

**Abstract.** Low-cost devices in the Internet of Things (IoT) can be integrated into Unmanned Aerial Systems (UAS) as a means of collecting remote data and forwarding to central collection points. However, sensitive data is subject to compromise, and should be protected using cryptography. In order to minimize threats to authenticity, data should be encrypted using session keys known only to participating nodes. In general, however, incorporation of capabilities required to both generate secure session keys and encrypt or decrypt sensitive data is difficult in low-cost IoT installations, due to resource and performance constraints. In this research, we implement a combined public and secret key secure data delivery system in a low-cost aerial platform, which incorporates cryptographic accelerators and required peripherals, in the Zybo Z7-10 System-on-Chip. Components are integrated using a flexible and extensible Applications Programming Interface (API) in a hardware-software design approach, and flown on a low-cost F450 ARF quad-copter drone. Resulting components consume 60% of the slice resources of the Zybo FPGA, and achieve a takeoff weight of 1.2 kg. A flight demonstration is performed, where sensitive data, collected at a remote sensor, is securely delivered to a host.

**Keywords:** ECC · AES · Codesign · Cryptography · Embedded systems · Encryption · Lightweight · Drone

## 1 Introduction

Smaller and lower-cost embedded devices comprising the Internet of Things (IoT) are revolutionizing information technology. IoT sensors can be emplaced at remote locations, and can provide monitoring of operational, environmental, or security conditions. Although some decisions can be made autonomously in an edge-centric network, much of remotely collected data must be forwarded to central locations for processing. While data forwarding can be accomplished using landline or higher-powered Radio Frequency (RF) networks, low budgets often preclude such arrangements.

The use of Unmanned Aerial Systems (UAS), i.e., “drones” or “Unmanned Aerial Vehicles” (UAVs), for data collection from sparse and geographically-distributed nodes, is an established trend in academia and industry. In particular, data collection using UAS can support environmental efforts (e.g., agricultural and conservation) [1], logistics functions (e.g., tracking of deliveries and traffic optimization) [2], and Critical Infrastructure Protection (CIP) (e.g., pipeline and power line monitoring) [3].

However, data collected by remote sensors and forwarded by a UAS is vulnerable to cyber exploitation. Risks could include adversary collection of information, or malicious alteration or denial of service by a malicious actor. Safeguarding of data, at rest and in motion, often requires cryptographic protections. Ideally, protections would include confidentiality (i.e., preventing an eavesdropper from reading a device’s communications), integrity (i.e., preventing substitution of any communication), and authenticity (i.e., assurance that communications come from the purported sender and not from a malicious third-party). While these cryptographic services can be achieved through a single secret key cipher (in conjunction with secure hashes or authenticated encryption), secret keys (e.g., session keys) utilized for the above services must be ephemeral – nodes must be able to securely, rapidly, and repeatedly regenerate device-to-device keys. Additionally, in a network consisting of many secure nodes, no two nodes should possess the same secret key, as this increases the possibility of an adversary co-opting a node to employ in a spoofing or man-in-the-middle attack. But centralized key management and distribution is logistically challenging. A solution would be to field a public key cryptosystem to operate in tandem with the secret key cipher, to provide and replenish session keys. However, public key ciphers are much more expensive in terms of power, energy, and resources than secret key ciphers, and are rarely incorporated in IoT devices [4].

In this research, we tackle the above challenges head-on through SADDLE: Secure Aerial Data Delivery with Lightweight Encryption. Specifically, we design and demonstrate a secure UAS, capable of securely collecting sensitive data at a remote “sensor” node, forwarding the data using a drone, and delivering the sensitive data to a central collection point, or “host” node. Each of the nodes in SADDLE is capable of public and secret key encryption, in order to generate bi-nodal secret session keys from private keys known only to individual nodes, and exchange sensitive data using high-speed encryption. For establishment of secret session keys, we use the U.S. National Institute of Standards and Technology (NIST) standard Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol [5]. For high-speed exchange of encrypted secure data, we use the NIST-approved Advanced Encryption Standard (AES) block cipher [6].

Since cost is a critical factor in design of SADDLE, we leverage low-cost components. Our secure data delivery system is implemented on Digilent Zybo Z7-10 System-on-Chips (SoC), and our aerial component is based on the DJI F450 ARF quad-copter drone, controlled with an mRo Pixhawk 2.4.6 flight controller, and remotely controlled by a drone pilot. Inter-nodal communications use

the short-range and low-power Bluetooth (IEEE 802.15.1) standard, facilitated by low-cost Digilent Bluetooth Zybo peripherals.

Our secure data delivery system, consisting of drone, sensor, and host nodes, is implemented using hardware-software (HW-SW) codesign. This optimizes available resources by permitting control-intensive processes to be rendered in SW, and data-intensive processes, such as cryptographic algorithms, to be implemented in HW. Specifically, we implement two cryptographic accelerators, one for Elliptic Curve Cryptography (ECC) point multiplication, and one for AES encryption and decryption. Our design also incorporates peripherals required for data acquisition, communication, and external monitoring, which are integrated into the design. Our SW model incorporates a flexible and extensible Application Programming Interface (API), in a baremetal approach (i.e., no operating system or kernel), to facilitate an easily understood command interface which can be upgraded with future capability. Our contributions in this research are as follows:

1. We develop a flexible and extensible SW API, applicable to all types of nodes (drone, sensor, host), which provides a flexible command interface, and is easily upgradeable with more advanced capability.
2. We integrate public and secret key cryptographic accelerators into a single lightweight System-on-Chip (SoC) using HW-SW codesign, to solve the problem of secure and repeatable bi-nodal session key generation.
3. We demonstrate incorporation of our secure data delivery system into an actual low-cost aerial platform, the F450 ARF quad-copter drone, and demonstrate its applicability for remote sensing while maintaining data security.

The paper is organized as follows: In Sect. 2, we provide background on public and secret key cryptography, and review previous explorations of data collection using UAS. In Sect. 3, we detail our methodology, including HW components, HW-SW codesign, SW model, and system integration. In Sect. 4 we present post-implementation results, and present conclusions in Sect. 5.

## 2 Background and Previous Work

### 2.1 Remote Sensor Data Collection with UAS

Many Wireless Sensor Network (WSN) applications have used UAS to either collect data from nodes or to supplement the system. These applications include environmental monitoring, agriculture and livestock optimization, and disaster recovery. Huang et al. [7] present a WSN deployment model using UAS in post disaster scenarios. In [8] Polo et al. use a UAS to collect data from WSNs used to gather data in an agricultural application. Further, in [9] Potter et al. use a UAS to collect data from sensor nodes deployed in a water stream for the purpose of collecting data such as water temperature, dissolved oxygen and pH level.

However, most lightweight UAS-WSN, designed for environmental and agricultural monitoring, or disaster relief, do not use security measures and protocols



in their systems. While these applications were conceived to operate in a non-threat environment, the exponential expansion of the IoT increases all cyber-attack surfaces, and necessitates that we go back and rethink the suitability of vulnerable systems. Some examples of cryptographic deployments in lightweight drones include [10], where authors implement linear homomorphic encryption (LinHAE) to provide a secure ground-to-air control loop, and [11, 12], in which authors devise a lightweight combination of public and secret key encryption for light drones.

High-end UAS of the military and intelligence collection services of many nations certainly employ secure communications, however, these come at high cost and complexity. The goals of our research are to investigate low-profile secure data delivery at very low cost, i.e., about USD 1,500 total cost.

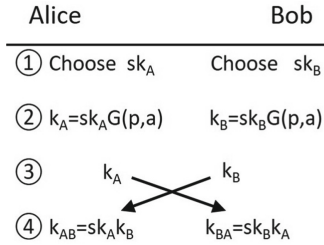
## 2.2 Elliptic Curve Cryptography

Public key ciphers generate a pair of related keys – one publicly available (i.e., the public key  $pk$ ), and one held privately (i.e., the private key  $sk$ ).  $pk$  is generated as a function of  $sk$  and other publicly-specified parameters of the system. The generation of  $pk$  is a “trap-door” function, where security of a public key cryptosystem relies on the fact that recovery of  $sk$  with knowledge of only  $pk$  and system parameters is computationally infeasible. The advantage of public key ciphers is that key exchange with new parties is simplified, using key exchange mechanisms such as Diffie-Hellman (DH) [13]. Additionally, the fact that each party possesses a unique  $sk$  facilitates cryptographic services such as digital signatures (e.g., Digital Signature Standard). A disadvantage of public key ciphers is that execution times are significantly longer. Therefore, public key ciphers are not viable for high-throughput or low-power communications. Rather, their intended use is to enable key exchange and generation of “session keys,” which then enable high-speed communication between participating parties using secret key ciphers.

ECC is a public key cryptosystem proposed in [14, 15]. In ECC, an elliptic curve  $E$  over a Galois Field  $GF(p)$  is a set of points fulfilling the equation of the curve. Such an equation is  $y^2 = x^3 + ax + b$ , where  $a$  and  $b$  are parameters of the curve, belonging to  $GF(p)$ . Two operations, addition and doubling, are performed on points  $A$  and  $B$  resulting in a third point  $C = A + B$ . A single point,  $A$ , can be doubled, giving  $C = 2A$ . Scalar multiplication  $kA$ , using key  $k$ , is equivalent to the sum of  $k$  instances of  $A$ .

ECC can be used with the ECDH protocol for exchanging session keys for use by secret key ciphers [13]. In ECDH, Alice and Bob first generate secret keys  $sk_A$  and  $sk_B$ . They then compute and exchange intermediate values  $k_A$  and  $k_B$  using point multiplications. Finally,  $k_{AB} = k_{BA}$ , is simultaneously derived by Alice and Bob through another point multiplication, as shown in Fig. 1, and represents a shared secret key known only to Alice and Bob.

ECC is a popular research topic from mathematical and implementation points of view. Lightweight implementations of ECC have been proposed in SW and HW. In [16] Zhou et al. present a lightweight implementation of ECC using



**Fig. 1.** Elliptic Curve Diffie-Hellman (ECDH) Key Exchange Protocol. Two users, Alice and Bob, each choose a secret key, and compute intermediate keys, according to agreed public parameters  $G(p, a)$ . After exchanging intermediate keys  $k_A$  and  $k_B$ , each compute shared session key  $k_{AB} = k_{BA}$ .

NIST prime P-256 targeting an 8-bit microcontroller. In [17] Al-Adhami et al. present a 256-bit ECC implementation suitable for RFID tags. The design uses the optimized GNU Multiple Precision (GMP) arithmetic library. The authors report the fastest scalar multiplication time on the target platform as a result of the optimizations they performed. In [18] Marzouqi et al. present another NIST P-256 implementation on FPGAs. They use Karatsuba multiplier with two levels of division and conquer approach to optimize their implementation. Rahman et al. [19] present another NIST P-256 implementation targeting FPGA. The design uses Jacobian coordinates to avoid the costly division operations.

Several studies have been conducted on the usage of ECC to secure WSNs. In [20] Ozgur presents a public-key infrastructure using ECC. The system uses multiple UAVs acting as Certification Authorities (CAs) to provide certifications during the pairwise key exchange. The system relieves the WSNs from the extra overhead such as storage. In [21] Lu et al. use public-key protocols to solve the orphan node problem in symmetric-key management. The system provides digital signatures as well as other public-key services while saving energy. Nadir et al. [22] use the TinyECC library to provide public keys to WSNs through ECC. They show that their system is suitable for different scenarios such as in the initial deployment phase and when introducing a new node to the system. They also show that the memory usage is minimal. In [23] Malathy et al. introduce a border surveillance clustered network secured through ECC. Their system uses digital signature algorithms for effective key-generation as a shared signature between the sender and the authenticated recipient. Hossain et al. [24] present a HW implementation of ECC for WSNs. They use polynomial basis over binary fields  $GF(2^m)$  for their proposed design and show that it is suitable for FPGA implementations.

### 2.3 Advanced Encryption Standard

Secret key ciphers use a secret key to encrypt information at origin and decrypt information at destination. Advantages of secret key ciphers include speed and

simplicity of engineering, in that encryption and decryption algorithms are usually closely related. For this reason, secret key ciphers are typically used in both high-throughput and low-power applications. One disadvantage of secret key ciphers is that all parties must possess the same key. This requires either embedding the secret key in all devices party to the communication, or using a key distribution mechanism to get the secret key to all senders and recipients.

AES is a worldwide standard for secret key encryption defined in [6], and uses key sizes of 128, 192, or 256 bits. AES-128 uses a 128-bit key, encrypts 128-bit blocks of plaintext, and consists of 10 rounds. There are four transformations which occur on a state, defined as a 4-by-4 matrix of bytes. The SubBytes transformation conducts one-to-one byte substitutions. The ShiftRows transformation permutes the state by rotating the  $i$ th row left by  $i$  bytes across rows 0 through 3. The MixColumns transformation is a column multiplication on each column of the state by a matrix of constants in  $GF(2^8)$ . In the final transformation, AddRoundKey, a 128-bit round key is added to the state through a bitwise xor operation. Implementation of AES is a mature research area, and examples of FPGA implementations are ubiquitous. Some examples include [25–28].

### 3 Methodology

In order to optimize heterogeneous components, some of which are control-intensive and some of which are data-intensive, we use a HW-SW co-design approach. In this approach, a flexible central control structure, with extensible API, is implemented in SW, while cryptographic accelerators and communications peripherals are implemented in HW. The extensible API ensures a rich and flexible interface environment for current and future accelerators and peripherals.

Our design consists of three types of nodes: *drone*, *sensor*, and *host*. The drone node is installed on the quad-copter, and must enable data encryption and decryption, as well as sufficient on-board storage. The sensor must contain peripherals to interface with a data collection element (e.g., temperature sensor), and data encryption capability. The host must allow for data decryption. All three nodes must enable wireless communications (e.g., Bluetooth), and session key establishment capability.

#### 3.1 Hardware Components

**Zybo Z7-10 System on Chip.** To meet the above requirements, we choose a low-cost SoC solution consisting of Digilent Zybo Z7-10, with Peripheral module (Pmod) set, and MicroSD flash memory storage. The Zybo Z7-10 includes a Bluetooth based communication framework (available through the BT2 Pmod peripheral), a Zynq-7010 FPGA, and dual-core ARM Cortex A9 processor. We also employ an Organic Light-Emitting Diode (OLED) as a convenient status and debugging tool.

**ARM Cortex A9.** The ARM Cortex-A9 Processor System (PS), operating at 667 MHz, is the central processor of the Zybo Z7 system, which is tightly coupled to the Xilinx 7-series FPGA logic and optimized for low power. In this research, the usage of the on-device Cortex A9 processor is instrumental in implementing our baremetal embedded system. By using Xilinx Vivado’s Software Development Kit (SDK), we are able to directly write, cross-compile, and program the respective HW.

**Zynq 7000 Programmable Logic.** The Zynq 7000’s programmable logic (PL) enables use of Intellectual Property (IPs) to allow HW-SW codesign to optimize resource usage and maintain extensibility. The Zynq 7000 has 17,600 LUTs, 35,200 Flip-flops, and 270 KB Block RAM. We use the Xilinx 7-series based PL to interface with all of the SADDLE system’s HW peripherals, including the Digilent Pmod Temperature Sensor, the Diligent Pmod Bluetooth 2, the Digilent Pmod OLED, the ARM Cortex A9 PS, and the Zybo Z7-10’s General Purpose Input Output (GPIO) pins.

**Digilent Pmod TMP3 Temperature Sensor.** In order to demonstrate the secure data delivery aspect of SADDLE, we must collect sensor data at a remote node, and deliver it to the host by means of the drone. We use the Digilent Pmod TMP3 temperature sensor, which is built around the Microchip TCN75A. This operates with up to 12-bit resolution, and senses ambient temperatures with up to 0.0625 degree C resolution. It interfaces with the Zybo Z7-10 through an 8-pin Pmod connection using the Inter-Integrated Circuit (I2C) interface. As an I2C device, the serial clock and serial data lines must be pulled up to logic high voltage levels through external pull-up resistors.

The use of temperature data is used to simulate “notionally secure data”. In real-world applications, sensitive data could consist of temperature, pressure, or electrical readings of critical infrastructure, intelligence-collecting sensors, etc.

**Digilent Pmod Bluetooth 2.** In order to ensure SADDLE maintains a standardized, secure, and extensible communication framework, we incorporate a Bluetooth-based system. In particular, we use the Digilent Pmod BT2: Bluetooth Interface peripheral. The BT2 uses a 12-pin UART interface, and has an associated lightweight IP, which requires 420 Xilinx Look-Up Tables (LUTs), and is freely available for use via Xilinx Vivado’s IP repository. We use the BT2 for all device-to-device communication, while also echoing outgoing messages via UART-to-USB, to allow for terminal-based communication, which facilitates debugging and external event monitoring. The Digilent Pmod BT2 uses the RN42 Class 2 Bluetooth radio, capable of up to 3 Mbps at 20 m distance.

The BT2 IP provides access to an interrupt-line, which, similar to a HW-based UART interrupt, allows for selective interrupts (in this case, RX, TX, and error-based interrupts) to be triggered upon satisfaction of a certain criteria. SADDLE capitalizes upon the HW interrupts to build messages from incoming data and serialize messages for outgoing data.

Specifically, all bytes are read in through the BT2 using an on-device UART. Upon receipt of a byte, the BT2 RX interrupt triggers an interrupt service routine (ISR) to be called from SW. In order to not waste valuable processor cycles within the ISR, a simple processing flag is set, which indicates a UART RX needs to be processed. Shortly thereafter, the main processing loop reads the UART RX processing flag, receives the byte from the UART RX buffer, and attempts to build a new message, or add to an existing message, after which, the remainder of the transaction is abstract to the BT2. Similarly, the BT2 TX interrupt can be triggered in order to indicate a message is ready to be sent through the BT2 module. As the BT2 module is UART based, the processor clock rate of the Cortex A9 (667 MHz) is significantly faster than any supported baud rate. Therefore, when transmitting a message, the BT2 module must receive each byte in a serial manner, the nature of which is abstract to the module and handled by user-defined SW.

**Diligent Pmod OLED.** In any embedded system, the ability to receive visual debugging output during runtime is critical for trouble-shooting the many potential SW and HW bugs which may be present. SADDLE incorporates the Diligent Pmod OLED not only due to its lightweight IP (498 LUTs), but also due to its usability. The Diligent Pmod OLED features a powerful, user-friendly API, which allows a programmer to easily interface with the device, saving valuable design, implementation, and debugging time. For extensibility, we provide a simple `OLED_DEBUG` C macro, which can be placed anywhere within the user code to update the embedded system’s state, which maintains the last line number and function name referenced by the `OLED_DEBUG` macro. Furthermore, we provide a `DEBUG_ERROR_MESSAGE` macro, which can be placed anywhere in the user’s code to trap the CPU, flash onboard LEDs, and display an appropriate debug message on the OLED. This powerful, yet intuitive interface, allows for a programmer to receive visual, temporal queues as to when their program failed, as well as visual, spatial queues as to where their program failed (identified by the C macros’ automatic inclusion of line number and function name).

## 3.2 Cryptographic Accelerators

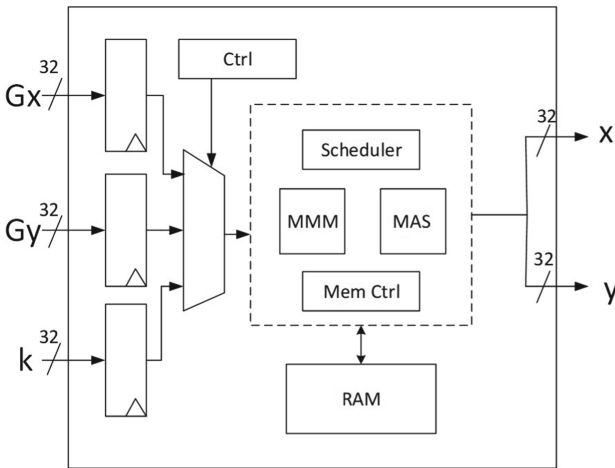
**ECC256 Point Multiplication Accelerator.** The number of bits required to represent a point on curve  $E$  in  $GF(p)$  represents the key strength of the ECC implementation. We employ parameters standardized in [5] using a fixed field size of 256 bits. Our design implements ECC256 point multiplication in a tailored HW accelerator, based on the design introduced in [29], and analyzed in [30]. We choose this implementation since it is designed to perform scalar multiplication with low area, and performs point addition and doubling over modified Jacobian coordinates, to avoid costly divisions.

This implementation uses the most popular method of facilitating multiplication in ECC, which is Montgomery Multiplication. Introduced in [31], a product  $abR \bmod N = aR \bmod N * bR \bmod N$  is computed, where  $aR$  and  $bR$  are

representations of  $a$  and  $b$  in the Montgomery Domain, based on a residue  $R$ . Although there is significant overhead in conversion of  $a$  and  $b$  to the Montgomery Domain, there can be savings in overall resources for large multiplications and exponentiations, which are prevalent in public key cryptosystems such as ECC.

[29] implements point multiplication using a variable number of processing elements (PE), where more PEs improve performance but require more resources. As ECC point multiplication is much more complex than any other operation in our design, we use the maximum allowable number of 8 PEs to minimize latency.

Our ECC point multiplier implementation is shown in Fig. 2. Key components of the implementation in [29] include the Montgomery Modular Multiplier (MMM), Modular Adder and Subtractor (MAS), scheduler, and memory control. Our contribution is to repackage the original ECC core, shown in dashed lines in Fig. 2, into an Advanced eXtensible Interface (AXI) memory-mapped slave device, which is easily configurable as a custom IP in a Xilinx block design. The initial point  $G$  (consisting of  $G_x$  and  $G_y$  components) and curve parameter  $a = -3$ , and precomputed  $R^2 \bmod p$  used in conversion to the Montgomery Domain, are adopted from [5, 29], respectively. Input  $G_x$ ,  $G_y$ , and  $k$  values, as well as output  $x$  and  $y$  values, are written and read by the SW API as 32-bit words at 8 memory-mapped locations, for a total size of 256 bits. Our custom IP also includes 1.2 KB of dedicated RAM, needed by (but not included in) [29], and used to store intermediate results of calculations.



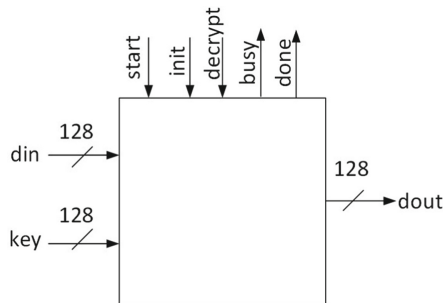
**Fig. 2.** ECC256 point multiplier, instantiated as a custom IP. The inputs consist of  $x$  and  $y$  coordinates of generator point  $G$ , and multiplicative scalar  $k$  (i.e., private key). This custom IP encloses design from [29], denoted by dashed lines.

**AES Accelerator.** We implement AES as a HW accelerator, using the design at [25], since it is publicly-available, license-free, and has an easily-tailorable external interface. This design is optimized primarily for throughput, and uses a

basic-iterative architecture, where one round executes in one clock cycle, requiring 10 clock cycles to encrypt or decrypt a 128-bit block of data. This implementation additionally precomputes round keys, which requires an additional 10 cycles as a start-up cost every time the key is changed (however, this cost is negligible for large messages).

Since we employ AES-128, only 128 bits of the 256-bit bi-nodal session key returned by the ECC256 module are required. According to NIST recommendations based on best-known analytic attacks, a 256-bit ECC-derived key is equivalent in strength to a 128-bit AES key [32].

The AES HW accelerator is depicted in Fig. 3. The AES design at [25] is directly incorporated as a custom IP in the Xilinx Block Design, where the IP design tool is used to automatically configure an AXI-capable wrapper compatible with the ARM processor.



**Fig. 3.** AES HW accelerator, incorporated from [25], where `din` and `key` are inputs, `dout` is the result, `init` signals a key initialization, `start` commences an encryption or decryption according to `decrypt`, and `done` asserts when result is complete.

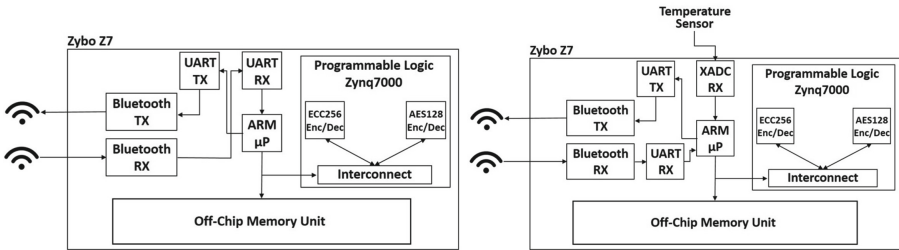
### 3.3 Secure Data Delivery Subsystem Configuration

The three separate SADDLE nodes (drone, sensor, host) each perform differently, and require specific HW to accomplish their respective purposes. From a SW standpoint, given the 8 GB of available space for user SW and bootloader data, and the 2.2 MB maximum required space for the fully equipped base SW, it is not practical to limit the SW resources of individual nodes. Thus, all nodes have the same SW, with only their initial enumerated variable state preset to an appropriate value, i.e., either a “drone state”, “sensor state”, or “host state” value.

**Drone and Host Node Hardware Configuration.** The SADDLE drone and host nodes require two peripheral modules: the Pmod OLED (for debugging) and the Pmod BT2 (for device-to-device communication). In order to interface with these peripheral modules, the system must also utilize the Zybo Z7’s on-board UART (to serialize and pack outgoing and incoming data, respectively) for

communication with both the Pmod BT2, as well as the onboard UART-to-USB connection. The Drone and Host node are each configured with cryptographic accelerators consisting of the ECC256 and AES IPs.

**Sensor Node Hardware Configuration.** The SADDLE sensor node requires three peripheral modules: the Pmod OLED (for debugging) the Pmod BT2 (for device-to-device communication), and the Pmod TMP3 (for obtaining sensor data). Similar to the drone and host nodes, in order to interface with these peripheral modules, the system must also utilize the Zybo Z7's onboard UART for communication with both the Pmod BT2, as well as the on-board UART-to-USB connection. The sensor node is also equipped with ECC256 and AES IP cryptographic accelerators. A simplified configuration of drone, sensor, and host nodes is depicted in Fig. 4. The detailed block design is shown in Fig. 5.

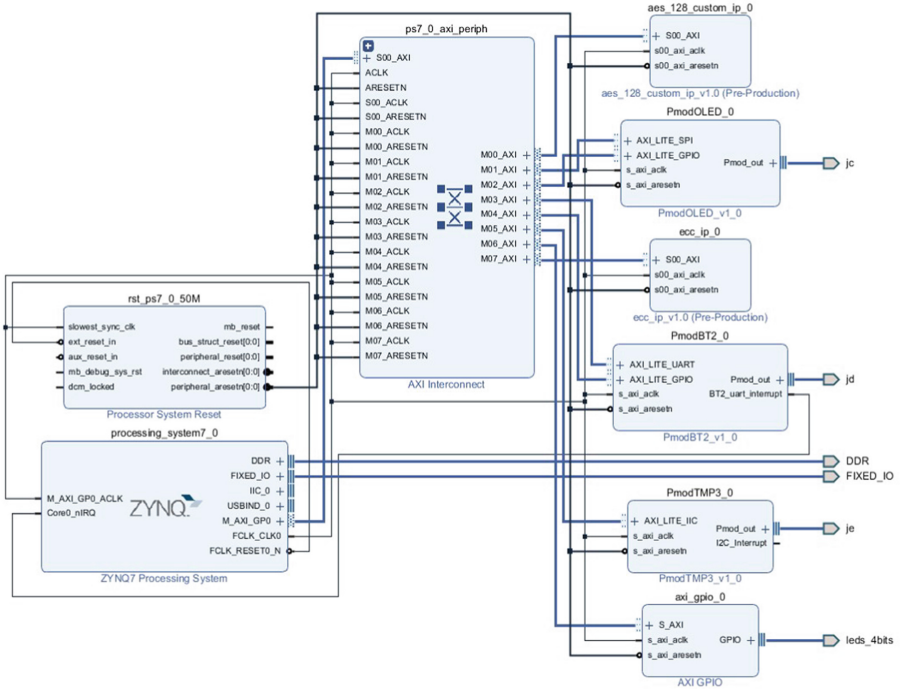


**Fig. 4.** Simplified configuration of Zybo Z7-10 SoCs, configured for drone or host nodes (left), or for sensor node (right).

**Software Model.** The SADDLE SW model is written for extensibility and modularity, such that specific operations are decomposed appropriately. This approach allows integration of additional resources and modules with ease. The main processing loop provides an excellent summary of the system. Upon program start, the system initializes required peripheral modules, onboard HW, and IPs. Afterwards, the system enters the main processing loop, in which the processor checks for flags which have been set in a given ISR (*processInterrupts()*) and then processes any pending communication messages from the current main message queue, updating the respective local Zybo data, as needed. We provide an excerpt of our main program in Fig. 6.

Our SW model relies upon message passing. To ensure synchronization of messages, we implement a SADDLE queue. As our lightweight system utilizes a baremetal processor (without a Real-time Operating Systems (RTOS) or Linux Kernel), the SADDLE system maintains its own queue format. As the system is capable of handling HW interrupts, we also want to ensure appropriate serialization and synchronization of individual messages and their respective orders. Therefore, we implement a *spin-lock*, which is utilized when pushing or popping from a given message queue.





**Fig. 5.** The SADDLE Vivado block design, showing integration of Processing System (PS) and Programmable Logic (PL). The ARM Cortex A9 is contained in Zynq7 PS at left, while cryptographic custom IP accelerators and Pmod peripherals are shown at right. The AXI Interconnect block, in center, establishes a memory-mapped matrix of AXI master drivers, which are accessed through memory-mapped locations by SW running on the ARM, and drive accelerator or peripheral slave devices.

```

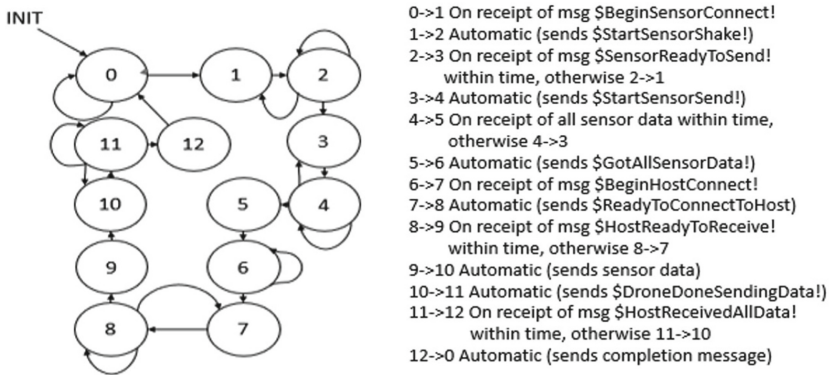
int main (int argc, char* argv[])
{
    OLED_DEBUG();
    initializeSystem();
    while (1)
    {
        OLED_DEBUG();
        processInterrupts();
        processMessages(&mainMessageQueue, &currentZyboData);
    }
    cleanup();
    return EXIT_SUCCESS;
}

```

**Fig. 6.** This figure represents the main SW function of the SADDLE system. Upon entry, the system initializes relevant HW and IPs. Next, the system processes any interrupts by checking flags which are set in Interrupt Service Routines (ISR), checks the main message queue for any pending messages, and updates respective Zybo data, such as current state, current encrypted data, or current AES key.

In a baremetal processor, performing a dynamic allocation of memory (such as that which occurs by using `malloc`) is not practical, as there is no operating system from which to request memory, i.e., users must manage their own memory. In order to solve this problem, the SADDLE message queue uses only stack-allocated memory, which is allocated at compile time. Furthermore, we include the spin-lock as a message queue internal, which is abstracted away from the user. By providing this simple interface, we greatly increase the extensibility of the system by ensuring we not only save resources, but also allow for additional programmers to integrate and develop within the system rapidly, if desired.

With the communication framework appropriately in place, the SADDLE system has a basis for essentially infinite state machine traversal. On this premise, we implement a “command-based” system state machine for each respective system node. Based on a given node’s respective type (drone, sensor, or host), which is set at compile time, a respective series of states will be traversed based on a series of received commands. In this implementation, a variety of handshakes take place to ensure nodes are appropriately synchronized and ready to send/receive data, as needed. In the event a message is, for some unknown reason, not fully sent or not fully received, the state machine periodically attempts re-synchronization until the system is able to connect; this will also persistently attempt to resend after a Bluetooth module’s potential disconnect, so the SADDLE system is quite robust. The drone node’s state transition diagram, as an example, is provided in Fig. 7.



**Fig. 7.** This figure represents the state machine for the drone node. The left portion represents the visual state transitions which are possible, including initialization (entry point), self loops, and single state backwards steps; the transitions and their triggers are explained verbosely at right.

### 3.4 Aerial Subsystem

Our SADDLE is implemented using the lightweight and low-cost DJI F450 ARF quad-copter drone, which has a maximum takeoff weight of 1.6 kg and a maximum diagonal span of 0.45 m. The drone is controlled by the mRo Pixhawk 2.4.6 flight controller, and powered by a 3S Lithium Polymer (LiPo) battery, capable of 5000 mAh at 11.1 V. While the Zybo SoCs used for sensor and host nodes are connected to, and powered by, attached PCs through a USB connection, the drone Zybo SoC instance must be autonomously powered. This is accomplished using a Poweradd Slim 2 portable USB charger, with 5000 mAh storage, and capable of sourcing 2.1 A at 5 V. As our bench testing of the drone Zybo instance showed a maximum current of 450 mA at 5 V, the Poweradd USB charger was determined to be sufficient to support mission requirements. The remote pilot communicates with the drone via the DX6i transmitter with AR6210 DSMX receiver. This employs DSMX wide band modulation with Frequency Hopping Spread Spectrum (FHSS) at 2.4 GHz.

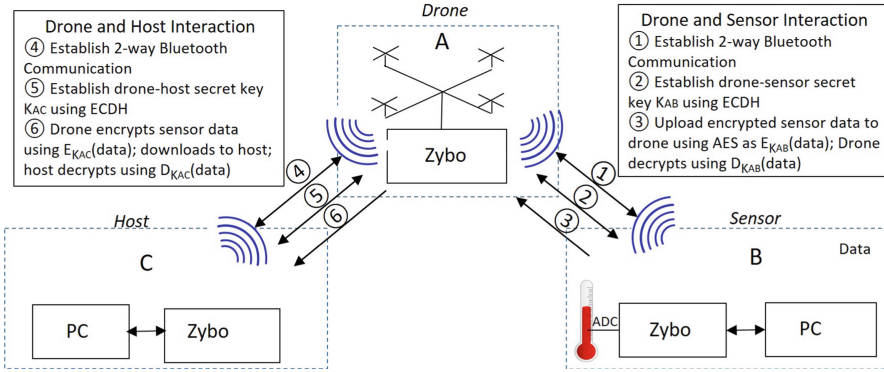
### 3.5 System Operation

System operation, and interaction between the drone, sensor, and host nodes, is shown in Fig. 8. A typical operational scenario begins with all nodes geographically separated. As the drone (A) approaches a sensor node (B), the two nodes' BT2 peripherals automatically establish a wireless link. At this point, the two nodes interact using ECDH to establish a bi-nodal session key  $K_{AB}$  unique to the two nodes. Once established, the sensor encrypts its temperature data using AES, and transmits encrypted data to the drone. The drone decrypts the data and places it in volatile storage. As the drone navigates away from the sensor, the drone-sensor BT2 connection is severed. When the drone approaches the host (C), a new two-way BT2 wireless link is automatically established. The drone and host interact using ECDH to generate a session key  $K_{AC}$  unique to these nodes. Next, the drone reencrypts sensor data using AES, and downlinks encrypted data to the host. Finally, the host decrypts sensor data using  $K_{AC}$ .

## 4 Results

### 4.1 Hardware Software Codesign of Secure Data Delivery Subsystem

The HW-SW codesign is assembled using Xilinx Vivado and accompanying SDK. Utilization statistics are shown in Table 1. The statistics show post-synthesis results for the ECC point multiplier and AES, and post implementation results for the completed design, since the cryptographic accelerators cannot be independently implemented (without special wrappers) due to high numbers of required I/O pins. All programmable logic devices run on a single clock frequency fixed at 50 MHz. In terms of FPGA LUTs, the area of the complete design exceeds the sum of the cryptographic accelerators by approximately 48%. This overhead



**Fig. 8.** Interaction between drone, sensor, and host nodes. Interaction between drone and sensor, and between drone and host, are explained at upper right, and upper left, respectively.

includes peripheral devices such as Pmod IPs and AXI interconnects. While fewer than 50% of total device LUT resources are consumed, our design uses 61% of slice resources. 61% is acceptable, but approaching an upper limit, since as the percentage of used slices increases, routing complexity increases exponentially, leading to long Place & Route (P&R) times.

Major performance statistics are also outlined in Table 1. It is evident that the public key component used for session key establishment accounts for the majority of system latency at 3.3 million clock cycles (or 65.6 ms at 50 MHz) per point multiplication; the secret key encryption or decryption (at least for a small amount of data) is insignificant. Additionally, two ECC256 point multiplications per node are required to generate a shared session key, according to ECDH. Therefore, total session key generation takes at least 131.2 ms.

The ECC256 accelerator, even with the maximum number of installed processing elements ( $PE = 8$ ), is heavily performance-constrained. For example, we operate our UARTs at a speed of 115,250 bps, whereas the throughput of our ECC256 is only 3,900 bps. The wireless Bluetooth link itself is capable of up to 3 Mbps. In contrast, the AES accelerator is communications-constrained, since total block latency of 10 clock cycles is less than the sum of clock cycles required to write to and read from the accelerator (approximately 16 clock cycles). Since we have used the maximum number of PEs for the ECC core at [29], future improvements could include use of a faster (but larger) ECC core, and smaller (but possibly slower) AES core.

## 4.2 Implementation of Aerial Subsystem

The F450 ARF quad-copter drone was assembled in accordance with vendor specifications. The drone instance of the Zybo Z7-10 SoC, together with USB power supply, were mated to the ventral surface of the drone at center of

**Table 1.** Utilization results for post-synthesis (ECC256 and AES accelerators), or post-implementation (complete design) for the Zybo Z7-10 HW-SW co-designed implementation in Zynq-7000 programmable logic. RAMB18 and RAMB36 refer to 18 KB or 36 KB block RAM instances, respectively. Cycles and latency refer to clock cycles and time, respectively, necessary to complete one block operation of the size indicated in the “bits” field.

Entity	LUT	Slices	Registers	RAMB18	RAMB36	Cycles	Latency	Bits
ECC256	2715	–	3808	1	0	3,280,175	65.6 ms	256
AES	2314	–	402	0	2	10	200 ns	128
Complete design	7445	2686	7251	1	2	–	–	–
Available	17600	4400	35200	120	60	–	–	–

gravity (CG), and secured using packing ties and masking tape. The weight of individual components, and total weight, are shown in Table 2. The F450 ARF specifications note a take off weight of 800–1,600 g. Thus our actual take off weight of 1,152 g is within specifications; however, flight performance can be marginal in these conditions, depending on environmental factors.

**Table 2.** Weights of individual components in the takeoff configuration of the F450 ARF quad-copter. “Motors” includes 4 total 50 g motors. “Battery” refers to the 3S LiPo 11.1 V battery used to power the drone, while “USB Battery” refers to the Poweradd Slim 2 USB stick used to power the Zybo.

Item	Frame	Motors	Battery	Pixhawk	Zybo	USB Battery	Total
Weight (g)	282	200	376	38	136	120	1,152

### 4.3 Demonstration

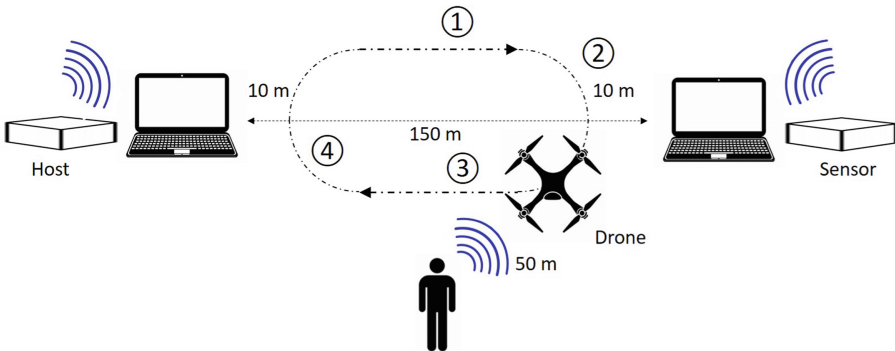
The fully-configured SADDLE was demonstrated in actual flight conditions. The demonstration occurred in a drone test field, with sensor and host nodes separated by 150 m, depicted in Fig. 9. The drone, controlled by a human pilot via remote control, lifted off at center field (approximately 75 m from either node), and cruised at a constant altitude of 5 m and velocity of 1 m/s toward its rendezvous with the sensor node. The drone and sensor acquired Bluetooth lock at 10 m, which can be observed by changing of the light pattern on the Pmod BT2 peripheral. The uplink of sensor information, observed via USB UART connection on the sensor PC, occurred within 10 s. The pilot then directed the drone on a path towards the host node. The sensor-drone Bluetooth link was observed to break at a distance of 40 m. Next, the drone-host Bluetooth link was established at a distance of 10 m. Session key establishment, downlink, and decryption of temperature data at the remote sensor was observed to occur within 10 s.

Since both the RN42 Bluetooth radios and DSMX receiver/transmitter system both use the Industrial, Scientific and Medical (ISM) bandwidth allocation

between 2.4 and 2.5 GHz, there is the possibility of Radio Frequency Interference (RFI) between flight control and mission components. However, both components use spread spectrum modulation to employ Signal-to-Noise Ratio (SNR) gains (since the ISM band is known to be crowded), and RFI was not observed. Although there is also the possibility of Electro-Magnetic Interference (EMI) caused by high-output electric motors, EMI was not observed. Future research could include establishing conditions at which these phenomena might be observed, in order to determine accurate RFI and EMI resistance.

#### 4.4 Comparison with Previous Work

There is sparse basis for direct comparison with related works, since design goals, choice of algorithms, implementation platforms, and evaluation methods are divergent. In [10], The linear homomorphic encryption (LinHAE) employed in a F550 DJI Hexarotor drone appears efficient, but is designed to protect only commands from ground to flight controller, whereas our design ensures secure transport of data between multiple sensor nodes. Additionally, [11,12] have similar design goals, namely, secure transfer of sensor information via drone, and their choice of public and secret key algorithms, Boyko-Peinado-Venkatesan (BPV) ECDH, and ChaCha20, (respectively) have lower latencies than our ECDH and AES, if implemented in the same architecture. However, research in [11,12] is statically benchmarked; it is not evaluated in a realistic flight environment. Finally, our choice of HW-SW codesign fundamentally differs from [10–12], in that we employ FPGA accelerators for ECDH and AES processes, while [10–12] are pure SW implementation inside ARM Cortex microprocessors, which makes direct comparison of performance and required resources difficult.



**Fig. 9.** Schematic of flight demonstration. The drone takes off at event (1), and completes uplink of secure sensitive data from the sensor node at event (2). The drone breaks communication link with the sensor at event (3), and is piloted to the host node. At event (4), the drone establishes communications link with the host node, and downlinks sensitive data to the host.

## 5 Conclusions and Future Work

Our Secure Aerial Data Delivery with Lightweight Encryption (SADDLE) was successfully demonstrated and met all test objectives. The HW-SW codesign facilitated the insertion of public and secret key cryptographic accelerators in a baremetal, modular, and extensible embedded encryption and decryption system with a reliable Bluetooth communication infrastructure. Approximately 60% of slice resources were consumed from available programmable logic resources of the Digilent Zybo Z7-10 System on Chip (SoC). This enabled successful bi-nodal session key establishment, using the Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol, and subsequent encryption and decryption of sensitive data using Advanced Encryption Standard (AES-128).

Three nodes, drone, sensor, and host, were established using the HW-SW codesign in the Zybo SoC. The autonomously powered and operated drone-instance of Zybo SoC was successfully mated to a F450 ARF quad-copter. The system was successfully demonstrated in a flight test, where notionally sensitive data was securely uplinked from a sensor node, transported to destination by the drone node, and downlinked and decrypted by a remote host node.

Future work will include improved design of public and secret key cryptographic accelerators, in order to balance resource and performance requirements of the two accelerators, i.e., to make the public key accelerator faster (and likely larger), while making the secret key accelerator smaller. Additionally, researchers will experiment with more powerful and capable drones, such as the Tarot X8 octo-copter (with up to 7 kg of payload capacity), and integration of higher-performance secure sensor applications, such as streaming video.

**Acknowledgment.** This work was funded by 4-VA, a collaborative partnership for advancing the Commonwealth of Virginia (<https://4-va.org>) – Spring 2019.

## References

1. Wang, C., Ma, F., Yan, J., De, D., Das, S.: Efficient aerial data collection with UAV in large-scale wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **11**, 286080 (2015)
2. Kanistras, K., Martins, G., Rutherford, M.J., Valavanis, K.: A survey of unmanned aerial vehicles (UAVs) for traffic monitoring, pp. 221–234 (2013)
3. Kochetkova, L.: Pipeline monitoring with unmanned aerial vehicles. *J. Phys. Conf. Ser.* **1015**, 042021 (2018)
4. Schneier, B.: Regulating the Internet of Things. In: RSA Conference USA 2017, San Francisco, CA, 14 February 2017
5. Federal Information Processing Standards Publication 186-4: Digital Signature Standard (DSS), National Institute of Standards Technology (NIST), July 2013
6. Advanced Encryption Standard: FIPS PUB 197, 26 November 2001
7. Huang, H., Wu, J.: A probabilistic clustering algorithm in wireless sensor networks. In: VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, September 2005, vol. 3, pp. 1796–1798 (2005)



8. Polo, J., Hornero, G., Duijneveld, C., Garcia, A., Casas, O.: Design of a low-cost wireless sensor network with UAV mobile node for agricultural applications. *Comput. Electron. Agric.* **119**, 19–32 (2015)
9. Potter, B., Valentino, G., Yates, L., Benzing, T., Salman, A.: Environmental monitoring using a drone-enabled wireless sensor network. In: 2019 Systems and Information Engineering Design Symposium (SIEDS), pp. 1–6, April 2019
10. Cheon, J.H., et al.: Toward a secure drone system: flying with real-time homomorphic authenticated encryption. *IEEE Access* **6**, 24325–24339 (2018)
11. Ozmen, M.O., Yavuz, A.A.: Dronecrypt – an efficient cryptographic framework for small aerial drones. In: IEEE Military Communications Conference (MILCOM 2018), Los Angeles, CA, pp. 1–6 (2018)
12. Ozmen, M.O., Behnia, R., Yavuz, A.A.: IoD-Crypt: a lightweight cryptographic framework for Internet of Drones. arXiv [arXiv:1904.06829](https://arxiv.org/abs/1904.06829) (2019)
13. Rescorla, E.: Diffie-Hellman key agreement method. In: RFC 2631, June 1999
14. Miller, V.: Use of elliptic curves in cryptography. In: Advances in Cryptology, CRYPTO 1985, pp. 417–426 (1986)
15. Koblitz, N.: Elliptic curve cryptosystems. *Math. Comput.* **48**, 203–209 (1987)
16. Zhou, L., Su, C., Hu, Z., Lee, S., Seo, H.: Lightweight implementations of NIST P-256 and SM2 ECC on 8-bit resource-constraint embedded device. *ACM Trans. Embed. Comput. Syst.* **18**(3), 23:1–23:13 (2019)
17. Al-Adhami, A., Ambroze, M., Stenget, I., Tomlinson, M.: A 256 bit implementation of ECC-RFID based system using Shamir secret sharing scheme and Keccak hash function. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 165–171, July 2017
18. Marzouqi, H., Al-Qutayri, M., Salah, K.: An FPGA implementation of NIST 256 Prime Field ECC processor. In: 2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), pp. 493–496, December 2013
19. Rahman, M.S., Hossain, M.S., Rahat, E.H., Dipta, D.R., Faruque, H.M.R., Fattah, F.K.: Efficient hardware implementation of 256-bit ECC processor over prime field. In: 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 1–6, February 2019
20. Sahingoz, O.K.: Large scale wireless sensor networks with multi-level dynamic key management scheme. *J. Syst. Architect.* **59**(9), 801–807 (2013)
21. Lu, H., Li, J., Guizani, M.: Secure and efficient data transmission for cluster-based wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **25**(3), 750–761 (2014)
22. Nadir, I., Zegeye, W.K., Moazzami, F., Astatke, Y.: Establishing symmetric pairwise-keys using public-key cryptography in wireless sensor networks (WSN). In: 2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), pp. 1–6, October 2016
23. Malathy, S., Geetha, J., Suresh, A., Priya, S.: Implementing elliptic curve cryptography with ACO-based algorithm in clustered WSN for border surveillance. In: 2018 4th International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), pp. 1–5, February 2018
24. Houssain, H., Badra, M., Al-Somani, T.F.: Hardware implementations of elliptic curve cryptography in wireless sensor networks. In: 2011 International Conference for Internet Technology and Secured Transactions, pp. 1–6, December 2011
25. CERG: Source code for AES, June 2016. [https://cryptography.gmu.edu/athena/index.php?id=CAESAR\\_source\\_codes](https://cryptography.gmu.edu/athena/index.php?id=CAESAR_source_codes)
26. OpenCores (Tiny AES). [https://opencores.org/projects/tiny\\_aes](https://opencores.org/projects/tiny_aes)
27. OpenCores (AES Core). [https://opencores.org/projects/aes\\_core](https://opencores.org/projects/aes_core)



28. Visengi Hardware Software Engineering (AES). <http://www.visengi.com/products/aes>
29. Salman, A., Ferozpuri, A., Homsirikamol, E., Yalla, P., Kaps, J., Gaj, K.: A scalable ECC processor implementation for high-speed and lightweight with side-channel countermeasures. In: 2017 International Conference on ReConFigurable Computing and FPGAs (ReConFig), Cancun, pp. 1–8 (2017)
30. Salman, A., Diehl, W., Kaps, J.: A light-weight hardware/software co-design for pairing-based cryptography with low power and energy consumption. In: 2017 International Conference on Field Programmable Technology (ICFPT), Melbourne, VIC, pp. 235–238 (2017)
31. Montgomery, P.: Modular multiplication without trial division. *Math. Comput.* **44**, 519–521 (1985)
32. NIST: Recommendation for Key Management, Special Publication 800-57 Part 1 Rev. 4 (2016). <https://www.keylength.com/en/4/>



# Malware Analysis with Machine Learning for Evaluating the Integrity of Mission Critical Devices

Robert Heras<sup>(✉)</sup> and Alexander Perez-Pons

Florida International University, Miami, FL 33165, USA

rhera003@fiu.edu

<http://www.myweb.fiu.edu/rhera003>

**Abstract.** The rapid evolution of technology in our society has brought great advantages, but at the same time it has increased cybersecurity threats. At the forefront of these threats is the proliferation of malware from traditional computing platforms to the rapidly expanding Internet-of-things. Our research focuses on the development of a malware detection system that strives for early detection as a means of mitigating the effects of the malware's execution. The proposed scheme consists of a dual-stage detector providing malware detection for compromised devices in order to mitigate the devices malicious behavior. Furthermore, the framework analyzes task structure features as well as the system calls and memory access patterns made by a process to determine its validity and integrity. The proposed scheme uses all three approaches applying an ensemble technique to detect malware. In our work we evaluate these three malware detection strategies to determine their effectiveness and performance.

**Keywords:** Task structure · System calls · Memory access patterns · Dual-stage classification

## 1 Introduction

Malware is often attributed as the cause for loss of data, data integrity, and financial damage for a company or institution. Due to the nature of malware and their individual uniqueness, the determination of a compromise or infection can be daunting and difficult as well as challenging to determine exactly how much damage will transpire because of it. There have been many advances in the industry including the rise of the prominent internet-of-things (IoT). IoT includes sensors, actuators, vehicles, home appliances and just about any device that are embedded with electronics in order to interconnect and exchange data. The increasing popularity of these devices makes them inevitable targets for malicious corruption.

With the emergence and rising popularity of IoT and ubiquitous embedded systems it is increasingly common for them to be infected. Many IoT devices are intended to be deployed to serve their function with little to no maintenance and

This is a U.S. government work and not under copyright protection in the U.S.;

foreign copyright protection may apply 2020

K. Arai et al. (Eds.): SAI 2020, AISC 1230, pp. 224–243, 2020.

[https://doi.org/10.1007/978-3-030-52243-8\\_18](https://doi.org/10.1007/978-3-030-52243-8_18)

overhead required. Once deployed, many of these devices do not have enough battery or processing power to perform complex operations much less use standard signature or heuristic methods to detect whether or not they have been infected and compromised as it is too inefficient and costly to monitor at the device.

We propose a framework which uses an embedded low overhead agent in each device to extract key feature data as the device is executing its tasks. This approach allows our machine learning algorithms to make a determination as to whether a device has been maliciously compromised by using behavioral analysis on extracted features.

The embedded agents connect to our remote framework in order to transfer the harvested information and our models analyze this information to determine the stability and integrity of each device. The analysis is done remotely and is specific to each device so it is persistent and is dynamic enough that we can detect various types of malware as they are executing even if the malware has attempted to delay its execution [16]. These IoT devices typically maintain an operating system, such as Android, and our system augments the OS with an agent that collects execution features of running tasks in order to transmit them for analysis. Other operating systems include Windows, Mac, and specific agents per operating system would have to be generated. These agents have an inherent impact delay in application execution as once the agent runs and at a specific rate, it will impact the native application on the device. This process is conducted on an ongoing basis, with the frequency and amount of information transmitted dependent on situational conditions. Once the malicious activity is detected the device can be halted or the task suspended from functioning within the network in order to prevent further execution and the possible spread of the malware. Furthermore, our framework is intended to work with as little overhead and interruption on the end devices.

Novel research into new ways of detecting malicious activity is needed and it is needed for a wide range of devices and device types. Researchers have looked into the use of machine learning models in order to create anomaly detectors and malware classifiers [3]. Specifically the use of feature extraction, system call extraction, and the observance of memory access patterns in order to train models that use algorithms such as Support Vector Machines, Random Forests, Logistic Regression, and Naive Bayes.

From a Forensics standpoint, infected devices pose a myriad of difficulties. Devices can have their data altered, erased, or hidden [19]. Forensics investigators must now also account for the rise of the prominent internet-of-things (IoT) such as smart fridges and smart home air conditioning (AC) units.

To combat malware in both home and enterprise environments, we will attempt to use machine learning via anomaly detection and decision tree classifier algorithms to determine if a process is benign or malicious. The framework models are constructed and maintained from extracted task structure features, system calls, and memory access patterns harvested from benign and malicious processes as they are executing to provide an understanding of typical executions.

The framework will analyze this information to determine the stability and integrity of each device. Once malicious activity is detected the device can be halted or the task suspended from functioning in order to prevent further execution and the possible spread of the malware. Furthermore, our framework is intended to work with as little overhead and interruption on the end devices.

## 2 Methodology

Our framework functions by analyzing extracted features using a sliding window approach for maximized accuracy. The values of our chosen task structure features, system calls, and memory access patterns are harvested, combined into individual vectors for the current sliding window, and then individually analyzed by our framework. While a malware may obfuscate what system calls it makes, it may not also attempt to cloud what memory access patterns occur or forge its task structure feature values. Using statistical probably built on the results of all three individual models, our framework can accurately determine a processes nature with high true positive rates and low false positives. This ensemble approach offers the maximum coverage of a system and its processes and increases the likelihood of our framework detecting malicious activity.

Our framework uses machine learning to analyze extracted features for the real time detection of malware in deployed devices. We have seen the success of task structure based feature extraction within a Windows [20] and Linux environment [19]. It has been shown that cloud-based services can provide security to mobile phone devices with limited resources [8] and that behavioral based malware detection systems can be used to detect anomalous activity [21]. Hybrid approaches for anomaly detection with Hidden Markov Models and naive Bayes models have also been explored and shown to be successful [1, 8, 16, 22].

Our approach differs from other tested approaches as we intend our framework to use an ensemble of features, analyzed via a sliding window in order to be an accurate and dynamic approach, focused on deployed devices. By extracting features from a processes task structure and by observing the system calls [9, 15] and the memory access pattern [6] made, our framework can analyze a devices behavior for anomalies. By observing all three features of a process, our framework offers dynamic detection of malware. Through the use of sliding windows of vectors created from these extracted features, we have increased accuracy for our models. Furthermore, if an anomaly is detected then a more in-depth analysis occurs. Thus far we have performed these extractions on Linux Kernel 4.10, Windows, and intend to further test on Android and various IoT devices. The selection of the Linux operating system as our target is based on the widespread use of Linux dialects for many IoT devices.

Ubiquitous embedded systems or as they're more commonly known, IoT devices, are devices with the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. They have sensors and tend to collect data or act on received information. Such devices are perfect targets for malicious activity. Due to the nature of our framework, a

potential application of the basic technology works well for deployed IoT devices. Most IoT devices run command-line interface (CLI) applications and therefore this study aims to build a malware detector for these types of application, as GUI based applications that exist for user interactions are not the norm for IoT devices. IoT devices will also have a minimal number of applications, if not only one. We can focus on verifying one application or a minimal number of applications in an IoT device being monitored. Furthermore, as we know what application(s) are running on an IoT device type, we can develop the anomaly detection models and use them to detect an attack (prior to the device being deployed) since we know the processes running on that specific IoT device.

Through the frameworks use of drivers, the feature data is first observed and extracted. Then the extracted data is formatted for the model and the classifier performs its classification.

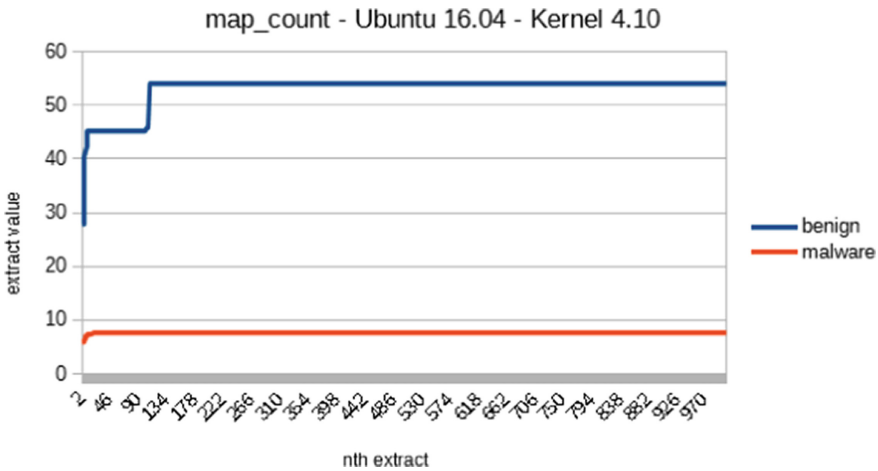
## 2.1 Task Structure Feature Extraction

For the purposes of malware detection with machine learning via feature extraction, a feature is a system property which should deviate greatly when observed from a benign process versus a malicious process. Each process is manifested in the OS through a data structure called task structure that maintains on an ongoing bases all information associated with the process, like open files, network activity, memory region, time spent in user and kernel space and many more. All of the information that is required to move a process from a running state to a waiting state and at a later time to the running state without affecting the process execution. By observing and cataloging the values of these task structure features, we can map the normal operating procedure the system has and observe any deviations as possibly malicious. The research performed in [17] determined 11 out of 118 feature which can discriminate between a benign and malicious process with high accuracy. Within a Linux operating system, a feature logger was used which periodically dumped 118 fields of task structure data every millisecond for 15 s. The results of the experiments performed show that a classification decision can be made within the first 100 instances as the process's execution enters a steady state afterwards. The classification power of using task structures for malware analysis is shown as the results were 93% accuracy with minimal overhead, from 50 to 70  $\mu$ s after every millisecond [17].

For our framework, we opted to extract the eleven identified features every millisecond over the course of fifteen seconds. The eleven features selected showed the most discrepancy between benign and malicious activity out of the identified 118 features. Extractions at a rate less than one millisecond didn't contain enough discrepancy when compared to extractions performed at a rate of one millisecond but caused more overhead due to more extractions over the given 15 s window. Extractions at a rate over one millisecond began to lose their discrepancy and thus resulted in less accuracy. Furthermore, we compared the extracted features of command line interface applications against graphical user interface applications. Due to the event based nature of graphical user interface applications, the models lost accuracy when they were trained on mostly command

line interface applications and vice versa. To ensure consistent accuracy, a broad spectrum of applications must be used to train the models.

Of the extracted features, one to note is `map_count` which is an integer value depicting the number of memory regions. Figure 1 clearly denoted the disparity between the `map_count` value of a benign and malicious program execution. The feature `hiwater_rss` is the high-water of resident set size usage which is used to show how much memory is allocated to that process and is in random access memory. The `utime` feature depicts how much time a process spends in user space where as the `stime` feature represents how much time a process spends in kernel space. Each feature in a processes task structure has a purpose and value which can denote its maliciousness. In order to achieve our observations depicted in Fig. 1, we observed the task structure of various benign processes and various malicious processes and harvested all eleven features. For each benign and malicious process, we graphed the average `map_count` value at each interval of extraction resulting in Fig. 1; denoting that the `map_count` value for a benign process deviates from the standard `map_count` value for a malicious process. It is worth noting that our other ten extracted features behaved similarly and support the research performed in [17].



**Fig. 1.** Time series statistics of the `map_count` task structure feature of a Benign vs. Malicious process

We determined 11 out of 118 task structure feature which can discriminate between a benign and malicious process with high accuracy [17]. The features we have identified are extracted using a device driver or through introspection [17, 19] and used to train and test our model. These features are then used to audit the running system and perform monitoring on system performance [26]. Table 1 depicts twelve possible features that can be extracted of the identified 118 in [17]. `Utime` for instance is the amount of time that a process spends in

userspace. Conversely, stime is the amount of time a process spend in kernelspace. Map\_count is an integer value depicting the number of memory regions used by the process. Each feature serves a specific purpose within the task structure of a process and the features value can denote whether that process is benign or malicious. The observations of such features offers a basis for the behavioral analysis of a deployed device.

**Table 1.** Twelve possible task structure features as identified by [17]

exec_vm	fscount	hiwater_rss	hiwater_vm
map_count	min_ftt	nlvcsm	nr_ptes
nvcsm	stime	total_vm	utime

In [26], a Feature Extraction and Selection Tool (FEST) was developed which used a feature-based machine learning approach for malware detection and tested on Android platforms. The researchers developed a means to look at the permissions, API actions, IP, and URLs within an android apk and using statistical analysis, collect the features they wish to use. The data-set contained 7972 apps with 50% being malware and the rest being benign. From their research, they discovered that the performance of the Naive Bayes (NB) algorithm is the worst of all algorithms because of its strictly independence limitation and continued further testing with the K-Nearest Neighbor (KNN) and the Support Vector Machines (SVM) algorithms which displayed increased accuracy.

Researchers have also compared the performance of Naive Bayes (NB), Random Forests (RF), and Support Vector Machines(SVM) and determined that the Random Forest algorithm had better classification of malware [12]. This was tested with 10-fold cross validation over a sample set of 25 benign and 84 malicious samples. The detection rate for their approach was calculated using five metrics, True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, and Accuracy. For static analysis, the Random Forest classifiers had an accuracy of 69.72% and was closely followed by Naive Bayes classifiers with an accuracy of 68.23%. With dynamic analysis, the Random Forest classifiers provided 63.3% accuracy and while the closest other classifier was the Support Vector Machines (SVM) with 60.55% accuracy. Overall, SVM was showed to have a better accuracy than Naive Bayes and Random Forest had the most accuracy among all the classifiers tested.

Evasion of feature extraction based detection might seem possible, however can be quite difficult. Author in [17] found that the accuracy of their model is unacceptable once eight features are forged, however each parameter depends on a particular configuration of the system - cache, RAM, paging, etc. The values can change from one host to another so a malicious actor must first estimate the values for a particular system and to do so would require the execution of a malware which hooks into sample fields for benign processes. In Linux, this is

only allowed to super user processes and as such, a malicious process would be unable to easily evade a feature extraction based malware analysis.

## 2.2 System Call Extraction

Instead of using feature extraction for task structure, researchers have focused on analyzing system calls as an alternative. System calls allow user level applications to interact with the kernel and benign processes interact with the kernel differently than malicious ones. By observing the system calls a process makes, we can determine if it is operating within normal functionality or if it acting in a malicious manner. Such research as in [5] compares the accuracy of n-gram analysis to that of bag-of-n-grams as well as ordered and unordered 2-g representations of the system call trace.

The conclusion of the [5] showed that the best extraction techniques used a minimum system call trace length of 1,000 ordered 3-g chosen using recursive feature elimination (RFE). Recursive feature elimination (RFE) is a feature selection method that fits a model and removes the weakest features until the specified number of features are reached. The algorithms with the highest accuracy were Logistic Regression (LR) and Support Vector Machine (SVM) which outperformed signature-based detectors. For classification, the Random Forest (RF), K-Nearest Neighbor (KNN), and Logistic Regression (LR) algorithms provided the highest accuracy, outperforming Naive Bayes (NB) and Nearest Centroid Classifier (NCC).

Along side task structure features, we observe the system calls executed by a process. System call extraction has been explored on various platforms such as Hadoop [14], QEMU [2], Android [13,22], and within Linux using a Loadable Kernel Module (LKM) [18] to protect against kernel level rootkits [7]. Our framework uses a form of introspection, similar to the extraction of task structure features, in order to extract the system calls made by a process. After testing, we have opted to format our extracted system calls into 3-g chunks instead of using an approach like bag-o-words for a higher accuracy within our sliding window [4,5,10].

The program we use to collect system calls is strace which is a diagnostic and debugging utility for Linux. It is used to monitor and tamper with interactions between processes and the Linux kernel. The strace output is a list of system calls made by a running process for a given duration. Each line in the trace contains the system call name, followed by its arguments in parentheses and its return value. For example “open(“/dev/null”, O\_RDONLY) = 3” is one of the outputs of the open system call when I ran the command “cat/dev/null”. For our purposes, we need the system call made and not the parameters or the return value. This is list of calls is then formatted into 3-g.

These overlapping system calls provide a depiction of the systems running state and by observing them in order, we can determine behavioral patterns.

Within [4], more exploration was done including Naive Bayes, SVMs, decision trees, and neural networks. System calls extracted and compiled into 3-g with



a trace length of 1500 calls and compared against a LR, RF, and SVM classifier. According to the results, the worst performing classifiers were Naive Bayes and nearest centroid conversely nearest neighbor and random forest classifiers providing nearly identical performance. The LR classifier performed similarly to the nearest neighbor and the random forest classifier, however the LR performance degraded significantly when the feature reduction techniques singular value decomposition (SVD) and linear discriminant analysis (LDA) were used. Classifier accuracy was shown to improve when  $n$  was increased, in particular there was a large increase in performance when moving from 1-g to 2-g indicating that the frequency information about system calls alone is insufficient for classification. At a trace length of 1500, the average execution time of the studied processes was 205 ms. This result indicates that high classification accuracy was achievable during the initial execution of the malware samples. As noted by the author, for a production deployment of a classification system, the LR classifier is preferred to the random forest and nearest neighbor.

Research into system calls is an ever expanding niche. Author in [5] tested bag-of-2-g representations of system call traces with the use of sliding windows. The ordered 2-g representation considers the local ordering of the calls within the sliding window, whereas the unordered 2-g representation ignores the local ordering. For  $n$  size 2, 3, and 4 the LR and SVM detectors offered significantly improved detection performance while LR and SVM detectors peaked around 1000 system calls for  $n$  sizes of three and four indicating that the studied malware sampled were detected during the early stages of their execution. The research further shows the results of testing a LR detector trained with 10-fold cross validation and  $n$ -gram sizes of three, containing 3,500 ordered system calls.

Malware analysis on Windows and Linux has been explored however in [14] the Hadoop platform was tested. Using a system calls analysis method with MapReduce, the Hadoop platform is used to analyze the system calls on a server rather than on the clients side. The author also proposes a more universal and persistent model to correlate system calls among modules. Some limitations with this research is if a persistent malware's behavior is completed without the use of system calls then the approach fails to detect the malware. Another limitation is that the cost of data transmission required to have the server analyze the collected system calls has not been measured. The research demonstrated the ability to improve upon previous research which resulted in a 28% increase in detection rate.

Within [2], we see the use of QEMU-emulated sandbox's virtual hard drive where the program Procmon was responsible for collecting system call information for all processes. Once collected, the system calls were analyzed and compared two at a time in a sliding window format. Specifically, the research used Jaccard distance due to good results and clear semantics. However, some limitations encountered was with the restrictions of a virtualized sandbox. Because some of the malware samples were unable to connect to the internet, some of their behavior may have gone unseen and at times malware may even terminate early resulting in insufficient traces which could lead to poorer results.

In [18], the researchers develop a lightweight monitoring infrastructure which extracts runtime information from the Linux kernel for the purpose of enhancing system reliability. The loadable kernel module titled KOMI observes the guest OS and intercepts system calls' to collect information from the kernel and suggests strategies to audit and protect the kernel with minimal CPU overhead. The researchers found the monitoring service easy to customize with minimal penalty to performance allowing for easy integration into existing embedded systems.

In [25], a Loadable Kernel Module (LKM) is proposed as part of a secure auditing system in which system calls are intercepted in order to suggest strategies to audit and protect the kernel. Within [7], a system is proposed with the intention of protecting the OS kernel from rootkits which hide their respective running processes and threads. This is done by generating signatures for the kernel's data structure and building a profile using this observed data. Twenty applications were tested and 221 fields were observed, 32 of which were never accessed during the execution of the profiled applications. The research encourages a systematic way of determining which features should be used when determining a data structure for research which extends to machine learning based applications.

With the expansion of embedded systems, smartphones are at risk as well. In [13], dynamic analysis was performed on an android device using system call extraction. System calls for twenty normal applications and forty malicious applications were recorded for thirty seconds and then sixty seconds while the total amount of times in which the system calls were made was observed and recorded. For example, the `epoll_wait` operation was called 2613 times by a benign application conversely to 4703 times from a malicious application. Similarly, `clock_gettime` was called 3796 times by a benign application and 17,251 times by a malicious one. Of note, malicious applications also request high-level permission from users in order to compromise the system by exploiting the granted permission. The research shows that observed system calls made by malicious applications can be used to create a signature to detect malware and although this research does not include machine learning components, it proved that the analyzing of system calls can aid in determining whether a system has been exploited.

In a similar research, [22], a Hidden Markov Model was used with system call extraction and key press patterns in order to attempt malware detection on a smartphone; which showed promise.

### 2.3 Memory Access Pattern

While feature and system call extractions have been shown to accurately detect whether a process is benign or malicious, other researchers have begun to explore the same functionality using memory access patterns. Memory access patterns are the patterns in which a system reads and writes data to secondary storage and random access memory. Our framework focuses specifically on virtual memory access rather than physical memory access. By observing what virtual memory access is done by a process and analyzing what portions of virtual memory are

interacted with, we can determine if a process has been tampered with. The collected data is compiled, similar to the 3-g approach of our system calls, and then passed to our first stage detector.

An epoch is the date and time relative to which a computer's clock and timestamp values are determined. Virtual memory is divided into memory regions based on the systems epoch which are used to build the systems memory model according to the type and frequency of access within these memory regions. Operating systems use pages which are the smallest fixed-length contiguous blocks of virtual memory possible. These pages are allocated together into a page frame which is the smallest fixed-length block of physical memory to which the pages are mapped to [23].

In this case, memory addresses are observed for changes that deviate from the norm in order to classify the process invoking the memory access. In [23], a novel framework is developed which includes techniques for collecting and summarizing memory access patterns. The framework also functions on a two-level classification architecture and observes kernel rootkits vs user-level malware and contains epoch-based monitoring. Thus far, ten rootkits have been observed using QEMU with the machine learning algorithms having been trained on four of those rootkits and asked to detect six rootkits it has never seen before. The two algorithms tested were Random Forest and Logistic Regression and both showed great accuracy and demonstrated the framework could detect new malware.

For our purposes, we use a program called Perf to observe memory access. The output contains references such as "LFB hit" or "L1 or L2 hit" or "Local RAM hit". These "hits" are memory access events that occur for both read and write events. Observing their order and occurrence gives us a standard for normal behavior when observing benign processes. We also observe whether these memory access patterns occur in userspace vs. kernel space. Each type of "hit" is given a correlating numerical value ("LFB hit" = 1, "L1 or L2 hit" = 2, etc) and this formatted output is used to train our models. Because of the limited number of occurrence types that exist ("LFB hit", "L1 or L2 hit", etc.), it is viable to observe these for behavioral analysis.

## 2.4 Data Extraction Process

For task structure feature extraction, we used a loadable kernel module which hooked into the kernel and intercepted task structure features when a process was executing. These features were stored in a buffer as the process was observed in its running state and then the buffer was saved to a file. The resulting feature values were stored as a comma-delimited file and compiled together to train our task structure models. Because each process that is observed creates its own output file, there is a resulting abundance of data files. As part of our framework we also developed a program to compile all the extracted task structure feature files into one file which could be used to train our machine learning models.

In order to extract the system calls performed by a process we used a program called strace. Strace is a diagnostic and debugging userspace utility for Linux which is used to monitor and tamper with interactions between processes and

the kernel via system calls. Strace is a intercepts and records the system calls made by a process and the signals which are received as a result. The name of the system call, its arguments, and its return value are all printed to data files. The extracted system calls must be parsed and our framework does this by iterating through each line in the data files and extracting each call.

The memory access patterns of the benign cli application and malicious executable are obtained using an application called Perf. This tool is a performance analyzing utility within Linux which provides a number of sub-commands that allow statistical profiling of the entire system. The commands used in conjunction with this tool recorded both read and write memory access events and outputted them to text files for further processing/formatting. These memory accesses were captured with their parameters and were outputted one per line. Table 2 depicts three lines of extracted memory access patterns though we have only included seven of the thirteen available data columns that are outputted by Perf. The symbol column describes if the memory hit is occurring in userspace as depicted by “[.]” or in kernelspace as depicted by “[k]”. The object column can inform us if a memory access event is occurring in the heap or the stack. The last column, “TLB access” describes the unique hits that occur to the translation lookaside buffer (TLB) which is a memory cache that is used to reduce the time taken to access a memory location.

**Table 2.** Example extracted memory access patterns for a benign application

Access type	Overhead	Samples	Local weight	Memory access	Symbol	Object	TLB access
Read	1.41%	1	2516	LFB hit	[.]	anon	L1 or L2 hit
Read	1.37%	1	2444	LFB hit	[.]	[heap]	L2 or L3 hit
Write	0.53%	1	940	Local RAM hit	[k]	[kernel.kallsyms]	L1 or L2 hit

Our framework formats and compiles all the memory accesses into readable data for our model. The formatter within our framework iterates through each line of the file created from the extraction phase and first adds necessary spacing in order to distinguish between features/columns. It then formats each column in the file by representing each possible value for each parameter by a number ranging from 1 to  $n$ , with  $n$  being the total number of possible values that parameter can take. Once each value has been replaced with a corresponding number, the script continues to iterate through the remaining lines in the file and performs the same actions on the data. The formatted file is then input to the machine learning models to make predictions as to whether or not a specified process is malicious. It operates by creating numerous decision trees during training.

## 2.5 Ensemble Comparison

Our framework uses feature extraction of task structure properties, system calls, and memory access patterns in order to facilitate a multi-pronged approach to malware detection and analysis. The task structure of a process describes the shared resources used such as an address space or open files. The kernel stores the list of processes in a linked list and each process contains a task structure which contains all the information about that specific process. System calls are the fundamental interface between an application and the Linux kernel and are invoked to perform an action. Some common system calls are read, write, and open. A memory access pattern is the pattern with which a system or program reads and writes memory to secondary storage.

We have identified various features within a Linux environment whose values deviate sufficiently over time when they are benign or malicious. These features consist of the smallest subset possible to provide detection of malware [5, 17, 23]. These features were selected after much testing and observation to determine their deviance in value when they were benign or malicious. Thus far we have performed these extractions on Linux Kernel 4.10, and intend to further test on Windows, Android, and various IoT devices. The selection of the Linux operating system as our target is based on the widespread use of Linux dialects for many IoT devices.

Our task structure, system call, and memory access features are harvested individually and across different system times. A vector is a collection of extracted features of a given feature type (task structure, system calls, memory access). Each vector has a given formatting requirement given which feature type it belongs to. For instance, task structure and memory access vectors are a collection of extracted features  $\{F_1, F_2, \dots, F_n\}$  while system calls are 3-g representations of extracted system call features  $\{\text{Call}_1, \text{Call}_2, \text{Call}_3\}$ . These vectors are independent of each other and there is a respective support vector machine, random forest, and logistic regression model for each feature type as they cannot be classified with a shared model. When sufficient vectors of a given feature type are collected, the vectors are passed to our models for analysis, regardless of the status of other feature type vectors. This results in independent collection time and data sampling for each feature type. If all feature types return a benign classification for their respective vectors then a device is considered to be benign. If any of the vectors are classified as malicious, then the device is exhibiting suspicious behavior. It is worth noting that a malware may attempt to spoof one of its feature types such as its task structure but may not attempt to spoof all three feature types at the same time resulting in a more resilient IoT device.

## 2.6 Dual-Stage Classification

Dual-level classification architectures have been shown to be successful in classifying malware [23, 24]. Besides classifying many aspects of a process, our framework uses a dual-stage approach. Our first stage is an anomaly detector which

has been trained solely on benign data and is highly sensitive. Our second stage occurs if the sliding window is found to be anomalous after being analyzed by our anomaly detector (first stage). It is passed to our classifier which has been trained on both benign and malicious data. This classifier attempts a more in-depth analysis to further assess the window as benign or malicious and makes a decision. Our first stage can mark a task as suspicious, limiting its execution, while the second stage will perform a more in-depth analysis to fully determine the intent of the task. Our framework contains a one-class support vector machine, a random forest, and a logistic regression model for each of our feature types (task structure, system calls, and memory access patterns) which analyze independently of each other. A vector is first passed to the one-class support vector machine and if the vector is determined not to be benign, the vector is passed to both of our random forest and logistic regression models for comparison. The one-class support vector machine is trained solely on benign datasets and the random forest and logistic regression models are trained on the same dataset containing both benign and malicious samples.

Once a stage-one alert is identified the agent will be notified to extract more data and to monitor more features in an effort to pass the additional information to stage-two before final determination is taken. A Quasi-malware state is a state in which a device has limited performance and network impact while it is in an intermediate evaluation state. The agent on the deployed device can be informed after a stage-one alert in order to restrict or modify various aspects of the device (Quasi-malware state) prior to the stage-two determination in order to prevent further malicious behavior or the propagation of malware. If stage-two finds no malware the agent is informed to return to normal operations. Throughout the process a visual representation of the networked devices is maintained denoting the health of these devices. A key feature of our dual stage approach is its ability to be diverse enough to accommodate the varying degrees of configuration of deployed devices. This diverse nature enables our first stage to function with a different sliding window size and threshold than our second stage. The first stage can be configured to analyze a sliding window and threshold that has been shown to be overly sensitive while our second stage can be configured for accuracy.

### 3 Results and Discussion

Due to our ensemble approach, our framework relies on various machine learning models, specifically one-class support vector machine, random forest, and logistic regression. The one-class support vector machine is our anomaly detector and as such was trained solely on benign data. Our random forest and logistic regression models are our classifiers trained using 10-fold cross-validation and were trained on the same set of benign and malicious processes.

For task structure features, the random forest model had a model result mean of 99.98% and an accuracy of 100%. The task structure logistic regression model had a model result mean of 65.98% and an accuracy of 65.35%.

For system calls, the random forest had a result mean of 99.40% and an accuracy of 99.42% while the logistic regression had a result mean of 98.82% and an accuracy of 98.92%.

For memory access pattern analysis, the random forest model displayed a result mean of 87.64% and an accuracy of 89.14%. The logistic regression model for the memory access pattern approach displayed a result mean off 87.67% and an accuracy of 89.26%.

When classifying a process as a whole, on average across all three techniques, our one-class support vector machine had an accuracy of 86% served as our anomaly detector. Due to our dual-stage approach, the remaining 14% of unknown or malicious occurrences were passed to our random forest and logistic regression models for further classification at which point either model came to its own respective classification for the tested data. In total, our random forest and logistic regression models showed overall more accuracy than our one-class support vector machine model however required more time to classify the data. By using our anomaly detector as a first stage, we reserved in-depth classification and overhead to only the processes that truly required it.

Table 3 lists some benign and malicious processes which were used for the testing of our models. We attempted to use processes which produced a minimum of 1000 task structure vectors, 100 system call vectors, 100 memory access vectors. On average we have 500 task structure vectors, 500 system call vectors, and 200 memory access vectors per process.

**Table 3.** Example benign and malwares that were used to the accuracy of our framework

Tested benign	Tested malwares
Ack-grep, grep, bmon, cat, cbm, ethstatus, htop, ls, top, htop, wget, fzf, fortune, bc, task	Backdoor_c, doffoo, linux.worm.lupper, sotas, kaiten

Due to our ensemble approach, our framework relies on various machine learning models, specifically one-class support vector machine, random forest, and logistic regression per feature type. Using the same malware and benign processes, Table 4 compares the accuracy of each of our individual model using the same dataset in order to demonstrate how our selected machine learning algorithms perform. These results are the averages of each feature type (task structure, system calls, and memory access) for each respective model.

We took our benigns and malwares and observed the average accuracy of each feature type for each given model. We further recorded the average speed it took to make a determination. The short time to determination we observed with the one-class support vector machine supports our use of that machine learning model as a first stage classifier as the first stage is required to be a rapid and robust anomaly detector. The increased accuracy of the random forest supports our use of it as a in-depth classifier.

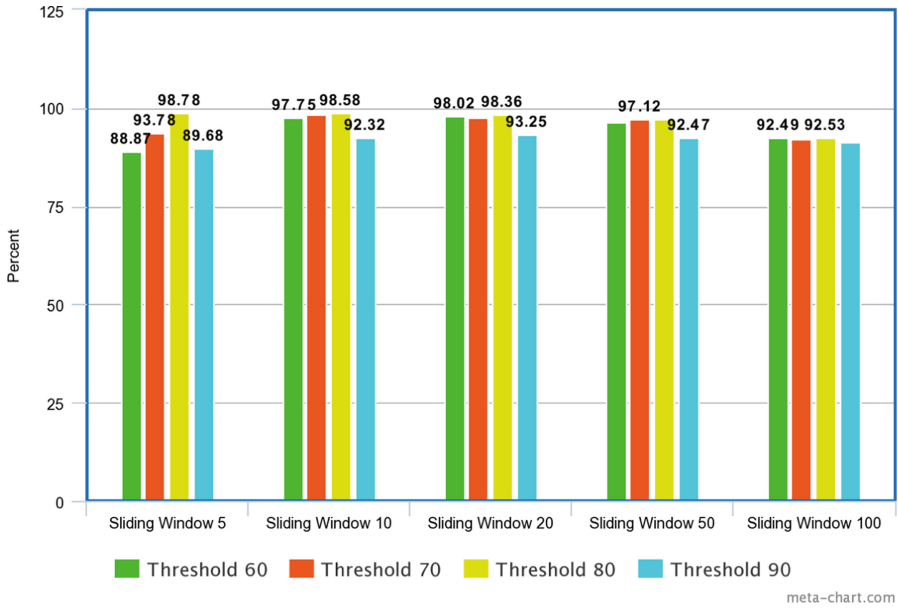
**Table 4.** Comparison of the accuracy of our models for each of our extracted feature types

	Task structure accuracy	System call accuracy	Memory access accuracy
One-class SVM	84%	93%	83%
Random forest	98%	99%	89%
Logistic regression	65%	98%	89%

To improve upon our frameworks accuracy and robustness, rather than classifying a process as a whole, we implemented the use of sliding windows. These sliding windows consist of chunks of our extracted features of a process and are passed in similar fashion to our one-class support vector machine and anomalies are then passed to our two classification models. By classifying a process in chunks rather than as a whole, we were able to make earlier determinations as to whether a process was benign or malicious. Furthermore, delayed execution of malicious activity eventually fell into a sliding window as we iterated through the extracted features and registered as anomalous, increasing our chances of detecting malicious activity. The use of sliding windows also allows us to continuously monitor an executing application and classify smaller chunks of data rather than a process as a whole which results in less computation.

Each sliding window can also have a threshold with which judgment is made upon the window as a whole. For instance, a sliding window of ten records can be said to have a threshold of 80% if eight evaluated records are determined to be benign and thus the whole window determined to be benign. We tested varying window sizes while testing each with a threshold of percent benign to percent malware within each window. The sliding window sizes tested were 5, 10, 20, 50, and 100. The thresholds tested were 90, 80, 70, and 60. This means that if the percent of benign vectors within the sliding window was determined to exceed the threshold then that whole sliding window is now classified as benign due to the prevailing percentage. An excerpt of this is shown in Fig. 2 which depicts the varying accuracy depending on threshold for sliding window sizes 5, 10, 20, 50, and 100. In other words, different sliding windows are more accurate with certain thresholds. The results depicted in Fig. 2 are the averages from all three of our feature types (task structure, system calls, and memory access) using the same benign and malicious processes in Table 3 in order to test the overall accuracy of various sliding window sizes and thresholds. We tested our benigns and malwares using a sliding window of five with the various thresholds and averaged the results of our models. We repeated our tests over the same dataset but with different thresholds and window sizes in order for the dataset to remain our constant. Figure 2 demonstrates that the accuracy of a sliding window is affected by the threshold selected for that sliding window and that each sliding window must be carefully analyzed to determine its best accuracy setting.





**Fig. 2.** Threshold comparison for varying sliding window sizes

Each of our individual models proved to be accurate in classifying a malicious process. Although our framework requires more training and testing, using probabilistic analysis, all three techniques are accounted for and a final determination can be achieved. Coupled with the use of sliding windows and a dual-stage approach for further accuracy, our framework shows great promise in determining when a device is compromised. Table 5 depicts the average accuracy of all three of our approaches for a given set of benign and malicious processes. Of note, the malware sotas was determined to be unknown by all three of our models and for the security and integrity of our device, if a process is unknown or anomalous it is assumed to be malicious.

**Table 5.** Average accuracy of our models over our three feature types for two benign and three malicious processes

	Class	Ensemble OC-SVM	Ensemble random forest	Ensemble logistic regression	Average detection
bmon	Benign	97.58% benign	96.72% benign	96.39% benign	Benign
ethstatus	Benign	98.42% benign	97.68% benign	96.33% benign	Benign
kaiten	Malware	89.85% unknown	93.94% malware	92.12% malware	Malware
linux.worm.lupper	Malware	87.37% unknown	91.94% malware	88.01% malware	Malware
sotas	Malware	98.96% unknown	86.41% malware	85.73% malware	Malware

IoT devices tend to run a single CLI application on a dialect of Linux and as such, our focus was on CLI applications. However, we further tested our models against various popular graphical user interface applications within a our x86 Linux system such as Chrome a web browser, VLC an audio player, Gimp a photo editor, and Skype a communications platform to ensure our models were accurate when introduced to new applications. Our models exceeded our expectations and accurately identified each of these as benign. However, a model trained purely on command line interface (CLI) applications will have a more difficult time classifying a benign graphical user interface (GUI) application as benign due to the difference in behavior between a CLI and GUI application. For instance, GUI applications are event driven whereas a CLI application usually performs their function once executed. This inherent difference in behavior results in models requiring more training depending on their intended target application type, CLI or GUI.

Past research has shown that the individual use of task structure, system calls, and memory access patterns are viable and effective approaches to the detection of malicious activity [5, 17, 23]. Each individual approach has its benefits and merits, however we explore the possible synergy of these approaches to augment their benefits and increase the work required by a malicious actor to circumvent each technique. Furthermore, we present this ensemble approach as a dual-stage classification framework which uses a highly sensitive first stage and a more in-depth analytical secondary stage.

A promising factor of our approach is that it is dynamic enough to allow for different window sizes and thresholds per deployed device for further usability in distributed networks that run varying device models. In other words, device types will have their own dual-machine learning system, customized to their respective window size and threshold value.

To further improve our framework, we propose exploring other available task structure features and system calls which can be used to train our model. Android based devices should also be tested [22] as well as windows based operating systems [20] and a complete cloud based solution for intrusion and compromise detection [11].

## 4 Conclusion

Malware is proving to be more dangerous than ever before and with the ubiquity of deployed IoT devices it is increasingly difficult to determine which devices have been maliciously compromised. We propose a framework which uses a dual stage behavioral analysis approach with increased accuracy of detection by combining the observance of task structure features, system calls, and memory access patterns into an ensemble approach. Coupled with a dual-stage anomaly detector and classifier, our framework offers high accuracy and malware detection to determine the integrity of an IoT device. These approaches show success on a variety of devices such as Windows, Linux, and Android. Various algorithms such as Support Vector Machines, Random Forest, and Logistic Regression have

been tested with varying yet promising accuracy when compared to Naive Bayes and Nearest Centroid algorithms.

Furthermore, these machine learning based malware detectors have been shown to require more work by a malicious actor in order to circumvent their detection. Our model showed great accuracy with limited training data. This approach allows a device to contain an embedded agent and extract key process data as the device executes its tasks. Furthermore, the data could be transmitted as a collected vector to a cloud framework for processing.

Within a multi-stage framework, the first, sensitive, anomaly detection stage flags malicious executions and passes them to a secondary stage for a more in-depth analysis and classification. Once the malicious activity is detected a device can be halted or the task suspended from functioning within the network in order to prevent further execution. Furthermore, once refined, this approach may work with little overhead and interruption on the end device. This near real-time detection would detect various kinds of malware and could stop them before they complete their execution even when a malware attempts to delay its execution to obfuscate itself.

Our approach shows promise as it can be combined with other approaches for more accuracy and a higher detection rate while also maintaining low battery and CPU costs. It is our hope that as the rapid evolution of technology in our sociality increases so does our defensive and countermeasure capabilities. Malware is attributed as the cause of massive data and integrity loss. Due to the dynamic and aggressive nature of Malware we require dynamic, vigorous, and progressive approaches to defend against such attacks.

## References

1. Aziz, A.S.A., Hassanien, A.E., Hanaf, S.E., Tolba, M.F.: Multi-layer hybrid machine learning techniques for anomalies detection and classification approach. In: 13th International Conference on Hybrid Intelligent Systems (HIS 2013), pp. 215–220, December 2013
2. Blokhin, K., Saxe, J., Mentis, D.: Malware similarity identification using call graph based system call subsequence features. In: 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops, pp. 6–10, July 2013
3. Cabrera, J.B.D., Lewis, L., Mehra, R.K.: Detection and classification of intrusions and faults using sequences of system calls. *SIGMOD Rec.* **30**(4), 25–34 (2001)
4. Canzanese, R., Mancoridis, S., Kam, M.: Run-time classification of malicious processes using system call analysis. In: 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), pp. 21–28, October 2015
5. Canzanese, R., Mancoridis, S., Kam, M.: System call-based detection of malicious processes. In: 2015 IEEE International Conference on Software Quality, Reliability and Security, pp. 119–124, August 2015
6. Carbone, M., Cui, W., Lu, L., Lee, W., Peinado, M., Jiang, X.: Mapping kernel objects to enable systematic integrity checking. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS 2009, pp. 555–565. ACM, New York (2009)

7. Dolan-Gavitt, B., Srivastava, A., Traynor, P., Giffin, J.: Robust signatures for kernel data structures. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS 2009, pp. 566–577. ACM, New York (2009)
8. Dorj, E., Altangerel, E.: Anomaly detection approach using hidden Markov model. In: *Ifost*, vol. 2, pp. 141–144 (2013)
9. Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.: A sense of self for Unix processes. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy, SP 1996, p. 120. IEEE Computer Society, Washington, DC (1996)
10. Fuyong, Z., Tiezhu, Z.: Malware detection and classification based on n-grams attribute similarity. In: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol. 1, pp. 793–796, July 2017
11. Houmansadr, A., Zonouz, S.A., Berthier, R.: A cloud-based intrusion detection and response system for mobile phones. In: Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops, DSNW 2011, pp. 31–32. IEEE Computer Society, Washington, DC (2011)
12. Jain, A., Singh, A.K.: Integrated malware analysis using machine learning. In: 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), pp. 1–8, August 2017
13. Jaiswal, M., Malik, Y., Jaafar, F.: Android gaming malware detection using system call analysis. In: 2018 6th International Symposium on Digital Forensic and Security (ISDFS), pp. 1–5, March 2018
14. Liu, S., Huang, H., Chen, Y.: A system call analysis method with MapReduce for malware detection. In: 2011 IEEE 17th International Conference on Parallel and Distributed Systems, pp. 631–637, December 2011
15. Rhee, J., Riley, R., Lin, Z., Jiang, X., Xu, D.: Data-centric OS kernel malware characterization. *IEEE Trans. Inf. Forensics Secur.* **9**(1), 72–87 (2014)
16. Rieck, K., Trinius, P., Willems, C., Holz, T.: Automatic analysis of malware behavior using machine learning. *J. Comput. Secur.* **19**(4), 639–668 (2011)
17. Shahzad, F., Bhatti, S., Shahzad, M., Farooq, M.: In-execution malware detection using task structures of Linux processes. In: 2011 IEEE International Conference on Communications (ICC), pp. 1–6, June 2011
18. Sun, L., Nakajima, T.: A lightweight kernel objects monitoring infrastructure for embedded systems. In: 2008 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 55–60, August 2008
19. Upadhyay, H., Gohel, H., Pons, A., Lagos, L.: Virtual memory introspection framework for cyber threat detection in virtual environment. *Adv. Sci. Technol. Eng. Syst. J.* **3**, 25–29 (2018)
20. Upadhyay, H., Gohel, H.A., Pons, A., Lagos, L.: Windows virtualization architecture for cyber threats detection. In: 2018 1st International Conference on Data Intelligence and Security (ICDIS), pp. 119–122, April 2018
21. Xie, L., Zhang, X., Seifert, J.-P., Zhu, S.: pBMDS: a behavior-based malware detection system for cellphone devices, pp. 37–48, March 2010
22. Xin, K., Li, G., Qin, Z., Zhang, Q.: Malware detection in smartphone using hidden Markov model. In: 2012 Fourth International Conference on Multimedia Information Networking and Security, pp. 857–860, November 2012
23. Xu, Z., Ray, S., Subramanyan, P., Malik, S.: Malware detection using machine learning based analysis of virtual memory access patterns. In: Design, Automation Test in Europe Conference Exhibition (DATE) 2017, pp. 169–174, March 2017

24. Yuan, X.: PhD forum: deep learning-based real-time malware detection with multi-stage analysis. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–2, May 2017
25. Zhao, K., Li, Q., Kang, J., Jiang, D., Hu, L.: Design and implementation of secure auditing system in Linux kernel. In: 2007 International Workshop on Anti-counterfeiting, Security and Identification (ASID), pp. 232–236, April 2007
26. Zhao, K., Zhang, D., Su, X., Li, W.: Fest: a feature extraction and selection tool for Android malware detection. In: 2015 IEEE Symposium on Computers and Communication (ISCC), pp. 714–720, July 2015



# Enhanced Security Using Elasticsearch and Machine Learning

Ovidiu Negoita and Mihai Carabas(✉)

University Politehnica of Bucharest, 313 Splaiul Independentei, 060042 Bucharest, Romania  
ovidiu.negoita95@gmail.com, mihai.carabas@cs.pub.ro

**Abstract.** The purpose of this paper is to highlight how can Elasticsearch be used to enhance the security of your applications and your cloud infrastructure by combining intrusion detection systems with machine learning techniques in order to detect possible attacks. It will cover the setup and configuration of a test environment for anomaly detection and network security alerting using Elasticsearch as the core for storing data. Snort is used for monitoring, alongside system and network analytics collected via Metricbeat and Packetbeat. Built-in machine learning jobs from Elastic will be used to find disturbances in the normal operation of the devices. To create a baseline dataset the Damn Vulnerable Web application is used to generate analytics and alerts upon exploiting the vulnerabilities exposed.

**Keywords:** OpenStack · Elasticsearch · Logstash · Kibana · Metricbeat · Packetbeat · Snort · DVWA · Hierarchical Temporal Memory

## 1 Introduction

With the continuous increase of cloud-based services and IoT connected devices there is also a need for better availability and reachability between those internet nodes. Migrating your applications and services to a scalable system, like the cloud environment will boost your performance, but will also increase your product's attack surface, generated by the infrastructure's vulnerabilities or by other applications running on the same server. To prevent such scenarios powerful and costly equipment is necessary, or software capable of detecting threats in real-time.

Common defence approaches include signature-based intrusion detection systems or web application firewalls, which can achieve real-time monitoring and analysis of network traffic, application and system logs. Most tools also provide file integrity checks, configuration and permissions assessments, policy violation and malicious behaviour detection, active threat prevention and attack surface reduction. But, because all of these are based on known vulnerabilities and exploits, hybrid solutions appeared that are enhancing those mechanisms with supervised or unsupervised machine learning techniques to better detect incoming attacks.

The main purpose of this research is to study popular security tools and how they integrate with Elasticsearch, and how can machine learning methods can be used to detect anomalies inside an infrastructure. The architecture proposed consists of three

virtual machines with two of those being Elasticsearch nodes and anomaly detectors, and the last one will act as both a physical device and an application container. Its normal operation parameters will be monitored with Metricbeat and Packetbeat and the network traffic will be analysed by Snort, configured with the community rules. The Damn Vulnerable Web Application will also be installed and used as a threat generator, to easily generate data for the basic vulnerabilities available in the application.

The following chapters will present relevant articles and proof-of-concepts that successfully achieved an increase in detection accuracy by combining both methods. Another focus will be on algorithms that train using time-series data with the intent of finding disturbances in the operation of physical devices, which are critical in a cluster. Based on those, our setup environment on OpenStack will be detailed, along with the proper configuration for every software used.

The structure of the paper is as follows: Sect. 2 presents the related work regarding Elasticsearch and machine learning and Sect. 3 presents the state of the art about elastic stack (Elasticsearch, logstash, kibana, machine learning). Section 4 describe the architecture we have used in order to implement IDS information ingestion into the elastic stack (Sect. 5).

## 2 Related Work

Achieving real-time anomaly detection in your infrastructure is detailed in numerous blog posts from the Elastic website, either by integrating your logging system with an intrusion detection system (IDS), like Wazuh or Suricata or by using the built-in functionality of creating machine learning jobs based on the metadata particularities of a threat, like the increased number of requests in a denial-of-service attack. So, the two focus points that were studied in the following papers were algorithms for detecting anomalies in a dataset and popular security tools for monitoring and threat prevention.

### 2.1 Improve Security Analytics with Elastic, Wazuh and IDS

Detailed in the following article [5], there is an implementation of an automated threat detection system, using Suricata sensors to monitor the network traffic and Wazuh agents for the system calls, file integrity and application logs monitoring. What makes Wazuh such a good choice is the built-in integration with Elasticsearch and Kibana and the versatility of the agents, which can be configured to parse network alerts from Suricata, so that all your monitoring data will be stored and indexed in Elasticsearch. They found a list of source IP addresses with abnormal activity by applying a “population analysis” job on the data, which calculated a baseline for the number of high-level alerts generated by each source IP. The analysis correctly identified traffic that matched multiple rules that ended in a firewall rule to block the specific source IP triggered by the Wazuh Active Response module.

Another approach for parsing the output of any IDS would be to use Filebeat [8] and depending on the data format you could use an existing parser or write your own. Most frequently used products, like Suricata, already have a module for ingesting data into Elasticsearch, with individual documentation for configuration and integration [9].

## 2.2 WebHound: A Data-Driven Intrusion Detection from Real-World Web Access Logs

In the context of web application security, the most common defence tool is a firewall, also known as WAF, which analyses HTTP traffic and eventually filters or blocks malicious requests from reaching your web application. It is using a set of rules defined based on known web vulnerabilities, like cross site scripting, SQL injection, denial of service or file inclusion. A good implementation of a web application filter is ModSecurity [14], which offers real-time HTTP traffic monitoring and access control, rule-based logging and attack surface reduction by selectively disabling unwanted HTTP features.

Since current defence approaches, like IDS or WAF use signature and pattern detection, they can detect only known malware. To tackle this problem, authors of article [6], analysed and stored information from web application logs in a graph structure and using unsupervised machine learning created a model to detect abnormal nodes inside a graph. This solution does not only offer better accuracy than ModSecurity, but also offers security experts the intrusion procedure and original entry points exploited by attackers.

## 2.3 Detecting Web Attacks Using Multi-stage Log Analysis

The importance of log analysis is further highlighted in [3], where authors proposed a hybrid solution, using both pattern matching to detect known vulnerabilities and a supervised machine learning technique. The first one provides a fast response to attacks, while data is gathered and annotated by experts for the training algorithm. The proof-of-concept was deployed on the Amazon Web Services stack and used Elasticsearch for storing log information, Kibana for pattern matching and Bayes Net for machine learning. Tests were conducted only for the SQL injection attack and based on a set of 10000 logs the detection accuracy of the combined solution was better than both used alone.

## 2.4 Unsupervised Real-Time Anomaly Detection for Streaming Data

Authors of [1] provide a real-time technique for anomaly detection and a benchmark designed for evaluating such algorithms on streaming data. The algorithm is based on a Hierarchical Temporal Memory network that was enhanced with two additional steps, error prediction to counter spontaneous shifts in data and anomaly likelihood where data is very noisy, and the first step is not enough. If we look at the CPU usage for example, whenever a process is started a spike in data will appear and be classified as an anomaly by the network, but the prediction error will also be high and if the CPU usage persists on a high value the prediction error will drop. This is used to change the algorithm's prediction in the training phase, so the network will treat changes in data differently and will not classify any spike as anomalous.

Another important contribution is the Numenta Anomaly Benchmark, which is designed for evaluating anomaly detection algorithms on streaming data. The repository contains a set of data streams from a variety of sources, ranging from server network utilization to temperature sensors on industrial machines to social media chatter. There



are also 3 scoring functions that focus on rewarding early predictions or false positives and penalizing false negatives.

### 3 State of the Art

As mentioned in the introductory chapter, the paper will focus on Elastic products and services, because they offer open source solutions which have built-in anomaly detection algorithms for your data. This is done with usage of unsupervised learning techniques that use data stored in Elasticsearch for training and pattern recognition. It is a very powerful approach if your data is enough and consistent, but this can be resolved easier by using the Beats data shippers, which helps with collecting information from devices.

The combination of Elasticsearch, Logstash, and Kibana, referred to as the Elastic Stack, is available as a product or service. Logstash provides an input stream to Elasticsearch for storage and search, and Kibana accesses the data for visualizations such as dashboards. Elastic also provides “Beats” packages which can be configured to provide pre-made Kibana visualizations and dashboards about various database and application technologies.

One of the major advantages over the competition is that Elasticsearch provides clustering almost seamlessly and the setup is easy. Also, the distributed nature of Elasticsearch allows customers to easily handle data that is too large for a single node to handle. By using multi-node clusters, we can also achieve uninterrupted work of our application, even if several machines are not available due to outage or administration tasks such as upgrade.

#### 3.1 Elasticsearch

Elasticsearch [7] is the most popular enterprise search engine, according to the DB-Engines ranking, and lets you store, search and analyse all kinds of documents. It can be used alongside a data collection and log-parsing engine called Logstash, an analytics and visualization platform called Kibana, and Beats, a collection of lightweight data shippers. The four products are designed for use as an integrated solution, referred to as the “Elastic Stack”.

The architecture behind this product is based on the Apache Lucene library and provides documented HTTP APIs as an easy way to communicate with any application. Elasticsearch stores its data as documents in one or more indices, which are like a database. A document is an object with one or more fields, each field has a name and multiple values. Before indexing, the document is analysed according to a mapping, tokens are extracted from the input text and saved in a map, where the key is the token and the value is a list of documents [16].

On a higher level, each server running an instance of Elasticsearch is called a node and will run by default in a distributed mode, searching for other nodes in the network by broadcasting a message containing the cluster id. There are three types of nodes and the primary one is the data node, which oversees indexing data and resolving queries. The master node is mandatory in a cluster, because it manages the requests and distributes them to the data nodes. The last type is a node acting as a bridge between multiple

clusters and is called a tribe node. Only the first type can store data, and it can be used together with the gateway module to achieve long-term persistence and store the state of the cluster and later be recovered in case of a server crash or restart [16].

### 3.2 Kibana

Kibana [10] is an open source plugin that offers data visualization capabilities on top of the information stored on your cluster. It provides easy-to-use features such as histograms, bar or line graphs, scatter plots, pie charts, heat maps, and built-in geospatial support. It also gives you the possibility to create shareable dashboards and reports that you can use to interactively navigate through large amounts of log data.

### 3.3 Logstash

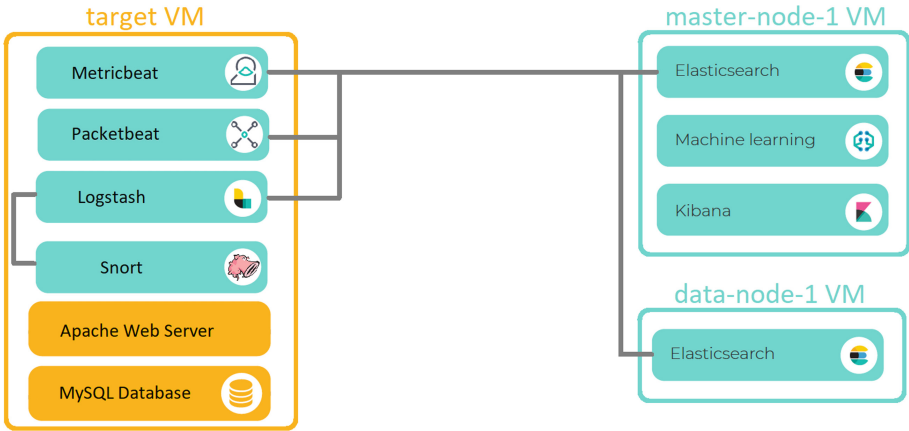
Logstash [11] is a tool to collect, process log and event messages and is the primary way of ingesting data into Elasticsearch. Collection is accomplished via configurable input plugins including raw socket/packet communication, file tailing, and several message bus clients. Once an input plugin has collected data it can be processed by any number of filters which modify and annotate the event data. Filters are a very good way to annotate and transform data for the training of an unsupervised model. Finally, Logstash routes events to output plugins which can forward the events to a variety of external programs including Elasticsearch, local files and several message bus implementations.

### 3.4 Machine Learning Recipes

Another important feature available in the Elastic Stack are the machine learning jobs, which can model any type of time-series data and can be configured to detect anomalies in your data, like a high number of unsuccessful logins in a short period of time. Detailed in [4] there is an example of a “recipe” for detecting DNS data exfiltration, cyberattack stage in which hacker tries to extract valuable data from the system he gained access to by using a DNS tunnel. The recipe is basically just a search query and a function, like count or sum, applied on the retrieved data.

## 4 Architecture

Motivated by results of certain articles [2, 3, 5, 6] to increase the security of your infrastructure this paper is proposing the usage of an IDS together with Elasticsearch for storing alerts, events, messages and network packet data. Upon all this data machine learning jobs, defined with the built-in module in Elasticsearch will run with the goal of having an automated process of detecting possible vulnerabilities and attacks. This module uses proprietary machine learning algorithms that automate the analysis of time-series data. They create accurate baselines of normal behaviour in the data and identify anomalous patterns.



**Fig. 1.** Deployed architecture on OpenStack

In terms of security products for monitoring, analysis and alerting, we are focusing on open source solutions, like OSSEC, Wazuh, Suricata and Snort. All of them can perform log analysis, integrity checking, Windows registry monitoring, rootkit detection, real-time alerting and active response. Wazuh is an OSSEC fork, with an extended functionality, including integration with Elastic stack, which makes it the first candidate. Snort is a good all-around rule-based detection system that can also be integrated with Elasticsearch, but also has the potential of tracking unknown attacks. Because it uses a standardized format for rule definition, they can even be generated by machine learning methods. So, in a scenario where we have Snort alerts alongside the network traffic data that caused them, we can then train a model to generate new alerts.

The environment will be setup using Openstack and it will contain three virtual machines, like stated in the introductory chapter and Fig. 1. Two of those will be data collectors, running Elasticsearch in cluster mode and one of them will also run Kibana for presentation and overview of the infrastructure and Logstash for ingesting and transforming data. The observed target will be our last VM and will act as potential victim/critical device, where Damn Vulnerable Web App will be deployed. It will have an IDS running, Beats for collecting system-level CPU usage, memory, file system, disk IO, and network IO statistics, as well as information about processes that are running on your system, network traffic metadata and analytics.

#### 4.1 Wazuh and Suricata

In the Fig. 2 below, the architecture presented in [5] is based on lightweight Wazuh agents that run on monitored systems, reporting to a centralized server where data analysis is done. In addition, it provides a complete Kibana plugin for configuration management, status monitoring, querying and alert data visualization. Suricata was added, because Wazuh lacks network monitoring, and is integrated with Elasticsearch with an agent that reads files, but it can also be used with a Filebeat module [9].

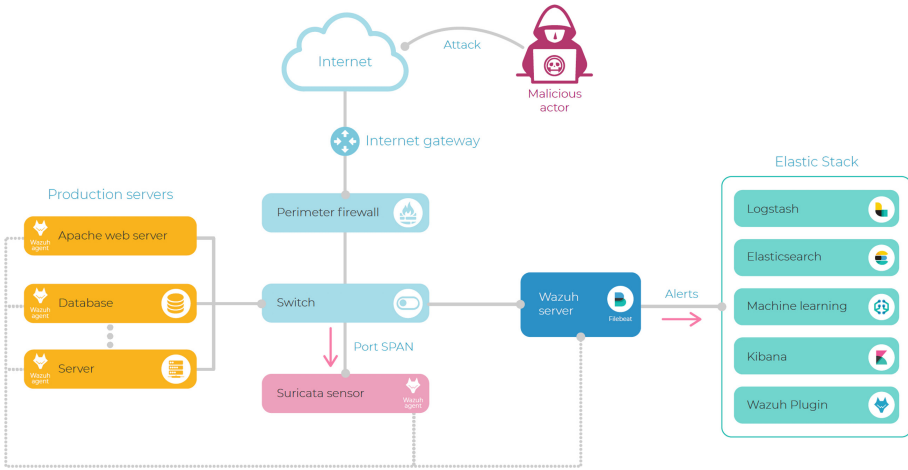


Fig. 2. Wazuh & Suricata monitoring system [5]

## 4.2 Snort

Snort can perform protocol analysis and content searching or matching. It can be used to detect a variety of attacks and probes, such as buffer overflows, semantic URL attacks, stealth port scans, OS fingerprinting attempts, and much more. It uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user specified file, a UNIX socket, or WinPopup messages to Windows clients.

Snort can be configured in three main modes: sniffer, packet logger, and network intrusion detection. In sniffer mode, the program will read network packets and display them on the console and in packet logger mode, it will log packets to the disk. In intrusion detection mode, the program will monitor network traffic and analyse it against a rule set defined by the user. The system will then perform a specific action based on what has been identified.

Because Wazuh doesn't offer network monitoring and analysis of traffic data, the proposed solution above had to use Suricata, but this paper is considering Snort as a better option. The configuration and integration with Elasticsearch will be detailed in the following chapter, and the usage will be to monitor the activity of the observed target, while the DVWA will be exposed to different attacks.

## 5 Implementation

For our first goal of using device sensor data to detect disturbances in the normal mode of operation, the focus is on the machine learning algorithm, which will be trained on sample data from Numenta anomaly benchmark and ranked using the scoring functions. The provided corpus contains data from a variety of sources, ranging from server network utilization to temperature sensors on industrial machines to social media chatter, totalling

58 data streams, each with 1000–22,000 records, for a total of 365,551 data points. Also included are some artificially-generated data files that test anomalous behaviours not yet represented in the corpus’s real data, as well as several data files without any anomalies.

Another collection of sample data that is going to be used for machine learning training can be found here [13]. This contains security related data, malicious files and static information about malwares, system and application logs from third parties, BRO and Snort log files, and other network related data. It is very good that most of the samples come from real-world applications and systems that were exposed to attacks due to certain vulnerabilities.

Our testing setup in Fig. 3 consists of three virtual machines running on Openstack, one of them being the observed system, on which the Damn Vulnerable Web Application is deployed to act as the target of different attacks. Alongside it, Snort will run in network intrusion detection mode and will output alerts and message in JSON format to be easier to transport to Elasticsearch, with the help of Logstash. Another two monitoring modules from Elastic installed are Metricbeat, for system information and Packetbeat, for HTTP traffic analytics. The other two VMs are running Elasticsearch in cluster mode and on the master node Kibana is also running.

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor
<input type="checkbox"/>	data-node-1	Ubuntu 16.04 Xenial	10.9.0.112	m1.medium
<input type="checkbox"/>	master-node-1	Ubuntu 16.04 Xenial	10.9.0.106	m1.large
<input type="checkbox"/>	target	Ubuntu 16.04 Xenial	10.9.0.103	m1.medium

Displaying 3 items

**Fig. 3.** Running virtual machines on OpenStack

With data collected by Beats, the built-in module of machine learning jobs from Elasticsearch can be used to detect basic vulnerabilities or anomalies inside the cluster by defining recipes based on the specificity of the attack. The algorithms can detect and score unusual behaviours for a member of a population (population analysis), statistical rarity and anomalies related to temporal deviations in values, counts, or frequencies. Examples of certain recipes can be found here [4], alongside the process for defining and running them in Elasticsearch.

The Beats module from Elastic can also offer templates for the canvas feature in Kibana, in which you can define a set of queries and their corresponding representation (histogram, graph, chart, etc.) to be displayed on the same page. This is a useful feature for monitoring your target system that was also added to our setup environment and can be seen in Fig. 4 and 5.

Snort is configured to write alerts in JSON format, by adding the desired fields in the configuration file, and is running with all the community rules enabled. Logstash is configured using the file from the article [15] and reads the alerts file, converts the JSON data and sends it to Elasticsearch.

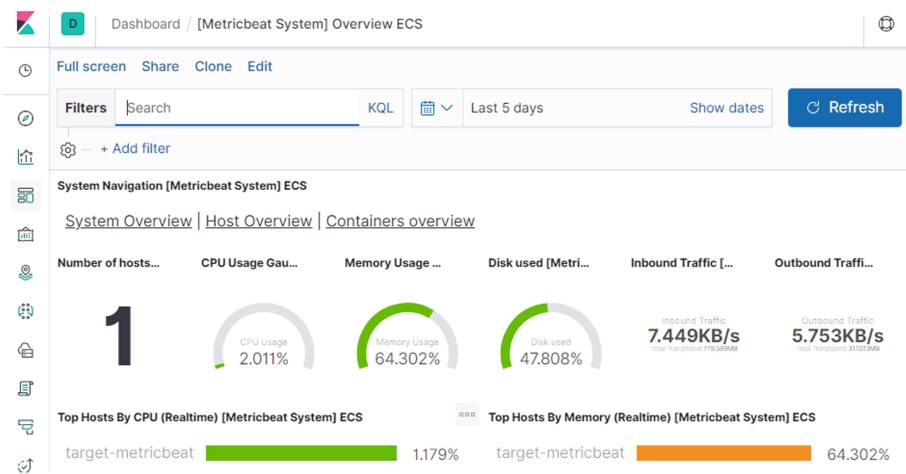
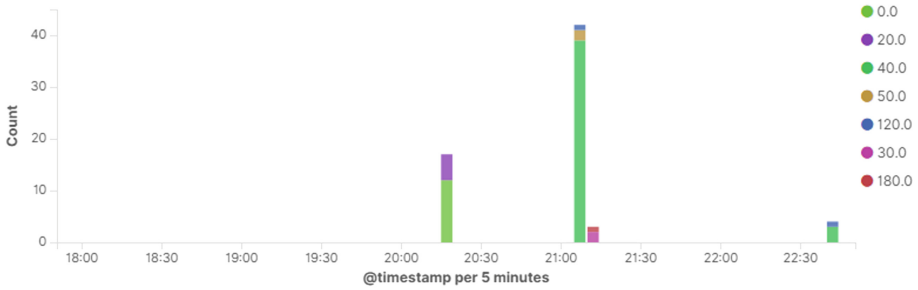
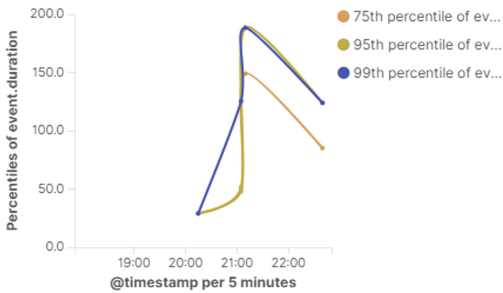


Fig. 4. Metricbeat overview dashboard in Kibana

Response times repartition [Packetbeat] ECS



Response times percentiles [Packetbeat] ECS



Errors vs successful transactions [Packetbeat] ECS

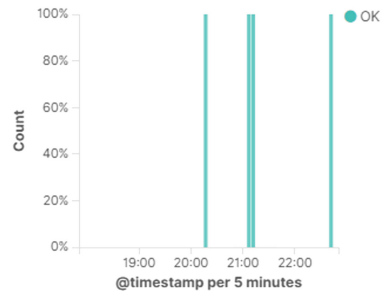


Fig. 5. Packetbeat response times in Kibana

## 6 Conclusions

Presented in the related work section, there is a clear advantage of using machine learning techniques, either by defining recipes for jobs using built-in module in Elasticsearch or by using an algorithm for anomaly detection like the ones presented in [1, 3, 6], to enhance the security of your infrastructure. An important mention is the Numenta anomaly benchmark, because it was specifically designed to score algorithms that detect anomalies in real-time and is providing a corpus of time-series data, which is exactly the type of data Elasticsearch is working with. This can be used to provide a large baseline for testing and training different classification algorithms.

The built-in functionality of machine learning jobs from Elasticsearch can be used to detect basic vulnerabilities or anomalies inside the cluster by defining recipes based on the specificity of the attack, but it requires manual configuration and it cannot be accurate for more complex attacks, as seen in [4]. A rule-based intrusion detection system can easily replace this solution and can be improved using machine learning for filtering. Because we are dealing with time-series, data like CPU usage, process or network information, gathered with the use of Beats module [12], can be used alongside alerts in training models for a better accuracy.

To conclude the architecture for the test environment and the configuration for the VMs is complete and Snort is now logging events and messages in Elasticsearch using Logstash. Also, system-level and network analytics are collected with the help of the Beats module from Elastic. DVWA is running on the Apache Web Server and is connected to an SQL database.

As future work, we will focus on ingesting the sample data into Elasticsearch and applying machine learning jobs to test the accuracy of the built-in module. Tests will also be done using collected data from Beats, while certain actions are performed against the DVWA to observe if patterns can be found while vulnerabilities are exploited. Another goal would be to integrate a custom algorithm, like the Numenta HTM [1] with Elasticsearch to be used instead of the proprietary algorithms that they offer.

**Acknowledgment.** The work has been funded by the Operational Programme Human Capital of the Ministry of European Funds through the Financial Agreement 51675/09.07.2019, SMIS code 125125.

## References

1. Ahmad, S., Lavin, A., Purdy, S., Agha, Z.: Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* **262**, 134–147 (2017)
2. Taylor, A.: Detect Beaconing with Flare, Elastic Stack, and Intrusion Detection Systems (2019). Austin Taylor. <http://www.austintaylor.io/detect/beaconing/intrusion/detection/system/command/control/flare/elastic/stack/2017/06/10/detect-beaconing-with-flare-elastic-arch-and-intrusion-detection-systems>. Accessed 9 Sept 2019
3. Moh, M., Pininti, S., Doddapaneni, S., Moh, T.-S.: Detecting web attacks using multi-stage log analysis. In: 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, India, pp. 733–738 (2019)

4. Paquette, M.: Using Machine Learning and Elasticsearch for Security Analytics: A Deep Dive. Elastic Blog, 02 May 2019. <https://www.elastic.co/blog/using-machine-learning-and-elasticsearch-for-security-analytics-deep-dive>. Accessed 08 Sept 2019
5. Bassett, S., Paquette, M.: Improve Security Analytics with the Elastic Stack, Wazuh, and IDS. Elastic Blog, 01 April 2019. <https://www.elastic.co/blog/improve-security-analytics-with-the-elastic-stack-wazuh-and-ids>. Accessed 08 Sept 2019
6. Wei, T.-E., Lee, H.-M., Jeng, A.B., Lamba, H., Faloutsos, C.: WebHound: a data-driven intrusion detection from real-world web access logs. *Soft Comput. Fusion Found.* **23**, 1–19 (2019)
7. Elastic.co: Elasticsearch Documentation (2019). <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>. Accessed 08 Sept 2019
8. Elastic.co: Filebeat Documentation (2019). <https://www.elastic.co/guide/en/beats/filebeat/current/index.html>. Accessed 08 Sept 2019
9. Elastic.co: Suricata module: Filebeat Reference [master] (2019). <https://www.elastic.co/guide/en/beats/filebeat/master/filebeat-module-suricata.html>. Accessed 08 Sept 2019
10. Elastic.co: Kibana Guide (2019). <https://www.elastic.co/guide/en/kibana/current/index.html>. Accessed 08 Sept 2019
11. Elastic.co: Logstash Documentation (2019). <https://www.elastic.co/guide/en/logstash/current/index.html>. Accessed 08 Sept 2019
12. Elastic.co: Metricbeat Documentation (2019). <https://www.elastic.co/guide/en/beats/metricbeat/current/index.html>. Accessed 08 Sept 2019
13. Secrepo.com: SecRepo - Security Data Samples Repository (2019). <https://www.secrepo.com>. Accessed 9 Sept 2019
14. ModSecurity: Open Source Web Application Firewall (2019). <https://www.modsecurity.org/about.html>. Accessed 9 Sept 2019
15. Combs, R.: Snort 3.0 with Elasticsearch, Logstash, and Kibana (ELK) (2019). Blog.snort.org. <https://blog.snort.org/2017/11/snort-30-with-elasticsearch-logstash.html>. Accessed 9 Sept 2019
16. Kuc, R., Rogozinski, M.: *Mastering Elasticsearch*, 2nd edn. Packt Publishing Ltd., Birmingham (2015)





# Memory Incentive Provenance (MIP) to Secure the Wireless Sensor Data Stream

Mohammad Amanul Islam<sup>(✉)</sup>

School of Computer Science and Technology, Xidian University, Xi'an 710071, China  
aman.cse.bd@gmail.com

**Abstract.** Evaluating the trustworthiness of the data in the wireless sensor network, data provenance has already been employed as a significant security feature. In this application, provenance based trust management involves the secret information transmission either with tempering the data or varying the transmission parameter. However, numerous challenges including secure transmission, storage overhead, bandwidth consumption have been perceived in this procedure. Thereby, a novel method has been proposed through modeling a real-time provenance framework based on the memory consumption in the designated payload of the sensor data packets. In the proposed method, trailed provenance is signified as a hierarchical relation between the data fields per packet and the data packets per data flow. The resiliency of this scheme indicates that it neither intrudes on the data nor the transmission parameters. Yet, the substantial experiment validates the proposed method and does not compromise the data accuracy and transmission efficiency.

**Keywords:** Bits · Memory · Provenance · Payload · Wireless sensor network

## 1 Introduction

The diverse nature of data in the wireless sensor network has always been observed based on its associated security concern since it traverses numerous intermediate nodes between source and destination. Besides this, the explosion of computing possibilities has greatly contributed to the development of the intelligence community. The inclusion of intelligent systems recently includes many application domains like financial analysis, location trace, environment monitoring, etc. It stimulates the tasks to collect potentially useful information from the streaming data in a sensor environment that has significance on numerous classification mapped to secret data provenance. Introducing data provenance has many applications for authorizing the data source, data flow, and ensures the integrity of data. The challenge is thus how to acquaintance the potential information to provenance to assess the trustworthiness of the above paradigms for the resource-constrained sensor environment. The importance of securing streaming data also highlighted in the research and development Challenges for

Critical National Infrastructure [1], which recommend research initiatives on developing a proactive and predictive security posture.

Though security and privacy [2] of data in the distributed systems [3,4] have already been explored, the accountability of data in this platform can ensure the privacy right. Enabling accountability can include the access and medication history of data that can be addressed by the data provenance. However, the definition of provenance is varied according to the context of its application. Data provenance in the domain of database, workflows, and cloud systems has already been introduced, where the provenance is referred to as the history of ownership or actions performed on the data. Hence, the provenance includes a vast amount of necessary information, thus its generation and validation increase the computational cost.

However, very few approaches have been studied in the context of network flow to secure the data stream. Proficient active timing [5–7] based data hiding without interfering the data is subjected to detection [8] and recovery of adversary acts. Another significant protocol, inter-packet delay (IPD) based provenance encoding proposed by Sultana et al. [9]. This scheme is resilient and provides scalability, imperceptibility, and robustness to attacks besides the protection of provenance forgery, insertion, deletion, alternation, replay, and integrity. The difference between the active timing and IPD based methods is, active timing scheme encodes a single secret message over the IPDs in a particular data flow, whereas the IPD method allows multiple nodes to encode the secret message over the same set of IPD. However, these schemes introduce the interference to the transmission parameter by delaying the data packets and hiding capacity per data flow is limited here. Hence, the characteristic of sensor data and the associated transmission constraints require extensive analysis to introduce provenance without interfering with the data and transmission channel.

In this paper, the problem of provenance provisioning with moderating sensor data or the transmission medium has been introduced and studied. This scheme analyzes the required bits for each data out of space limited in each segment (data field) of the data payload of a sensor data packet. However, introducing provenance to sensor network imposes a set of challenges:

- Computing provenance for the data stream and its validation must adapt fast processing and low computational cost;
- It must imitate the security (confidentiality, authenticity, integrity) of provenance information and the data in an attacker channel;
- Network parameters (e.g., delay, packet receiving rate) must not introduce significant overhead on processing;
- Data provenance must not increase the storage cost and energy-efficient.

Pursuing the aforementioned challenges, the proposed provenance framework was designed based on analyzing the size of sensor data against the primitive data type that initiated for both the data and the data fields of the pre-defined payload of a data packet. Data provenance in this method is introduced as a label with summarized binary status i.e., 1(high) or 0(low) on memory consumption in two distinct extents: (i) between the segmented data fields of a given payload

of a data packet (ii) between the data packets in terms of payload usage in a particular flow. The memory usage is accounted according to minimum required bits (utilized bits, or actual bit width) of each sensory data. In this methodology, the utilized bits refer to the only bits that belong to the sensor value out of the predefined size of its data type. Assume the binary of 11 within the 8 bits space is 00001011. The proposed scheme estimates only ...1011 as the utilized bits in binary. Thus, the actual bit-width of the digit 11 stands for 4 bits out of 8 bits.

However, at the end of accumulating the number of utilized bits for a particular data set, the number of incidents of each unique quantity of utilized bits is examined between the data fields of the payload per data packet i.e., data transport. Likewise, the data source computes the total utilized bits against the total payload of each packet and inspects the number of incidence of each distinct quantity of utilized bits among the data packets in a particular data flow. Thereafter, the product of any particular quantity of utilized bits and its number of incidence between the data fields per packet, and among the data packets per data flow is observed based on numerous usage threshold by the data source. However, such threshold-based inquiry deliberates the status of memory usage with a binary label as the provenance for each data packet and its corresponding data flow that hierarchically correlated.

Moreover, the proposed method considers the threat of modifying the data by keeping the perceived size of utilized bits for each sensor data. Hence, besides ensuring the data confidentiality, the security protocol of this method initiates counter values as a bit guard that correspond to each binary digit of sensor data. In this function, a set of initial counter values have been pre-shared between the data source and the sink node and verified based on a key synchronization procedure. Thus, the proposed method can detect error on access violation against the attacks that may cause of violating the integrity of data and accumulated utilized bits.

Furthermore, the proposed framework introduces a logistic regression-based binary provenance classification methodology at the data receiver, where the quantity utilized bits and their incidence has been initialized as two features vector. However, the proposed memory incentive provenance (MIP) framework requires an offline analysis of sensor data rather than altering the data or any transmission parameters to initiate the provenance as an indicator of memory consumption. Thus, there was no risk of data degradation and no additional complexity in diagnosing the channel parameters. In summary, the contributions of this paper include:

- Introducing provenance by analyzing the actual bit width of data against the length of primitive data type that has defined for data and data fields, and confirms the trustworthiness of data
- Provisioning provenance avoids its transmission as inline metadata, also evades tempering any parameter that affects the transmission;
- Determining provenance as a storage overhead indicator without acquiring additional storage;

- Designing security properties to confirm the integrity of data and the accrued utilized bits of each sensor data;
- Introduce learning-based provenance identification approaches with low complexity;

## 2 Related Works

The notion of data provenance is well established in the various scientific domain; however, the definition of provenance varies depending on the specific application domain [10]. Introducing data provenance in different fields of computer science has been initially discussed in the seminar paper by Becker et al. [11]. After that, its application has been found in the domain of database [12–14] and cloud [15]. Afterward, the importance of provenance and its application also has been observed in the domain of WSN [16,17] to describe the data source nodes (i.e. access points, router) and the operation on the data being transmitted [10].

As a basic provenance requirement for WSN a chain model of data provenance [18]. In this method, each node in the packet path appends the provenance information to the current provenance. Though this mechanism is simplest and easiest, it expands provenance size too fast and makes unaffordable the provenance transmission overhead. Another method of encoding data provenance within the data set by a data source as inline metadata was proposed by Chong et al. [19].

However, two-block provenance schemes: inter-packet delay (IPD) and probabilistic provenance flow (PPF) were proposed by Sultana et al. [9] and Fahmy et al. [20]. A longer provenance has introduced as composed of smaller blocks, where each small block is identified as the provenance encoded by each node on the travel path of a data packet. Although the IPD approach doesn't append any extra message to the data packet, both IPD and PPF schemes have a dependency on limiting the data packets.

However, another two methods CAPTRA (Coordinated Packet Traceback) [21] and CTrace (Contact-based traceback) [22] proposed the provenance in a distributed approach. In these methods, a data receiver doesn't receive the entire provenance with the data packet as it spread on the nodes along the packet path from the data source and destination. Though the energy and bandwidth overhead is low due to carrying a limited provenance with the packet, compromising nodes and link failure may cause provenance decoding failure in this approach.

However, the proposed method focuses the study into the following two classes: tracing the data payload overhead for per data packet and data flow and introducing provenance as an indicator of memory usage. It significantly differs from the above approaches in several aspects. (i) analyzes the usage of the data payload of a data packet regarding the actual bit width of data, (ii) the calculated provenance between the data fields per packet is hierarchically related to the provenance among the data packets per data flow, (iii) introduce a learning-based approach to verifying the pre-shared provenance at sink node. Since the provenance computed in this method based on only the original form of data, it has no dependency on any underlying security protocol e.g., encryption, decryption. An idea of detecting and recovering the attack was proposed by

**Table 1.** Important notations for the MIP method

Notation	Description	Notation	Description
$\varepsilon$	encrypt	$\mathcal{D}$	decrypt
$pk_i$	single data packet	$df_i$	single data field
$udf_i$	quantity of utilized bits per $df_i$	$dfu_i$	distinct quantity of utilized bits per $df_i$
$upk_i$	quantity of utilized bits per $pk_i$	$pku_i$	distinct quantity of utilized bits per $pk_i$
$pv_i$	provenance bit in binary digit	$k_i$	secret key for each $d_i$
$d_i$	single sensor data	$b_i$	each binary digit of $d_i$
$n_i$	single source node	$at_i$	arrival time of single $d_i$
$L$	size of $df_i$	$M$	size of data payload
$t_i$	single threshold	$DF_i$	single data flow

Wang et al. [8] by delaying the transmission of some packets. On the contrary, the introduced protocol illustrates a revision based bit protection approach that can detect the changes of the data if the data fields of a data packet is tempered.

### 3 System Model and Preliminaries

#### 3.1 Data Model

The proposed MIP framework demonstrates the provenance as a hierarchical relation between the data packet and data flow in terms of memory consumption within the data payload of a data packet by a data source. Such relation is verified at the SN to determine whether the data and the data packet in a particular data flow are secured. Table 1 lists important notations used in the rest of this paper for convenience. The MIP scheme comprises two significant parameters in analyzing the utilized bits:

1. size of each data field  $df_i \in dF$  is  $size(df_i) = L$  according to the defined primitive data type for both the data  $d_i$  and the data field  $df_i$ ;
2. size of the data payload of a data packet  $pk_i \in PKT$  is  $M = n * L = \sum_{i=1}^n size(df_i)$  bits, where  $n$  is the total number of segmented data fields of a given payload per  $pk_i$ .

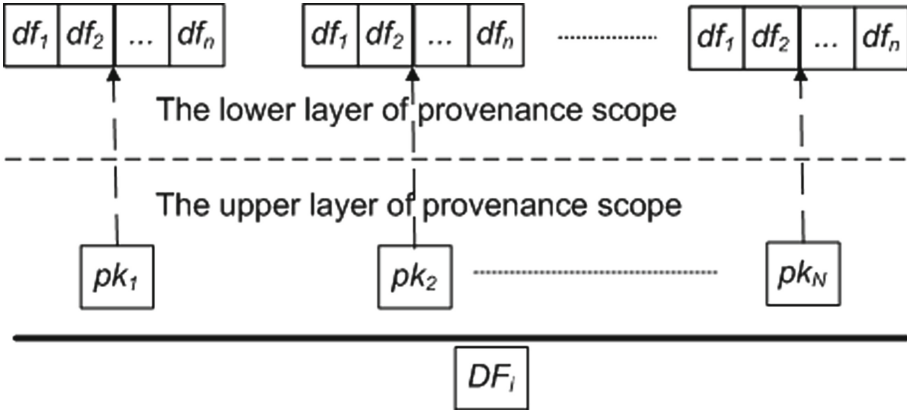
**Remark:** A primitive data type of data filed incurs  $L$  bits memory for the data of the same data type that indicates each binary digits of a certain data indices within  $[0, L]$  sequentially.

Hence, the consumed utilized bits per data field and data packet can be calculated as  $udf_i$  and  $upk_i$ , respectively according to the equation number (1) and (2) as,

$$udf_i = L - nub_i \quad (1)$$

$$upk_i = \sum_{i=1}^n udf_i \quad (2)$$

Here,  $nub_i$  denotes the quantity of non-utilized bits obtained in  $i^{th}$  data field.



**Fig. 1.** Provenance origination scope in a sensor data flow

**Definition 1.** *Non-utilized bits*  $nub_i = \sum_{\gamma_i \in \gamma_S} 1$  can be stated as a set of redundant bits [23],  $\gamma_S = \{\gamma_1, \gamma_2, \dots, \gamma_{|\gamma_S|}\}$ , where  $1 \leq |\gamma_S| \leq L$  and  $\forall \gamma_i \in \gamma_S$  indicates a single binary value 0.

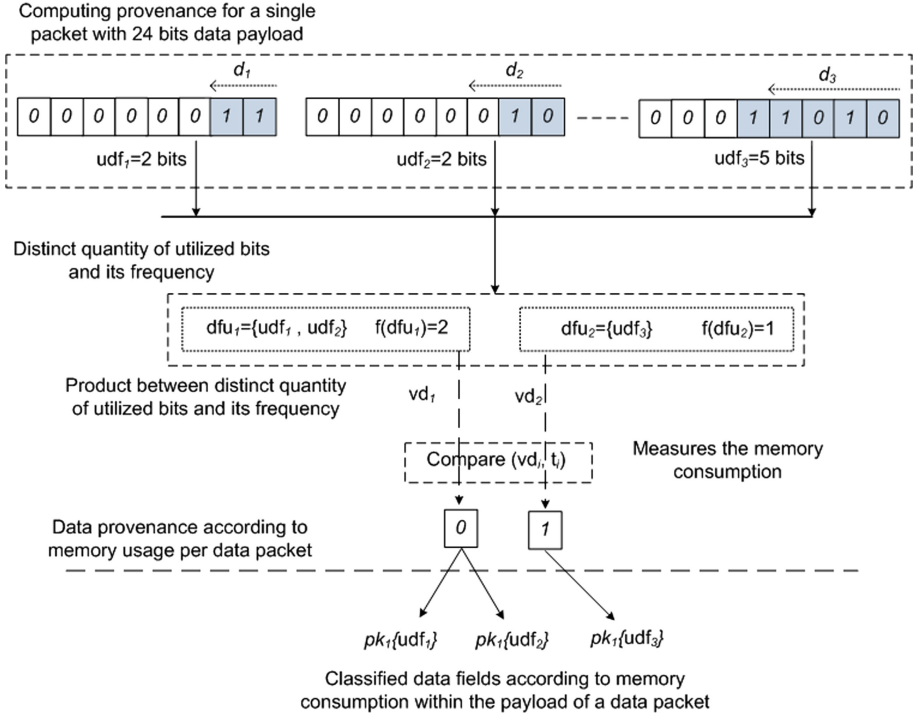
In this scheme, the provenance can be computed in offline with obtaining some pre-advised information: a number of data fields, size of the data payload per packet, calculated utilized bits of each sensory data.

However, conferring the relationship between the data packet and data flow, the proposed framework identified two extents or layers: lower layer and upper layer as a provenance stating scope that illustrated in Fig. 1. Here, analyzing the data fields per packet illustrated as the lower layer of the data stream, and analyzing the data packets demonstrated as the upper layer for the same data flow. Here, the number of data fields of a data packet might be a finite set, but the quantity of data packets might not be the same in several data flows.

Hence, the MIP framework can be illustrated as composed of various functional acts: computing utilized bits, distinguishing unique quantity of utilized bits, accumulating the incidence of each unique quantity of utilized bits, calculating product of each unique quantity of utilized bits and its frequency and originating the notification on memory consumption in each layer of provenance initiation scope. In Fig. 2, the origination of provenance has been described for a particular data set in a 24 bits payload of a data packet and traced the data fields according to storage status in the lower layer. In Fig. 3 the origination of provenance has been described for a particular data flow in the upper layer, where each packet has structured with the equal data payload and illustrated the relationship between a particular set of data packets and data flow.

### 3.2 Encoding Provenance

The idea of data provenance in this scheme is illustrated as the label information on the utilization of payload memory of data transport in a transmission channel.



**Fig. 2.** Provenance structure in lower layer

Thus, the query that functioning on memory consumption requires to compute some data parameters:

- distinct utilized bits and their frequency between the data fields per data packet;
- distinct utilized bits and their frequency among the data packets in a particular flow.

We assumed the structure of the data payload of a data packet is composed of number of segmented data fields as,  $pk_i = \{df_1 \parallel df_2 \parallel \dots \parallel df_{|n|}\}$ , Since, each  $df_i$  contains a particular sensor data  $d_i \in D$  during transmission, the payload utilization status based on consumed utilized bits according to the equation number (1) in each  $df_i$  of a data packet can be stated as,  $udf_{pk_i} = \{udf_1, udf_2, \dots, udf_{|n|}\}$ , where  $0 \leq udf_i \leq L - 1$ . Since, each  $udf_i \neq udf_j$  always, a set of distinct quantity of utilized bits can be accumulated from the data set  $udf_{pk_i}$  as,  $udU_{pk_i} = \{dfu_1, dfu_2, \dots, dfu_{|udU_{pk_i}|}\}$ , where  $udf_i \in dfu_i \neq udf_j \in dfu_j$ . Hence, the total number of elements in each  $dfu_i$  is accounted as the frequency of each distinct quantity of utilized bits  $f(dfu_i)$ . Thus a set of  $f(dfu_i)$  per data packet can be stated as,  $udF_{pk_i} = \{f(dfu_1), f(dfu_2), \dots, f(dfu_{|udF_{pk_i}|})\}$ .

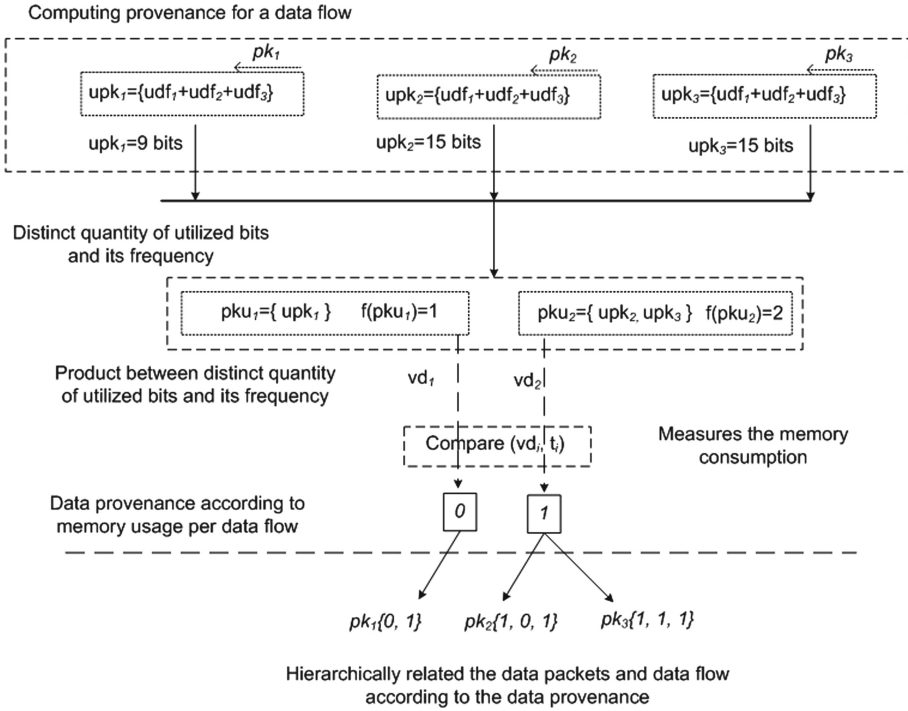


Fig. 3. Provenance structure in upper layer

However, a set of uniqueness of utilized bits  $udU_{pk_i}$  for each of  $N$  data packets in a single data flow  $DF_i$  might be perceived as  $udU_{pk_i} \subseteq udU_{DF_i}$ , where  $udU_{DF_i} = \{udU_{pk_1} \parallel udU_{pk_2} \parallel \dots \parallel udU_{|udU_{pk_N}|}\}$ .

Therefore, the data provenance based on the consumed memory between the data fields of a given payload of a single packet can formally be defined as:

**Definition 2.** Provenance is the desired binary codes  $pv_i \in \{0, 1\}$  as storage overhead  $\{low, high\}$ . Its existence is investigated on the product of each element of  $udU_{pk_i}$  and their corresponding frequency in  $udF_{pk_i}$  based on a certain threshold of data usage from  $\{t_1, t_2, \dots, t_{|t_{pk_i}|}\} \in t_{pk_i}$ , where  $t_{pk_i}$  denotes a set for memory threshold values per data packet.

Furthermore, the construction of each data stream should be considered as compose of several data packets as  $DF_i = \{pk_1, pk_2, \dots, pk_{|N|}\}$ . Due to consuming numerous quantity of utilized between the data fields of a data packet, the measured utilized bits in the payload of a data packet according to the equation number (2) also not always be equal to one another. Thus, the payload utilization according to consumed utilized bits among the data packets in a certain  $DF_i$  can be stated as,  $upk_{DF_i} = \{upk_1, upk_2, \dots, upk_{|N|}\}$ , where  $0 \leq upk_i \leq M-1$ . Since, each  $upk_i \neq upk_j$  always, a set of distinct quantity of utilized bits from the data



set  $upk_{DF_i}$  can be accumulated as,  $upU_{DF_i} = \{pku_1, pku_2, \dots, pku_{|upU_{DF_i}|}\}$ , where,  $upk_i \in pku_i \neq upk_i \in pku_j$ . Hence, the total number of member in each  $pku_i$  is also accounted as the incidence of each distinct quantity of utilized bits  $f(pku_i)$ . Thus a set of  $f(pku_i)$  per data flow can be stated as,  $upF_{DF_i} = \{f(pku_1), f(pku_2), \dots, f(pku_{|upF_{DF_i}|})\}$ . Thus, the data provenance based on the payload utilization among the data packets of a single data flow can formal be defined as,

**Definition 3.** *Provenance is the desired binary codes  $pv_i \in \{0, 1\}$ . Its existence is investigated on the product of each element of  $upU_{DF_i}$  and their corresponding incidence in  $upF_{DF_i}$  based on a certain threshold from  $\{t_1, t_2, \dots, t_{|t_{DF_i}|}\} \in t_{DF_i}$ , where  $t_{DF_i}$  denotes a set of memory threshold values per data flow.*

## 4 The Proposed Framework

### 4.1 Overview

The proposed memory incentive provenance (MIP) introduces a decentralized approach of encoding data provenance that exhibits as a result of a query on the amount of usage of payload memory of a data packet with binary status. The key idea of this scheme has developed on diagnosing the utilization of the segmented data fields of a predefined payload of a data packet. The MIP scheme demonstrates the provenance manifestation in two distinct layers of a data flow to secure the sensor data stream according to Fig. 1. In the lower layer, the MIP methodology explores the distinct utilized bits and their frequency among the data fields per data packet to define the data provenance according to Definition 2. Such assessment drives this application to explore the distinct utilized bits and their frequency among the data packets in a particular flow to form the data provenance according to Definition 3 in the upper layer.

However, any alteration on data can be a cause of having an inaccurate quantity of utilized bits at the SN. Therefore, a revision based bit protection is introduced besides encrypting the data by the data source in this framework. The encryption procedure includes a data specific secret key which has been pre-shared. In this work, the secret key is comprised of data arrival time and the counter values that initialized corresponds to the utilized bits of each data. Hence, before determining the provenance, SN can determine the received data whether they altered or not based on a key synchronization procedure.

Hence, the tailed procedures of memory incentive provenance provisioning by a data source and its detection at the sink node can be trailed in below steps according to Fig. 4:

- Accumulation of distinct utilized bits and their frequency.
- Compute the product values and several threshold parameters.
- Manifesting the data provenance.
- Security initiatives for data provenance.
- Data decryption and provenance identification.

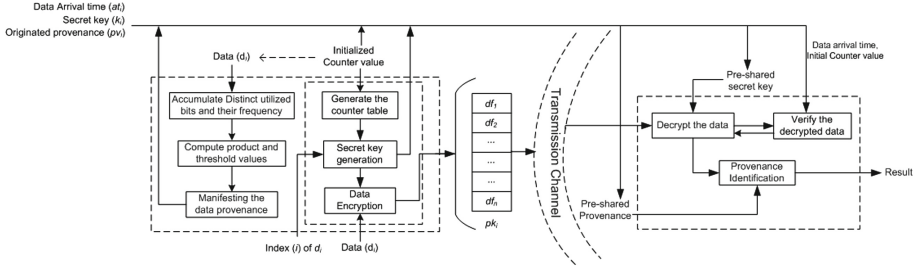


Fig. 4. Provenance initiation and identification phases

### 4.2 Accumulation of Distinct Utilized Bits and Their Frequency

As the first step provenance allocation, a data source  $n_i$  initiates a set of sensor data stream  $D = \{d_1, d_2, \dots, d_{|D|}\}$  to transmit with  $N$  number of data packets  $\{pk_1, pk_2, \dots, pk_{|N|}\}$  through a secure channel. Assume that, each data packet  $pk_i$  has structured with  $M$  bits data payload, which is composed of which is composed of equal length of ( $L$  bits)  $n$  number of data fields. Thus, following such predefined structure of data payload of a data packets,  $n_i$  forms a data set per  $pk_i$  as  $D_{pk_i} = \{df_1(d_1) \parallel df_2(d_2) \parallel \dots \parallel df_{|n|}(d_{|n|})\}$  and a set of data packets in a data flow can be expressed as  $D_{DF_i} = \{pk_1(D_{pk_1}) \parallel pk_2(D_{pk_1}) \parallel \dots \parallel pk_{|N|}(D_{pk_N})\}$ . Since, each  $d_i \in D$  resides into each  $df_i$  of a  $pk_i$ , the quantity of data fields per  $pk_i$  in a data flow can be calculated as,

$$n = \frac{|D|}{N} \tag{3}$$

However,  $n_i$  originates a set of  $\{udf_1, udf_2, \dots, udf_{|n|}\} \in R^{n \times L}$  by analyzing  $n$  data items for a data packet and  $\{upk_1, upk_2, \dots, upk_{|N|}\} \in R^{N \times M}$  by analyzing  $n * N$  number of data items for a data flow according to the equation number (1) and (2) respectively. Afterward, the MIP framework queries the distinct utilized bits per data packet based on grouping each dissimilar  $udf_i$  and  $udf_j$ , and accounts the number of elements in each group as the frequency of each distinct quantity of utilized bits. Thus,  $n_i$  adopts a Gaussian radial basis function [24, 25] called Gaussian kernel function to form a similarity based clusters between the  $udf_i$  and  $udf_j$  as,

$$k(udf_i, udf_j) = \exp\left(\frac{-\|udf_i - udf_j\|^2}{2 * \sigma^2}\right) \tag{4}$$

$$k(upk_i, upk_j) = \exp\left(\frac{-\|upk_i - upk_j\|^2}{2 * \sigma^2}\right) \tag{5}$$

Where,  $\sigma$  limits the width of Gaussian kernel and  $k(udf_i, udf_j), k(upk_i, upk_j) \in [0, 1]$ . Note that, the similar value as  $udf_i == udf_j$  and  $upk_i == upk_j$  occur, while  $k(udf_i, udf_j) = 1$  and  $k(upk_i, upk_j) = 1$  respectively. Hence, the MIP

method adopts the Gaussian kernel function that defined in the equation number (4) and (5) with  $\sigma = 1.74$  such as [25, 26].

At the end of grouping, a single element from each cluster is selected to be a cluster head. Thus,  $n_i$  generates a set of cluster head from each element of  $udU_{pk_i}$  and  $upU_{DF_i}$  as  $\{dfu_1, dfu_2, \dots, dfu_{|udU_{pk_i}|}\} \in R^{|udU_{pk_i}|}$  and  $\{pku_1, pku_2, \dots, pku_{|upU_{DF_i}|}\} \in R^{|upU_{DF_i}|}$ , respectively.

$$f(dfu_i) = \sum_{udf_i \in dfu_i} 1 = |dfu_i| \quad (6)$$

$$f(pku_i) = \sum_{upk_i \in pku_i} 1 = |pku_i| \quad (7)$$

Yet, the cumulative frequency for each  $dfu_i$  and  $pku_i$  is accrued from its individual cluster according to the equation number (6) and (7) respectively.

Thus, the MIP framework accumulates a set incidence constraints for each distinct quantity of utilized bits per data packet and data flow as  $\{f(dfu_1), f(dfu_2), \dots, f(dfu_{|udF_{pk_i}|})\} \in R^{L \times n}$  and  $\{f(pku_1), f(pku_2), \dots, f(pku_{|upF_{DF_i}|})\} \in R^{M \times N}$ , respectively.

### 4.3 Compute the Product Values and Threshold Parameters

In this section,  $n_i$  initially computes the product between each  $dfu_i \in udU_{pk_i}$  and its corresponding incidence  $f(dfu_i) \in udF_{pk_i}$ , and originates a set of product values as,  $V_{pk_i} = \{vd_1, vd_2, \dots, vd_{|V_{pk_i}|}\}$  per data packet. Hence, each  $vd_i (i = 1, 2, \dots, |V_{pk_i}|)$  can be stated as  $vd_i = dfu_i * f(dfu_i) \in V_{pk_j}$ , where,  $pk_j (j = 1, 2, \dots, N)$ . Each  $vd_i \leq L$  (in bits) is examined to determine whether it is higher (or lower) than threshold  $t_i \in t_{pk_i}$ . In this method, each  $t_i$  is equated as  $t_i = dfu_i * \frac{n}{2}$  that illustrates the memory usage with a certain quantity of utilized bits between a certain number of data fields belong to the data payload of a data packet. However, the number of threshold values in a set of  $t_{pk_i}$  remain as  $|t_{pk_i}| \leq |udU_{pk_i}|$ .

**Remark:** The utilized bits per data packet can be calculated according to the equation number (2), in the proposed framework it also can be measured with the sum of all the  $vd_i$  values per  $pk_i$  as,  $upk_i = \sum_{i=1}^{|V_{pk_i}|} vd_i$ .

At the end of calculating  $vd_i \in V_{pk_i}$  and  $t_i \in t_{pk_i}$  generation process,  $n_i$  computes the product of each  $pku_i \in upU_{DF_i}$  and its corresponding incidence  $f(pku_i) \in upF_{DF_i}$ , and generates a set of  $V_{DF_i} = \{vpk_1, vpk_2, \dots, vpk_{|V_{DF_i}|}\}$  per data flow. Hence, each  $vpk_i (i = 1, 2, \dots, |V_{DF_i}|)$  can be equated as  $vpk_i = pku_i * f(pku_i) \in V_{DF_j}$ , where,  $DF_j (j = 1, 2, \dots, |DF_j|)$ . Each  $vpk_i \leq M$  (in bits) is examined to determine whether it is higher (or lower) than threshold  $t_i \in t_{DF_i}$ . In this case, each  $t_i$  is equated as  $t_i = pku_i * \frac{n}{2}$  that describes the consumed memory usage with a certain quantity of utilized bits between the data payloads belong a certain number of data packets in a data flow. However, the number of threshold values in a set of  $t_{DF_i}$  remain as  $|t_{DF_i}| \leq |upU_{DF_i}|$ .

### 4.4 Manifesting the Data Provenance

At this step, the data source  $n_i$  initially manifest the  $pv_i \in \{0, 1\}$  as consequence of examining each element of  $\{vd_1, vd_2, \dots, vd_{|V_{pk_i}|}\}$  in respect to its corresponding threshold value seized in  $\{t_1, t_2, \dots, t_{|t_{pk_i}|}\}$ . Hence, the tagged data provenance by analyzing each measured  $vd_i \in V_{pk_i}$  can be stated as,

$$pv_i = \begin{cases} 0 & \text{if, } vd_i \leq t_i \in t_{pk_i} \\ 1 & \text{otherwise} \end{cases}$$

Thus, a set of data provenance per data packet is attained as  $PV_{pk_i} = \{pv_1, pv_2, \dots, pv_{|PV_{pk_i}|}\}$ , where the number of provenance bits in its set is  $|PV_{pk_i}| == |V_{pk_i}|$ .

Moreover, the  $n_i$  also examines each element of  $\{vpk_1, vpk_2, \dots, vpk_{|V_{DF_i}|}\}$  in respect to its corresponding threshold value seized in  $\{t_1, t_2, \dots, t_{|t_{DF_i}|}\}$ . Hence, the tagged data provenance by analyzing each measured  $vpk_i \in V_{DF_i}$  can be equated as,

$$pv_i = \begin{cases} 0 & \text{if, } vpk_i \leq t_i \in t_{DF_i} \\ 1 & \text{otherwise} \end{cases}$$

Thus, a set of data provenance for a particular data flow is accrued as,  $PV_{DF_i} = \{pv_1, pv_2, \dots, pv_{|PV_{DF_i}|}\}$  for a particular data stream, where the quantity of provenance bits in its set is  $|PV_{DF_i}| == |V_{DF_i}|$ .

### 4.5 Security Initiatives for Data Provenance

In this section, the MIP framework initiates the security initiatives on the data intend to transmit to other node. Therefore, before initiating the transmission, each is encrypted based on XOR encryption algorithm [27]. A message transformation function *enc* is followed by the MIP method to encrypt each  $d_i$  with each data specific key  $k_i$ , and thus an encrypted form of data message  $\varepsilon(d_i)$  is computed as,

$$\varepsilon(d_i) = enc(d_i; k_i)$$

The MIP framework also follows the one way hash function [28] to generate each data specific secret key  $k_i$  as,

$$k_i = H(i, at_i, CV_{d_i})$$

where  $at_i, i, CV_{d_i}$  are the arrival time, the position index and groupXOR [23] of the counter values correspond to each binary digit of  $i^{th}$  data  $d_i$  respectively.

However, before dealing with the encryption protocol, the proposed scheme initializes the initial counter values for a set of binary digits  $\{b_1, b_2, \dots, b_L\}$  of each single  $d_i \in D$ . Thus,  $n_i$  constructs a  $n \times L$  counter matrix  $CV$  for a data packet.

$$CV = \begin{pmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} & \dots & \lambda_{1L} \\ \lambda_{21} & \lambda_{22} & \lambda_{23} & \dots & \lambda_{2L} \\ \dots & \dots & \dots & \dots & \dots \\ \lambda_{n1} & \lambda_{n2} & \lambda_{n3} & \dots & \lambda_{nL} \end{pmatrix}$$

where,

$$CV_{d_i} = \text{groupXOR}(\lambda_{11} \oplus \lambda_{12} \oplus \dots \oplus \lambda_{1L})$$

Here,  $\{\lambda_{11}, \lambda_{12}, \dots, \lambda_{1L}\} \in R^{n \times L}$  is a set of counter values for any  $i^{th}$  data  $d_i \in D$ , and each vector  $\lambda_{ij}$  denotes an initialized counter value for each  $b_j$  belongs to the binary value 0 or 1.

Afterward, the encryption operation performed in a manner, where the index of  $k_i$  and  $d_j$  should be the same as,  $i = j$ . Furthermore, the SN will regenerate  $k_i$  as  $k_i^{new}$  with a reciprocal relation between the received  $\widehat{k}_i$  and the  $CV_{d_i}$ , which is a key synchronization policy to determine whether it does match or not. If it does match, the status will be regarded as the counter values belong to the binary sequence of  $i^{th}$  received data have not been altered.

#### 4.6 Data Decryption and Provenance Identification

In this section, the SN performs a decryption operation on the received encrypted data set per packet  $\varepsilon(\widehat{D}) = \{\varepsilon(\widehat{d}_1), \varepsilon(\widehat{d}_2), \dots, \varepsilon(\widehat{d}_n)\}$  with a set of pre-shared data specific keys  $\{\widehat{k}_1, \widehat{k}_2, \dots, \widehat{k}_n\}$ . However, the decrypt function  $dec$  uses each  $\widehat{k}_i$  as an input for each  $\varepsilon(\widehat{d}_i)$  to recover decrypted sensor data  $\mathcal{D}(\widehat{d}_i)$  as,

$$\mathcal{D}(\widehat{d}_i) = \text{dec}(\varepsilon(\widehat{d}_i); \widehat{k}_i)$$

However, the MIP framework initiates two steps provenance identification procedure for a particular received data set  $\widehat{D} = \{\widehat{d}_1, \widehat{d}_2, \dots, \widehat{d}_{|\widehat{D}|}\}$  that obtained from several received data packets in a flow:

1. Validating the integrity of each bits of  $\widehat{d}_i \in \widehat{D}$  based on the counter values.
2. Identifying provenance with a regression analysis.

**Validating the Integrity of Received Data Based on the Counter Values:** In this process, the SN initially forms a self-generated counter matrix  $\widehat{CV}$  following the same procedure of  $n_i$  according to the dimension of  $n \times L$  or  $(n * L) \times L$  according to the number of data fields of a data packet  $pk_i$  or  $DF_i$ . Thus, a set of counter values for any  $i^{th}$  data  $\widehat{d}_i \in \widehat{D}$  can be stated as,  $\{\widehat{\lambda}_{11}, \widehat{\lambda}_{12}, \dots, \widehat{\lambda}_{1L}\} \in R^{n \times L}$ , where each vector  $\widehat{\lambda}_{ij}$  is initialized with pre-shared counter value that belongs to each binary digit  $\widehat{b}_j$  (0 or 1) of  $\widehat{d}_i$ . Additionally, the SN computes the groupXOR of all the counter values that correspond to each  $\widehat{b}_j$  of  $i^{th}$  data  $\widehat{d}_i$  as,

$$\widehat{CV}_{\widehat{d}_i} = \text{groupXOR}(\widehat{\lambda}_{11} \oplus \widehat{\lambda}_{12} \oplus \dots \oplus \widehat{\lambda}_{1L})$$

However, to examine the integrity of received  $\widehat{d}_i$ , the SN perform a key synchronization procedure, where each  $\widehat{d}_i \in \widehat{D}$  specific secret key  $\widehat{k}_i$  is regenerated as  $k_i^{new}$  and determines whether they matched or not. The SN follows the same

approach that  $n_i$  has been followed, and thus, generating  $k_i^{new}$  also includes position index  $i$  of each  $\widehat{d}_i$  in a particular received data set, its pre-share arrival time  $\widehat{at}_i$  and  $\widehat{CV}_{\widehat{d}_i}$ . If the above components are matched appropriately, SN regenerated  $k_i^{new}$  will be synchronized correctly to its conformed  $\widehat{k}_i$  as,  $k_i^{new} == \widehat{k}_i$  that indicates the pre-shared information and the transmitted data have not been altered in mid-transmission.

**Regression Based Provenance Identification:** In this section, the provenance identification has been evaluated with a logistic regression [29–31] approach, where the probability of predicted  $pv_i$  is related to a set of explanatory variable ( $\widehat{dfu}_i, f(\widehat{dfu}_i)$ ) or ( $\widehat{pku}_i, f(\widehat{pku}_i)$ ). Therefore, at the end of data decryption and its integrity confirmation, the SN originates a set of utilized bits  $\widehat{udf}_{pk_i} = \{\widehat{udf}_1, \widehat{udf}_2, \dots, \widehat{udf}_n\} \in R^{n \times L}$  per received  $pk_i$  and  $\widehat{upk}_{DF_i} = \{\widehat{upk}_1, \widehat{upk}_2, \dots, \widehat{upk}_N\} \in R^{N \times M}$  per  $DF_i$  according to the equation number (1) and 2 respectively. Afterward, each distinct quantity of utilized bits is classified from the above two sets as  $\{\widehat{dfu}_1, \widehat{dfu}_2, \dots, \widehat{dfu}_{|\widehat{udF}_{pk_i}|}\} \in R^{|\widehat{udU}_{pk_i}|}$  and  $\{\widehat{pku}_1, \widehat{pku}_2, \dots, \widehat{pku}_{|\widehat{upU}_{DF_i}|}\} \in R^{|\widehat{upU}_{DF_i}|}$  according to the equation number (4) and (5) correspondingly. Finally, SN accumulates the  $\widehat{udF}_{pk_i} = \{f(\widehat{dfu}_1), f(\widehat{dfu}_2), \dots, f(\widehat{dfu}_{|\widehat{udF}_{pk_i}|})\}$  according to the equation number (6) and  $\widehat{upF}_{DF_i} = \{f(\widehat{pku}_1), f(\widehat{pku}_2), \dots, f(\widehat{pku}_{|\widehat{upF}_{DF_i}|})\}$  according to the equation number (7) by probing the element of  $\widehat{udU}_{pk_i}$  and  $\widehat{upU}_{DF_i}$  respectively.

However, the MIP framework arrange a data set that is composed of all the collected ( $\widehat{dfu}_i, f(\widehat{dfu}_i)$ ), ( $\widehat{pku}_i, f(\widehat{pku}_i)$ ) along with some training samples. Each sample is initialized as a set of feature inputs  $X$ , where  $X = x_i (i = 1, 2, \dots, P)$ . Here,  $x_i$  and  $x_{i+1}$  correspond to explanatory variables  $\widehat{dfu}_i$  and  $f(\widehat{dfu}_i)$  or  $\widehat{pku}_i$  and  $f(\widehat{pku}_i)$ . Hence, each  $x_i$  is examined to predict the corresponding  $pv_i \in \widehat{PV}_{pk_i}$  or  $pv_i \in \widehat{PV}_{DF_i}$ , which have been pre-shared to SN to determine utilization status as a tagged binary value.

The desired  $pv_i \in \widehat{PV}_{pk_i}$  and  $pv_i \in \widehat{PV}_{DF_i}$  is identified as  $y_{pv_i}$  to probabilities, and thus, the logistic function [32] or sigmoid function specified as,  $y_{pv_i} = \sigma(z) = \frac{1}{1+e^{-z}}$  where the result  $y_{pv_i}$  of  $f(z)$  the value in between [0, 1]. The value of  $y_{pv_i}$  mapped to predicted  $pv_i$  either 0 and 1 based on a certain threshold value  $T = 0.5$  [19, 20]. Hence, the hypothetical approach of  $\sigma(z)$  can be defined as,

$$h_\beta(x) = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-\beta^T X}} \tag{8}$$

where,  $\beta$  represents the logistic regression coefficients [21, 22, 33] and it will be updated iteratively for each  $(x_i, y_{pv_i})$  as,

$$\beta = \beta + \alpha \nabla_\beta \ln P(Y | X; \beta) \tag{9}$$

Here, updating  $\beta$  depends on the learning rate  $\alpha$  and partial derivative of log likelihood of  $\beta$ . The value of  $\sigma(z) \rightarrow 1$  if  $z \rightarrow \infty$  from  $T$  (i.e.,  $z \geq T$ ) and

$\sigma(z) \rightarrow 0$  if  $z \rightarrow -\infty$  from  $T$ . Hence, a first order derivative of  $\sigma(z)$  minimizes the error and forms the function to find an optimal values as,

$$\sigma'(z) = \frac{e^{-z}}{(1 + e^{-z})^2} = \frac{1}{1 + e^{-z}} * (1 - \frac{1}{1 + e^{-z}}) = \sigma(z) * (1 - \sigma(z))$$

So the conditional distribution of data provenance with a prediction function can be written as,

$$P(Y | X) = h_\beta(X)^{y_{pv_i}} (1 - h_\beta(X))^{1-y_{pv_i}} \tag{10}$$

where,  $P(Y | X) = h_\beta(X)^{y_{pv_i}}$  while,  $Y = y_{pv_i} = 1$  and  $P(Y | X) = (1 - h_\beta(X))^{1-y_{pv_i}}$ , while,  $Y = 0$ . Assume the estimate value of the probability  $\varphi(y_{pv_i} | X)$  is  $P_{y_{pv_i}}(X; \beta)$ , where the values of  $\beta, \beta_i (i = 1, 2, \dots, Q)$  maximizes the equation number (10).

However, a maximum likelihood function [34] to determine the optimal values of  $\beta$  can be specified as,

$$\begin{aligned} L(\beta) &= P(X; \beta) = \prod_{i=1}^Q P(y_{pv_i} | x_i, \beta) \\ &= \prod_{i=1}^Q h_\beta(x_i)^{y_{pv_i}} (1 - h_\beta(x_i))^{1-y_{pv_i}} \end{aligned} \tag{11}$$

Hence, to make the computation simpler, the cost function takes the a log likelihood [8] of the equation number (11) as,

$$\begin{aligned} \ln(L(\beta)) &= \ln P(Y | X; \beta) \\ &= \sum_{i=1}^Q y_{pv_i} \ln(h_\beta(x_i)) + (1 - y_{pv_i}) \ln(1 - h_\beta(x_i)) \end{aligned} \tag{12}$$

However, computing the value of  $\beta$  entails a partial derivative of the equation number (12) with respect to  $\beta$  as,

$$\begin{aligned} \frac{\delta}{\delta \beta} (\ln(\beta)) &= \frac{\delta}{\delta \beta} (y_{pv_i} \ln(h_\beta(x_i))) + \frac{\delta}{\delta \beta} ((1 - y_{pv_i}) \ln(1 - h_\beta(x_i))) \\ &= \frac{\delta}{\delta \beta} (y_{pv_i} \ln(\frac{1}{\sigma(\beta^T x_i)})) + \frac{\delta}{\delta \beta} ((1 - y_{pv_i}) \ln(1 - \frac{1}{\sigma(\beta^T x_i)})) \\ &= (y_{pv_i} \frac{1}{\sigma(\beta^T x_i)}) - (1 - y_{pv_i}) (\frac{1}{1 - \sigma(\beta^T x_i)}) (\frac{\delta}{\delta \beta_j} (\sigma(\beta^T x_i))) \\ &= (y_{pv_i} (1 - \sigma(\beta^T x_i)) - (1 - y_{pv_i}) (\sigma(\beta^T x_i))) x_j \\ &= (y_{pv_i} - \sigma(\beta^T x_i)) x_j \\ &= (y_{pv_i} - h_\beta(x_i)) x_j \end{aligned} \tag{13}$$

where, the substitution of  $h_\beta(x_i)$  follows the equation number (8).

Accordingly, by substituting the value from the equation number (13) into the equation number (9), the value of  $\beta_i$  for each training vector  $(x_i, y_{pv_i})$  is

updated by following an stochastic gradient descent [35] approach according to equation number (14).

$$\beta_j = \beta_i + \alpha(y_{pv_i} - h_\beta(x_i))x_j \tag{14}$$

Here, the distribution of  $P(1|X) = h_\beta(X)^{y_{pv_i}}$  and  $P(0|X) = (1 - h_\beta(X))^{y_{pv_i}}$  are estimated from the statistic of  $S$  and  $S'$  respectively. Hence, the data provenance  $y_{pv_i} = 1$  along with pragmatic samples are accrued in  $S$ , and the samples with those are not along with, accrued in  $S'$ . Thus, the prediction accuracy for the samples that indicate the tagged provenance  $y_{pv_i} = 0$  can still be performed as  $\frac{|S'|}{|S|+|S'|}$ .

## 5 Performance Evaluation and Security Analysis

### 5.1 Experimental Result

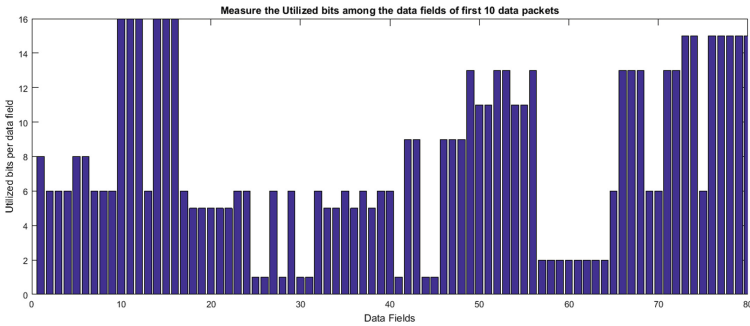
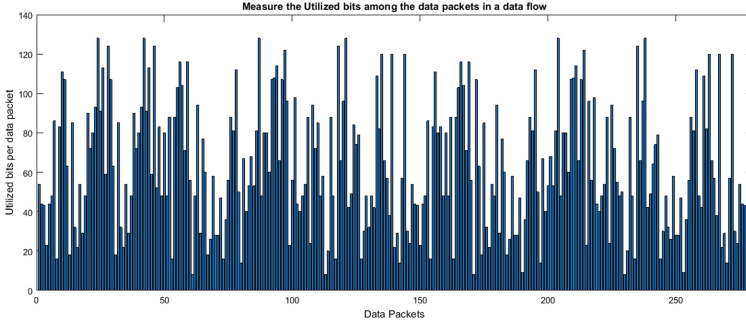


Fig. 5. Measured utilized bits among the data fields

In this experiment, several data flows have been originated that include several numbers of data packets based on numerous volumes of sensor data set. However, a particular data flow with 280 data packets has been analyzed to determine the performance of this scheme. Figure 5 depicts the several quantities of utilized bits between the  $8 * 10 = 80$  data fields for first 10 data packets that accumulated according to the equation number (1), where the quantity of utilized bits in any  $df_i$  has been observed between  $[0, 16] \leq$  in bits. Moreover, numerous lengths of utilized bits among the data packets in this particular flow also have been observed in Fig. 6. Here, each  $upk_i$  has been accumulated according to the equation number (2) and the quantity of utilized bits in any  $pk_i$  has been observed between  $[0, 128] \leq M$  in bits.





**Fig. 6.** Measured utilized bits among the data packets

However, the number of provenance bits in this scheme can be accounted for according to the number of threshold values that have been initiated by a data source to quantify memory utilization. The MIP framework computes the threshold values based on the distinct quantity of utilized bits and its frequency in each extent of provenance origination scopes. It has been observed in Fig. 5 that the measured utilized bits from the data fields  $df_{57}$  to  $df_{64}$  (belong to  $pk_8$  in Fig. 6) are same that indicates a single  $dfu_i$  with  $n$  incidence between the data fields of a  $pk_i$ . Moreover, each single  $df_i$  of a  $pk_i$  could attain  $n$  distinct  $udf_i$  values, then the frequency for each single  $dfu_i$  is 1. Likewise, a variation of threshold and provenance quantity between the  $N$  data packets in a data flow also might be in a range between  $[N - (N - 1), N]$  based on the number of occurrences of each  $pku_i$ .

Thus, according to the procedure of defining data provenance based on the threshold value, a minimum and maximum number of elements in each set of  $PV_{pk_i}$  and  $t_{pk_i}$  can be calculated as,

$$|PV_{pk_i}| = |t_{pk_i}| = \begin{cases} 1^{(min.)} & \text{if } f(dfu_i)=n, \text{ where } udf_i===udf_j \\ n^{(max.)} & \text{if } f(dfu_i)=1, \text{ where } udf_i \neq udf_j \end{cases}$$

However, assuming the similar procedure of tagging the data provenance based on the several utilization thresholds among the data packet per data flow, a minimum and maximum number of elements in each set of  $PV_{DF_i}$  and  $t_{DF_i}$  can be specified as,

$$|PV_{DF_i}| = |t_{DF_i}| = \begin{cases} 1^{(min.)} & \text{if } f(pku_i)=N, \text{ where } upk_i===upk_j \\ N^{(max.)} & \text{if } f(pku_i)=1, \text{ where } upk_i \neq upk_j \end{cases}$$

Hence, it can be stated from the above facts that the variation ratio of provenance and threshold quantity is proportional to the frequency uniqueness of a distinct quantity of utilized bits, and the above properties of the MIP framework signify the scalability and adaptability of this scheme.

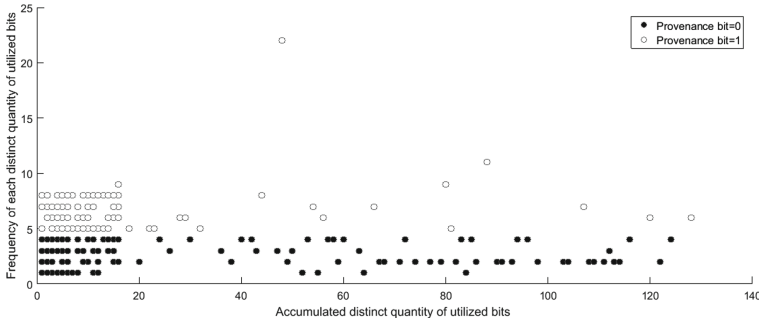


Fig. 7. Plotted provenance based on the accrued collective distinct utilized bits and their relative frequency among the data fields and the data packets in a flow

However, it is very reasonable that there are many sensor environments, where the transmitted data varies in a range of values. Although the memory for the data is always constrained to the Definition 1, the utilized bits for each sensor data might vary in several ranges. Thus, any random length of utilized bits between a particular data set can be examined to determine the usage within the data payload, and the usage status can also be defined as labeled data provenance. Moreover, the illustration indicates the quantity of data provenance also associated with enlarging the size of data payload and number of segmented data fields.

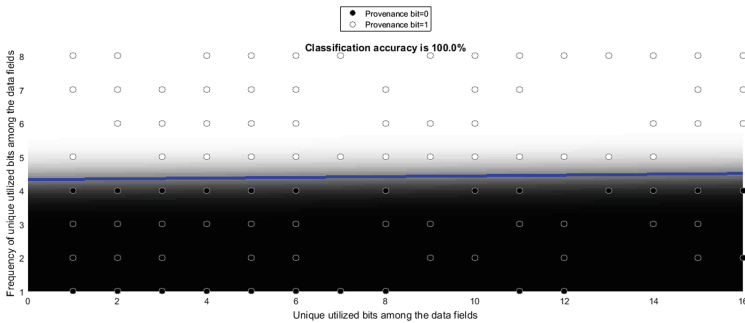
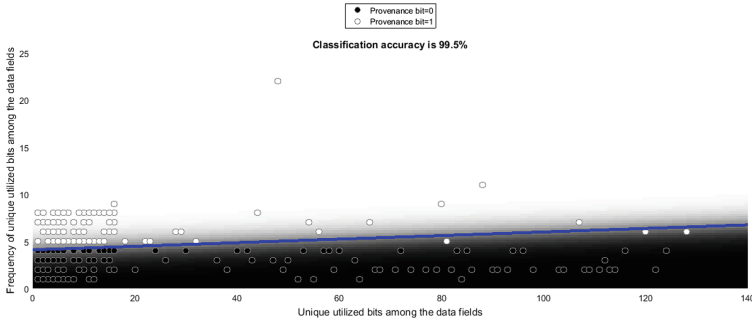


Fig. 8. Decision boundary and accuracy on classifying the provenance bits that initiated on memory usage among the data fields only

Figure 7 demonstrates the plotted provenance bits based on the distinct utilized bits and their relative frequency that have been observed among the data fields and the packets from the aforesaid data stream. The decision boundary along with the classification accuracy on distinguishing a binary class of data provenance has shown in Fig. 8 and 9. In Fig. 8, the provenance classification accuracy illustrated for the samples accumulated, where the projected distinct

utilized bits and their frequency have been observed between the data fields of all the data packets in the data flow.

However, the classification accuracy in Fig. 8 shows 0.5% higher than the accuracy observed on a collective set of samples illustrated in Fig. 9 that includes estimated distinct utilized bits and their frequency among the data fields and data packets together. Since, the discrimination removal cost reduces the prediction accuracy [36], increasing the maximum possible correct training samples corresponding  $pk_{u_i}$  and  $f(pk_{u_i})$  in the set of samples that used in Fig. 9 can increase the maximum possible accuracy.



**Fig. 9.** Decision boundary and accuracy on classifying the provenance bits that initiated on memory usage among the data fields and the data packets both

## 6 Discussion

An analysis has been evaluated on the capacity of provisioning data provenance per node and compared with the inter-packet delay (IPD) [9] approach. This scheme was proposed to ensure the security of each sensor data flow rather than concerning the integrity data. In this method, each node whether it is a data source or aggregator in a route path, can encode PN (pseudo-noise) sequence into the consecutive delays between the data packets. Thus, the base station (BS) can trace all the traversed started from the data source over the travel path for the received data packets. In this methodology, each node requires  $N + 1$  data packets to encode  $N$  bits node identity as a data provenance, where the first data packet  $pk_i$  doesn't introduce any delay. Assume that a set of inter-packet delays  $\{\Delta_1, \Delta_2, \dots, \Delta_N\}$  has originated to transmit  $N$  bits provenance, where each  $\Delta_i$  signifies the delay between  $i^{th}$  and  $(i + 1)^{th}$  packet and a single bit of each node's identity is added to it. Hence, a predetermined delay perturbation  $v_i[j]$  is added to general delay  $\Delta_i$ , where each  $v_i[j]$  is composed of a normally distributed real number and each bit of PN values. Moreover, the generation of  $v_i[j]$  can be done offline. On the contrary, the proposed protocol allocates the data provenance  $pv_i$  as labelled information on payload memory utilization, and reports its presence by exploring two correlated scopes:  $n$  number of  $df_i$  per  $pk_i$

and  $N$  number of  $pk_i$  per  $DF_i$  according to the Definition 2 and Definition 3 based on the notion of Definition 1.

Since, the distinct quantity of utilized bits accumulated in  $udU_{pk_i}$  and  $upU_{DF_i}$  and the frequency of each element of these two sets are the key properties, the number of provenance bits is equivalent to the volume of these two sets. Thus, calculating the quantity of  $pv_i$  among the  $df_i$  per  $pk_i$  can be stated as,  $|PV_{pk_i}| = \sum_{i=1}^N \sum_{j=1, pv_i \in PV_{pk_i}}^{|udU_{pk_i}|} 1$  and the provenance among the  $pk_i$  per  $DF_i$  calculated as,  $|PV_{DF_i}| = \sum_{pv_i \in PV_{DF_i}} 1$ . Table 2 evaluates the provenance encoding efficiency between the schemes based on IPD and MIP schemes. It shows an increased volume in the MIP approach compared to the IPD scheme.

**Table 2.** Provenance capacity for a particular data set with  $N$  number of packets

Scheme	Number of data flow	Provenance capacity(in bits)
The MIP scheme	1	$N + 1(min.)$ and $(n * N) + N(max.)$
The IPD scheme	1	$N - 1$

Moreover, since the network transmission protocols are not designed to add delays between the data packets, some usual packet transmission events e.g., packet loss, packet aggregation, etc, can restrict the IPD channels. The proposed method doesn't deal with altering any parameter related to network (e.g., delay, size of payload or data fields) rather observing the payload memory consumption of sensor data packet based on analyzing the utilized bits. Hence, decoding error or the detection of an adversary threat can be identified by the SN originated groupXOR of counter values for the received data and validating it through the key synchronization process. Moreover, computing the utilized bits for a particular data set and tagging the data provenance can also be prepared in offline with the knowledge of predetermined payload properties (e.g., length or data type of each  $df_i$ , the quantity of segmented  $df_i$ ) per data packet. Meanwhile, the generation of secret keys includes additional computation cost to proposed scheme.

Assuring the trustworthiness of sensor data with or without interfering with the sensor data has already been explored. Hence, the proposed scheme may conduct a broader range of data streams to initiates a variable length of data provenance per data packet and data flow without tempering the data or transmission channel parameter. In this scheme, the initiated data provenance signifies the utilization of data payload of a data transport being carrying the data only. In this method, any alteration on data can be detected by validating the counter table using the pre-shared initial counter value at the SN for the received data per packet. Moreover, the altered data also can be recovered to attain more accuracy in determining valid provenance. However, the provenance identification and analyzing data separately also implied a reversible technique [37] data security. To apply the MIP scheme in sensor environments, a secure mechanism

is required to distribute the secret keys, the initial counter values, timestamps of data and the source node initiated provenance to the designated recipient.

## 7 Conclusions

In this paper, a novel problem of manifesting a noiseless data provenance has been addressed, where provisioning provenance neither disrupts the data nor the transmission channel in the wireless sensor network. Though the proposed method exhibits the provenance as a status based on the payload memory consumption of a data packet, it illustrates the security of sensor data and data flow through a hierarchical relation between the data fields per packet and the data packets per data flow.

Experiment outcomes have been conducted to the provenance classification accuracy besides provenance encoding capacity, which confirms the scalability of this scheme. The proposed method is semantically secure as long as the underlying data encryption mechanism is semantically secure. Moreover, the initiation of counter values to protect the binary sequence of data and its integrity confirmation approach not only makes the security properties resilient against adversary acts but also confirms the integrity of data.

The proposed method could extend its robustness by varying the threshold values as a memory usage indicator and adopting suitable learning methods in identifying the provenance. Additionally, it opens the prospect of securing data stream based on the provenance concerning the utilized bits of the original form of data rather than its other form.

## References

1. Maglaras, L., Ferrag, M.A., Derhab, A., et al.: Threats, protection and attribution of cyber attacks on critical infrastructures. arXiv preprint [arXiv:190103899](https://arxiv.org/abs/190103899) (2019)
2. Anderson, A.: Effective management of information security and privacy. *Educause Q.* **29**(1), 15–20 (2006)
3. Tan, Y.S., Ko, R.K.L., Holmes, G.: Security and data accountability in distributed systems: a provenance survey. In: *IEEE International Conference on Embedded and Ubiquitous Computing*, pp. 1571–1578 (2013)
4. Kundur, D., Luh, W., Okorafor, U.N., et al.: Security and privacy for distributed multimedia sensor networks. *Proc. IEEE* **96**(1), 112–130 (2007)
5. Houmansadr, A., Kiyavash, N., Borisov, N.: Multi-flow attack resistant watermarks for network flows. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1497–1500. IEEE (2009)
6. Cabuk, S., Brodley, C.E., Shields, C.: Ip covert timing channels: design and detection. In: *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pp. 178–187. ACM (2004)
7. Wang, X., Chen, S., Jajodia, S.: Network flow watermarking attack on low-latency anonymous communication systems. In: *IEEE Symposium on Security and Privacy (SP'07)*, pp. 116–130. IEEE (2007)

8. Wang, X., Reeves, D.S.: Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 20–29. ACM (2003)
9. Sultana, S., Shehab, M., Bertino, E.: Secure provenance transmission for streaming data. *IEEE Trans. Knowl. Data Eng.* **25**(8), 1890–1903 (2012)
10. Lim, H.-S., Moon, Y.-S., Bertino, E.: Provenance-based trustworthiness assessment in sensor networks. In: Proceedings of the Seventh International Workshop on Data Management for Sensor Networks, pp. 2–7. ACM (2010)
11. Becker, R.A., Chambers, J.M.: Auditing of data analyses. *SIAM J. Sci. Stat. Comput.* **9**(4), 747–760 (1988)
12. Kamel, I.: A schema for protecting the integrity of databases. *Comput. Secur.* **28**(7), 698–709 (2009)
13. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In: Proceedings of the Twenty-Sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 31–40 (2007)
14. CTan, W.: Provenance in database: past, current and future. *IEEE Data Eng. Bull.* **30**, 3–13 (2007)
15. Zhang, O.Q., Kirchberg, M., Ko, R.K., et al.: How to track your data: the case for cloud computing provenance. In: 2011 IEEE Third International Conference on Cloud Computing Technology and Science, pp. 446–453. IEEE (2011)
16. Sultana, S., Ghinita, G., Bertino, E., et al.: A lightweight secure provenance scheme for wireless sensor networks. In: 2012 IEEE 18th International Conference on Parallel and Distributed Systems, pp. 101–108. IEEE (2012)
17. Wang, C., Hussain, S.R., Bertino, E.: Dictionary based secure provenance compression for wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **27**(2), 405–418 (2015)
18. Hasan, R., Sion, R., Winslett, M.: The case of the fake picasso: preventing history forgery with secure provenance. In: Proceedings of the Conference on File and Storage Technologies (FAST)
19. Chong, S., Skalka, C., Vaughan, J.A.: Self-identifying sensor data. In: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, pp. 82–93. ACM (2010)
20. Alam, S.I., Fahmy, S.: A practical approach for provenance transmission in wireless sensor networks. *Ad Hoc Netw.* **16**, 28–45 (2014)
21. Rothenberg, C.E., Macapuna, C.A.B., Magalhães, M.F., et al.: In-packet bloom filters: Design and networking applications. *Comput. Netw.* **55**(6), 1364–1378 (2011)
22. Zhang, Q., Zhou, X., Yang, F., Li, X.: Contact-based traceback in wireless sensor networks. In: 2007 International Conference on Wireless Communications, Networking and Mobile Computing, pp. 2487–2490 (2007)
23. Sun, X., Su, J., Wang, B., et al.: Digital watermarking method for data integrity protection in wireless sensor networks. *Int. J. Secur. Appl.* **7**(4), 407–416 (2013)
24. Liu, Z., Zuo, M.J., Xu, H.: A Gaussian radial basis function based feature selection algorithm. In: 2011 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSAP) Proceedings, pp. 1–4. IEEE (2011)
25. Ghaddar, A., Razafindralambo, T., Simplot-Ryl, I.: Algorithm for data similarity measurements to reduce data redundancy in wireless sensor networks. In: 2010 IEEE International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM), pp. 1–6. IEEE (2010)

26. Scott, D.W.: *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, New York (2015)
27. Huo, F., Gong, G.: Xor encryption versus phase encryption, an in-depth analysis. *IEEE Trans. Electromagn. Compat.* **57**(4), 903–911 (2015)
28. Jizhi, W., Shujiang, X., Min, T., et al.: The analysis for a chaos-based one-way hash algorithm. In: *2010 International Conference on Electrical and Control Engineering*, pp. 4790–4793. IEEE (2010)
29. Liu, L., Luo, G., Qin, K., et al.: An algorithm based on logistic regression with data fusion in wireless sensor networks. *EURASIP J. Wirel. Commun. Networking* **2017**(1), 10 (2017). <https://doi.org/10.1186/s13638-016-0793-z>
30. Schumacher, M., Roßner, R., Vach, W.: *Neural networks and logistic regression: Part i*. *Comput. Stat. Data Anal.* **21**(6), 661–682 (1996)
31. Hosmer Jr., D.W., Lemeshow, S., Sturdivant, R.X.: *Applied Logistic Regression*, vol. 398. Wiley, New York (2013)
32. Ansong, M.O., Yao, H.-X., Huang, J.S.: Radial and sigmoid basis function neural networks in wireless sensor routing topology control in underground mine rescue operation based on particle swarm optimization. *Int. J. Distrib. Sensor Networks* **9**(9), 376931 (2013)
33. Archibald, R., Ghosal, D.: A covert timing channel based on fountain codes. In: *IEEE 11th International Conference on trust, Security and Privacy in Computing and Communications*, pp. 970–977. IEEE (2012)
34. Korkmaz, M., Güney, S., Yigîter, Y.: The important of logistic regression implementation in the turkish livestock sector and logistic regression implementations or fields. *J. Agric. Fac. HRU.* **16**(2), 25–36 (2012)
35. Cohen, K., Nedić, A., Srikant, R.: On projected stochastic gradient descent algorithm with weighted averaging for least squares regression. *IEEE Trans. Autom. Control* **62**(11), 5974–5981 (2017)
36. Kamiran, F., Calders, T., Pechenizkiy, M.: Discrimination aware decision tree learning. In: *2010 IEEE International Conference on Data Mining*, pp. 869–874. IEEE (2010)
37. An, L., Gao, X., Li, X., et al.: Robust reversible watermarking via clustering and enhanced pixel-wise masking. *IEEE Trans. Image Process.* **21**(8), 3598–3611 (2012)



# Tightly Close It, Robustly Secure It: Key-Based Lightweight Process for Propping up Encryption Techniques

Muhammed Jassem Al-Muhammed<sup>1</sup>(✉), Ahmad Al-Daraiseh<sup>1</sup>,  
and Raed Abuzitar<sup>2</sup>

<sup>1</sup> Faculty of Information Technology, American University of Madaba,  
Madaba, Jordan

{m.almuhammed, a.daraiseh}@aum.edu.jo

<sup>2</sup> College of Engineering and Information Technology, Ajman University,  
Ajman, United Arab Emirates  
r.abuzitar@ajman.ac.ae

**Abstract.** Securing invaluable information has been, and will be, the highest priority whether for individuals or organizations. Researchers are working diligently to meet this priority by offering different types of protection techniques. The encryption techniques stand out as de-facto mechanisms for ensuring proper protection for information. Many encryption techniques are available that have passed basic security tests and ensure reasonable levels of protection. The greatest challenge to these techniques is the formidably-ever-advancing cryptanalysis tools. Given this real challenge, we believe that these encryption techniques will sooner or later face the same destiny as other techniques (e.g. DES). That is, unless we keep boosting their capabilities, these techniques may fail to resist the tricky cryptanalysis tools, offering perfect opportunity for privacy-intruding lovers to threaten the information's privacy. This paper addresses this problem by offering a specific way. In particular, it proposes a closing stage that forms an additional (and highly effective) line of defense against security attacks by concealing the final output of the encryption techniques in highly random and enormously complicated codes. This method can be integrated with any encryption technique as a final stage to increase its resistance against cryptanalysis tools. The proposed method is implemented and subjected to rigorous security testing. These tests showed that the method provides very effective camouflaging mechanisms to hide data.

**Keywords:** Encryption techniques · Bolster encryption methods  
security · Key-echo generation · Hiding exploitable patterns

## 1 Introduction

Despite the fact that users nowadays enjoy the easiness of accessing and posting information than any time ever, they have legitimate worries about their



sensitive information. It is true that the digital technologies have simplified the information dissemination, but do expose this information to highly hostile environment. To leverage the blessings of the digital world, it is essential to come up with protection techniques that effectively beat the hostility of the digital world and reassures the users about the privacy of their information. Encryption techniques provide effective protection mechanisms [1–14, 19–22]. These techniques seem—based on the security testing that they passed—to offer a reasonable protection for information and thus mitigate the users’ worries.

Although encryption techniques adopt different computational models to hide the information, they all transform their inputs in many stages with the aim that the output becomes too confusing to be predicted. In spite of that each transformation stage contributes to the overall security of the information, the closing stage is typically of a special importance. It should firmly seal the output and eliminate any indicative exploitable patterns; otherwise the security of the information is likely to be compromised [23, 24]. Effective closing stage should provide means for burying the output of the previous stages in enormously complex codes that not only boost the randomness of the output, but also absorb any potential patterns that may be left by the previous stages. If this closing stage is properly implemented, it will eliminate the leaky and vulnerable points that adversaries can possibly exploit.

This paper offers a specific way to implement a highly effective closing stage that can be incorporated as a final stage in any encryption technique to provide a powerful line of defense. Specifically, the proposed method intercepts the final output of the encryption technique (ciphertext) and greatly conceals it in an enormously complicated codes. Our method fundamentally depends on the encryption key without compromising its security. It adopts an innovative computational model that relies on fuzzy distortion and mapping operators to create key echoes from the encryption key—a sequence of symbols whose relation to the original keys is greatly diminished. These sequences of codes are deeply processed using the fuzzy distortion and complicated mapping operators to ensure two objectives. First, this deep processing hides the traces of the original key and therefore preserves its security. Second, it ensures the randomness of the key echo and hence yields camouflaging codes that show no patterns from one hand and dissipate the patterns in the output of the encryption technique from the other hand.

To this end, the contribution of this paper is that it offers an effective way to secure the output of encryption methods; thereby providing an additional and powerful layer of defense against cryptanalysis techniques. This method is light-weight process that incurs no significant time on the encryption process.

We present our contributions as follows. Section 2 briefly introduces the major components of our proposed solution. Section 3 discusses the random generator. Sections 4 and 5 deeply discuss the technical details of the key echo generator and camouflaging code generator. We test the performance of our system in Sect. 6. We conclude and give directions for future work in Sect. 7.

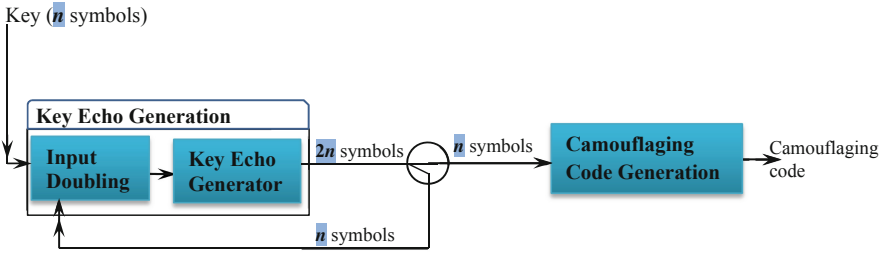


Fig. 1. The system overview.

## 2 System Architecture

Figure 1 highlights the two major components of our system. The Key Echo Generator initially receives the encryption key ( $n$  symbols) and expands this sequence to  $2n$  symbols. The output of the echo generator is split out into two sequences each with  $n$  symbols. The first sequence is directed as an input to the echo generator for producing more  $2n$  sequences and the second sequence is directed as an input for camouflaging code generation. Both components use different fuzzy computational models to accomplish their tasks. The following sections describe the details of these two components.

## 3 Key-Based Random Generator

Fundamental to our approach is the random generator. The random generator produces sequences of random numbers that support the functionality of several operations and influence their performance. As we will see next, the performance of the different operations greatly relies on the quality of the random number sequences. Accordingly, we chose the random number generator reported in [25] because it has very powerful properties that make it suitable. First, it has a long period: generates really long sequences without repeating the same sequence. Second, as reported in [25], it passed important randomness tests. Third, the sequences that are generated using different seeds (encryption keys) are not correlated. Fourth, it is very efficient: high speed with minimal memory requirements. Fifth, the generator possesses a fundamental property that makes it highly effective in the security field: it is highly sensitive to the changes of the key regardless whether the change is in a single bit or more. Figure 2 shows the pseudo-code for the random generator (excerpted from [25]).

In step 1, the generator extends the key to 64 symbols so that the period of the generator increases. The key expansion method uses two operations: *Substitute* operation and *Manipulate* operation. These two operations as described in [25] can take any sequence and extend it into arbitrary length. Steps (2)–(7) define the major functionality of the generator. Steps (2)–(4) apply substitution, flipping, and shifting operations to prepare a seed for the next steps. Steps (5) and (6) sum up the symbols of the resulting seed by multiplying the integer

---

```

Input: Key
Output: sequence of key-based numbers with an arbitrary length.
1. Extend the key to 64 symbols to get the Seed.
2. Seed = Substitute (Seed)
3. Seed = FlipR (Seed, n, m) /*flip the right n bits in the symbol Seed[m]*/
4. Seed = ShiftL (Seed, k) /* circular left shift symbols of Seed k positions*/
5. For i=1 to |Seed| /*|Seed| is the length of Seed*/
6. SUM += 256|Seed|-i × (INT) Seed[i] /*Seed[i] is the symbol at index i*/
7. RAND = SUM mod P /*mod is module operation and P is the range of the numbers*/
8. If More numbers needed, GO TO 2

```

---

**Fig. 2.** The algorithmic steps for the key-based number generator.

value of the seed's symbol at index  $i$  with the power of 256. The reason for taking the power of 256 is that radix for the symbols is 0 to 255 is 256. Therefore, this summation never yields the same Sum value for different seeds. (Interested readers may refer elsewhere [25] for more details.)

## 4 Key Echo Generation

The key echo generation takes the initial encryption key as an input and applies effective substitution and distortion operations to its input to produce sequences of an arbitrary length, called the key echo. Two routines support the functionality of this operation. First, the operation makes use of the input doubling routine, which generates sequences of symbols by doubling its input. Second it makes use of the key echo generator, which imposes deeper manipulations to the output of the first routine. Both routines work synergistically to highly cut any relationship between the encryption key and the final output (the key echo).

Before we discuss these two routines (input doubling and key echo generator), we introduce the mapping table, which is used in the subsequent sections. We also define a substitution operation that uses the mapping table to map symbols.

### 4.1 Symbol Substitution Using Mapping Table

The mapping table  $MAP-TAB$  is a  $n \times n$  array whose entries are filled with all possible permutations of the byte. For the purpose of this paper, there are two instances of the mapping table  $MAP-TAB_M$  and  $MAP-TAB_F$ . These two instances have identical elements but the order of the elements in the first instance is different from that of the second instance. The elements of the first instance ( $MAP-TAB_M$ ) are organized like the S-Box in AES encryption method [6]. The second instance ( $MAP-TAB_F$ ) is derived from the first by shifting the rows to the left and shifting the columns down using a sequence of random numbers obtained from the random generator.

The substitution of an input symbol  $O_i$  using the mapping table is done as follows. The bits of  $O_i$  are split into two halves. The left half is used to index one of the mapping table's rows and the right half is used to index one of its columns. The value in the designated cell is the substitution for the input symbol.

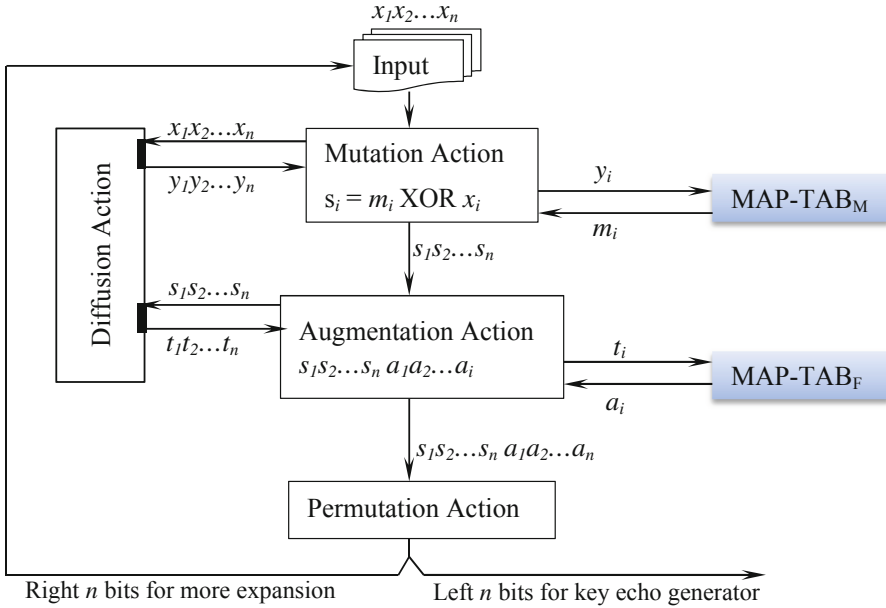


Fig. 3. Self-feeding input doubling operation algorithmic steps.

### 4.2 Self-Feeding Input–Doubling Operation

The operation receives the encryption key ( $n$  symbols) and expands the input to  $2n$  symbols. Figure 3 shows the four actions of this operation: *Diffusion*, *Mutation*, *Augmentation*, and *Permutation*. As Fig. 3 shows, the operation splits its input ( $2n$ ) into two parts. The prefix (left)  $n$  symbols provide an input for the key echo generator and the suffix  $n$  symbols provide an input to doubling operation for producing more  $2n$ -symbol sequences.

**Diffusion Action (D-Action).** The diffuse action sniffs changes in the input bits and effectively reflects this to large changes in the output (impact every symbol in the output). In other words, the action produces very different outputs for different inputs regardless of the magnitude of the differences in the input (a single bit or more). The diffusion action algorithmic steps are illustrated in Fig. 4. The action performs two-pass substitutions that ensure the maximum sensitivity to input change: *forward* and *backward*. The forward substitution substitutes the input symbol  $b_1$  to yield a new symbol  $c_1$ . For every subsequent input symbol  $b_i$  ( $i > 1$ ), the forward substitution first XORs  $b_i$  with the result of the previous substitution  $c_{i-1}$  and substitutes the outcome of the XOR operation, yielding the symbol  $c_i$ .

The output of the forward substitution  $c_1c_2...c_n$  is passed to the backward substitution. The backward substitution uses identical logic to that of the forward except that it starts from the end of its input. Therefore, it first substitutes the last symbol in the input  $c_n$  to yield the output symbol  $s_n$ . For the rest of

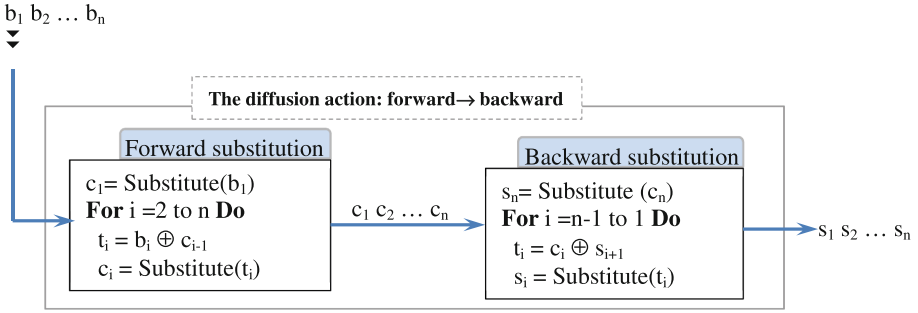


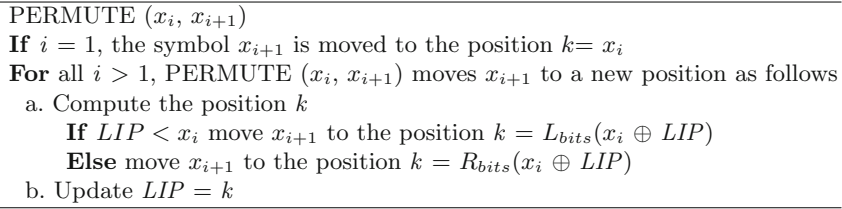
Fig. 4. The algorithmic steps of the diffusion action.

the input symbols  $c_i$  ( $i=n-1, n-2, \dots, 1$ ),  $c_i$  is first XORed with  $s_{i+1}$  and the result of the XOR operation is substituted again to yield  $s_i$ .

With the forward substitution, the symbol  $b_{i-1}$  impacts the substitution of all the subsequent symbols  $b_j$  ( $j=i, i+1, \dots$ ). This means that if some symbol, say  $b_k$ , changes, this change essentially impacts the substitution of all the successor symbols. Likewise, with the backward substitution, the symbol  $b_i$  impacts the substitution of all its predecessors  $b_j$  ( $j=i-1, i-2, \dots, 1$ ). Accordingly, by virtue of these two passes, if some symbol  $b_i$  in the block  $b_1 b_2 \dots b_n$  changes, the diffusion action ensures that this change impacts every symbol in the output block.

**Permutation Action (P-Action).** The permutation action scrambles its input block. The permutation action adopts an innovative functionality described in Fig. 5. The symbol  $x_i$  influences the position to which the next symbol  $x_{i+1}$  is moved. The permutation action has two cases. For  $i=1$ , the symbol  $x_2$  is moved to the position determined by  $x_1$ . For  $i > 1$ , the position to which the symbol  $x_{i+1}$  is moved depends not only on its predecessor  $x_i$  but also on the last point of insertion  $LIP$ . When  $x_i$  is greater than  $LIP$ , the symbol  $x_{i+1}$  is moved to the position  $k = L_{bits}(x_i \oplus LIP)$ , where  $L_{bits}$  is an operator that selects a number of bits from the leftmost bits of the outcome of the XOR operation that is sufficient to index any symbol in the output. (For instance, if the input is 16 symbols, this operator selects the leftmost 4 bits since 4 bits are adequate to index any of the 16 symbols.) When  $LIP \geq x_i$ , the symbol  $x_{i+1}$  is moved to the position determined by  $k = R_{bits}(x_i \oplus LIP)$ , where  $R_{bits}$  is the same as  $L_{bits}$  except it selects the bits from the right.

**Mutation/Augmentation Actions.** These actions use both the diffusion action and the M-TAB to perform their functionality. The mutation action invokes the diffusion action, which processes the input  $x_1 x_2 \dots x_n$  and produces the output  $y_1 y_2 \dots y_n$ . The mutation action flips each symbol  $x_i$  by substituting  $y_i$  using the M-TAB and then XORing the resulting symbol  $m_i$  with  $x_i$ . The augmentation action does the same steps except that the outcome of the substitution  $a_i$  is appended to the end of the input  $s_1 s_2 \dots s_n$ .



**Fig. 5.** The algorithmic steps of the permutation action.

Using these four actions, the Self-input doubling operation works as follows. The mutation action mutates the input  $x_1x_2\dots x_n$ . The Augmentation action doubles its  $n$ -symbol input to produce a sequence of  $2n$  symbols. The permutation action scrambles the output of the augmentation action ( $2n$  symbols). Finally, the right  $n$  symbols are fed back to the input doubling action for producing further  $2n$ -symbol sequences and the left  $n$  symbols are passed on the key echo generator (discussed next) to produce key echo sequences.

### 4.3 Key Echo Generator

This key echo generator processes its input—the output of the input doubling operation—in two stages. Each stage manipulates the symbols of the input causing drastic changes to them. Figure 6 shows the two stages of the generator. The first stage consists of two operations: *Diffusion* and *Re-Directives*. The second processing stage consists of a single operation: *Mutation*, which uses a probabilistic model to perform fine-grained modifications to some of its input symbols.

The Diffusion Action uses  $\mathcal{D}$ -Action (Subsect. 4.2) to propagate any change in the input so that this change impacts every symbol in the output. This is very important operation as it highly increases the diffusion and avalanche effect (very important security properties [26,27]).

The Re-Directives operation is a multistage operation that is composed of  $T$  distortion layers. Each layer is populated with integers from 0 to some integer  $(2^p - 1)$ , where  $p$  is the number of bits that represent a symbol. The order of the integers in each layer is independently scrambled using a sequence of random numbers  $r_i$  ( $i=1, 2, \dots, 2^p$ ), where the integer at index  $k$  is swapped with the integer at the index  $r_k$ . The input to the first layer is a symbol  $s_i$  and the output is a symbol  $x_i$  indexed by  $s_i$ . The output of the layer  $L_{i-1}$  is first manipulated by the diffusion action and is then passed as an input for the next layer  $L_i$ .

The Mutation operation is a stochastic process that uses a probabilistic model to intercept the output symbols and possibly flip bits of some of these intercepted symbols. This operation is triggered with the probability of  $\gamma \in [0, 1]$ . We call  $\gamma$  the intensity of the mutation. When  $\gamma = 0$ , no symbol is mutated. When  $\gamma = 1$ , all the symbols are mutated. One way to effectively implement this probabilistic model is to define a list  $F^{pr}$  with  $D = 2^p$  entries. This list is populated with  $H$  ( $\leq D$ ) replications of mutation operation and the remaining entries are

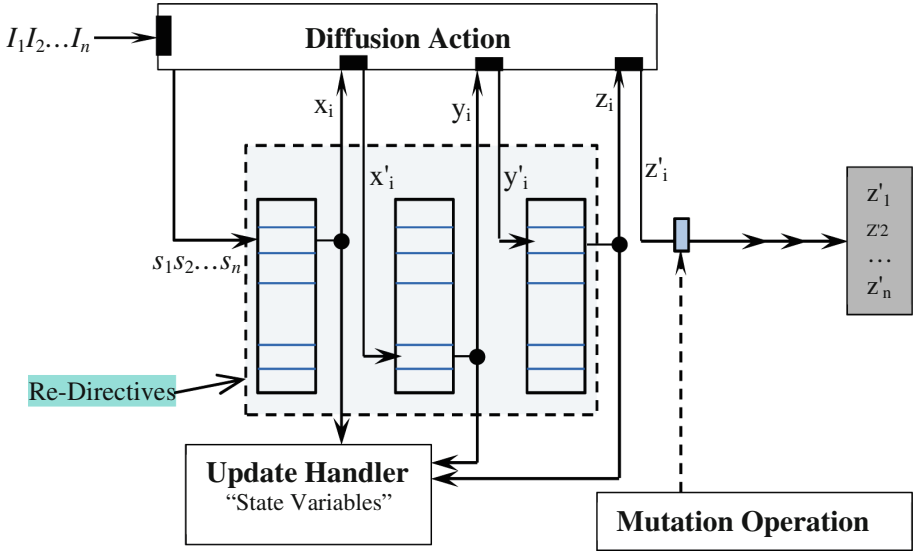


Fig. 6. The key echo generator.

populated with a null operation (does nothing). The content of this list is randomly scattered. We then can define the intensity of mutation by  $\gamma = H/D$ . Note, because of the random reordering of the elements in the list, the probability of selecting the mutation operation is  $H/D$ . In addition, the intensity of the mutation can be adjusted by changing  $H$ . Larger  $H$  grants higher chances for the mutation operation to fire and impact the input symbol. We set  $H$  to  $D/2$  in our preliminary experiments, which means that the mutation operation is invoked with a probability of  $\frac{1}{2}$ . We believe, however, that the choice of  $H$  should be based on a selection mechanism that relies on the encryption key (planned for future work).

The update handler maintains a set of  $M$  state variables for supporting the functionality of a set of actions that manipulate the re-directive layers  $L_i$ 's and mutation operation list  $F^{prt}$ . We attach a state variable  $V_{L_i}$  with each layer  $L_i$ . These state variables are used to perform some reordering to the elements of the corresponding layer. We assign two state variables  $V_{M1}$  and  $V_{M2}$  to the mutation operation, where the first variable  $V_{M1}$  checks the applicability of the mutation operation and the second variable  $V_{M2}$  selects the mutation pattern (discussed next).

Each state variable  $V$  is initially set to 0 (zero). The update handler continuously renews their values (after processing each input symbol) as follows. Every state variable  $V_{L_i}$  is updated by XORing its previous value with the output of the layer  $L_i$  just before passing this output to the diffusion action. The state variables  $V_{M1}$  and  $V_{M2}$  are updated using values from the layers  $L_i$  in a round-robin manner. In particular,  $V_{M1}$  is updated by XORing its previous values with the content of the cell  $L_j[V_{M1}]$  and  $V_{M2}$  is updated likewise using the content

of the cell  $L_{j+1}[V_{M2}]$  ( $j=0, 1...T$ ). The rationale behind this update mechanism is that we want the two processing stages to be highly influenced by the input symbols.

After discussing the two processing stages and the update handler, we describe how the key echo generator works. Suppose a sequence of  $n$  symbols  $I_1 I_2 ... I_n$  received from the input doubling operation. These input symbols are processed by the diffusion action, yielding the new sequence  $s_1 s_2 ... s_n$ . Each symbol  $s_i$  undergoes successive distortions through the layers  $L_i$ 's. Each layer  $L_i$  maps its input to a new output symbol. This new output symbol is used to update the state variable  $V_{L_i}$  corresponding to the layer  $L_i$  and is then passed to the diffusion action (recall the diffusion action uses  $\mathcal{D}$ -Action) before mapping it to the next layer. The output of the first stage (the re-directives) may be further distorted by applying the mutation operation (second stage). The state variable  $V_{M1}$  accesses the list  $F^{pr}$ . If the accessed operation is null, no distortion is performed on the current symbol. Otherwise, the mutation operation flips bits of the input symbol by XORing this symbol with the content of the cell  $L_y[V_{M2}]$ , where  $y = 0, 1...T$ . Regardless of whether the mutation operation is invoked or not, the two state variables ( $V_{M1}$  and  $V_{M2}$ ) must be updated as described above and the index  $y$  is incremented by one.

Before processing the next sequence of symbols  $I_1 I_2 ... I_n$  (received from the input doubling operation), the structure of each layer must be modified by partially reordering its elements. In particular, the layer  $L_i$  is first left shifted one position and the content of the first cell (i.e.  $L_i[0]$ ) is swapped with the content of the cell indexed by  $V_{L_i}$  (i.e. with the cell  $L_i[V_{L_i}]$ ).

## 5 Camouflaging Code Generator

This process receives sequences of symbols from the key echo generator and sharply modifies the individual input symbols and the structure of the output sequence. By so doing, the resulting output is highly random and provides enormously sophisticated camouflaging code in which the output of an encryption method is concealed. Figure 7 shows the main constituent components of the process. The dashed-shaded shapes represent data sources, the solid-line shapes represent subprocesses, and double-line shape represents an output list. Before we present the technical details of each of component, we briefly describe how the process works. Referring to Fig. 7, the **Mapping** subprocess invokes the **Substitution** action to map the input symbol  $X$  to a new symbol  $s_i$  using the  $MAP-TAB_M$ . It additionally invokes the **Feedback Handler** to calculate a feedback symbol  $f$ . The **Input Noising** subprocess utilizes the feedback symbol to (1) further modify the outcome of the mapping (i.e. the symbol  $s_i$ ) using one of the **Bitwise Operations** and (2) identify the insertion point for this processed symbol in the output list.

The **Flirt and Mate** subprocess works as a controller for the **Output Noising** and the **Internal Update** action. It sends invocation signals that carry state information to invoke the output noising and internal update actions and



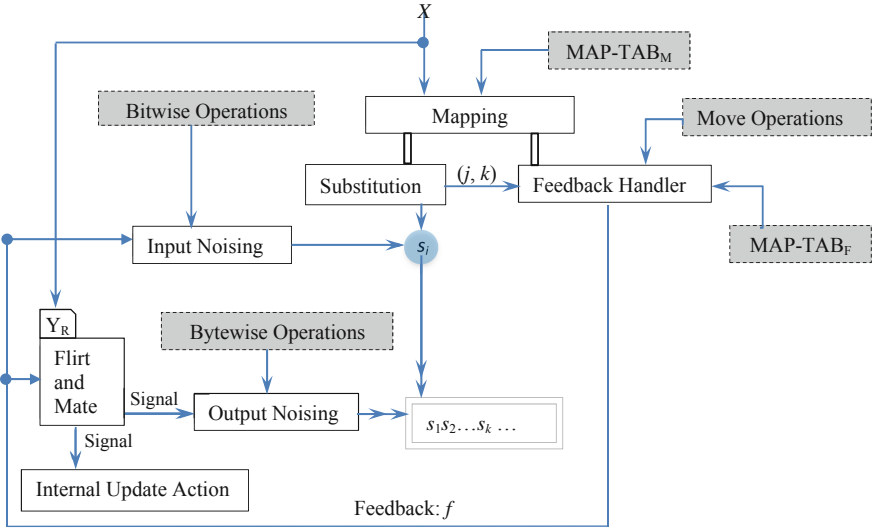


Fig. 7. The Camouflaging code generator subprocesses.

control their functionality. The **Flirt and Mate** subprocess is triggered when a variable  $X$  flirts its chromosome  $Y_R$ . Once triggered, it sends invocation signals to (1) the **Internal Update** action to update the variable  $Y_R$  and (2) the **Output Noising** action to handle the output list using one of the **Byte-wise Operations**.

After briefly describing how the process (code generator) works, we present the technical details of each component.

### 5.1 Move Operations

The move operations add significant fuzziness to mapping the input symbols to  $MAP-TAB$ . Table 1 shows the eight proposed move operations and succinct descriptions of their functionality. Referring to the table, the move operations represent all the move directions that could be taken starting from some cell in  $MAP-TAB$ . For instance, the operation  $Top(k)$  moves up  $k$  positions starting from some location in the  $MAP-TAB$  while  $TLC(k)$  moves  $k$  positions along the top left corner of some location in  $MAP-TAB$ .

### 5.2 The Input Noising

This subprocess performs dual tasks that largely affect the input symbols and partially affect the output list. First, it alters the input symbols using one of the bitwise operators. Second, it finds an insertion point within the output list to insert the processed symbol.

**Table 1.** The move operations.

Operation	Functionality
<i>Left(k)</i>	Move $k$ positions from the current position to the left-wrap if reaching the border
<i>Right(k)</i>	Move $k$ positions from the current position to the right-wrap if reaching the border
<i>Top(k)</i>	Move $k$ positions from the current position to the top-wrap if reaching the border
<i>Bottom(k)</i>	Move $k$ positions from the current position to the down-wrap if reaching the border
<i>TLC(k)</i>	Move $k$ positions from the current position along the top left corner-wrap if reaching the border
<i>BLC(k)</i>	Move $k$ positions from the current position along the bottom left corner-wrap if reaching the border
<i>TRC(k)</i>	Move $k$ positions from the current position along the top right corner-wrap if reaching the border
<i>BRC(k)</i>	Move $k$ positions from the current position along the bottom right corner-wrap if reaching the border

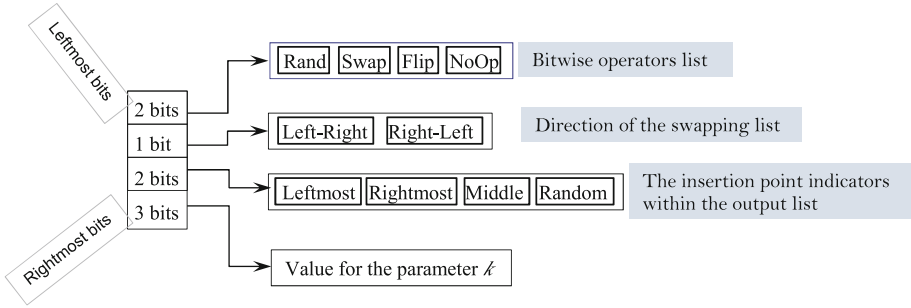
**Table 2.** The input noising operations.

Operation	Functionality
<i>Rand(r)</i>	Flips randomly selected bits by XORing its input symbol $s_i$ with the random number $r$
<i>Swap(k)</i>	Swaps the left $k$ bits of the symbol $s_i$ to the right or the right $k$ bits to the left
<i>Flip(k)</i>	Flips the left $k$ bits of its input $s_i$
<i>NoOp()</i>	Does no processing (idempotent)

The bitwise operators—as succinctly described in Table 2—cause different patterns of change to the bits or to the bits organization of the input symbols. In particular, the operators *Flip* ( $k$ ) and *Rand* ( $r$ ) modify some or all the bits of the input symbol  $s_i$  by XORing this symbol with respectively a specific value  $k$  or a random value  $r$ . While the *Swap*( $k$ ) operator modifies the input symbol  $s_i$  by changing the order of its bits. The operator *NoOp*() does no actual processing and is added for the purpose of bringing more fuzziness to the input noising subprocess.

To process the input symbol  $s_i$ , the input noising subprocess utilizes the feedback symbol  $f$  to select the bitwise operator and do any data binding necessary for the functionality of the selected operator. The selection of an operator and the data binding are performed according to the procedure described in Fig. 8.<sup>1</sup> Referring to Fig. 8, the input noising subprocess uses the two leftmost bits (top) to select one of the four bitwise operators. If the selected operator is *Rand*( $r$ ), the input noising obtains a random number from the random generator and invokes this operator to handle the symbol  $s_i$ . If the selected operator is *Swap*( $k$ ), the input noising binds the parameter  $k$  to the value of the three rightmost bits (bottom), determines the direction of the swap using the third bit (from top), and finally calls the *Swap* operator to process the input symbol  $s_i$ . If the selected operator is *Flip*( $k$ ), the input noising binds the parameter  $k$  to the value of the three rightmost bits and calls this operator to process the input

<sup>1</sup> Please note: we assume in Fig. 8 that each symbol is represented by 8 bits and the bits are organized from top (leftmost bit) to bottom (rightmost bit).



**Fig. 8.** Procedure for selecting a bitwise operator and binding values for the parameters of the selected operator.

symbol  $s_i$ . Finally, if the operator is  $NoOp()$ , the subprocess passes the symbol  $s_i$  without processing.

After processing the symbol  $s_i$ , the input noising subprocess determines an insertion point within the output list for inserting the output symbol  $s_i$ . To do so, it uses the fourth and fifth bits to choose one of the four insertion point indicators—**Leftmost**, **Rightmost**, **Middle**, and **Random** and inserts the symbol  $s_i$  in the output list according to the selected insertion point indicator. (The meaning of the insertion point indicators is: **Leftmost** appends the symbol  $s_i$  as a prefix to the output list, **Rightmost** appends  $s_i$  as a suffix, **Middle** inserts  $s_i$  in the middle of the output list, and finally **Random** inserts the symbol  $s_i$  in a random position obtained from the random generator.)

To maximize its effectiveness, the input noising subprocess updates the order of the elements in three lists (bitwise operators, direction of swapping, and the insertion point) after processing each symbol  $s_i$ . The update is done by left shifting the contents of each list, where the shifting amount is determined using the bits of the feedback symbol as follows. It uses the three rightmost bits to compute the amount of shifting the bitwise operators list, the three leftmost bits to compute the amount of shifting the insertion point list, and the fourth and fifth bits to compute the amount of shifting the direction of swapping list.

### 5.3 The Mapping

As the Fig. 7 shows, this subprocess relies on two actions: Substitution action and Feedback Handler. The Substitution subprocess is described in Subsect. 4.1. We therefore explain in this subsection only the details of the feedback handler.

The feedback handler produces a feedback  $f$ , an essential value to support the functionality of other subprocesses (e.g. input noising). Figure 9 illustrates the logic of the feedback handler. The handler defines three rings  $C_1$ ,  $C_2$ , and  $C_3$  each of which consists of  $n$  entries. Each of the rings  $C_1$  and  $C_2$  contain the integers from 0 to  $n-1$  (e.g.  $n = 256$ ). The outer ring  $C_3$  contains descriptors that designate the eight possible adjacent cells for any specific cell  $(a, b)$  in

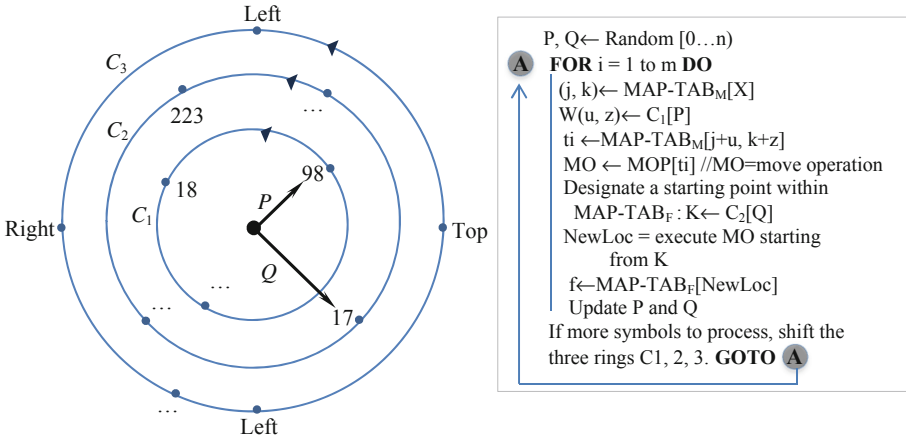


Fig. 9. The feedback action logic.

the mapping table  $MAP-TAB$ . These descriptors are: *top*, *bottom*, *left*, *right*, *left/right top corner*, and *left/right bottom corner*. These descriptors have an obvious meaning. For instance, descriptor *left* refers to the cell that is directly at the top of some specific cell  $(a, b)$  while the descriptor *top right corner* refers to the cell that directly is at top right corner of some specific cell  $(a, b)$ . The contents of the three rings are randomly scattered using different sequences of random numbers.

The feedback handler additionally maintains two pointers  $P$  and  $Q$  to select entries from the three rings. Each of the pointers  $P$  and  $Q$  is initially set to a random number but continuously updated in a way to be made clear next.

The feedback handler is triggered when the substitution action uses  $MAP-TAB_M$  and substitutes the input  $X$  with a value  $s_i$ . Suppose the substitution location is  $(j, k)$ . The handler receives the location  $(j, k)$  and uses the pointer  $P$  to index the ring  $C_1$  and obtain a value  $W = C_1[P]$ . It then uses both  $W$  and the location  $(j, k)$  to create a new location  $(j+u, k+z)$ , where  $u$  is the decimal value of the left half bits of  $W$  and  $z$  is the decimal value of the right half bits of  $W$ . The feedback handler accesses the mapping table  $MAP-TAB_M$  at the row  $j+u$  and the column  $k+z$  to obtain a value  $t_i$ . The value  $t_i$  is used to index one of the move operations. The feedback handler executes the selected move operation to perform a move within the mapping table  $MAP-TAB_F$  starting from the point designated by  $C_2[Q]$ . The value at the sink cell is extracted, which is the feedback value  $f$ .

The feedback handler updates the pointers  $P$  and  $Q$  before producing any new feedback symbol. It uses the ring  $C_3$  to find the adjacency descriptors  $C_3[P]$  and  $C_3[Q]$ . The handler uses descriptor  $C_3[P]$  to identify the proper adjacent cell to the cell (within  $MAP-TAB_F$ ) from which  $f$  was obtained and extracts the value  $v$  from this adjacent cell. The pointer  $P$  is updated by XORing its previous value with  $v$  (i.e.  $P = P \oplus v$ ). Likewise, the handler uses the descriptor

**Table 3.** Chromosome evolution handling operators.

Operation	Functionality
$Crossover(m, flag)$	The chromosome $Y_R$ and the flirting variable $X$ exchange $m$ bits based on $flag$ . The $flag$ can be either value: $LL$ (Left-Left), $RR$ (Right-Right), $LR$ (Left-Right), $RL$ (Right-Left)
$Flip()$	XORes the chromosome $Y_R$ , the flirting variable $X$ , and the feedback symbol $f$ ( $Y_R \oplus X \oplus f$ )

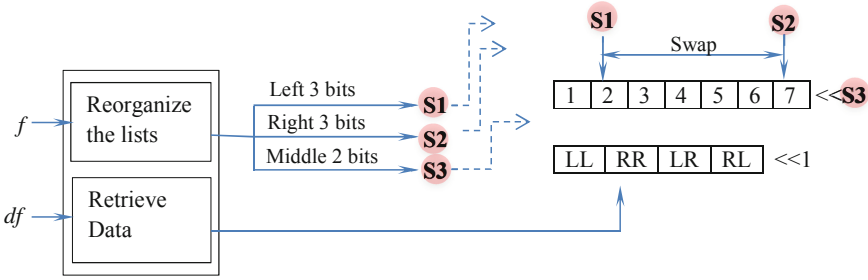
$C_3[Q]$  to designate the proper adjacent cell (within  $MAP-TAB_M$ ) to the cell from which the value  $t_i$  was obtained and extract its value  $w$ . The pointer  $Q$  is updated by XORing its previous value with the value  $w$ .

To exacerbate the fuzziness of the feedback handler action, the content of the three rings  $C_1$ ,  $C_2$ , and  $C_3$  are counterclockwise shifted ( $\triangleleft$ ) after processing  $m$  input symbols. (The integer  $m$  is the length of the block as specified by the encryption technique.) In particular, the rings  $C_1$ ,  $C_2$ , and  $C_3$  are counterclockwise shifted by respectively 1, 2, and 3.

### 5.4 Flirt and Mate Technique

This subprocess is composed of a single  $n$ -gene chromosome and an internal mechanism to control the evolution of the chromosome. The process is triggered when some variable flirts the chromosome. Upon the flirtation, the mate subprocess checks whether the flirting variable and the chromosome are eligible to mate, where the mate eligibility is defined in terms of the genetic diversity adequacy, which in turned is measured by the number of genes that differ in the corresponding positions. (We call the number of different genes the degree of fitness, or  $df$ .) If the degree of fitness exceeds some prespecified threshold  $n/2$ , the flirting variable and the chromosome are eligible to mate and we call this state effective flirtation; otherwise we call this state (eligibility to mate) ineffective flirtation. Whether the flirtation is effective or ineffective, the mate subprocess triggers its internal update action to mutate the internal chromosome using one of the mutating operators in Table 3. The operator  $Crossover(m, flag)$  causes  $m$ -gene exchange between the chromosome  $Y_R$  and a flirting variable. The genes to be exchanged are fully controlled by the flag state. The flag can assume any of four states:  $LL$  (Left-Left),  $RR$  (Right-Right),  $LR$ , and  $RL$ . For instance, when the flag has the state  $LL$ , the crossover operator replaces (updates) the left  $m$  genes of the chromosome with the left  $m$  genes of the flirting variable, yielding a new chromosome. The operator  $Flip()$  alters (updates) the genes of the chromosome  $Y_R$  by XORing its previous value with both the flirting variable and the feedback symbol  $f$ .

This biologically inspired subprocess offers a very powerful means for influencing the functional behavior of both the internal update action (evolution mechanism) and the output noising subprocess (see Fig. 7). These two subprocesses are discussed next.



**Fig. 10.** The binding procedure: binding the parameters of the Crossover operator with the proper values.

### 5.5 Internal Update Action

This action controls the way in which the chromosome evolves. It fires upon receiving a control signal from the flirt and mate subprocess and updates the chromosome. The update is fully determined by the state carried by the received signal. If the flirtation is ineffective, the internal action updates the chromosome by calling the operator *Flip()*. If the flirtation is effective, the internal action updates the chromosome using the Crossover operator. To invoke the Crossover operator, the internal action must first bind the parameters *m* and *flag* to appropriate values.

The paper proposes the procedure in Fig. 10 to bind values for the two parameters of the Crossover operator.<sup>2</sup> The procedure maintains two lists: a list to store the number of genes (1, 2,...7) that could possibly be exchanged between the chromosome and the flirting variable and another list to store the possible states of the flag. The binding procedure receives two inputs, the feedback symbol *f* and degree of fitness *df*. It uses the feedback symbol to modify the order of the elements in the first list (top list). In particular, it uses the three leftmost bits (*S*<sub>1</sub>) and the three rightmost bits (*S*<sub>2</sub>) of the feedback symbol to designate two locations in the list and exchange their contents. It additionally uses the middle two bits (*S*<sub>3</sub>) to shift the content of the top list to the left by *S*<sub>3</sub> positions. The second list is left shifted but by one each time the procedure is invoked. When the order of the elements for both lists changed, the degree of fitness *df* indexes the first list to retrieve a value and binds it to *m* and indexes the second list to retrieve a value and binds it to the *flag*. Once binding the parameters of the Crossover operator with proper values, this operator is invoked and updates the chromosome.

Observe that the flirt and mate subprocess induces fuzziness to the behavior of the internal update action. As discussed above, the mate subprocess bases the functionality of the internal update action on the state of flirtation (effective or not) and the degree of fitness; both involve fuzziness.

<sup>2</sup> For the sake of simplifying the presentation, it is assumed that each symbol is represented by 8 bits.

**Table 4.** Output manipulation operators.

Operation	Functionality
<i>Permute</i> ( <i>h</i> )	This operator performs <i>h</i> swaps on the output list
<i>Shift</i> ( <i>k</i> )	This operation left shifts the output list <i>k</i> positions

### 5.6 The Output Noising

The output noising gradually breaks the order of the symbols in the output list without mutating the symbols per se, leading to a more randomized output. This subprocess is triggered when it receives a state signal from the flirt and mate subprocess. The output noising makes a decision regarding whether to actually preserve or change the order of the symbols in the output list based on the state carried by the received signal (the state of the mating). If the state indicates *ineffective* flirting (no mate), the output noising performs no reordering on the output list. It, however, accumulates the current feedback symbol with the previous ones by an XOR operation to create accumulated feedback history.

If the state indicates *effective* flirting however, the output noising changes the order of the symbols in the output list by executing first the operator *Permute*(*h*) and then the operator *Shift*(*k*). These two operators are defined in Table 4 and function as follows. The *Permute* (*h*) operator performs *h* swaps, where *h* is the degree of fitness. Each swap exchanges the element at index *i* (*i*= 0, 1...*h*-1) with the symbol at index  $j = \frac{f_c \oplus x_i}{2^p} * L_{out} \pm H * x_i$ . The symbol *f<sub>c</sub>* is the most recent feedback symbol, *p* is the number of bits that represents a symbol, *x<sub>i</sub>* is the Unicode value of the symbol at location *i*, *L<sub>out</sub>* is the length of the current output list, and  $H = (H \oplus f_k) / 2^p$  (*k*=1, 2...*c*-1) is the accumulated history of the previous feedbacks (the outcome of XORing all the previous feedbacks). The offset *H* \* *x<sub>i</sub>*, which is created by multiplying the feedback history *H* and the value of the symbol *x<sub>i</sub>*, is added (+) or subtracted (-) if *x<sub>i</sub>* is respectively even or odd.

The *shift* (*k*) operator moves the symbols of the output list by *k* positions to the left. The number of positions *k* equal to *H* \* *x<sub>i</sub>* after adding the effect of the most recent feedback to *H*.<sup>3</sup>

We conclude this section by emphasizing the fact that the flirt and mate subprocess fuzzily impacts the behavior of output noising subprocess. From one hand, the output noising subprocess depends on the state of the flirtation (effective or not) to determine the way of handling the output list (alter the order of some of its elements). From the other hand, it depends on the state of flirtation and the degree of fitness (between the chromosome and the flirting variable) to control the behavior of the *Permute* operator.

<sup>3</sup> Observe that the new index *j* depends on both the impact of the current feedback symbol *f<sub>c</sub>* and the accumulated history of all the previous feedbacks. This makes the computation of each index *j* involve plenty of fuzziness. Furthermore, the shifting operator maximizes the effectiveness of the *Permute*(*h*) operator by changing the symbols that will be influenced by every permutation.

## 6 Performance Analysis

We report in this section our conducted simulations using the prototype implementation of our technique. We analyze the desired properties of the technique's output that are fundamental for effectively securing the ciphertexts (output of encryption methods). These properties are roughly summarized by the following two points.

1. The output of the proposed technique must pass fundamental randomness tests regardless of the encryption key. This property is so important because random output will boost the randomness of the supposedly-random ciphertext.
2. Two different keys should result in two independent sequences (not correlated) regardless of the key differences (tiny-one bit or large-many bits). This property is crucial for two reasons. First, the existence of the correlation is tantamount to the existence of patterns, which is very serious problem that may jeopardize the security of the ciphertext. Second, this property shows that the technique is highly sensitive to key changes (an extremely important property [26]).

### 6.1 Statistical Hypotheses

We want to test the following hypotheses about the output of the proposed technique.

**H0:** The generated code sequence by the proposed technique is random.

**H1:** The generated code sequence by the proposed technique is not random.

The null hypothesis ( $H_0$ ) asserts that the tested data is random while the alternative hypothesis ( $H_1$ ) asserts that the tested data is not random. Accepting  $H_0$  or  $H_1$  depends on comparing two values: p-value and the significance level  $\alpha$ . The p-value is computed by the statistical test based on an input sequence. The significance level  $\alpha$  is prespecified by the tester (e.g. 0.00001, 0.001, 0.01, 0.05 are typical values for  $\alpha$ ). If  $p\text{-value} \geq \alpha$ ,  $H_0$  is accepted and  $H_1$  is rejected. If otherwise  $H_0$  is rejected and  $H_1$  is accepted. We set the significance level  $\alpha$  to 0.05 in all of our tests.

### 6.2 Randomness Tests

We use fundamental randomness tests to examine the randomness properties of the proposed technique's output and to measure the correlation among pairs of the output sequences. The chosen randomness tests are the core tests obtained from the battery of randomness tests recommended by National Institute for Standards and Technology-NIST [17]. All the definitions were excerpted from [17].



- **Runs test:** determines whether the number of runs of ones and zeros of various lengths is as expected for a random sequence.
- **Frequency test (Monobit):** determines whether the number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence.
- **Discrete Fourier Transform Test (*Spectral*):** detects periodic features (i.e. repetitive patterns that are near each other) in the tested sequence that would indicate a deviation from the assumption of randomness.
- **Serial Test:** determines if the number of occurrences of the  $2^m$   $m$ -bit overlapping patterns is approximately the same as would be expected for a random sequence. Random sequences have uniformity in a sense that each  $m$ -bit pattern has an equal chance of appearing as every other  $m$ -bit pattern.
- **Cumulative Sums Test:** determines if the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of that cumulative sum for random sequences. The cumulative sums may be considered as random walks. If the sequence is random, the excursions of the random walk should be near zero.

### 6.3 Data Preparation

The data that we want to test include code sequences, which is generated by our technique using different encryption keys. The greatest challenge is that: how many encryption keys are sufficient to ensure a reasonable test? It is certainly infeasible to try our technique on all possible encryption keys; there are almost infinite number of them. The best we can do is to find a fair approximation that reasonably covers the space of the possibilities. One reasonable way, which we adopt here, is to split the space of all possible keys into classes and randomly select representative keys from each class.

For the purpose of this paper, we split the space into three classes. The first class consists of 20 groups created as follows. We started with a key whose bits are all zeros (128 zeros). We then created different keys by randomly setting  $p$  bits ( $1 \leq p \leq 128$ ). Table 5 shows statistics about these keys. For instance, we created 128 different keys by setting a single bit of the original key, 1000 different keys by setting two bits in randomly selected positions, and so on. The total number of keys in this class is 17130. The second class consists of 3000 128-bit keys generated randomly using online generator [15]. The third class consists of 240 handcrafted 128-bit keys created by our students at AUM. The total number of keys in the three classes is 20370. We checked in-class keys and across-classes keys to make sure that they are all different.

Other important configurations that impact the system performance are set as follows. The number of layers in re-directive component (Fig. 6) is three layers. The length of all the used keys were 16 bytes (symbols). We finally set the mutation intensity  $\gamma$  (Subsect. 5.3) to 0.45.

**Table 5.** The groups of keys (Class 1)

Group	Description	Keys (chosen)	Group	Description	Keys (chosen)
1	1 bit was set	128	11	60 bits were set	1000
2	2 bits were set	1000	12	70 bits were set	1000
3	5 bits were set	1000	13	80 bits were set	1000
4	10 bits were set	1000	14	90 bits were set	1000
5	15 bits were set	1000	15	100 bits were set	1000
6	25 bits were set	1000	16	110 bits were set	1000
7	35 bits were set	1000	17	115 bits were set	1000
8	40 bits were set	1000	18	120 bits were set	500
9	45 bits were set	1000	19	125 bits were set	500
10	50 bits were set	1000	20	128 bits were set	1

### 6.4 Code Sequence Randomness Analysis

The proposed technique used the keys to generate camouflaging code sequences of two sizes: sequences with 64000 symbols and sequences with 128000 symbols. These sequences were tested for randomness using the aforementioned NIST’s randomness tests. Since NIST’s randomness tests assume binary input strings [16], all the generated sequences were converted to binary sequences (consisting of zeros and ones). Due to the fact that we used only symbols within the range [0, 255], these sequences were straightforwardly converted to binary sequences by finding the 8-bit equivalent of each symbol (e.g. the binary equivalent to the symbol “A” is 01000001).

We applied the randomness tests (Subsect. 6.2) to these sequences. Tables 6 and 7 show the results for respectively the sequences of size 64000 and 128000. Each of the two tables displays the used randomness tests, the number of code sequences that passed the respective test (Successes), the number of code sequences that failed the respective tests (Failures), and the Success Rate. The level of significance is set to 0.05 for all the five randomness tests. Statistically, the significance level of 0.05 implies that, ideally, no more than 5 out of 100 code sequences may fail the corresponding test. However, in all likelihood, any given data set will deviate from this ideal case [18]. For a more realistic interpretation, a confidence interval (CI) is used for quantifying the proportion of the binary sequences that may fail a randomness test at the significance level 0.05. We therefore computed the maximum number of binary sequences that are expected to fail the corresponding test at significance level of 0.05 and presented the result in the rightmost column of each table. For instance, a maximum of 1111.82 (or 1112) code sequences are expected to fail each of the randomness tests.<sup>4</sup>

<sup>4</sup> The maximum number of binary sequences that are expected to fail at the level of significance  $\alpha$  is computed using the following formula [18]:  $S \cdot (\alpha + 3 \cdot \sqrt{\frac{\alpha(1-\alpha)}{S}})$ , where  $S$  is the total number of sequences and  $\alpha$  is the level of significance.

**Table 6.** Randomness tests outcome for sequences of size 64000 symbols.

Randomness test	Successes	Failures	Success rate	Upper limit of CI (0.05)
Runs test	20370	0	100%	1111.82
Monobit test	20363	7	99.9%	1111.82
Spectral test	19378	992	95.1%	1111.82
Serial test	20133	257	98.8%	1111.82
Cumulative sums test	19929	441	97.8%	1111.82

**Table 7.** Randomness tests outcome for sequences of size 128000 symbols

Randomness test	Successes	Failures	Success rate	Upper limit of CI (0.05)
Runs test	20370	0	100%	1111.82
Monobit test	20370	0	100%	1111.82
Spectral test	19869	501	97.5%	1111.82
Serial test	20346	24	99.8%	1111.82
Cumulative sums test	20156	214	98.9%	1111.82

According to the performance figures in Tables 6 and 7, our proposed technique performed really well. From one hand, the minimum percentage of the sequences that passed the randomness test is 95.1%. From the other hand, the number of sequences that actually failed the randomness test is less than the maximum number expected by the 95% confidence interval. It is worth noting that as the size of the sequences increases, so does (though slightly) the success rate.

## 6.5 Correlation Analysis

Given the large number of sequences (20370), it is quite difficult to analyze the correlation between all the possible different pairs of sequences. As such, the best we can do is to draw random samples and analyze the correlation using these samples. To analyze the correlation among the sequences in each class (recall we have three classes of sequences), we randomly sampled 1000 pairs from the first class, 1000 pairs from the second class and 1000 pairs from the third class. To analyze the correlation across classes, we randomly select a sequence from the first class and a sequence from the second class, a sequence from the second class and a sequence from the third class, and continued in round-robin manner. The sample size was 1500 pairs  $(x, y)$ , where  $x \in class_i$  and  $y \in class_j$  and  $i \neq j$  ( $i, j=1, 2, 3$ ). All the samples are drawn from the classes with size 128000 symbols.

To test for the correlation (whether between the sequences that belong to the same class or different classes), we performed an XOR operation on the sequences of each pair. The outcome of the XOR operation is tested for randomness using the random tests (Subsect. 6.2).

**Table 8.** Randomness tests outcome for the correlation between sequences.

Randomness test	Successes	Failures	Success rate	Upper limit of CI (0.05)
Runs test	4500	0	100%	268.86
Monobit test	4500	0	100%	268.86
Spectral test	4414	86	98.1%	268.86
Serial test	4499	1	99.97%	268.86
Cumulative sums test	4491	9	99.8%	268.86

**Table 9.** Randomness test outcome for DES ciphertexts.

Randomness test	Class 1 (keys)		Class 2 (keys)		Class 3 (keys)	
	<i>p-value</i>		<i>p-value</i>		<i>p-value</i>	
	Average	Min	Average	Min	Average	Min
Runs test	0.61	0.185	0.64	0.17	0.581	0.306
Monobit	0.59	0.210	0.501	0.305	0.541	0.169
Spectral	0.34	<b>1.3E-12</b>	0.391	<b>0.0025</b>	0.280	<b>6.0E-08</b>
Serial test	0.401	0.12	0.511	0.222	0.623	<b>4.5E-05</b>
Cumulative sums	0.44	0.201	0.410	0.187	0.377	<b>1.11E-4</b>

Table 8 shows the results of the randomness test. Based on the Table 8’s figures, it is evident that only a very small ratio of the pairs failed the randomness test (show correlation). Generally speaking, the proposed technique generate code sequences that do not correlate.

### 6.6 Integrating the Proposed Technique with DES: Impacts on Des

For further discussing the capabilities of the proposed technique, we integrated this technique with a widely used encryption technique: Data Encryption Standard (DES). The purpose is to analyze the impact of the proposed technique on the randomness of the DES’s output (ciphertext). We used 3 text files of size 128000 symbols (extracted from Wikipedia). We then used the DES technique to encrypt each of three files using 500 different keys from class 1, 500 different keys from class 2, and 100 keys from class 3. These keys are chosen randomly. We tested the ciphertexts for randomness using the aforementioned random tests. Table 9 shows the results. As it could be seen, the DES technique generated ciphertexts that are random except for five ciphertexts failed the Spectral test, two failed the cumulative sums, and one failed the serial test.

We then added the effect of the camouflaging codes to the ciphertexts. For each key used to encrypt a text, we chose the camouflaging code generated by this key.<sup>5</sup> The effect of the code was added to the corresponding ciphertext

<sup>5</sup> We ignored the keys that did not result in random camouflaging codes.

**Table 10.** Randomness test outcome for DES ciphertexts after adding the effect of the camouflaging code.

Randomness test	Class 1 (keys)		Class 2 (keys)		Class 3 (keys)	
	<i>p-value</i>		<i>p-value</i>		<i>p-value</i>	
	Average	Min	Average	Min	Average	Min
Runs test	0.83	0.589	0.88	0.506	0.791	0.499
Monobit	0.92	0.709	0.96	0.881	0.94	0.837
Spectral	0.56	0.145	0.603	0.228	0.499	0.333
Serial test	0.386	0.212	0.667	0.495	0.629	0.478
Cumulative sums	0.697	0.508	0.736	0.419	0.52	0.232

by simply performing a symbol-wise XOR operation between them. We then applied the randomness tests to the resulting ciphertexts. Table 10 shows the results. As the Table 10's figures show, all the ciphertexts passed the randomness tests. Moreover, the randomness indicators (p-values) also greatly improved. This improvement to the output of the DES techniques can be attributed to the high degree of randomness of the proposed technique output.

Before we conclude this section, we comment on the performance analysis results. First, the benchmark keys that we used constitute a quite reasonable coverage of the keys space. As discussed, the benchmark consists of keys that are slightly different (single bit or several bits), randomly generated, and handcrafted by actual users. We think that these variations of the keys fairly imitates the set of keys that may be used in reality (when encrypting texts). For all these variations of the keys, the proposed technique produced code sequences that are random with no correlation among the pairs of the generated sequences.

In addition, the figures in Tables 9 and 10 well justify the validity of adding the proposed new layer. Prior to adding the effect of the proposed technique to DES's ciphertexts, few ciphertexts failed some of the randomness tests. When we added the effect of the proposed technique, the results significantly improved. All the ciphertexts (after adding the noise caused by the proposed technique) passed the used randomness tests and the randomness itself is significantly improved.

## 7 Conclusions and Future Work

The paper proposed a technique for generating sequences of camouflaging codes. The technique can be intergraded as a closing stage with encryption techniques to (1) boost the randomness of their outputs (ciphertexts) and (2) provide a powerful line of defense against cryptanalysis techniques.

The technique is implemented and reasonably tested. The performance analysis showed that, in general, our technique produces random camouflaging code sequences with no correlation among these sequences. Additionally, the performance analysis highlighted an important property of our technique: it can

effectively melt undesired patterns in the ciphertext and therefore make the ciphertext random. This was evident when our technique improved the randomness of the already random ciphertexts and randomized those ciphertexts that failed some of the randomness tests.

We have three directions for future work. First, we plan to do more testing to better evaluate the true performance of the technique. Second, we want to study how our technique impacts the performance of encryption techniques (e.g. AES). In particular, we are interested in evaluating how our technique can improve the randomness of encryption techniques' output. Third, we are currently investigating many ideas to find a better model for hiding the symbols of ciphertexts (the output of encryption techniques) in the camouflaging code.

## References

1. Al-Muhammed, M.J., Abuzitar, R.: Intelligent convolutional mesh-based encryption technique augmented with fuzzy masking operations. *Int. J. Innov. Comput. Inf. Control* (2019). (to appear)
2. Al-Muhammed, M.J., Abuzitar, R.: Dynamic text encryption. *Int. J. Secur. Appl. (IJSIA)* **11**(11), 13–30 (2017)
3. Bogdanov, A., Mendel, F., Regazzoni, F., Rijmen, V.: ALE: AES-based lightweight authenticated encryption. In: Moriai, S. (ed.) *Fast Software Encryption, FSE, LNCS*, vol. 8424. Springer, Heidelberg (2013)
4. Knuden, L.R.: Dynamic encryption. *J. Cyber Secur. Mob.* **3**, 357–370 (2015)
5. Mathur, N., Bansode, R.: AES based text encryption using 12 rounds with dynamic key selection. *Procedia Comput. Sci.* **79**, 1036–1043 (2016)
6. Daemen, J., Rijmen, V.: *The Design of RIJNDAEL: AES-The Advanced Encryption Standard*. Springer, Berlin (2002)
7. Nie, T., Zhang, T.: A study of DES and blowfish encryption algorithm. In: *Proceedings of IEEE Region 10th Conference*, Singapore (2009)
8. AL-Muhammed, M.J., Abuzitar, R.:  $\kappa$ -lookback random-based text encryption technique. *J. King Saud Univ.-Comput. Inf. Sci.* **2019**(31), 92–104 (2019)
9. Patil, P., Narayankar, P., Narayan, D.G., Meena, S.M.: A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and blowfish. *Procedia Comput. Sci.* **78**, 617–624 (2016)
10. N.I.S.T. Special Publication, 800–67 Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher Revision 1.: Gaithersburg, MD, USA, January (2012)
11. Bogdanov, A., Mendel, F., Regazzoni, F., Rijmen, V., Tischhauser, E.: ALE: AES-based lightweight authenticated encryption. In: Moriai, S. (ed.) *FSE 2013, LNCS*, vol. 8424, pp. 447–466. Springer, Heidelberg (2014)
12. Stallings, W.: *Cryptography and Network Security: Principles and Practice*, 7th edn. Pearson, London (2016)
13. Anderson, R., Biham, E., Knudsen, L.: *Serpent: a proposal for the advanced encryption standard* (2018). <http://www.cl.cam.ac.uk/~rja14/Papers/serpent.pdf>. Accessed Feb 2018
14. Burwick, C., Coppersmith, D., D'Avignon, E., Gennaro, R., Halevi, S., Jutla, C., Zunic, N.: *The MARS Encryption Algorithm*. IBM, August 1999
15. Online Random Key Generator Service. <https://randomkeygen.com>

16. Soto, J.J.: Randomness Testing of the AES Candidate Algorithms (2019). <http://csrc.nist.gov/archive/aes/round1/r1-rand.pdf>. Accessed July 2019
17. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST special publication 800-22, National Institute of Standards and Technology (NIST), Gaithersburg, MD (2001)
18. Soto, J.: Randomness Testing of the Advanced Encryption Standard Candidate Algorithms. NIST IR 6390, September 1999
19. Ashwak, M.A., Faudziah, A., Ruhana, K.: A competitive study of cryptography techniques over block cipher. In: 13th International Conference on Computer Modelling and Simulation, Cambridge, UK (2011)
20. Juels, A., Restinpart, T.: Honey encryption: security beyond the brute-force bound. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2014), Copenhagen, Denmark, pp. 293–310 (2014)
21. Bose, P., Hoang, V.T., Tessaro, S.: Revisiting AES-GCM-SIV: multi-user security, faster key derivation, and better bounds. In: Nielsen, J., Rijmen, V. (eds.) *Advances in Cryptology – EUROCRYPT 2018*, EUROCRYPT 2018, LNCS, vol. 10820, pp. 468–499. Springer, Cham (2018)
22. Ksasy, S.M., Takieldean, A., Shohieb, M.S., Elteny, H.A.: A new advanced cryptographic algorithm system for binary codes by means of mathematical equation. *ICIC Express Lett.* **12**(2), 117–124 (2018)
23. Cheng, H., Zheng, Z., Li, W., Wang, P., Chu, C.-H.: Probability model transforming encoders against encoding attacks. In: *USENIX Security Symposium* (2019)
24. Jo, H.-J., Yoon, J.W.: A new countermeasure against brute-force attacks that use high performance computers for big data analysis. *Int. J. Distrib. Sens. Netw.* **2015**, 7 (2015)
25. Al-Muhammed, M.J., Abuzitar, R.: Mesh-based encryption technique augmented with effective masking and distortion operations. In: *Proceedings of the computing conference 2019*, London, United Kingdom, 17–18 July 2019, vol. 998, pp. 771–796 (2019)
26. Shannon, C.E.: Communication theory of secrecy systems. *Bell Syst. Tech. J.* **28**, 656–715 (1949)
27. Shannon, C.E.: A mathematical theory of cryptography. *Bell Syst. Tech. J.* **27**, 379–423, 623–656 (1945)



# Statistical Analysis to Optimize the Generation of Cryptographic Keys from Physical Unclonable Functions

Bertrand Cambou<sup>(✉)</sup>, Mohammad Mohammadi, Christopher Philabaum, and Duane Booher

Northern Arizona University, Flagstaff, AZ 86011, USA

{Bertrand.cambou, mm3845, cp723, Duane.booher}@nau.edu

**Abstract.** Physical unclonable functions are not easy to integrate into cryptographic systems because they age, and are sensitive to environmental interferences. Excellent error correcting schemes were developed to handle such drifts, however the computing power needed at the client level can leak information to opponents, and are difficult to deploy to networks of ultra-low power Internet of Things. Response-based cryptography methods, which are server based, use search engines to uncover the erratic keys generated by the physical unclonable functions, minimizing the consumption of electric power at the client level. However, when the defect densities are high, the latencies associated with search engines can be prohibitive. The statistical analysis presented in this paper shows how the fragmentation of the cryptographic keys can significantly reduce the latencies of the search engine, even when error rates are high. The statistical model developed, with Poisson distribution, shows that the level of fragmentation in sub-keys can handle up to 15% error rates. The methodology is generic, and can be applied to any type of physically unclonable functions with defects in the 15% range, or lower.

**Keywords:** Security primitives · Internet of Things · Physical unclonable functions · Error correction

## 1 Introduction

Physical Unclonable Functions (PUFs) [1–7], have been developed to provide additional layers of protection to cyber physical systems, in particular for access control. They act as the “fingerprints” of microelectronic components, and of the internet of things (IoTs). PUF were successfully introduced with logic gates to protect field programming gate arrays FPGA [8]. Since then, PUFs have been successfully designed with static random access memories (SRAMs) [9], a component available in most electronic devices. However, such solutions are not always tamper resistant; PUFs are designed with Dynamic RAMs [10], flash memory devices [11, 12], Resistive RAMs [13–16], and Magnetic RAMs [17, 18]. PUFs provide excellent authentication, and access control solutions, without the problem associated with key distribution. They are more difficult to use



as cryptographic keys, because PUFs are subject to aging, temperature drifts, electromagnetic interactions, and various environmental effects [19]. Typically, these effects produce 2–10% error rates between the initial readings of the PUFs that are stored as references during enrollment cycles, and the responses generated by these PUFs. The work presented in this work is agnostic about the selection of a particular type of PUF. SRAM PUFs were used because they are widely available, and relatively easy to design.

Error correcting (ECC) algorithms can correct most of these errors, and generate usable cryptographic keys from the PUFs with low false reject rates (FRR), as single-bit mismatches are not acceptable [20, 21]. Methods such as the ones using Polar codes have been successfully implemented [16, 22]. Other methods such as those using fuzzy extractors are extremely powerful, they can correct PUF responses to generate reliable keys; however some IoT devices may not have enough computing resources to run such codes [23, 24]. In several cases, helper data [25] is generated upfront by the server, from error free keys stored in look up tables, to “help” the client device accelerate error correction schemes. The length of the helper data has to be increased when the PUF error rates are high.

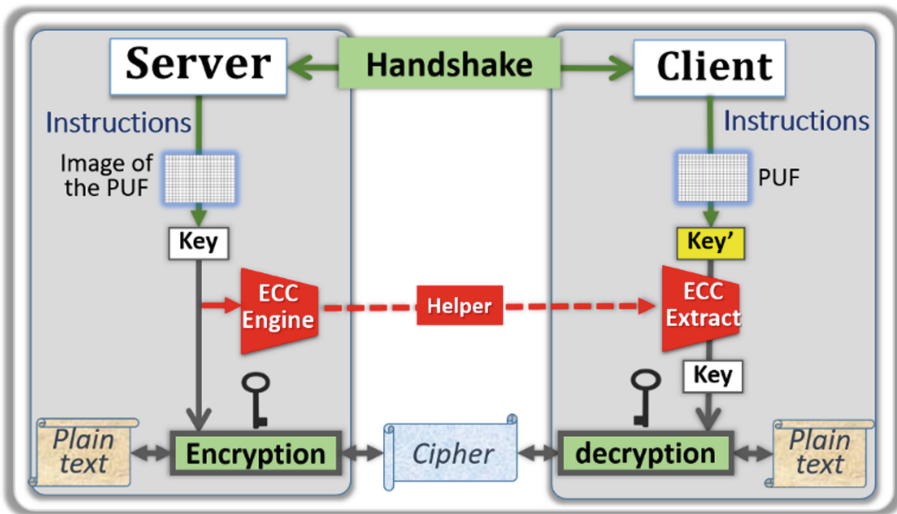
Response based cryptographic (RBC) methods have the potential to eliminate the need to use error correction at the client level, as it generates cryptographic keys directly from the un-corrected responses of the PUFs [26, 27]. This technology relies on the implementation of efficient search engines, driven by secure servers, which can uncover the keys extracted from PUF responses by the client device. With 256-bit long keys, RBC search engines can uncover keys with up to 1% error rates. The elimination of the weaker cells of a PUF during enrollment cycles [28, 29] can reduce the error rates below 10<sup>-4</sup>, which is low enough for a reliable RBC implementation. It has also been suggested that the fragmentation of the keys into sub-keys can expand the applicability of RBC in the 15% error range rate [30, 31]. Key fragmentations can reduce the latencies of the RBC search engine at high error rates, but adds complexity at low error rates. It is therefore desirable to use key fragmentation only when needed.

The objective of the work presented in this paper is to provide a statistical model, and to optimize PUF response based cryptographic schemes at various levels of error rate and fragmentation. Section 2 provides the background information describing error correcting methods, and the response based cryptography needed to be able to exploit PUFs as generators of cryptographic keys. In Sect. 3, a statistical model of RBC latencies under various conditions is developed. The effect of the level of random defects present in the PUF responses on RBC latencies is analyzed for 256-bit long keys. After presenting methods to fragment these keys into sub-keys, the impact of the level of fragmentation on the RBC latencies is analyzed, with the objective of developing predictive tools. An experimental validation of the suggested models is presented in Sect. 4. The measurements based on commercially available SRAM-based PUFs greatly validate the accuracy of the statistical models when the defect densities are large enough to necessitate RBC searches exceeding 100 ms. Finally, we are concluding, presenting how RBC has the potential to become mainstream, to generate defect free cryptographic keys from PUFs for networks of low power IoTs.

## 2 Background Information

### 2.1 Error Correction Methods for PUFs

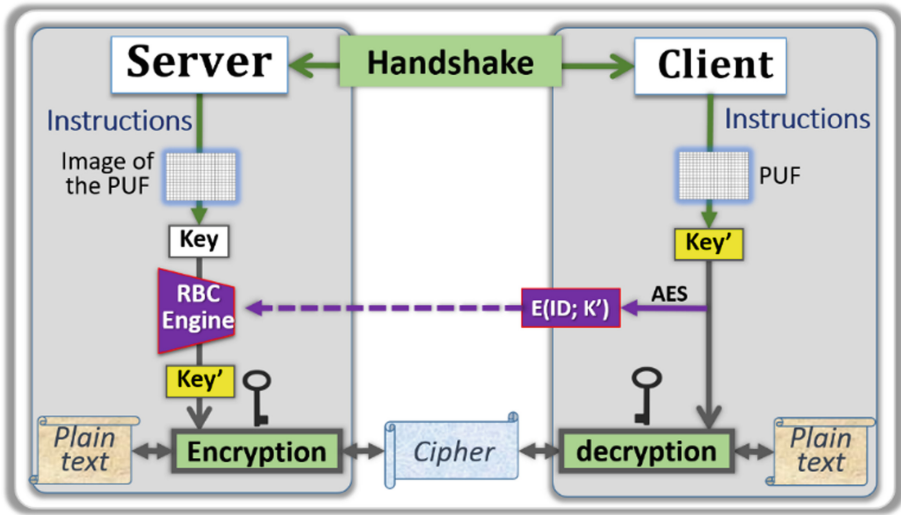
An example of architecture is shown in Fig. 1. The server downloads the “images” of the PUFs into look up tables during initial enrollment cycles [1]. The term “image” of the PUF is generic, and will have a different meaning for each PUF. In certain cases, the image stored in the server will be a set of challenge-responses describing the PUF. In this study, the image stored in the server is the set of preferential states of each cell of the SRAM memory array, zero or one, after repetitive power-off-power-on cycles. The “handshake”, is defined as a set of instructions generated by the server to independently find a particular address in the PUF, and concurrently read the same 256 cells on each side [28]; the keys generated from the PUF can contain errors. Each handshake could point to different addresses, and new keys. Hashing functions and multi-factor authentication are described in [29] to protect the handshake protocol. In the architecture shown in Fig. 1, the server generates the helper data [25] from the keys generated with the look up tables, and the ECC engine. The transmission of the helper is often protected by other cryptographic schemes. The client devices correct the keys generated from the PUF, with the helper data, and error correction schemes such as fuzzy extractors [23, 24]. The helper data is usually as long as the keys to correct. This method consumes computing power at the terminal level, which may not be available for certain IoT devices. Error correcting methods can also leak information to the opponents through side channel analysis.



**Fig. 1.** Example of PUF-based architecture with ECC. The keys generated from look up tables by the server differ from the ones generated from the PUF. The server send helper data to the client device, which uses ECC to retrieve the same key.

## 2.2 Response Based Cryptography

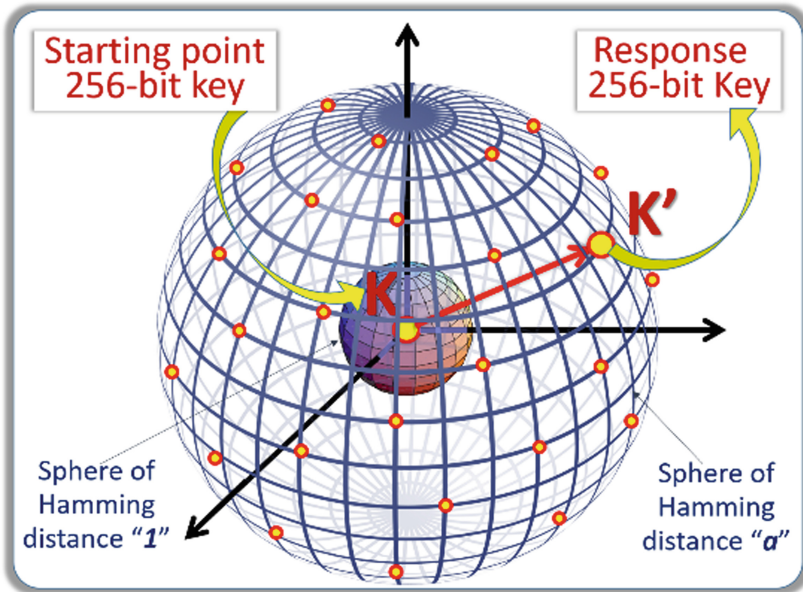
An architecture similar to the one presented Sect. 2.1, with RBC instead of ECC is shown in Fig. 2. After each handshake, both communicating parties independently generate their keys, the one extracted from the PUF contains errors. The client uses its key  $\mathbf{K}'$  to encrypt a user  $\mathbf{ID}$ , for example with the Advanced Encryption Standard (AES), and send to the server the cipher text  $\mathbf{E}(\mathbf{ID}; \mathbf{K}')$ . The RBC engine, compares this cipher text with  $\mathbf{E}(\mathbf{ID}; \mathbf{K})$ , the cipher text generated by its error free key  $\mathbf{K}$ . Both cipher texts are different unless the two keys are identical. The purpose of the RBC engine, which is described below, is to find in an iterative way the key generating the same cipher text  $\mathbf{E}(\mathbf{ID}; \mathbf{K}')$ , thereby uncovering  $\mathbf{K}'$  [26].



**Fig. 2.** Example of PUF-based architecture with RBC. The client device encrypt its user ID with its key  $\mathbf{K}'$ . The RBC engine retrieve this erratic key from its key  $\mathbf{K}$ , and  $\mathbf{E}(\mathbf{ID}, \mathbf{K}')$ .

The starting point of the algorithm used by the RBC search engine, as shown in Fig. 3, is the key  $\mathbf{K}$  extracted from the image of the PUF, and the cipher text  $\mathbf{E}(\mathbf{ID}, \mathbf{K}')$ , which is openly transmitted by the client device. If the key  $\mathbf{K}'$  is at the hamming distance " $a$ " of  $\mathbf{K}$ , the RBC has to test first all possible keys with Hamming distances of  $i$  from  $\mathbf{K}$ , with  $i \in \{0, a - 1\}$  by generating ciphers with each key, and comparing them with  $\mathbf{E}(\mathbf{ID}, \mathbf{K}')$ . The server repeats the process within the sphere of Hamming distance  $a$ . For 256-bit long keys, the number of possible keys located on this sphere is  $\binom{256}{a}$ , which is very large when  $a$  is greater than 4. The RBC is limited to PUFs with low defect densities, its latency becomes excessive when  $a$  is greater than 4, for 256-bit long keys, which correspond to a defect density of only  $4/256 = 1.5\%$ . The value of the combination .. is  $1.75 \cdot 10^8$ , if each matching cycle takes  $1 \mu\text{s}$ , the RBC latencies are in the three minute range, which is not acceptable. In this protocol, the cipher text  $\mathbf{E}(\mathbf{ID}; \mathbf{K}')$

is transmitted through unsecure communication channels, however, this is secure with cryptographic protocols such as AES-256. The only way to uncover  $\mathbf{K}'$  is to possess an image of the PUF. Previous work presented how the fragmentation of the keys expands the applicability of RBC to higher defect densities [30, 31], however it is desirable to minimize the level of fragmentation to reduce the computing power at the client level; this is the objective of this work. The RBC scheme could be sensitive to potential serious bias due to the lack of robustness of complex networks under attacks [32], which could add errors to the handshakes, and the cipher texts transmitted by the client devices. Mitigation of such attacks should be comprehended in a final implementation of the RBC. The inclusion of redundant correcting schemes is an example of possible remedy.



**Fig. 3.** Graphical representation of the RBC engine. The starting point is the key  $\mathbf{K}$  generated from the look up table. The iterative search is looking at the key  $\mathbf{K}'$  with Hamming distance “ $a$ ” from  $\mathbf{K}$ , by matching the ciphers.

### 3 Modeling the Latencies

#### 3.1 Effect of Random Defects on RBC Latencies

The RBC matching algorithm compares the cipher text sent by the client device  $\mathbf{E}(\mathbf{ID}, \mathbf{K}')$ , with the cipher text computed by the server. The cipher text of the client device is computed with the key  $\mathbf{K}'$  directly generated from the responses of the PUF. The initial cipher text of the server  $\mathbf{E}(\mathbf{ID}, \mathbf{K})$ , is computed with the key  $\mathbf{K}$  generated from the image of the PUF that is stored in the look up table as reference. If the two ciphers are different,

the server generates the 256 cipher texts from the 256 keys having a Hamming distance of one from the PUF challenges, and compares them to the cipher text transmitted by the client device from the responses.

The process is iterated with keys having higher Hamming distances to find the matching cipher. If the 256-long keys contain exactly  $X$  errors, and the latency of one encryption and matching cycle is  $\tau_o$ , the average latency  $L_{(X,1,256)}$  of the RBC search is given by:

$$L_{(X,1,256)} = \tau_o \left[ \sum_{i=0}^{i=a} \binom{256}{i} - \frac{1}{2} \binom{256}{X} \right] \approx \frac{\tau_o}{2} \binom{256}{X} \tag{1}$$

For example if  $X = 4$

$$L_{(4,1,256)} = \tau_o + \tau_o \binom{256}{1} + \tau_o \binom{256}{2} + \tau_o \binom{256}{3} + \frac{\tau_o}{2} \binom{256}{4} \tag{2}$$

$$\approx \frac{\tau_o}{2} \binom{256}{4} = 8.7410^7 \tau_o \tag{3}$$

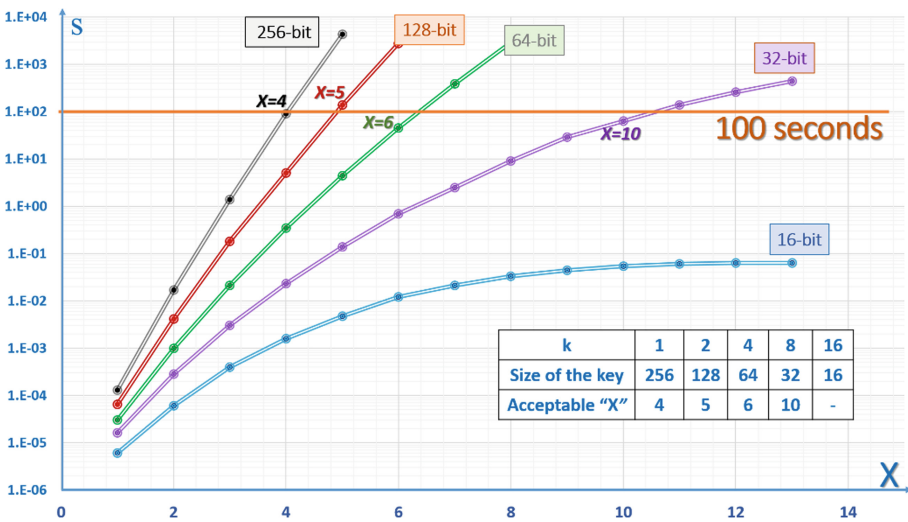
Using (1), the average RBC latencies  $L_{(X,1,256)}$ ,  $L_{(X,1,128)}$ ,  $L_{(X,1,64)}$ ,  $L_{(X,1,32)}$ , and  $L_{(X,1,16)}$  are computed, and summarized in Fig. 4, as a function of the number of error  $X$  per key, with an assumption that  $\tau_o = 1 \mu s$ . In the experimental work, we use generic AES-256 to generate the cipher texts from 256-bit long cryptographic keys, which has a latency of 500 ns with PC powered with quad-I7 chips from Intel.

$X$	% error	$L_{(X,1,256)}$	% error	$L_{(X,1,128)}$	% error	$L_{(X,1,64)}$	% error	$L_{(X,1,32)}$	% error	$L_{(X,1,16)}$
1	.39	1.3 E-4	.78	6.4 E-5	1.56	3.0 E-5	3.12	1.6 E-5	6.25	8.0 E-6
2	.78	1.7 E-2	1.56	4.1 E-3	3.12	1 E-3	6.25	2.8 E-4	12.5	6.0 E-5
3	1.17	1.4	2.34	1.8 E-1	4.69	2.1 E-2	9.38	3.0 E-3	18.7	4.0 E-4
4	1.56	9.0 E1	3.12	5.1	6.25	3.5 E-1	12.5	2.3 E-2	<b>NA: error rates out of range</b>	
5	1.95	4.4 E2	3.91	1.4 E2	7.81	4.4	15.6	1.4 E-1		
6	<b>Latencies are too long</b>		4.69	2.8 E3	9.38	4.5 E1	18.7	7.0 E-1		
7				10.9	3.9 E2					
8				12.5	2.9 E3					

Fig. 4. Average RBC latency with various  $X$ 's, and key size. With 256 & 128-bit long keys, the latencies with  $X$  greater than 5. With 32 & 16-bit long keys, the cryptography is exposed to brute force attacks.

The RBC algorithm is efficient when the error rates are small enough. The latencies of the RBC search engines are too slow to process PUFs with error rates higher than 1%, which is the case of most PUFs without other correcting methods. The RBC search fails when the latency exceeds a time limit. In this implementation, the server rejects the authentication of the client device when the latency exceeds 10 s, and initiates a new handshake. If the false reject rate FRR at each cycle is below 1%, and three attempts are judged acceptable, the cumulative FRR is below an acceptable 1 part per million (1 ppm) level. The maximum acceptable level of defect  $X$  for the RBC with 258-bit long keys is 3, which is acceptable with PUF technologies showing low defect rates. The SRAM-based PUFs used in the experimental section of this paper have defect rates below  $10^{-4}$  when the fuzzy cells are removed from the distribution during the enrollment cycle. The RBC scheme in this simple form is not applicable to mainstream SRAM-based PUFs having error rates in the 2–10% range.

Figure 5 is a set of graphs showing the value of the average RBC latencies  $L_{(X,I,256/k)}$  at various numbers of errors  $X$  with key lengths of  $256/k$ , in which  $k$  is respectively equal to 1, 2, 4, 8, and 16. Shorter keys have average RBC latencies that are much lower at equivalent error rates. For example, the RBC searches are always below 100 ms with 16-bit long keys. However, these shorter keys are not secure enough against brute force attacks.



**Fig. 5.** RBC latencies  $L_{(X,I,256/k)}$  as a function of  $X$ , and various key length: 256-bit ( $k = 1$ ), 128-bit ( $k = 2$ ), 64-bit ( $k = 4$ ), 32-bit ( $k = 8$ ), and 16-bit ( $k = 16$ ).

### 3.2 Poisson Distribution for RBC Latencies

For a given defect density, the number of bad bits vary key to key. Using Poisson distribution, with a density of error  $D$ , and a 256-bit long key, the coefficient  $\lambda = 256D$  is used to calculate the probability  $P_\lambda(X)$  to have  $X$  erratic bit in the key:

$$P_\lambda(X) = e^{-\lambda} \frac{\lambda^X}{X!} \tag{4}$$

For example if  $\lambda = 4$ , the probability  $P_\lambda(X)$  versus  $X$  (x-axis) is represented in Fig. 6:

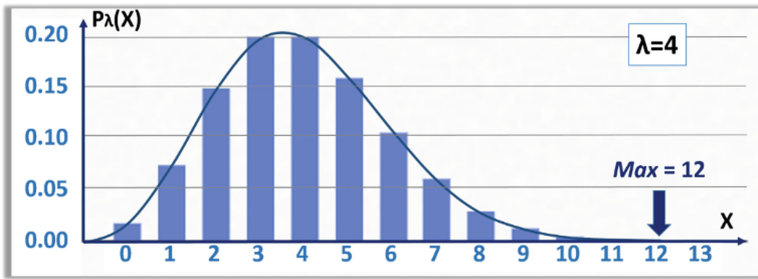


Fig. 6. Poisson distribution  $P_\lambda(X)$  when  $\lambda = 4$ .

When  $Max(\lambda)$  of  $X$  is defined as the value of  $X$  with meaningful probability to occur, the average RBC latency  $A_{(\lambda,1,256)}$  for the coefficient  $\lambda$ , and the average defect density  $D = \lambda/256$  affecting 256-bit long keys is given by:

$$\begin{aligned} A_{(\lambda,1,256)} &= \tau_0 \sum_{X=0}^{X=Max(\lambda)} P_\lambda(X) \cdot L_{(X,1,256)} \\ &= \tau_0 \sum_{X=0}^{X=Max(\lambda)} e^{-\lambda} \frac{\lambda^X}{X!} \left[ \sum_{i=0}^{i=X} \binom{256}{i} - \frac{1}{2} \binom{256}{X} \right] \\ &\approx \frac{\tau_0}{2} \sum_{X=0}^{X=Max(\lambda)} e^{-\lambda} \frac{\lambda^X}{X!} \binom{256}{X} \end{aligned} \tag{5}$$

The value  $Max(\lambda)$ , with  $\lambda$  varying from 1 to 6, is shown in Fig. 7. The parameter  $\lambda$  is a real number while the value of  $X$ , and  $Max(\lambda)$  are natural integer numbers.



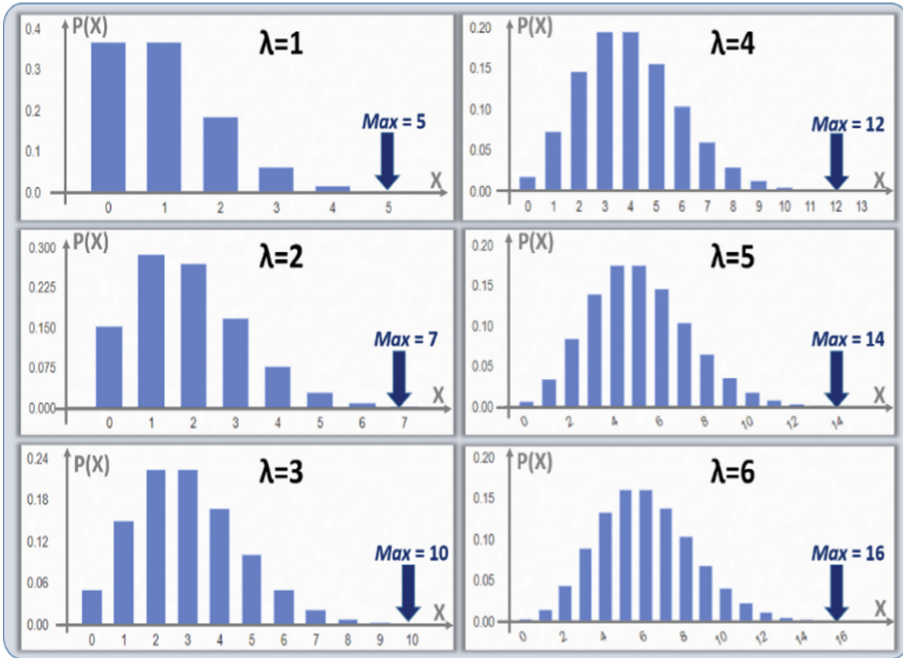


Fig. 7. Distribution  $P_\lambda(X)$  with  $\lambda = 1, 2, 3, 4, 5,$  and  $6$ .

## 4 Fragmentation into Sub-keys

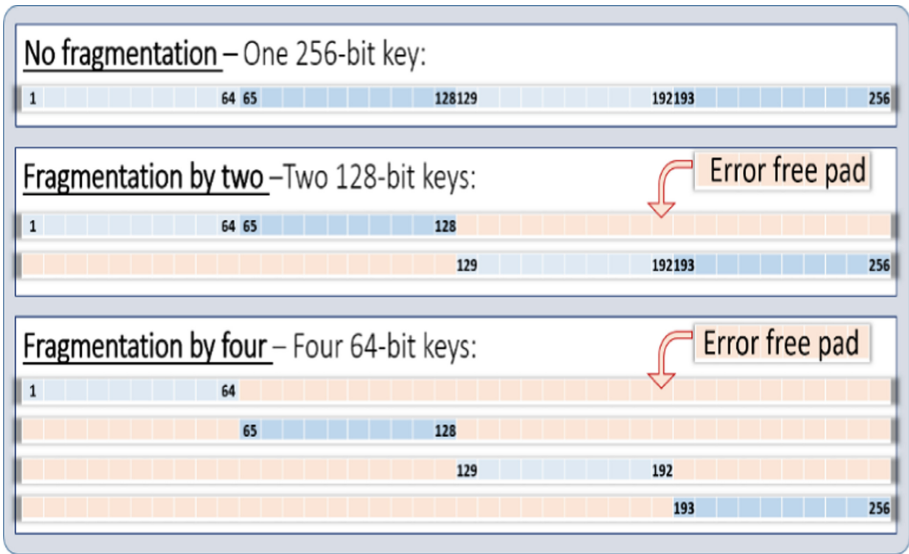
To reduce the latencies at higher error rates, we propose the fragmentation of the keys generated by the PUF into sub-keys, also 256-bit long, which are padded with random numbers.

### 4.1 Padding of the Sub-keys

In a fragmentation by two, the first sub-key is generated by keeping the first 128 bits of the 256-bit long key generated by the PUF, filled with a 128-bit long pad containing no errors. The last 128 bits of the PUF, also combined with a 128-bit long pad, generate the second sub-key. Statistically, the two sub-keys show error rates that are half those of full key error rates. In the RBC scheme, the client device sends two cipher texts generated by the two 256-bit long sub-keys. The RBC search engine can process the resulting two ciphers much faster to find the erratic key generated by the PUF. In a fragmentation by  $k \in \{2, 4, 8, 16\}$ , the first sub-key is generated by keeping the first  $256/k$  bits of the key generated by the PUF, filled with a  $(256 - 256/k)$ -bit long pad containing no errors. The subsequent  $k$  sub-keys are generated in a similar way. Such a method is shown in Fig. 8. The use of padding is widely adopted in many cryptographic schemes to enhance entropy [33]. The padding technology is of prime importance to be able to safely fragment the keys into sub-keys of equivalent strength, however, it is not part of the work presented in this paper, to study the effectiveness of various padding methods.



In the prototype developed here, the padding schemes are secretly shared between the communicating parties during the handshake cycles. The server XORed 512 randomly selected the position of the ternary states of the image of the PUF with the message digest, which is a result of the hashing of a salted random number. The salting technology is implemented with multi-factor authentication. This 512-bit long data stream is needed by the terminal device to eliminate the fuzzy cells of the PUF, and cherry pick the most stable cells of the PUF. The data stream is called a mask. Out of the 512-bit long mask, a 128-bit block is generated to pad the first 128-bit long sub-key, thereby resulting with a full 256-bit long key, in which only 128 bits can be impacted by an erratic PUF. The salted message digest changes during each handshake, and is kept secret. The fragmentation proposed in this paper is agnostic on the preferred padding technology, as long as it is error free, and contains enough entropy.



**Fig. 8.** Use of padding for key fragmentation. The padding use secret information exchanged during the handshake cycles.

### 4.2 Statistical Considerations for the Key Fragmentation

The parameter  $A_{(\lambda,k,256)}$  is defined as the average latency to find the matching keys with RBC, a Poisson coefficient  $\lambda$ , a fragmentation by  $k$  sub-keys, and 256-bit long keys. Each sub-key are filled with error free padding to form 256 keys. The average density of error per sub-key is  $D' = \frac{1}{k} \cdot D$ , with  $D$  being the average defect density observed on 256-bit keys before fragmentation. However, the average defect density affecting the fragment of the sub-keys generated by the original key is still  $D$ . Therefore, the coefficient of Poisson per sub-key is:  $\lambda' = \lambda/k$ . The corresponding distribution of probability  $P_{\lambda'}(X)$  given by (4).

To estimate the average latency  $A_{(\lambda,k,256)}$  for a group of  $e$  handshakes  $h$  with  $h \in \{1, e\}$ , is assumed that after handshake  $h$ , the  $k$  sub-keys  $K_i$ , with  $i \in \{1, k\}$ , have each a number of errors  $h(i)$ , which is following the distribution of Poisson with the coefficient  $\lambda/k$ . The RBC latency  $\mathcal{T}_h$  of handshake  $h$  is given by:

$$\begin{aligned} \mathcal{T}_h &= \sum_{i=1}^{i=k} L_{(h(i),1,256/k)} = \tau_0 \sum_{i=1}^{i=k} \left[ \sum_{j=0}^{j=h(i)} \binom{256/k}{j} - \frac{1}{2} \binom{256/k}{h(i)} \right] \\ &\approx \frac{\tau_0}{2} \sum_{i=1}^{i=k} \left[ \sum_{j=0}^{j=h(i)} \binom{256/k}{h(i)} \right] \end{aligned} \tag{6}$$

The average latencies  $A_{(\lambda,k,256)}$  for the group of  $e$  handshakes are summarized in Table 1, and given by:

$$\begin{aligned} A_{(\lambda,k,256)} &= \frac{1}{e} \sum_{h=1}^{h=e} \mathcal{T}_h = \frac{1}{e} \sum_{h=1}^{h=e} \sum_{i=1}^{i=k} L_{(h(i),1,256/k)} \\ &= \sum_{i=1}^{i=k} \frac{1}{e} \sum_{h=1}^{h=e} L_{(h(i),1,256/k)} \end{aligned} \tag{7}$$

When  $e$  is large enough, the  $k$  terms describing the average latency if the sub-keys  $K_i$ , with  $i \in \{1, k\}$ , of (7) are equal, and described by the Poisson distribution of (5):

$$\frac{1}{e} \sum_{h=1}^{h=e} L_{(h(i),1,256/k)} = A_{(\lambda/k,1,256/k)} \tag{8}$$

**Table 1.** Average latency of  $k$  sub-keys, after  $e$  handshakes.

Handshake	Sub-key					Sum
	K1	...	Ki	...	Kk	
1	$L_{(1(1),1,256/k)}$	...	$L_{(1(i),1,256/k)}$	...	$L_{(1(k),1,256/k)}$	$\mathcal{T}_1 = \sum_{i=1}^{i=k} L_{(1(i),1,256/k)}$
2	$L_{(2(1),1,256/k)}$	...	$L_{(2(i),1,256/k)}$	...	$L_{(2(k),1,256/k)}$	$\mathcal{T}_2 = \sum_{i=1}^{i=k} L_{(2(i),1,256/k)}$
...	---	...	---	...	---	---
h	$L_{(h(1),1,256/k)}$	...	$L_{(h(i),1,256/k)}$	...	$L_{(h(k),1,256/k)}$	$\mathcal{T}_h = \sum_{i=1}^{i=k} L_{(h(i),1,256/k)}$
...	---	...	---	...	---	---
e	$L_{(e(1),1,256/k)}$	...	$L_{(e(i),1,256/k)}$	...	$L_{(e(k),1,256/k)}$	$\mathcal{T}_e = \sum_{i=1}^{i=k} L_{(e(i),1,256/k)}$
Sum	$\sum_{h=1}^{h=e} L_{(h(1),1,256/k)}$	...	$\sum_{h=1}^{h=e} L_{(h(i),1,256/k)}$	...	$\sum_{h=1}^{h=e} L_{(h(k),1,256/k)}$	$\sum_{h=1}^{h=e} \mathcal{T}_h$
Av.	$A_{(\lambda/k,1,256/k)}$		$A_{(\lambda/k,1,256/k)}$		$A_{(\lambda/k,1,256/k)}$	$A_{(\lambda,k,256)} = kA_{(\lambda/k,1,256/k)}$

With  $A_{(\lambda/k,1,256/k)}$  been the RBC search latency of a 256/k bit long key with a coefficient of Poisson  $\lambda/k$ , and without fragmentation. Equation (7) can be written as:

$$A_{(\lambda,k,256)} = k \cdot A_{(\lambda/k,1,256/k)} \tag{9}$$

It is assumed that  $\mathbf{X} = \text{Max}(\lambda/k)$  is the highest number of error per 256/k bit long sub-key, with the meaningful probability to occur, following the distribution  $\mathbf{P}_{\lambda/k}(\mathbf{X})$ , and with the coefficient  $\lambda' = \lambda/k$ . The average latency  $A_{(\lambda,k,256)}$  can be written as:

$$A_{(\lambda,k,256)} = k \tau_0 \sum_{X=0}^{X=\text{Max}(\lambda/k)} \mathbf{P}_{\lambda/k}(\mathbf{X}) \left[ \sum_{i=0}^{i=X} \binom{\frac{256}{k}}{i} - \frac{1}{2} \binom{\frac{256}{k}}{X} \right] \quad (10)$$

With Poisson distribution, this is written as:

$$\begin{aligned} A_{(\lambda,k,256)} &= k \tau_0 \sum_{X=0}^{X=\text{Max}(\lambda/k)} e^{-\lambda'} \frac{\lambda'^X}{X!} \left[ \sum_{i=0}^{i=X} \binom{\frac{256}{k}}{i} - \frac{1}{2} \binom{\frac{256}{k}}{X} \right] \\ &\approx \frac{k \tau_0}{2} \sum_{X=0}^{X=\text{Max}(\lambda/k)} e^{-\lambda'} \frac{\lambda'^X}{X!} \binom{256/k}{X} \end{aligned} \quad (11)$$

### 4.3 Effectiveness of the Key Fragmentation

The effectiveness  $\eta$  of a fragmentation by  $k$  is given by the ratio of the latency before and after fragmentation:

$$\eta = A_{(\lambda,1,256)} / k A_{(\lambda/k,1,256/k)} \quad (12)$$

The effectiveness is summarized as follow:

- The average defect densities of each sub-key after fragmentation is  $D/k$ . The coefficient  $\lambda'$  is  $k$  time lower than  $\lambda$ , which pushes the entire Poisson population, including  $\mathbf{X} = \text{Max}(\lambda/k)$ , toward smaller values.
- The portions of the sub-keys for the RBC search with errors are  $k$  times smaller. The RBC latencies are computed with factors proportional to  $\binom{256/k}{i}$  versus  $\binom{256}{i}$ , which are much lower.
- The incorporation of a multiplication by  $k$  on (7) and (8) has a small impact, unless the defect densities are extremely small.

### 4.4 Predictive Model

Equations (10) and (12) are appropriate to model RBC average latencies, as a function of the defect density  $\lambda$ , and the level of key fragmentation  $k$ . The Poisson distribution at various level of defects is summarized in Table 2. The average latency as computed in Fig. 5 allows a rough estimate of the expected RBC average latencies:

- Without fragmentation,  $k = 1$ , the RBC probability  $P(X > 3)$ , in color in Fig. 11, is approximately 2% when  $\lambda = 1$ ; 15% when  $\lambda = 2$ ; and 40% when  $\lambda = 3$ . The latency for  $P(X = 3)$ , see Fig. 6, with  $\tau_0 = 1 \mu\text{s}$  is close to one second. Assuming that the acceptable latency is 10 s, and that the location of the defect density follows a Poisson distribution the maximum acceptable defect will be around  $\lambda = 2$ , a defect density  $D = 2/256$  of about 1%.

- With a fragmentation by two,  $k = 2$ , the RBC latency for  $P(X = 3)$  is 10 times lower, and the latency for  $P(X = 4)$  is below ten seconds. The defect density of  $\lambda = 2$ , is  $D = 2/(256/2) \approx 1.6\%$ , and the maximum acceptable defect density is about 2%.
- With a fragmentation by four,  $k = 4$ , the RBC latency for  $P(X = 5)$  is below ten second. The defect density of  $\lambda = 3$ , is  $D = 3/(256/4) \approx 4.6\%$ , and the maximum acceptable defect density is about 6%.
- With a fragmentation by eight,  $k = 8$ , the RBC latency for  $P(X = 7)$  is below ten seconds. The defect density of  $\lambda = 5$ , is  $D = 5/(256/8) \approx 15.6\%$ , and the maximum acceptable defect density is about 16%.

With a fragmentation by 16,  $k = 16$ , the RBC latencies stays below ten seconds. The scheme should be able to handle any defect density.

**Table 2.** The color code of the table is showing when a given fragmentation results in RBC latencies below 10 s; gray for  $k = 1$ ; orange for  $k = 2$ ; purple for  $k = 4$ ; blue for  $k = 8$  and 16.

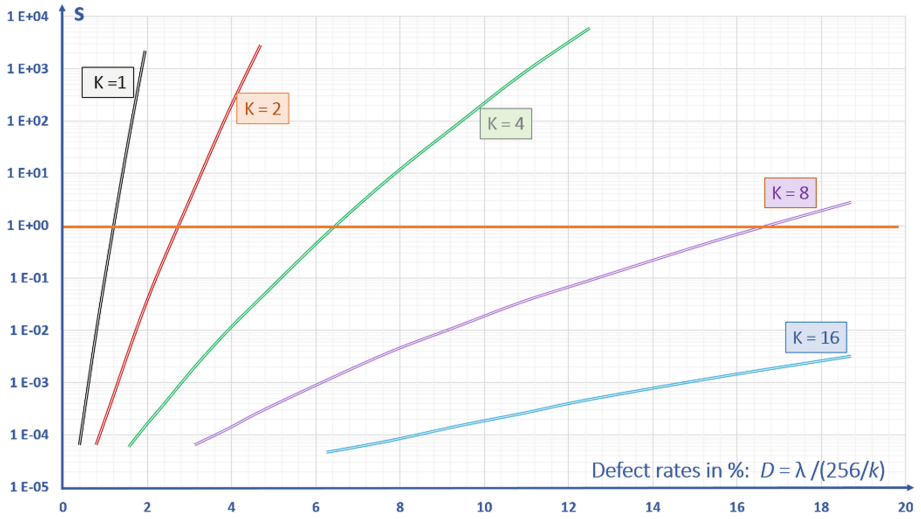
P(X) %	$\lambda=1$	$\lambda=2$	$\lambda=3$	$\lambda=4$	$\lambda=5$	$\lambda=6$
X=0	37	14	5	2	0.6	0.2
X=1	37	27	15	7	3.3	1.5
X=2	18.4	27	22	15	8.4	4.5
X=4	6	18	22	20	14	9
X=4	1.5	9	17	20	17.5	13
X=5	0.3	3.6	10	16	17.5	16
X=6		1.2	5	10	15	16
X=7		0.4	2.2	6	10	14
X=8			0.8	3	6.5	10
X=9			0.3	1.3	3.6	6.9
X=10				0.5	1.8	4.1
X=11				0.2	0.8	2.2
X=12					0.3	1.1
X=13					0.1	0.5
X=14						0.2

A simplified model is proposed based on (9), to estimate the average RBC latencies for integers  $\lambda = 1, 2, 3, 4$ , and 5, and with  $k = 1, 2, 4, 8$ , and 16:

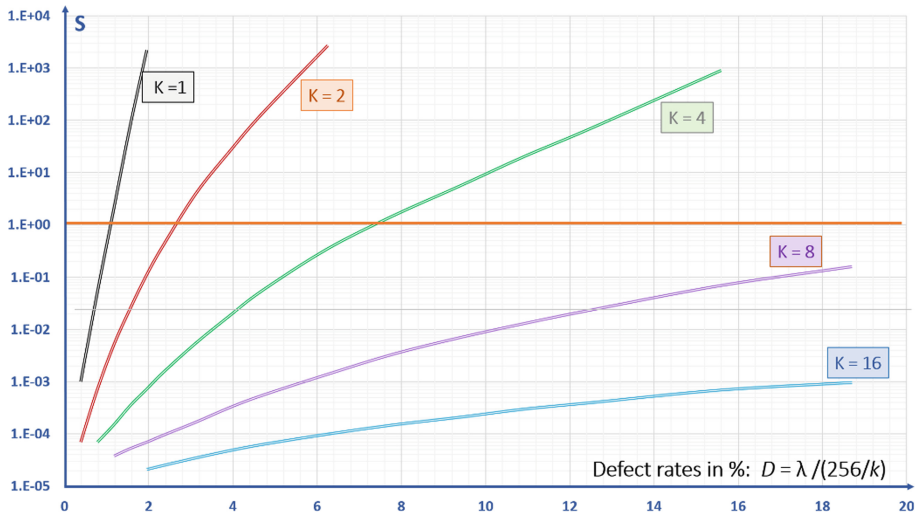
$$A_{(\lambda,k,256)} = k \cdot A_{(\lambda/k,1,256/k)} \approx k \frac{\tau_0}{2} \binom{256/k}{\lambda} \tag{13}$$

These estimated RBC latencies  $A_{(\lambda,k,256)}$ , as a function of the defect densities  $D = \lambda/(256/k)$ , are plotted in Fig. 9.

These estimated RBC latencies (y-axis) using Eq. (11) with Poisson distribution, as a function of the defect densities  $D = \lambda/(256/k)$ , (x-axis) are plotted in Fig. 10.



**Fig. 9.** Modelling the RBC latencies,  $A_{(\lambda,k,256)} \approx k \frac{\tau_0}{2} \left( \frac{256/k}{\lambda} \right)$ , based on the natural values of  $\lambda$  between 1 and 10. The curves are fitted around these values.



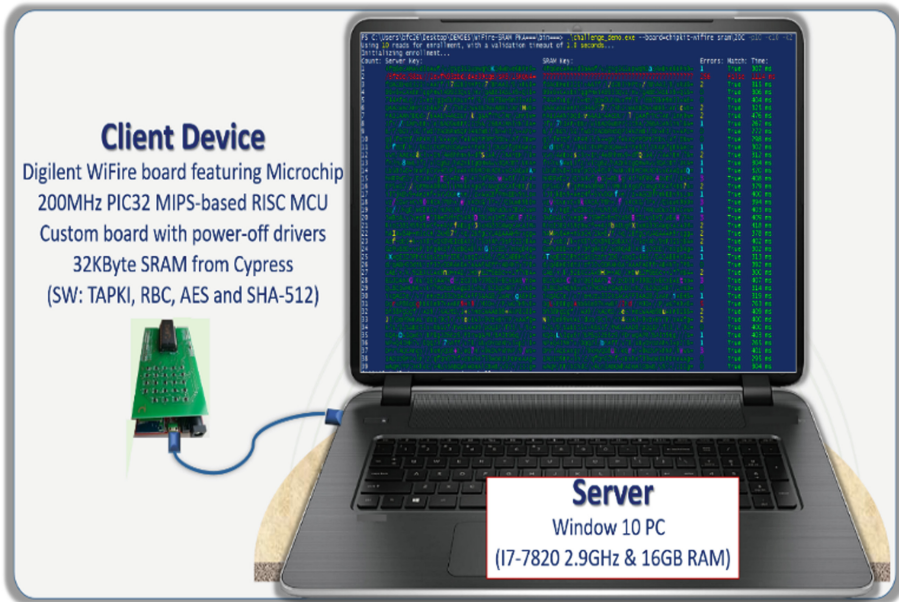
**Fig. 10.** Modelling the RBC latencies with Poisson distribution and  $A_{(\lambda,k,256)} = k A_{(\lambda/k,1,256/k)}$ .

## 5 Experimental Validation

### 5.1 Development Board

The experimental set up is shown in Fig. 11. To validate experimentally the effectiveness of the RBC with fragmentation, commercially available 32-Kbyte SRAM devices

from Cypress semiconductor were used as PUFs. During enrollment, the SRAMs are submitted to power-off-power-on cycles, and the resulting patterns are stored in look up tables. About half of the flip-flop of the SRAM cells are mainly responding to such power-off cycles as a “0” state, about half as a “1” state. The error rates of such PUFs are in the 2% to 10% range, due to the cells that have instability in their responses.



**Fig. 11.** Development board with SRAM-based PUF, and 200 MHz RISC MCU. The PC with 2.9 GHz quadcore processor performs the RBC searches.

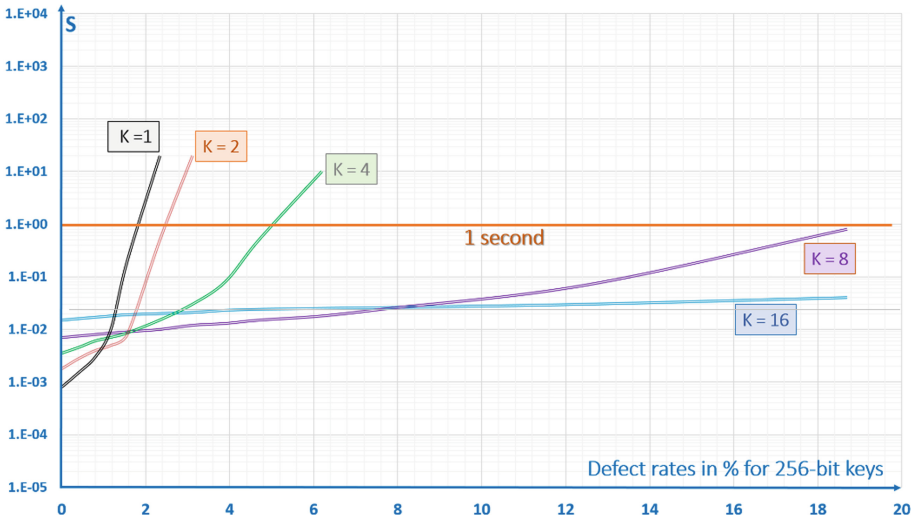
The error rates can be reduced to levels below  $10^{-5}$  by cycling the SRAM multiple times during enrollment, and by eliminating the cells that are not consistent. During 1,000 power-off cycles, about 20% of the cells are showing instability, while the remaining 80% are responding without errors. The error rates of the SRAM PUFs can be adjusted by tracking the position of the cells in the array, and selecting the group of cells with defect densities between 0%, and 20%. The mapping of the cells, and their respective defect density is stored in the server in look up tables. The client device does not memorize any of the enrollment information; it is assumed that the client device can be lost to the enemy. The SRAM PUFs, which are used in this analysis, are not tamper resistant; however, they are appropriate for this experimental work.

WiFire development boards from Digilent, powered by Microchip, were used to drive the SRAM PUFs. These boards contain 200 MHz 32-bit RISC microcontrollers from MIPS, ADC/DAC converters, 2 MB flash, and 512 KB RAM. The embedded software and cryptographic protocols to extract 256-bit keys from the PUFs were written in C, with software implementation of AES-256, and SHA-256.

On the server side, Window 10 PCs powered by Intel I7 quad core processors were used, they can process an AES cycle in less than one microseconds. The cryptographic protocol is randomly pointing at 256 cells in the PUF, every 2 s; the RBC search engine operates with various levels of fragmentation, and PUF defect rates. For each configuration fragmentation-level of defect, the PC typically performed 100 reads with different handshakes and randomly selected 256-bit long keys, and stored the resulting data for analysis.

### 5.2 Experimental Data

To acquire statistically significant data, the RBC latencies are averaged over 100 handshakes, and the PUF error rates vary from 0 to 20% by increment of 1%. For each level of error rate, and new handshake, the server, and the client device independently generate 256-bit long keys. The RBC search engine of the server experimentally measure the latency needed to find a key matching the one generated by the PUF, with fragmentations by 1, 2, 4, 8, and 16. The data used in Fig. 12 was generated with the measurement of 500,000 cells randomly selected, and 10,000 cycles of RBC search.



**Fig. 12.** Experimental measurement of RBC average latencies with 256 Kbit SRAM-based PUF, 200 MHz RISC MCU development board, and Window-10 PC with 2.9 GHz I7 quad.

The results of the experimental work can be summarized as follows:

- At low defect densities, below 1% rate, the quad-core processor of the PC faces delays due to initialization cycles of 1 ms, and the management of the multi-tasking operations of the PC. The fragmentation increases proportionally with these initialization cycles up to 20 ms for  $k = 16$ . In this range, the RBC is faster without fragmentation, and associated overhead.

- Above defect densities of 1.5%, the fragmentation schemes are needed to keep RBC latencies acceptable. A fragmentation by two is enough to handle defect densities of 2.5%, a fragmentation by four handles 5%, and a fragmentation by 8 handles defect densities higher than 16%.
- The fragmentation by 16 can handle any level of defects within less than 100 ms.

The SRAM-based PUFs characterized have defect densities in the 2% to 8% range, therefore a fragmentation by 8 yields consistently reliable RBC searches. With elimination of the erratic cells, and defect densities in the  $10^{-5}$  range, the fragmentation schemes are not needed.

### 5.3 Comparison Between Experimental and Models

Below 20 ms latencies, the model does not take into consideration the response time of the PC; above 10 s, the latencies are not acceptable to a normal user case. Therefore, the back-to-back comparison between experimental and modelling data is focusing in the 100 ms to 10 s range, as shown in Table 3:

**Table 3.** Defect densities needed to reach a given latency. Experimental versus model.

Defect densities to reach 0.05s to 10s latencies		$k=1$	$k=2$	$k=4$	$k=8$	$k=16$
@0.05s	Experimental	1.3%	1.9%	3.5%	11%	
	Simple Model	0.9%	1.9%	4.7%	11.5%	
	Poisson	0.8%	1.7%	4.5%	14.5%	-
@0.1s	Experimental	1.4%	2.0%	4.0%	13%	-
	Simple Model	1.0%	2.2%	5.2%	12.8%	-
	Poisson	0.9%	1.9%	5.2%	16.8%	-
@1.0s	Experimental	1.7%	2.4%	5.0%	18.5%	-
	Simple Model	1.2%	2.7%	6.3%	16.8%	-
	Poisson	1.2%	2.6%	7.2%	-	-
@10s	Experimental	2.2%	3.0%	6.4%	-	-
	Simple Model	1.5%	3.3%	7.7%	-	-
	Poisson	1.4%	3.5%	10%	-	-

- The defect densities needed to reach latencies between 0.1 s and 10 s measured experimentally are roughly similar to the ones anticipated by the model.
- For  $k = 1$ , about 1% defect rates requires 0.1 s latency for an RBC search, 1.5% requires 1 s, and 2% requires 10 s. The measurements are slightly faster than the model at the same defect density.



- For  $k = 2$ , about 2% defect rates requires 0.1 s latency for an RBC search, 2.5% requires 1 s, and 3% requires 10 s. The measurements are slightly slower than the model at the same defect density.
- For  $k = 4$ , about 4% defect rates requires 0.1 s latency for an RBC search, 5% requires 1 s, and 7% requires 10 s. The measurements are slower than the model at the same defect density.
- For  $k = 8$ , about 13% defect rates requires 0.1 s latency for an RBC search, 18% requires 1 s, and all other defect densities are matched by the RBC engine in less than 10 s. The measurements are slightly faster than the model at the same defect density.
- Both experimental data, and modelling data, show that a fragmentation by 16 always yield successful searches in less than 100 ms, regardless of the defect densities.

## 6 Conclusion and Future Work

The objectives of the statistical models developed in this work to predict accurately the efficiency of key fragmentation where achieved. The experimental measurements based on SRAM-based PUFs are validating these models, when the latencies are greater than 100 ms, to alleviate the inherent latencies of the PC.

The resistance based cryptography is applicable to PUFs with defect densities below 1% (for 256-bit keys), which is the case when fuzzy cells are mapped during enrollment, and removed during key generation. This eliminates the need to use error-correcting methods, helper data, and thereby simplifies PUF-based cryptographic protocols, and enhances security of networks of low power internet of things. Methods to fragment PUF-generated keys enhance the effectiveness of RBC algorithms; a fragmentation by 8 can handle error rates in the 15% range.

The statistical models developed in this work can be used for the following applications:

- Design and optimization of networks of power-constrained IoTs, secured by PUFs. The level of fragmentation is minimized to comprehend the quality of the PUFs used in the system. The level of fragmentation can be coupled with machine learning algorithms to follow the aging of the PUFs, and drifting environments.
- Unequally powered cryptographic systems to protect terminals subjected to variable threats, with noise injection. Noise can be added to the keys generated by the client device to increase the difficulty of the RBC search. High Performance Computers (HPC) are then used at the server level placing at a disadvantage opponents with inferior computing power.
- Enhancement of the digital signature schemes to secure the blockchain technology. A network of distributed IoTs will use the PUF technology to generate public-private key pairs. Each IoT generate public keys from the keys extracted from the PUFs, and the RBC engine of the certificate authority validate the public keys.

We see the need to complete this research work with various PUFs, which may not have the same distribution of errors as SRAM-based PUFs. The focus is on tamper resistant PUFs to enhance security when the client devices are under side channel analysis.

We are studying alternative statistical distribution beside Poisson distribution, which will describe the behavior of various PUFs. We are also conducting experiment with High Performance Computing (HPC), and the implementation of parallel computing to further reduce latencies.

Finally, the deployment of the technology to industry will require the design of custom secure microcontroller, protecting the client devices from side channel analysis. The statistical models enable the development, and optimization of improved RBC schemes mitigating various attacks.

**Acknowledgments.** The authors are thanking the Information Directorate of the US Air Force Research Laboratory (AFRL) for their support of this research work. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFRL. The authors are also thanking the contribution of several graduate students at Northern Arizona University, in particular Vince Rodriguez, and Ian Burke.

## References

1. Papakonstantinou, I., Sklavos, N.: Physical unclonable functions (PUFs) design technologies: advantages and trade offs. In: Daimi, K. (ed.) *Computer and Network Security Essentials*. Springer, Cham (2018). ISBN 978-3-319-58423-2
2. Herder, C., Yu, M., Koushanfar, F.: Physical unclonable functions and applications: a tutorial. *Proc. IEEE* **102**(8), 1126–1141 (2014)
3. Pappu, R., Recht, B., Taylor, J., Gershenfeld, N.: Physical one-way functions. *Science* **297**(5589), 2026–2030 (2002)
4. Maes, R., Verbauwhede, I.: Physically unclonable functions: a study on the state of the art and future research directions. In: Sadeghi, A.R., Naccache, D. (eds.) *Towards Hardware-Intrinsic Security*. Springer, Heidelberg (2010)
5. Jin, Y.: Introduction to hardware security. *Electronics* **4**, 763–784 (2015). <https://doi.org/10.3390/electronics4040763>
6. Gao, Y., Ranasinghe, D., Al-Sarawi, S., Kavehei, O., Abbott, D.: Emerging physical unclonable functions with nanotechnologies. *IEEE* (2016). <https://doi.org/10.1109/ACCESS.2015.2503432>
7. Rahman, M.T., Rahman, F., Forte, D., Tehranipoor, M.: An aging-resistant RO-PUF for reliable key generation. *IEEE Trans. Emerg. Top. Comput.* **4**(3), 335–348 (2016)
8. Guajardo, J., Sandeep, S.K., Geert, J.S., Pim, T.: PUFs and PublicKey crypto for FPGA IP protection. In: *Field Programmable 2007*, pp. 189–195 (2017)
9. Holcomb, D.E., Burleson, W.P., Fu, K.: Power-up SRAM state as an Identifying Fingerprint and Source of TRN. *IEEE Trans. Comput.* **57**(11) (2008)
10. Christensen, T.A., Sheets II, J.E.: Implementing PUF utilizing EDRAM memory cell capacitance variation. Patent No.: US 8,300,450 B2, 30 October 2012
11. Plusquellic, J., et al.: Systems and methods for generating PUF's from non-volatile cells. WO20151056887A1 (2015)
12. Prabhu, P., Akel, A., Grupp, L.M., Yu, W.-K.S., Suh, G.E., Kan, E., Swanson, S.: Extracting device fingerprints from flash memory by exploiting physical variations. In: *4th International Conference on Trust and Trustworthy Computing* (2011)
13. Chen, A.: Comprehensive Assessment of RRAM-based PUF for Hardware Security Applications. 978-1-4673-9894-7/15/IEDM IEEE (2015)

14. Cambou, B., Afghah, F., Sonderegger, D., Taggart, J., Barnaby, H., Kozicki, M.: Ag conductive bridge RAMs for physical unclonable functions. In: 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), McLean, USA (2017)
15. Cambou, B., Orłowski, M.: Design of physical unclonable functions with ReRAM and ternary states. In: Cyber and Information Security Research Conference, CISR 2016, Oak Ridge, TN, USA (2016)
16. Korenda, A., Afghah, F., Cambou, B.: A secret key generation scheme for Internet of Things using ternary-states ReRAM-based physical unclonable functions. In: International Wireless Communications and Mobile Computing Conference (IWCMC 2018) (2018)
17. Vatajelu, E.I., Di Natale, G., Barbaresi, M., Torres, L., Indaco, M., Prinetto, P.: STT-MRAM-based PUF architecture exploiting magnetic tunnel junction fabrication-induced variability. *ACM J. Emerg. Technol. Comput. Syst. (JETC)* **13**(1), 1–21 (2015)
18. Zhu, X., Millendorf, S., Guo, X., Jacobson, D.M., Lee, K., Kang, S.H., Nowak, M.M., Fazla, D.: PUFs based on resistivity of MRAM
19. Becker, G.T., Wild, A., Güneysu, T.: Security analysis of index-based syndrome coding for PUF-based key generation In: 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), Washington, DC (2015)
20. Boehm, H.M.: Error correction coding for physical unclonable functions. In: Austrochip 2010, Workshop in Microelectronics (2010)
21. Maes, R., Tuyls, P., Verbauwhede, I.: A soft decision helper data algorithm for SRAM PUFs. In: 2009 IEEE International Symposium on Information Theory (2009)
22. Chen, T.I.B., Willems, F.M., Maes, R., van der Sluis, E., Selimis, G.: A robust SRAM-PUF key generation scheme based on polar codes. [arXiv:1701.07320](https://arxiv.org/abs/1701.07320) [cs.IT] (2017)
23. Taniguchi, M., Shiozaki, M., Kubo, H., Fujino, T.: A stable key generation from PUF responses with a Fuzzy Extractor for cryptographic authentications. In: IEEE 2nd Global Conference on Consumer Electronics (GCCE), Tokyo, Japan (2013)
24. Kang, H., Hori, Y., Katashita, T., Hagiwara, M., Iwamura, K.: Cryptographic key generation from PUF data using efficient fuzzy extractors. In: 16th International Conference on Advanced Communication Technology, Pyeongchang, Korea (2014)
25. Delvaux, J., Gu, D., Schellekens, D., Verbauwhede, I.: Helper data algorithms for PUF-based key generation: overview and analysis. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **34**(6), 889–902 (2015)
26. Cambou, B., Philabaum, C., Booher, D., Telesca, D.: Response-based cryptographic methods with ternary physical unclonable functions. In: 2019 SAI FICC. IEEE, March 2019
27. Cambou, B., Philabaum, C., Duane Booher, D.: Response-based Cryptography with PUFs, NAU case D2018-049, June 2018
28. Cambou, B., Flikkema, P., Palmer, J., Telesca, D., Philabaum, C.: Can ternary computing improve information assurance? *Cryptography* **2**(1), 6 (2018)
29. Cambou, B., Telesca, D.: Ternary computing to strengthen information assurance, development of ternary State based public key exchange. In: IEEE, SAI 2018, Computing Conference, London, UK (2018)
30. Cambou, B.: Unequally powered cryptography with PUFs for networks of IoTs. In: IEEE Spring Simulation Conference, May 2019
31. Cambou, B., Philabaum, C., Booher, D.: Replacing error correction by key fragmentation and search engines to generate error-free cryptographic keys from PUFs. In: CryptArchi 2019, June 2019
32. Shang, Y.: Subgraph robustness of complex networks under attacks. *IEEE Tans. Syst. Man Cybern. Syst.* (2019)
33. European Payments Council: Guideline on cryptographic algorithms usage and key management. EPC342-08, November 2017



# Towards an Intelligent Intrusion Detection System: A Proposed Framework

Raghda Fawzey Hriez<sup>1</sup>, Ali Hadi<sup>2</sup>, and Jalal Omer Atoum<sup>3</sup>(✉)

<sup>1</sup> King Hussein School of Information Technology, Princess Sumaya University for Technology, Amman, Jordan

r.hraiz@psut.edu.jo

<sup>2</sup> Computer and Digital Forensics, Champlain College, Burlington, USA

ahadi@champlain.edu

<sup>3</sup> Department of Mathematics and Computer Science, East Central University, Ada, OK, USA

jomer@ecok.edu

**Abstract.** With the fast increase in network connectivity and reliance on information systems, the number of sophisticated threats has increased rapidly, hence demanding the development of intelligent security protection systems that are resilient to these new threats. This research has been conducted as an improvement to the Intrusion Detection Systems (IDS) detection methodology; it aims to design not only a framework for an intrusion detection system but also to make this system interact intelligently. The proposed IDS could self-customize itself to adopt different network topologies and network traffic situations and serve as a self-learner, which is a feature not seen in most commercial and open-source intrusion detection systems.

**Keywords:** IDS · Intrusion detection system · Intelligent agent · Network security · Threat detection

## 1 Introduction

Millions of worms, viruses and other malware are created every day, according to McAfee, the percentage of spam in email traffic in the third quarter of 2017 reached 55.9%, also there were more than 57.6 million new malware samples, the WannaCry attacks alone infected more than 300,000 computers in over 150 countries in less than 24 h [1]. Adversaries are relentlessly targeting every corner of the digital world, from computer networks to bank's websites, spreading to social networks and mobile devices. Their attacks didn't steal information or damage systems only, but costed the global economy as much as 400 billion dollars a year [2].

Researchers, specialists, and companies are continuously trying to create and develop various tools to respond to these vicious onslaught attacks targeting today's digital systems. Intrusion Detection Systems (IDS) are considered as a critical component for network security which can deal with internal and external attacks. They support security specialists with the ability to perform real-time security monitoring and identify

abnormal patterns. Many research papers have been conducted to improve the quality of intrusion detection systems, but they still have their problems, especially when it comes to high false-positive ratios, operational issues in high-speed environments, working with encrypted traffic and the difficulty of detecting unknown threats.

In this paper, a new intrusion detection framework is proposed that could not only detect intrusions but benefit from intelligent agents' structures and knowledge base systems to detect new and unknown attack patterns. This will improve the security posture of the guarded network and provide further confidence to network security engineers about their system's detection capability. The proposed intrusion detection framework will serve as a self-learner, which is a feature not seen in most commercial and open-source intrusion detection systems.

## 2 Related Work

IDSs are classified into three categories based on their approach of detecting and analyzing intrusions: Misuse/Signature-based IDS, Anomaly-based IDS, and hybrid-based IDSs.

Signature-based IDSs are based on a set of signatures that identify hostile traffic of known security threats. These signatures are used to analyze the data streams traversing through the network; when a flow matches a signature, the flow is marked as an intrusion [3].

Signature-based IDSs are used widely because they are effective and accurate in combating against the known security threats. In addition, they are easy to use and have low false-positive rates. However, they remain completely ineffective against unknown attacks, because signature-based IDSs can combat attacks only if their signatures are known. Therefore, signature-based IDSs require updating the signatures database regularly [4, 5].

The misuse detection concept traditionally implemented using a rule-based approach, but many data mining approaches have been suggesting automating the process of rules generation and pattern extraction. Other researchers use data mining techniques to build effective models to classify traffic and recognize malicious data instead of rules.

Using a rule-based approach involves preparing a set of rules that describe the behavior of known security threats. The rules are used to analyze the data streams traverse through the network. If the data in the traffic matches the rule, it is considered as intrusion and appropriate action is taken. The main advantage of this approach is the low false-positive rate achieved and high accuracy in detecting known attacks. In addition, this approach facilitates tracking down the cause of an alarm. It provides detailed information about the current attack. The attack information is specified when the rules have been written. On the other hand, rule-based techniques can only detect intrusions when their details are available. Unknown attacks can't be detected because there are no rules for them in the rule base, so the rule base needs to be updated constantly. Also inspecting packets to form rules is expensive in terms of time, money, and resources. Security experts would be required to study and analyze the malicious traffic, write down observations, findings, and patterns, then write the corresponding rule. Some researchers suggest the use of machine learning techniques such as association rules, evolutionary computation and fuzzy logic to automate and enhance the process of rules generation.

Association rules mining can be used to analyze malicious network traffic and find correlation between attributes, hence be used to extract attacks patterns and forming rules. Using Association rules mining for intrusion detection reduces the efforts done by security experts to analyze attacks and write corresponding rules [6]. Alternatively, the amount of data needed to extract useful patterns is huge. It should contain enough samples for each type of attack. Another issue with applying association rules mining for intrusion detection is that it often generates lots of irrelevant rules and redundant rules because of the similarity properties found within network traffic.

Another issue with rule-based IDSs is that it can be evaded if the attacker applies slight changes in his intrusion behavior; because these IDSs use strict signature matching systems, and any slight change in the pattern will lead to an evasion of detection system. As a countermeasure, fuzzy logic is used to make the IDS more flexible, and give it the ability to recognize attacks even though their behavior has slightly changed [7]. While the concepts of partial truth and uncertainty of fuzzy logic will improve the flexibility of the IDS, it requires tremendous efforts to build the fuzzy system. For each input, multiple fuzzy sets should be defined with their corresponding membership functions. Also, a set of fuzzy rules should be prepared to be used in the fuzzy inference process [8].

Some researchers suggested the use of evolutionary computation algorithms for misuse intrusion detection. Evolutionary algorithms are effective in rules devising and preparing process. The problem with this technique is the mapping between the intrusion detection problem and the solution (evolutionary computation algorithms), how to convert the rules into chromosomes, what fitness function to use, how to generate the random initial generation. Authors in [9] and [10] use the principles of evolution in a Genetic algorithm (GA) to generate the rules for misuse IDS. GA starts with randomly generated rules; these rules will evolve until a set of very effective rules is selected. Author in [11] used artificial neural network (ANN) and fuzzy clustering to enhance the detection accuracy and stability for low-frequent intrusions. Their proposed procedure consists of three stages: first, a fuzzy clustering technique is used to generate multiple training subsets. Based on these training sets, different ANNs are trained to generate base models in the second stage. These models will have lower complexity. This will enhance the learning process and make it more accurate and robust, especially for low-frequent attacks. In the third stage; the different results are aggregated using a fuzzy aggregation module. This approach is promising in detecting already known attacks, and it raises the detection rate for infrequent attacks in the training data set, However, this IDS will face some problems in detecting unknown attacks, because if new behavior (normal or abnormal) is seen and differs from what the fuzzy clustering algorithm learned, the IDS will fail in classifying this event correctly.

The second approach for intrusion detection is anomaly-based IDSs. Anomaly-based IDSs raises alarm when the detected object is falling outside a predefined description of normal behavior. It expects that malicious activities are different from normal activities. It relies on the distance between the two behaviors [12]. This approach is promising because it can detect unknown attacks since it doesn't rely on known attacks signatures. Alternatively, anomaly-based IDSs are capable of classifying network traffic as normal or abnormal only; hence, it can not specify the type of the attack even if it is an already

known attack. In addition, using anomaly detection increases the false-positive rate, because some normal data shows the same patterns as malicious data, and because the systems and network are changes every day, what is normal today may become abnormal and vice versa. Anomaly detection can be implemented using three main techniques, rule base detection, data mining models and statistical detection.

Using a rule-based in anomaly detection requires preparing a set of rules that describes each possible normal behavior. Rules could be written by experts or automatically generated using association rules mining, fuzzy logic, and evolutionary computation algorithms. The main problem with this approach is the difficulty of designing a rule base that could completely cover all types of normal behavior. The insufficient number of rules will increase the false-positive rate because any normal activity that didn't have a corresponding rule in the rule base will be considered as an intrusion [7]. As in Misuse rule-based intrusion detection, association rules and fuzzy logic could be used to enhance the system. Author in [8] proposed an anomaly-based intrusion detection system using fuzzy logic and association rules mining. The proposed approach starts by splitting the dataset into two sets; training data and testing data. The training data will be mined to generate fuzzy rules. The generated rules will be stored in the fuzzy rule base which is an important part of the fuzzy system. In the test phase, the fuzzy system will process the testing data and classify it as either normal or abnormal data. This method utilizes the flexibility and uncertainty of fuzzy logic without the need for a security expert to design and write the fuzzy rules; it reduces the efforts and time required in building the IDS. Alternatively, it didn't specify the type of attack even if it is already known. It only raises an alarm when an intrusion is detected, without any information about the intrusion.

Data mining algorithms can be trained to generate effective models that can identify and recognize normal behavior. Those models are more flexible than rule-based detection and easier to implement and can learn from incomplete data. This approach faces two challenges: features selection and data set preparation. To train machine learning algorithms, a set of effective features should be defined and extracted from network traffic. If the selected features are trivial or not strongly affecting the result of the classification, the generated model will be inaccurate and will generate a high false-positive rate. Also, the data set which will be used to train the data mining algorithm should contain a good diverse of normal activities and should be authentic. If the used data set contains malicious data, the generated model will consider this malicious data as a normal activity, and fail in detecting such an attack [7]. Author in [12] proposed multiple classifier systems for accurate payload-based anomaly detection, the system uses a combination of One-Class SVM Classifiers to learn the normal behavior based on some payload statistics. The research aims to achieve high detection rate and low false-positive rate in shellcode detection, especially the case of polymorphic attacks, where the shellcode is capable of creating different and multiple variants of malicious code before sending it to the victim, so it's very hard to be detected by signature-based IDS. The third type of anomaly detection is statistical anomaly detection. It uses statistics to differentiate between normal and abnormal behavior. It can be implemented as threshold detection or profile-based systems. Threshold detection requires calculating the number of occurrences of an event or attribute over an interval of time. If the number is greater

than the predefined threshold value, the intrusion is assumed and an appropriate alarm is raised [13]. While threshold detection is good in detecting flooding and probing attacks. It is not efficient in the detection of other types of intrusions. Profile-based systems require creating a model for normal user behavior, then comparing the recent behavior of the user with this model. Any significant deviation is considered as intrusion [14]. The profile for normal user behavior should balance between detection rate and false-alarm rate. A wide profile can decrease the detection rate, whereas a narrow profile can cause a high rate of false alarms [7].

To overcome the shortages and problems of misuse IDSs and anomaly-based IDSs, researchers have suggested the use of a hybrid IDS. They aim to integrate the flexibility and intelligence of anomaly detection methods with the accuracy and reliability of misuse detection methods. When implementing hybrid IDSs, two main issues should be considered: selecting suitable methods for anomaly and misuse detection. Since there are various methods available for each type of detection, selecting good pairs is challenging and may be very difficult. The second issue is determining the integration framework that specifies how the two methods work together. There are several available integration framework methods.

Anomaly–misuse sequence is designed to reduce the false-positives rate by excluding alarms that are not classified as alarms by the misuse detection system. But this sequence will cancel the role of Anomaly detection; it is equal to the use of misuse detection only. Misuse–anomaly sequence aims to detect unknown intrusions missed by misuse detection systems, but this approach cannot solve the problem of a high false-positive rate. Parallel detection allows each type of detection to work alone, and then it correlates the results to provide a stronger detection decision [7].

### 3 Proposed System Overview

Intelligent agents have the property of adaptability and scalability; they are able to learn from their own experiences and environment [15]. This research utilizes the concepts and features of intelligent agents and applies them in the intrusion detection field.

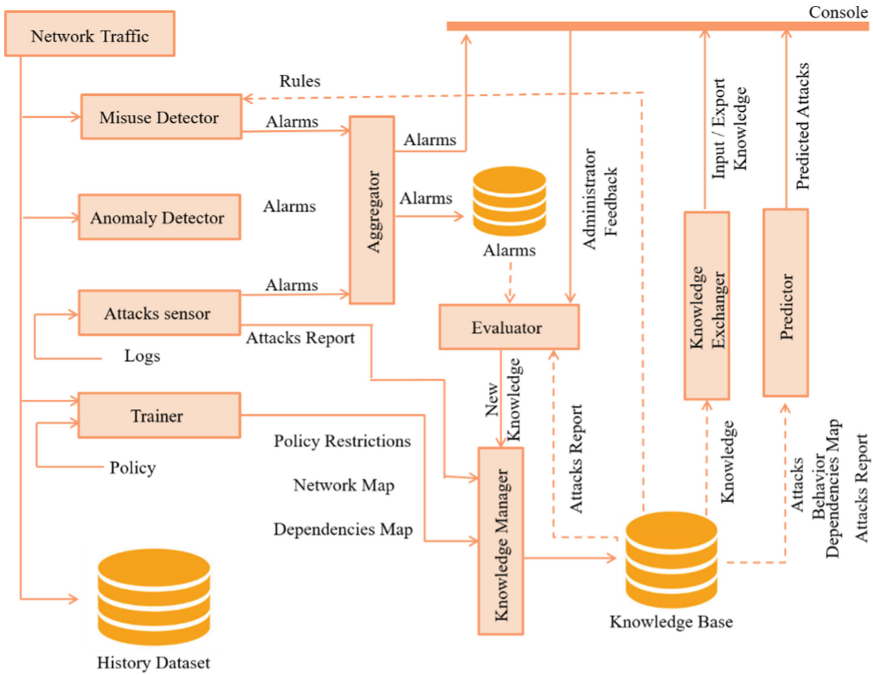
Intelligent agents' structures vary from a simple Table-driven agent to a utility-based agent. Table-driven agents are very simple agents that depend on percept/action lookup tables, whereas utility-based agents have a goal and aim to achieve it [15].

In this paper, the proposed intrusion detection system is a utility-based agent that aims to maximize the detection rate and reduce the false-positive rate. It evaluates its decisions and learns from the results, then changing its behavior to enhance the detection in the future.

### 4 System Components

Intelligent Intrusion Detection System (IIDS) consists of nine components and three data stores; Fig. 1 shows the architecture of the proposed IIDS and the relation between its components.





**Fig. 1.** Proposed system components.

The following subsections provide a detailed description of each component.

#### 4.1 Trainer

This component is responsible for customizing the IDS to make it suitable for the containing environment. The trainer accomplishes the following tasks:

- 1) **Network Discovery:** The trainer will draw a complete map of the network, by identifying the properties of each asset in the network. The map contains the IP address, MAC address, role (DNS server, HTTP server, host/workstation, etc.), services and running the operating system of each device in the network.

The trainer sends the map to the Knowledge Manager where it will be used to add a suitable set of rules to detect abnormal flows; for example, if device A is a host/workstation according to the map, and the IDS detect DNS request sent to A. This activity will be considered as intrusion, even though the DNS request is not malicious in itself, but sending it to a normal host is the abnormal activity.

The map is also valuable for the anomaly detector component. The anomaly detector computes how many packets of a specific type sent to/from critical devices, such as the number of HTTP GET requests that have been sent to the HTTP servers in a specific period. The anomaly detector needs the map to know the IP address of the HTTP servers, DNS server, etc.

## 2) Policy understanding:

The trainer extracts the following information from the security policy of the organization:

- a. Illegal interaction with digital systems.  
Some organizations prevent the employees from using the organization's digital resources for their benefits, for example, some organizations do not allow browsing social media websites during official working hours, other determines internet download limit, and such restrictions should be extracted from the acceptable use policies for the IDS to raise an alarm when any violation occurs.
- b. Holiday and vacation schedule and work time  
The trainer extracts holiday and vacation schedules and official working schedules because the traffic rate in holidays differs from the rate during a normal workday. In addition, some critical devices should not be accessed outside the work time.
- c. Important security-relevant events' schedules  
The IDS should know the schedule of any security-relevant event such as scanning and penetration testing, etc. to avoid generating alarms or dropping such traffic. To extract this vital information, the trainer should have the ability to understand the structure of the used security policy template, or it could ask the administrator to enter this information using GUI or predefined XML format, or the trainer can use text analysis techniques to extract this knowledge from these policies. After that, the trainer will send it to the Knowledge Manager to update other components.
- d. Extract entities' dependencies and relations.  
Some devices in the network cannot accomplish their tasks without the help of other devices, for example it common to find web application servers depending on a backend database server or server to answer client's requests. The trainer determines these dependencies and relations by analyzing traffic flows between devices. After preparing a dependency map, the trainer will send it to the knowledge manager, so that it will be available for the predictor component which needs this map to complete its work.

## 4.2 Knowledge Manager

Knowledge Manager is the component responsible for manipulating and managing knowledge of IIDS. It receives the requests for change from other components and applies the requested update on the knowledge base. Each time it receives a request it records the modification event. This will facilitate knowledge exchange and knowledge recovery operations. It also provides a good reference for the knowledge evolving process.

## 4.3 Misuse Detector

This component is one of the two core components of IIDS; the aim of using the misuse detector is to benefit from its accuracy and reliability in detecting known attacks.

Instead of creating a new methodology for misuse detection, rule-based IDSs such as snort can be used, because it is one of the most commonly used IDSs in signature-based intrusion detection and prevention systems and they proved their effectiveness in detecting known attacks.

#### 4.4 Anomaly Detector

This vital component is designed to detect unknown attacks that can evade the misuse detector. Unlike misuse detectors, it does not need any information about the attack behavior. It has good knowledge about the normal behavior and any behavior differs from this learned normal behavior, will be detected and marked as an attack.

The proposed anomaly detector is a protocol-based anomaly detector, where for each protocol a complete anomaly detector should be designed. Figure 2 illustrates the complete picture of the anomaly detector. Each anomaly detector consists of three parts, RFC Validator, Data Mining Classifier and statistics anomaly detector as shown in Fig. 3.

The RFC validator compares the traffic to the standard specifications (RFC), which is a technical and organizational document that describes the structure and the content of each protocol (Request for Comments (RFC)) [17]. If the traffic violates these specifications it will be considered abnormal. This type of validation is not enough, because some malicious traffic falls within these specifications and standards. To detect such attacks, data mining techniques are used to learn the normal behavior and generate flexible models that can recognize normal behaviors. Besides the data mining models, statistical anomaly detectors are needed, because, in some attacks, the abnormal property is not in the content of the packet but in the total number of packets.

The IIDS anomaly detector is designed as a protocol-based anomaly detector to reduce the false-positives rate, which is usually high in anomaly detection techniques. Learning each protocol alone will simplify the learning process and make it more focused in detection because the protocols' traffic differs from each other. When data mining techniques try to learn all these protocol characteristics together it will be confused and generate inaccurate models, because what is considered normal in HTTP may not be normal in SMTP, and the distinctive attribute that can be used to distinguish the normal traffic from the abnormal traffic in HTTP, differs from the distinctive attribute of SMTP, so it's better to create separated anomaly detector for each protocol.

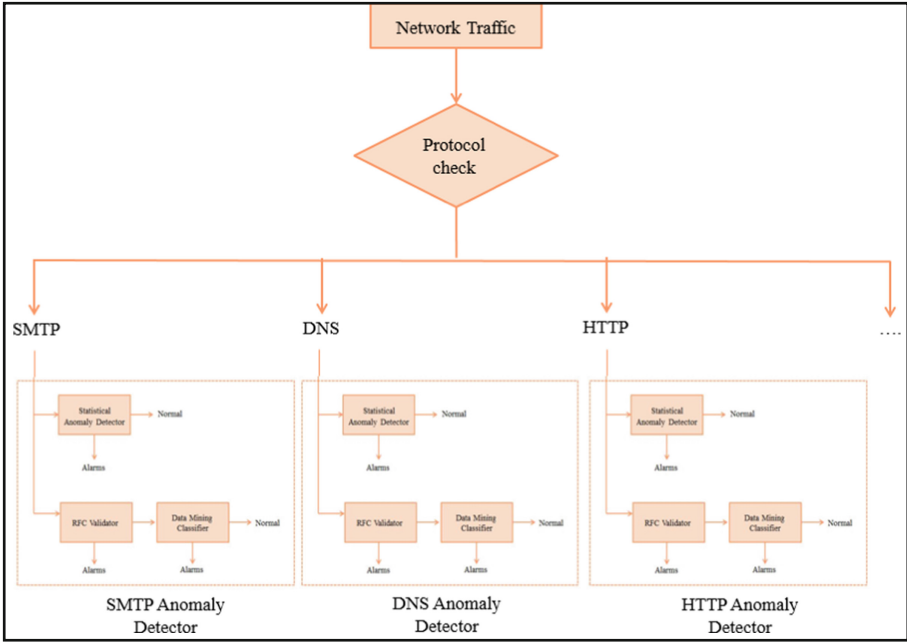


Fig. 2. Anomaly detector.

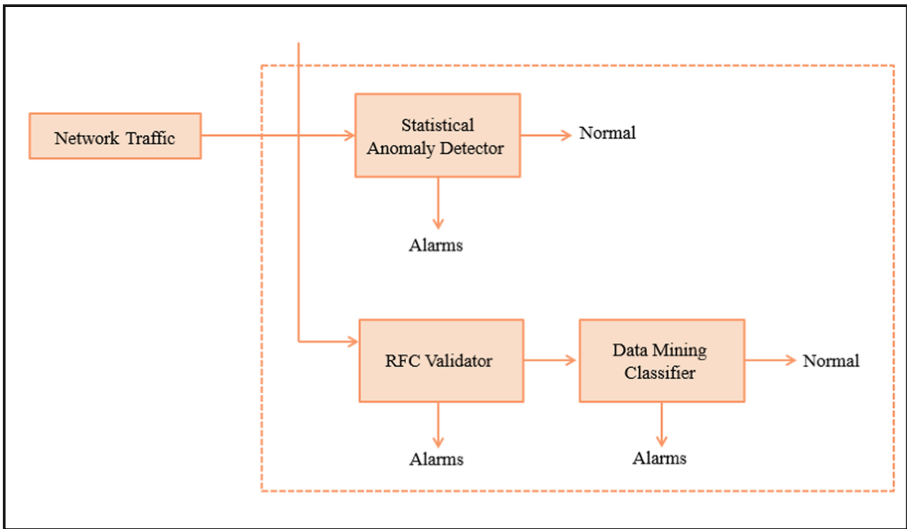


Fig. 3. Anomaly detector components.

## 4.5 Attacks Sensor

This component is responsible for generating attacks' report which summarizes the attacks that happened in the network and has not been detected by the anomaly detector and misuse detector. The detection methodology that is used relies on analyzing logs searching for attacks' results. It receives logs from different important applications and anti-virus software running on selected devices within the network. It then searches through these logs to find any error, failure, suspicious programs detected by anti-virus or any other indication of attacks' existence.

Attacks sensor is different from the intrusion detection systems that detect attacks based on logs. Those IDSs search in the logs for attack behavior, whereas attacks sensor, searches for attacks results. For example, to detect Denial of service attacks, IDSs will search for a huge number of requests sent to the application in a short amount of time, where attacks sensor will search for resource limitation warning messages in the logs.

The second objective of attacks sensor is to generate a report that describes the current state of the network and provides a list of attacks that targeted the network hosts. This report is important for the evaluator because it provides the answer to "what my actions do?", also it is needed by the predictor to know the current state of the network.

## 4.6 Evaluator

The evaluator could be considered as the brain of the proposed IIDS. It is responsible for learning from previous threat experiences and intelligently detects and discovers new threat patterns.

Initially, the evaluator determines if the previous decisions that have been taken by the IIDS are correct or not. This is done by comparing the attacks' report and CIRT team's feedback with the previously raised alarms. CIRT team's feedback contains a list of alarms marked as false-positive by the CIRT teams and another list of alarms marked as solved by the CIRT team. The solved alarms will be considered correct decisions and false-positive alarms will be considered as wrong decisions. Other alarms, the evaluator will compare it with the attacks report to know if they were correct or not. After determining the correctness of previous decisions, the evaluator will benefit from this experience and update the knowledge of the IIDS, this is done as follows:

If the decision was wrong, the evaluator will check the type of failure; false-positive or false-negative. False-positive means that there was normal traffic identified as an intrusion by the IIDS, and false-negative means that the IIDS didn't detect an intrusion [18].

If the type of the failure is false-positive, the evaluator will check the source of this false alarm, if the source is the rule-based misuse detector, the evaluator will update the ruleset. If the source is the anomaly detector, the IIDS will update the parameters of the used data mining technique. Alternately, if the type of failure is a false-negative, the evaluator will notify the anomaly detector and change the parameters of the data mining technique. Also, the evaluator will try to form a special rule for the new attack, by analyzing the traffic and extracting distinctive patterns. Figure 4 illustrates the overall process.

To be adaptable to changes, the evaluator generates new data mining models periodically. It stores samples of the traffic with its accurate labels and uses data mining algorithms to extract new features from this new and accurately classified traffic. After that, the evaluator generates new classification models using these new features and data set. The new model will stay under test until the evaluator ensures its maturity. Once it's mature, it will be used instead of the old model.

#### 4.7 Predictor

Predictor aims to predict what type of attacks and failures will happen in the future, and which host will be the target of the attack. This information will help the CIRT team in protecting hosts and responding to security incidents. The predictor generates predictions based on the current state of the network, dependency map and predefined knowledge that summarizes the attackers' behavior. The current state of the network is provided by the attacks sensors report, alarms raised by the misuse detector and the anomaly detector and CIRT team's feedback. The current state will help the predictor in deciding which hosts are infected or crashed. Dependencies map is important because if host A depends on host B to accomplish its task, and host B is down, then host A will not be able to complete its tasks. In addition, the predictor will rely on a set of rules defined based on the known behavior of attackers. Those rules will be identified by studying multiple attacks' scenarios and analyzing their phases and steps.

#### 4.8 Knowledge Exchanger

As any intelligent agent, the knowledge of IIDS will be evolved over time. IIDS will be consistently learning about attacks and intrusions behaviors. The knowledge exchanger component is responsible for sharing this vital knowledge with other IIDSs. This component will give the IIDS the ability to export and import knowledge.

### 5 Evaluation

This section evaluates the IIDS according to a set of requirements that should be met during the designing and building phases. These requirements are: adaptability, effectiveness in detection, and interoperability.

- 1) Adaptability: this requirement is required for the IDS, to compete with the changes in the network; the following features improve the adaptability of the IIDS:
  - a. The trainer will scan the network periodically to discover any change in the network map and update the rules according to the discovered change.
  - b. The evaluator will update the knowledge rules and data mining parameters based on the evaluation results.
  - c. The knowledgebase contains lists of commonly used values for some protocol fields. These lists are used in some data mining features in the anomaly detector, for example, the set of common methods for HTTP is {GET, POST, HEAD,

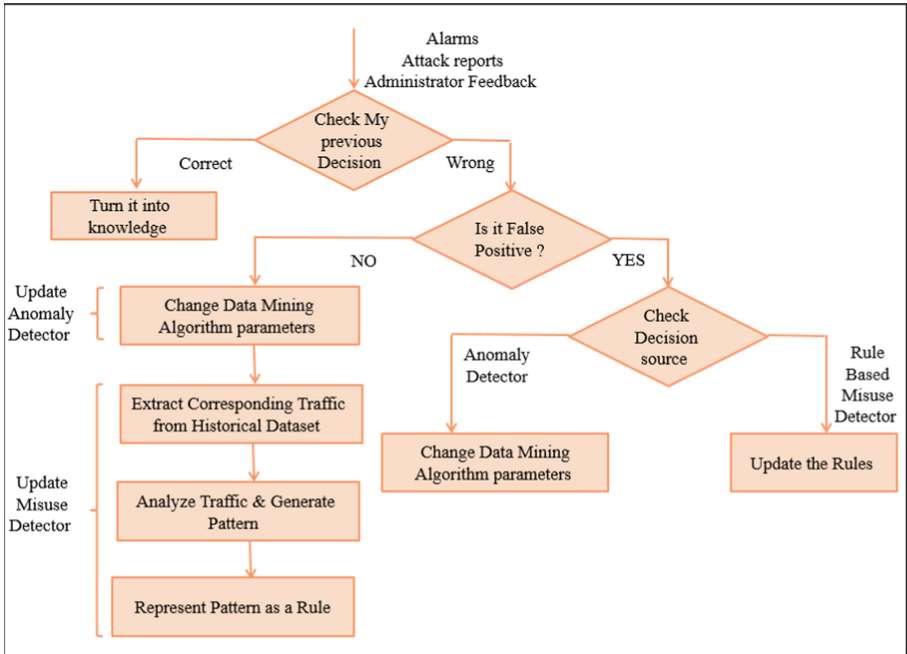


Fig. 4. Evaluator.

CONNECT, OPTIONS}, this set could be extended in the future. When the evaluator discovers a new common value, it will be added to the list. In addition, the administrator can add values to these lists.

- d. The design of the RFC validator can be enhanced, by allowing the administrator to store the validation in XML format, and then the RFC validator will validate the traffic according to the XML file. This will allow the administrator to update the IIDS when a new RFC version is published.
  - e. The evaluator will choose new features and build new models periodically, because the features which were considered robust today may become ineffective and useless in the future.
- 2) Interoperability: This requirement is indicated by the ability of IDS in working with other IDSs [16]. The proposed IIDS has the ability to exchange knowledge with other IIDSs via the knowledge exchanger component. In addition, the IIDS can interpolate with the conventional rule-based IDSs, because it has a misuse detector and a rule set and it can interchange the rules with them. Also, it can exchange knowledge with IDSs that use data mining techniques. IIDS can benefit from the set of features used by these IDSs, the evaluator can evaluate these features and use them to build new models.
  - 3) Effectiveness: This requirement represents the accuracy of IDS in attack detection. It can be measured by different metrics, such as: detection rate, false-positive rate [16]. The IIDS is expected to have a high detection rate because the misuse detector will

detect the known attacks and the anomaly detector will detect unknown attacks. In addition, the detection rate of the IIDS will be enhanced over time, because IIDS can learn from the bad experience, and create new patterns and new customized model for the misclassified attacks. The false-positive of IIDS will depend on the anomaly detector effectiveness, because most of the IDSs, which use anomaly detector, have a high false-positive rate. However, to reduce the false-positive rate, the IIDS employs a protocol-based anomaly, where a model for each protocol is created. Which expected to reduce the false-positive rate.

## 6 Conclusion

This research proposes a new intrusion detection framework that is capable of intelligently detect and discover new threat patterns. The proposed IDS can self-customize itself to adopt different network topologies and network traffic situations and can be trained to improve its detection accuracy, by learning from previous experiences.

## References

1. McAfee. Threats Report. mcafee.com, September 2017. [ps://www.mcafee.com/enterprise/en-us/assets/infographics/infographic-threats-report-sept-2017.pdf](https://www.mcafee.com/enterprise/en-us/assets/infographics/infographic-threats-report-sept-2017.pdf)[pdfs://www.mcafee.com/enterprise/en-us/assets/infographics/infographic-threats-report-sept-2017.pdf](https://www.mcafee.com/enterprise/en-us/assets/infographics/infographic-threats-report-sept-2017.pdf)[ps://www.mcafee.com/enterprise/en-us/assets/infographics/info](https://www.mcafee.com/enterprise/en-us/assets/infographics/info)
2. Cyber Crime Costs Projected to Reach \$2 Trillion by 2019, 17 January 2016. <https://www.forbes.com/sites/stevemorgan/2016/01/17/cyber-crime-costs-projected-to-reach-2-trillion-by-2019/#7e8423463a91>
3. Dhurpate, N.B., Lobo, L.M.R.J.: Network intrusion detection evading systems using frequent pattern matching. *Int. J. Eng. Trends Technol. (IJETT)* **4**(8), 3571–3575 (2013)
4. Bace, R., Mell, P.: NIST Special Publication on Intrusion Detection Systems (2001)
5. Kumar, V., Sangwan, O.: Signature based intrusion detection system using SNORT. *Int. J. Comput. Appl. Inf. Technol.* **1**(3), 35–41 (2012)
6. Tsai, F.: Network intrusion detection using association rules. *Int. J. Recent Trends Eng.* **2**(2), 202 (2009)
7. Dua, S., Du, X.: *Data Mining and Machine Learning in Cybersecurity*. Auerbach Publications, New York (2011)
8. Shanmugavadivu, R., Nagarajan, N.: Network intrusion detection system using fuzzy logic. *J. Comput. Sci. Eng. (IJCSSE)* **2**(1), 101–111 (2011)
9. Mbikayi, H.K.: An evolution strategy approach toward rule-set generation for intrusion detection systems (IDS). *Int. J. Soft Comput. Eng. (IJSCE)*, 201–205 (2012)
10. Hashemi, V., Muda, Z., Yassin, W.: Improving intrusion detection using genetic algorithm. *Inf. Technol. J.* **12**(11), 2167–2173 (2013)
11. Wang, G., Huang, L., Hao, J., Ma, J.: A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Syst. Appl.* **37**(9), 6225–6232 (2010)
12. Perdisci, R., Ariu, D., Fogla, P., Giacinto, G., Lee, W.: McPAD: a multiple classifier system for accurate payload-based anomaly detection. *Comput. Netw.* **53**(6), 864–881 (2009)
13. Kumar, S.A.P., Kumar, A., Srinivasan, S.: Statistical based intrusion detection framework using six sigma technique. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **7**(10), 333–342 (2007)



14. Abraham, A., Grosan, C., Martin-Vide, C.: Evolutionary design of intrusion detection. *Int. J. Netw. Secur.*, 328–339 (2007)
15. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 3rd edn. Prentice Hall, Upper Saddle River (2009). 200
16. Axelsson, S.: The base-rate fallacy and its implications for the difficulty of intrusion detection. *ACM Trans. Inf. Syst. Secur. (TISSEC)*, 186–205 (2000)
17. Instructions to Request for Comments (RFC) Authors. [rfc-editor.org](https://www.rfc-editor.org), 1 August 2004. <https://www.rfc-editor.org/old/instructions2authors.txt>
18. Intrusion Detection Overview. [pearsonitcertification.com](http://www.pearsonitcertification.com), 18 June 2004. <http://www.pearsonitcertification.com/articles/article.aspx?p=174342>



# LockChain Technology as One Source of Truth for Cyber, Information Security and Privacy

Yuri Bobbert<sup>1,2(✉)</sup> and Nese Ozkanli<sup>3</sup>

<sup>1</sup> Antwerp Management School, Antwerp, Belgium

yuri.bobbert@uantwerpen.be

<sup>2</sup> ON2IT B.V., Waardenburg, The Netherlands

<sup>3</sup> Open University, Heerlen, The Netherlands

nese.ozkanli@gmail.com

**Abstract.** Implementing and maintaining Information Security (IS) in a digitized ecosystem is cumbersome. Multiple complex frameworks and models are used to implement IS, but these are perceived as hard to implement and maintain in digitized dynamic value chains and platforms. Most companies still use spreadsheets to design, direct and monitor their information security function and demonstrate their compliance. Regulators too use spreadsheets for supervision. This paper reflects on longitudinal Design Science Research (DSR) on IS and describes the design and engineering of an artefact architecture, coined as LockChain, which can emancipate boards from silo-based spreadsheet management and improve their visibility, control and assurance via integrated dash-boarding and a reporting tool. LockChain is not a traditional Information Security Management System (ISMS) but is used for the design and specification of information security requirements and measures and privacy requirements. We elaborate “Why” we used Design Science Research into valorisation of the concept of LockChain, we explain “What” we have established in terms of the technology of LockChain and “How” it is applied and the added value LockChain brings for companies on cost savings, Security and Privacy by Design engineering culture and Digital Assurance.

**Keywords:** Information security controls · Security requirements · Security measures · Security by design · Privacy by design · Digital assurance

## 1 Introduction

When starting this research journey in 2008, security was mainly IT-oriented and the main focus was on using IT controls to mitigate or detect security vulnerabilities. Research has shown that the number of security incidents has increased [1] over the years, as has the financial impact per data breach [1]. Mastering emerging technologies such as big data, Internet of Things [2], social media and combating cybercrime [3], while protecting critical business data, requires a team instead of a single IT person [4]. To protect this data, security professionals need to know about the value of information and the impact if it is threatened [4]. IT risk management requires different capabilities, knowledge and expertise from the skills of IT security professionals [5]. Hubbard [5] refers to the failure

of ‘expert knowledge’ in impact estimations and to the importance of experience beyond risk and IT security, such as collaboration and reflection [6].

## 1.1 Problem Statement

In the past [7] IT security controls were implemented based on best practices prescribed by vendors, without a direct link to risks or business objectives [7]. These controls depended on technology and the audits and assessments (in spreadsheets) were used to prove their effectiveness [8]. The problem with this approach lay in the limitations of mainly IT-focused security and security experts working in silos with limited, subjective views of the world [9]. This is important, as information security is subject to many different interpretations, meanings and viewpoints [10]. In the case of IS, this refers to interactions and reflection between actors e.g. the business, data owners and industry peers on the appropriate level of risk appetite and security maturity [9]. Thus objectivity relates to reality, “truth reliability”, testability and reproducibility, while subjectivity refers to the quality of personal opinions. Intersubjectivity involves the agreements between social entities and the sharing of subjective states by two or more individuals [11].

The state of security in 2010 shifted towards “information security”. ISO specifies information security as “*protecting information assets from a wide range of threats in order to ensure business continuity, minimise business risk and maximise return on investment and business opportunities*” [12]. Its core principles are Confidentiality, Integrity and Availability (CIA) [12]. Later non-repudiation and auditability were added to comply with audit and compliance regulations. Thus Information Security should ensure a certain level of system quality and assurance [13]. In 2010 many organisations used spreadsheets to practice risk and security management and also proof their assurance via spreadsheets [14, 15].

The scope of Information Security was then expanded to other disciplines in the enterprise since digital became more and more common in our way of doing business [16]. In their book “Information Security Governance”, Von Solms and Von Solms describe the growing number of disciplines involved in IS [17]. By 2011 IT managers and IT security managers were increasingly urged to engage with business to determine risk appetite and the desired state of security. In 2005 ITGI proposed to co-develop IS together with the business [4]. Since 2011, the role of culture [10], awareness [18], compliance [19] and knowledge sharing [9] has also been included in security strategy frameworks [20]. Due to research on IT governance at the Antwerp Management School (AMS) [21], relational mechanisms such as culture, behaviour and knowledge were incorporated in the COBIT 5 Information Security Framework [22] in 2012.

IT staff still find it difficult translating security controls into concrete actions in the initial phase of a design and build of software [23]. Because of this complex processes, employees focus on continuous maintenance of documentation to please internal and external regulators, instead of value creation for customers. Khan states in his paper “*Due to constantly shifting regulations, businesses today are having to audit their IT compliance requirements on average four and a half times per year. Now more than ever, the act of adhering to regulatory requirements requires an ongoing commitment* [24]”. Without an automated process security & privacy by design and continuous delivery

will not be possible [25]. Compliance processes are complex and time consuming, often manual and the evidence has to be found numerous times for different audits, reviews and different regulators [24].

Up to 2016, the subjective silo approach to IS was designed, maintained and reported via spreadsheets [8]. Experts mapped multiple control frameworks [26] from ISO, ISF, COBIT 5 in spreadsheets and these are still used by regulators such as the Dutch Central Bank [27]. Powell et al. [28] discovered in 483 error instances in 50 spreadsheets. The Powell research is one of the largest examinations into spreadsheet errors. They have identified; Mechanical errors arising from typing or pointing errors, logic errors arising from choosing the wrong function or creating the wrong formula and omission errors arising from misinterpretation of the situation to be modelled. Volchkov stated that collecting evidence of effectiveness of the controls via spreadsheets has limitations [29] and pose a risk on its own. So Governance Risk and Compliance (GRC) tools moved towards information risk, due to the Sarbanes-Oxley Act, and were designed for large enterprises. GRC implementations are complex and their maintenance requires dedicated staff [30]. Integration of GRC tools with operational data via Security Information and Event Management (SIEM) functionality is reserved for companies with extensive budgets and sufficient staff [30].

Filling in spreadsheets with answers to questionnaires is subject to manipulation [28] because it is not a closed-locked-down cycle. Spreadsheets are stored –sometimes double versions- on decentral systems, sometimes not well protected which makes evidencing unreliable. Spreadsheet data is limited to subjective opinions and there is little room for reflection. Spreadsheet data cannot always be gathered from the original sources, which reduces authenticity and integrity [31]. Intersubjective aspects were missing from past timeframes, unless companies used third parties to interpret the data. Objective aspects are not covered, since the various objects (operational processes and data) are not interconnected. Javid Khan quotes *“The use of smarter and more intuitive tools and technologies, along with automating processes, will enable organisations to gain the benefits they are seeking, such as real-time alerts, better reporting and bringing all data sources together. Going forward, there will be increased demand for this type of technology that can optimise the compliance process, both from a management and maintenance point of view [24]”*.

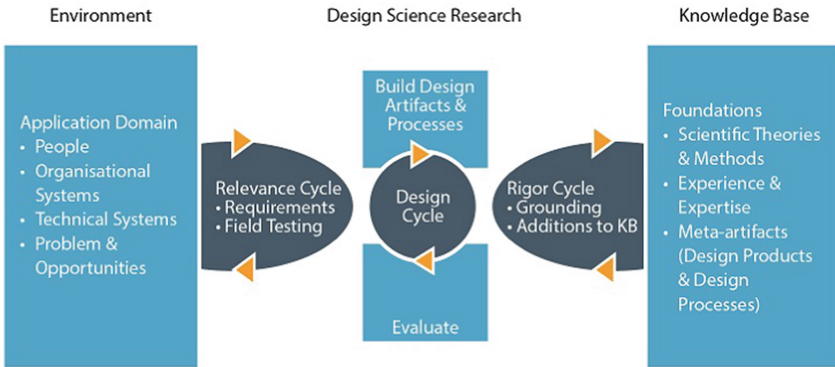
This brings us to the following problem statement:

*“Maintaining a realtime security administration (e.g. insight and oversight) on the end to end digital assurance is cumbersome. Specifically in a large Agile oriented enterprise with multiple DevOps teams that is subject to multiple regulations.”*

## **1.2 Design Science Research to Design and Engineer the LockChain Artefact**

As mentioned above traditionally, security and risk processes are being implemented by IT or security people only. Most of the time via spreadsheets and Microsoft Word files residing all over the organization with a lack of proper central administration [6]. Khan states *“Given that compliance is such a complex and time-intensive task, automating some of the processes can make realizing compliance on a continuous basis easier to achieve. It can also reduce the potential for human error and make the entire process more accurate and more efficient [24].”* Over the period 2016-2019 we have established

an artefact which addresses all of these future digital assurance problems via Design Science Research (DSR). According to Hevner, DSR is based on three major domains: the “Knowledge base” domain, the “Environment” domain and the “Design Science” domain [32]. The first is concerned with knowledge items produced and maintained with academic rigor. Theories, frameworks, models and techniques are produced in science and contribute to such rigor. These are then applied via the design science cycle to the practical environment, which includes organizations, systems and people with real-life problems. At the heart of the DSR framework is the design science cycle, which is concerned with receiving input from the knowledge base, applying this in environments and receiving feedback, in order to master problems and establish artefacts. The three cycles at the center of the framework represent the continuous feed-forward and feedback cycles which strengthen the design and development of the artefact. The main function of this design cycle is to establish and maintain the artefact and the main purpose of the artefact is to solve problems. The process of assessing and refining the artefact requirements is necessary to continuously test the artefact for its relevance to the practical environment (mainly to solve problems) and its contribution to the academic rigor (knowledge base). Creating business value due to the application of DSR artefacts is described as “valorization” and demonstrated in our earlier work [33].



**Fig. 1.** Hevner’s design science research framework for the design and engineering of security artefacts [32].

Hevner et al. [32] produced a broad framework which is used worldwide to perform and publish DSR work. This framework is visualized in Fig. 1 contrasts two research paradigms in information system research: *behavior sciences* and *design sciences*. Both domains are relevant for Business Information Security (BIS) because the first is concerned with soft aspects such as the knowledge, attitudes and capabilities required to study and solve problems. The second is concerned with establishing and validating artefacts. To put it more precisely, Johannesson and Perjons distinguish between the design, development, presentation and evaluation of an artefact [34]. Wieringa distinguished many methods for examining numerous types of problems, e.g. design problems and knowledge problems [35]. In this project we used Hevner’s work as a frame of reference to establish, build, test and valorize the artefact coined “LockChain”.

## 2 Lock the Chain of Evidence

LockChain is based on addressing three major problems in BIS [6] being a) silo based thinking and working, b) lack of central administration and oversight on information security requirements on paper versus implementation “one single source of truth” in a scaled agile environment, c) distributed decision making on risk and security without having a clear chain of evidence “end-to-end trust”. Departing from these problems we started establishing the first functionalities according to the architecture framework visualized below in Fig. 2. LockChain departs from the principle that security, risk and privacy requirements are being maintained and registered in a central administrative repository. This repository is an integrated database of requirements definition as well as implementation. The result is a more reliable data and therefore better verifiable and auditable. The repository is being accessed and written into by authorization of multiple stakeholders, with clear role based access. Nowadays, with autonomous DevOps teams, it is required to do this in a more distributed and automated manner. This is facilitated for DevOps teams via LockChain since the process of design, requirement setting is done in LockChain and develop, build, deploy, testing and logging changes in version controls is completely automated in the Continuous Delivery Pipeline (CDP) [36] and not in LockChain. The architecture including the functionalities is displayed in Fig. 2. The principle of LockChain is that the central repository is being fed by control objectives originated by regulating bodies, community bodies, security frameworks and/or auditors referred to as “the Body of Knowledge”. Normally the design of these controls in IT systems is something the IT department does but since IT is an integrated part of our day to day business more and more people are involved in designing and building new business models and associated systems, regardless if they are in the “Cloud” are on premise. To enable DevOps teams designing and building new applications the LockChain technology guides the autonomous team to design the IT chain end to end from cloud providers to external suppliers. By explicating the “End to End” assets and their owners this enables the asset owner and privacy officer to identify what kind of data is being processed. The level of Confidentiality, Integrity and Availability of the data being processed and the location of the application determines the level of security controls. The LockChain technology presents the required security controls per asset and requires the associated officers to authorize before going live. Traditionally the 1<sup>st</sup> line security officer, 2<sup>nd</sup> risk officer and data privacy officers or compliance officer all need to review, endorse and/or approve the going live of the application into production, depending on the internal effectuation of the Three Line of Defense model [37]. This chain of approvals as evidence, into the technology basically locks down (time and name stamping) the required assurance you need in order to demonstrate compliance, this refers to the terminology “Lock the Chain”. Since LockChain also enables 3<sup>rd</sup> parties to access the system you are also able to involve Cloud providers or third parties that need to adopt and comply to your company security standards. LockChain technology enables on the one hand one source of truth for Cyber, Information Security and Privacy, and facilitates on the other hand privacy and security by design. The real time administration of the technology ensures near real-time end to end trust, oversight and enables efficient assurance.

### 3 LockChain in Practice

In practice, more and more business models are changed and disrupted due to Information Technology (IT) [38]. Business owners, business developers and assets owners sometimes seek their own way in acquiring new technologies when they feel the IT department does not enable them but blocks innovations and new business models, Silic et al. refer to “Shadow IT as IT behind the curtain” [39]. Therefore the need for early involvement of IT and Security staff in designing these new business models and their associated technologies is needed. Normally the assessment of information security risks begins with a Business Impact Assessment (BIA) on the digital asset, making it a business driven activity. The terminology Business Information Security (BIS) [40] also means you involve business and IT and security departments end-to-end across company silos. The DevOps team members such as developers, product owners, engineers are responsible for the development of their solution, supported by the business and asset owners and the craftsmanship the need to develop and maintain for this [41]. LockChain enables this end to end BIA process as well as determining the security requirements. In the section below we describe how it works and how it contributes and demonstrates its value:

As outcomes from the LockChain technology and the rigorous process team members need to follow, the team will establish the CIA rating (Confidentiality, Integrity and Availability) which determines the level of security controls is required. As a result to the BIA process and CIA rating a complete data register, based on current privacy regulations (including the General Data Protection Regulation (GDPR)), a list of security requirements and associated measures is presented. In such a way it is fully aligned with the company’s policies. The LockChain technology also presents you the residual security risks that remain “open” after application of the security controls. This residual risk is determined based upon the Threat and Vulnerability Analysis (TVA). In case it is required, an additional Data Protection Impact Assessment (DPIA) can be performed, involving the Data Protection Officers (DPO) who can sign of on it.

Although common in mature organizations, the LockChain technology allows to reduce the burden and cost of information security risk management. A case study at a large financial institute indicated that the overall time spent on traditional security and risk processes is reduced by 50%, all roles and responsibilities that need to approve and sign off included. It also reduces cost of maintenance by 60%. As an example, a Business Impact Assessment (BIA) without existing documentation is considered to take an average of 44 h from initial steps to complete review. With LockChain, this time is reduced to an average of 22 h. As another example, the establishment of security requirements and the inherent Threat and Vulnerability Analysis (TVA) consumes an average of 69 h without pre-existing documentation. LockChain narrows this down to an average of 31 h. With existing documentation, assessment process takes an average of 31 h, time is reduced to 12 h, including the complete review process. In terms of expenses, LockChain reduces the cost of a complete process on a single application/solution from 10K euros to 4.8K Euros for a new asset, without any existing documentation. For an existing asset, cost goes down from 4.3K euros to 1.8K euro’s in average. Considering a large enterprise with 5000 application, in theory can realize a reduction of the total risk, security and compliance expenses of 56%.

## 4 Future Developments

The development of the LockChain technology will continue in an Agile manner. Allowing the users and stakeholders (environment) provide feedback to the design and development team to further enrich and scrutinize the Body of Knowledge, as proposed by Hevner et al. [32] and extended in other work of the authors on building LockChain alike artefacts [6, 42]. Extensive additional literature research has been conducted in 2019 as part of a Master in Science project at Open University to gain new insights into future requirements. Future research and development efforts will primarily focus on expanding the automation of control testing evidence. This Information Security Management System (ISMS) functionality connects with the Configuration Management Database (CMDB), allowing to automatically export the security configuration set up to the assessment and cross-examine the information already residing in the documentation. This will increase the reliability of the evidence, lower the level of manual labor, lower the error-rate caused by spreadsheet usage, and lower the frustration currently being experienced when collecting multiple spreadsheets and Word versions. It will also decrease the subjective discussions on the quality of evidence. The Service Application feature of the LockChain technology is expected to reduce redundancy in the documentation and ease communication between the DevOps Teams. Future research and development will also be focused on how LockChain can orchestrate and further automate operational security processes. An example is the design of security control User Access Management “user verification”, this will be designed in the LockChain technology and automatically kicks of periodical process of verifying users based on pre-defined criteria. Collecting evidence back from these processes back in the LockChain technology can be facilitated via an Application Programmable Interfaces (API). On the privacy management part it is planned to automate the detection of personal data flows between solutions. In combination with relevant metrics, and role base access, the intent is to facilitate the audit by third-parties and even regulator bodies. Therefor enabling “API based supervision” by the regulator instead of sending spreadsheets and documents. Additional privacy requirements and measures will be added, facilitating an application to the new extension of ISO/IEC 27001 and ISO/IEC 27002, the ISO 27701:2019 for privacy information management.



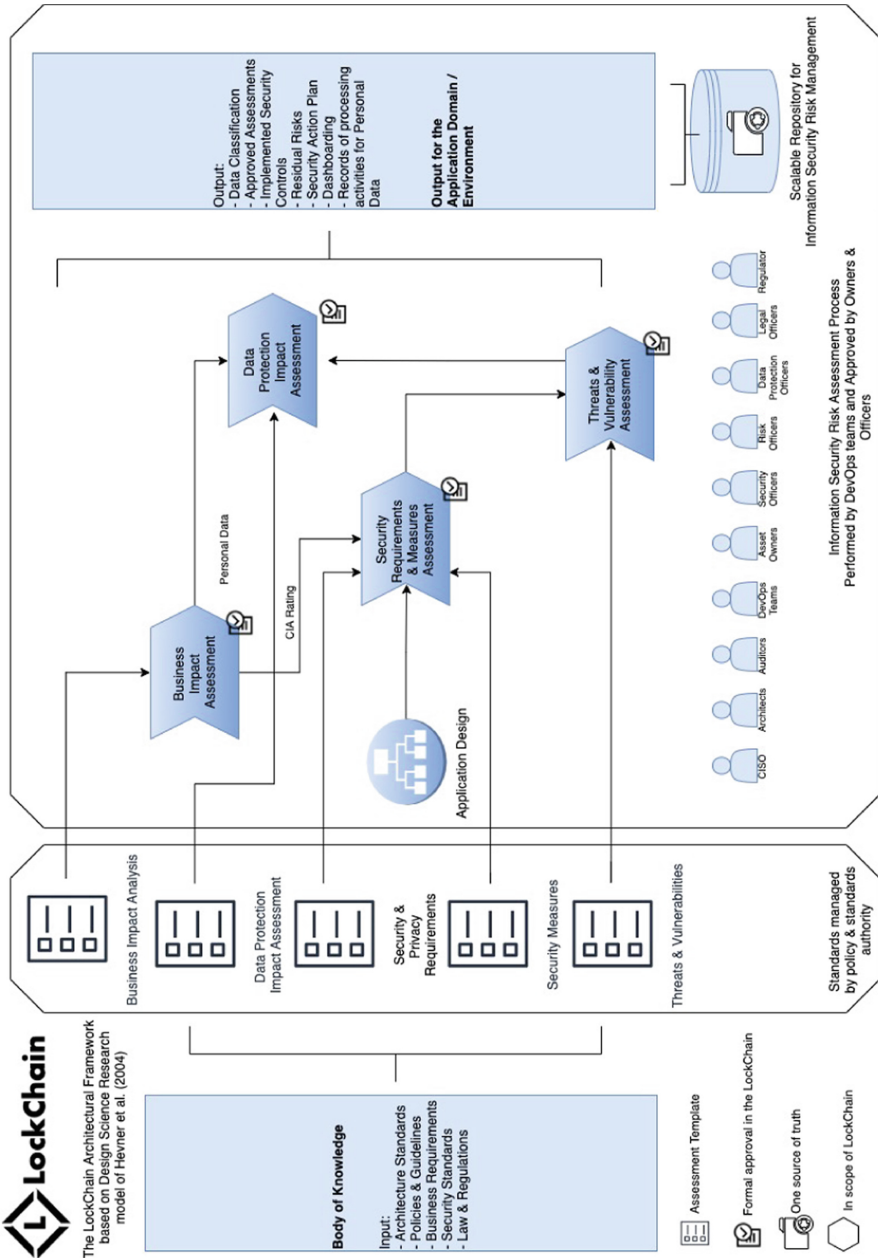


Fig. 2. The LockChain architecture framework based on Hevner et al.

## 5 Conclusion

The case study used for this paper allows to reduce the cost of security and risk processes by 60%. The case study company onboarded more than 1200 applications in less than a year and facilitates more than 1100 users, and includes the daily connection of 100 unique users, designing, maintaining and approving the security of critical assets. The development was based on a scrum organization of the work in a state of the art Continuous Delivery Pipeline (CDP) according to the CDP autonomy prescribed by Humble and Farley [41], with releases of updates every two weeks and a presentation of the new update by the developers themselves (End of Sprints). This method allows both the users and the development team to receive feedback on the use and needs of the product and its requirements. With rituals like these End Of Sprints (EoS) it establishes a close alignment with the business.

With Security first being practiced only in IT it is now transformed to Business Information Security, where business takes ownership over their critical assets and collectively with security teams designs, orchestrates and applies the requirements. This collectively designing and orchestration of automation is referred to as SOAR According to Gartner's SOAR market guide, "by year-end 2022, 30% of organizations with a security team larger than five people will leverage SOAR tools in their security operations, up from less than 5% today".

While collectively designing and developing the controls on assets, it also encourages ownership and stimulates craftsmanship throughout the company; ownership because each valuable and knowledgeable party is consulted and tracked in its analysis; craftsmanship because the DevOps teams are "by design" guided to the security of their application.

A positive side effect of LockChain is the development of a "Security by Design" culture in the DevOps teams of an enterprise. The simplification of security administration, led to an increase of comments and concerns, awareness on security at the very early stages. The same is observed regarding privacy topics when processing personal data. After a time of usage of LockChain, the challenge of security and privacy measures tends to "shift to the left", gradually reducing the time to review and administer. Ultimately resulting in improved oversight, visibility and control into Security, Risk and Compliance (SRC reporting), via dashboards for the Chief Information Security Officer (CISO), Chief Risk Officer (CRO) and Data Protection Officer (DPO). Figure 3 and 4 display two dashboard examples present in the LockChain artefact.



Fig. 3. Screenshot of the artefact dashboard function, general dashboard available to all users

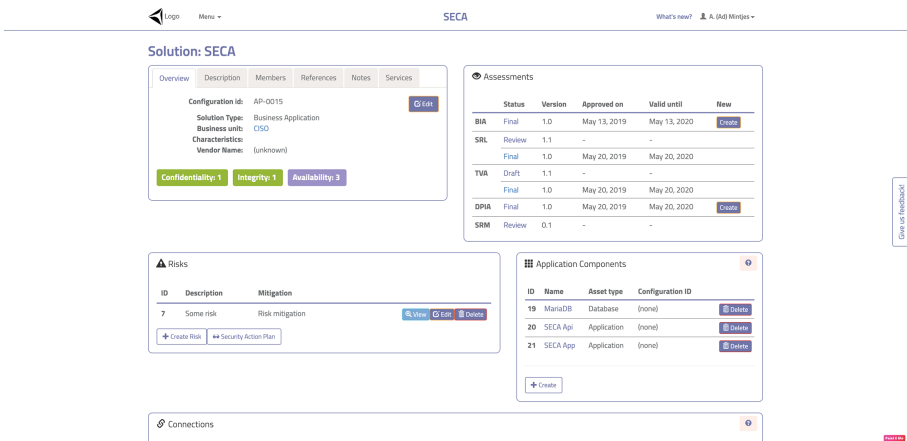


Fig. 4. Screenshot of the artefact dashboard: all information about a specific asset/solution compiled in one place.

## References

1. Ponemon: Cost of Data Breach Study: Global Analysis. Ponemon Institute LLC, United States (2016)
2. Conti, M., Dehghantanha, A., Franke, K., Watson, S.: Internet of Things security and forensics: challenges and opportunities. *Future Gener. Comput. Syst. Int. J. eSci.* **78**, 544–546 (2018)
3. Cashell, B., Jackson, W., Jickling, M., Webel, B.: *The Economic Impact of Cyber-Attacks*. Congressional Research Service, The Library of Congress, United States (2004)
4. ITGI: *Information Risks; Who’s Business are they?* IT Governance Institute, United States (2005)

5. Hubbard, D.: *The Failure of Risk Management*. Wiley, Hoboken (2009)
6. Bobbert, Y.: *Improving the Maturity of Business Information Security: On the Design and Engineering of a Business Information Security Administrative Tool*. Radboud University, Nijmegen (2018)
7. Yaokumah, W., Brown, S.: An empirical examination of the relationship between information security/business strategic alignment and information security governance. *J. Bus. Syst. Gov. Ethics* **2**(9), 50–65 (2014)
8. Zitting, D.: Are You Still Auditing in Excel? *Sarbanes Oxley Compliance J.* (2015). [http://www.s-ox.com/dsp\\_getFeaturesDetails.cfm?CID=4156](http://www.s-ox.com/dsp_getFeaturesDetails.cfm?CID=4156)
9. Flores, W., Antonsen, E., Ekstedt, M.: Information security knowledge sharing in organizations: investigating the effect of behavioral information security governance and national culture. *Comput. Secur.* **2014–43**, 90–110 (2014)
10. Van Niekerk, J., Von Solms, R.: Information security culture: a management perspective. *Comput. Secur.* **29**(4), 476–486 (2010)
11. Seale, C.: *Researching Society and Culture*, 2nd edn. Sage Publications, Thousand Oaks (2004). ISBN 978-0-7619-4197-2
12. ISO/IEC27001:2013: *ISO/IEC 27001:2013 Information technology – Security techniques – Information security management systems – Requirements*. ISO/IEC, Geneva (2013)
13. Cherdantseva, Y., Hilton, J.: A reference model of information assurance & security. In: *IEEE Proceedings of ARES*, vol. SecOnt Workshop, Regensburg, Germany (2013)
14. GOV.UK: *The Security Policy Framework (SPF)*. Statement of Assurance questionnaire in Excel - [Gov.uk](http://gov.uk)
15. Halkyn: *ISO27001 Self Assessment Checklist hits record downloads*, 19 February 2015
16. ISF: *Corporate Governance Requirements for Information Risk Management*. Information Security Forum, UK
17. von Solms, S., von Solms, R.: *Information Security Governance*. Springer, New York (2009). ISBN 978 0 387 79983 4
18. Al-Omari, A., El-Gayar, O., Deokar, A.: Information security policy compliance: the role of information security awareness. In: *Proceedings of the American Conference on Information Systems*, US (2012)
19. Al-Omari, A., El-Gayar, O., Deokar, A.: Security policy compliance: user acceptance perspective. In: *Proceedings of the 45th Hawaii International Conference on System Sciences*, Maui (2012)
20. Stackpole, B., Oksendahl, E.: *Security Strategy*. Auerbach Publications, Boca Raton (2011)
21. Van Grembergen, W., De Haes, S., Guldentops, E.: Structures, processes and relational mechanisms for IT governance. In: *Strategies for Information Technology Governance*, pp. 1–36. Idea Group Publishing, Hershey (2004)
22. ISACA: *COBIT5 for Information Security*, United States: Information Systems Audit and Control Association, ISACA (2012)
23. Visser, J.: *Building Maintainable Software*. O'Reilly Media Inc., Sebastopol (2016)
24. Khan, J.: The need for continuous compliance, pp. 14–15, June 2018
25. Forsgren, N., Humble, J.K.G.: *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. Lean IT Strategies LLC, Portland, Oregon (2018)
26. ITGI: *COBIT Mapping: Mapping of CMMI for Development V1.2 With COBIT*. IT Governance Institute, ISBN 1-933284-80-3, United States of America (2007)
27. Koning, E.: *Assessment Framework for DNB Information Security Examination*. De Nederlandsche Bank, Amsterdam (2014)
28. Powell, S., Baker, K., Lawson, B.: Errors in operational spreadsheets. *J. Organ. End User Comput.* **21**(3), 24–36 (2009)

29. Volchkov, A.: How to measure security from a governance perspective. *ISACA J.* **5**, 44–51 (2013)
30. Papazafeiropoulou, A.: Understanding governance, risk and compliance information systems the experts view. *Inf. Syst. Front.* **18**(6), 1251–1263 (2016)
31. Deloitte: Spreadsheet Management, Not what you figured (2009)
32. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* **28**(1), 75–105 (2004)
33. Bobbert, Y., Mulder, J.: Enterprise engineering in business information security. A case study & expert validation in security, risk and compliance artefact engineering. A comparative analysis of a security measurement tool. In: *EEWC 2018. LNBIP*, vol. 334, pp. 1–25. Springer (2019)
34. Johannesson, P., Perjons, E.: *An Introduction to Design Science*. Springer, Cham (2014). Stockholm University
35. Wieringa, R.: Design science as nested problem solving. In: *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, New York (2009)
36. Bass, L., Holz, R., Rimba, P., Tran, B., Zhu, L.: Securing a deployment pipeline. In: *3rd International Workshop on Release Engineering*. IEEE ACM (2018)
37. COSO: *Leveraging COSO Across the Three Lines of Defense*. The Committee of Sponsoring Organizations of the Treadway Commission (COSO), United States (2015)
38. McKinsey: *Disruptive technologies: advances that will transform life, business, and the global economy*. The McKinsey Global Institute (2013)
39. Silic, M., Back, A.: Shadow IT – a view from behind the curtain. *Comput. Secur.* **45**, 274–283 (2014)
40. Bobbert, Y.: *Maturing Business Information Security*. IBISA, Utrecht (2010)
41. Humble, J., Farley, D.: *Continuous Delivery*. Pearson Education Inc., New York (2011)
42. Bobbert, Y.: Defining a research method for engineering a Business Information Security artefact. In: *Proceedings of the Enterprise Engineering Working Conference (EEWC) Forum*, Antwerp (2017)



# Introduction of a Hybrid Monitor for Cyber-Physical Systems

J. Ceasar Aguma<sup>1</sup>(✉), Bruce McMillin<sup>2</sup>, and Amelia Regan<sup>1</sup>

<sup>1</sup> University of California, Irvine, CA 92617, USA  
{jaguma, aregan}@uci.edu

<sup>2</sup> Missouri University of Science and Technology, Rolla, MO 65401, USA  
ff@mst.edu

**Abstract.** Computing systems and mobile technologies have changed dramatically since the introduction of firewall technology in 1988. The internet has grown from a simple network of networks to a cyber and physical entity that encompasses the entire planet. Cyber-physical systems (CPS) now control most of the day to day operations of human civilization from autonomous cars to nuclear energy plants. While phenomenal, this growth has created new security threats. These are threats that cannot be blocked by a firewall for they are not only cyber but cyber-physical. In light of these cyber-physical threats, this paper proposes a security measure that promises to enhance the security of cyber-physical systems. Using theoretical cyber, physical, and cyber-physical attack scenarios, this paper highlights the need for additional monitoring of cyber-physical systems as an extra security measure. Additionally, we illustrate the efficiency of the proposed monitor using a Shannon entropy proof, and a multiple security domain nondeducibility (MSDND) proof.

**Keywords:** Cyber security · Intrusion detection · Smart grids · Energy networks

## 1 Introduction

In May of 2017, a ransomware attack held most of the developed world hostage, crippling healthcare systems, manufacturing systems, and multiple critical infrastructures across the globe. The British National Health Services was forced to limit health care to only emergency cases [7]. If not for a timely kill switch, the attack could have brought forth catastrophic damage to nuclear plants, air transportation systems, and many other infrastructures. The Wannacry [7] ransomware attack is a recent example of a now critical threat. A great many cyber and physical attacks keep cropping up all over the world, most notably; the Iran stuxnet attack [4], which, according to Iran's civil defense agency, was still a threat in October of 2018 [18], the byzantine replay attack [19], and the Ukraine power grid attack that left more than 230,000 people without electricity [20]. The author in [22] provides an extensive list of typical

security threats that are facing smart-city CPS and detailed countermeasures available to defend against these. Because cyber-physical systems (CPS), are physical entities with cyber functionality, traditional cybersecurity measures are simply not sufficient to mitigate the threat posed by this new wave of cyber-physical attacks [1]. While traditional cyber attacks were easily deducible and susceptible to prevention by means of a firewall or antivirus software, it has been shown that recent attacks like the Iran Stuxnet attack could go undetected for long periods of time [5]. A search for a solution to these threats should, therefore, focus on making the occurrence of such attacks almost impossible, and if the attacks remain possible, then they should at least be swiftly deducible.

The protection of cyber-physical systems cannot depend on the effectiveness of a single detection mechanism [5]. However, the majority of the proposed Cyber-physical security measures have centered around the notion of a single monitoring unit. The Shadow Security Unit (SSU) [16] proposed by Cruz et al. is a viable idea, but considering that the SSU is a single unit that employs only cybersecurity measures, a cyber attack that targets the central monitoring unit itself, if not detected early, could be fatal to the rest of the CPS. Scaglione, Peisert, and McParland acknowledge the need for both a centralized and distributed monitor but the proposed monitor is only an algorithm [14]. While it's a great algorithm, it's still a cyber measure which will inevitably be vulnerable to some cyber attack. The same could be said about the Intrusion Detection Systems (IDSs) [15], that is, IDSs are also a single cyber measure. For an extensive look at the many cyber attacks, industry CPS models, and common cyber measure, we direct the reader to [19].

Other than purely cyber measures, some scholars have proposed the use of physics based measures to detect attacks, but these very rarely provide ways to mitigate the attacks or prevent them in the first place [21]. The primary idea of those methods is that physical properties of the system models can be used to detect attacks. The author in [21] presents a detailed survey of recent physics-based attack detection schemes in CPS models. Our research proposes the addition of a hybrid monitor spread over virtual nodes with randomized features. This addition to a CPS would provide a much-needed auxiliary layer of security and also enhance attack deductibility.

This paper is arranged as follows: Sect. 2 introduces the tools used in testing the viability of the proposed hybrid monitor as a security measure for CPS. Section 3 gives a comprehensive look at the hybrid monitor, listing its features and the reasoning behind each feature. Section 4 explains the methodology used to demonstrate the efficiency and effectiveness of the hybrid monitor. Section 5 presents the test scenarios and proofs. Section 6 wraps up with a short conclusion.

## 2 Background

This research employs the Future Renewable Electric Energy Delivery and Management (FREEDM) [2] System as a model CPS. Shannon entropy [8] is used

as a tool to test the effectiveness of the monitor as a security measure. The multiple security domain nondeducibility (MSDND) [1] is used as a tool to test the effectiveness of the monitor in detecting attacks.

The FREEDM system center is an engineering research center funded by the National Science Foundation and spanning number of universities including but not limited to North Carolina State University, Missouri University of Science and Technology and Florida State University. The research center developed an energy management and distribution smart grid testbed located at the North Carolina State University that is also referred to as FREEDM [2]. The heart of this energy system is the Distributed Grid Intelligence (DGI) [2], an intelligent algorithm that implements energy management and distribution using modular adapters to interact with devices in a smart grid over different interfaces. In this paper, the FREEDM system is employed as a test subject for the implementation of the hybrid monitor proposed here.

The MSDND model is a security model that tests the integrity and confidentiality of a cyber-physical architecture. The MSDND model uses logic proofs to test information flow security; that is, how information moves among user groups within the security domains (SDs) that make up the system [1]. Howser and McMillin show that maintaining information flow security in CPS is challenging because the flow is irrevocably linked among the CPS' cyber and physical units [1]. Therefore, the MSDND model is employed to account for both cyber and physical information flow paths. The model defines two system properties namely: MSDND secure and notMSDND secure [1]. An MSDND secure system implies that for information flowing from entity A to entity B, entity B cannot deduce whether the information is valid or erroneous. This also means while an MSDND secure system is desirable if the goal is to maintain confidentiality, it can be an indicator of the possibility of an attack going undetected. A notMSDND secure system implies the alternative, that is, entity B can evaluate the correctness of information obtained from entity A. Therefore, in the event of an attack, Howser and McMillin show that a notMSDND secure system could easily detect the occurrence of an attack [1]. Thudimila and McMillin demonstrate the superiority of MSDND over traditional electronic and cryptographic solutions [3] when applied to detection of attacks in Automatic Dependent Surveillance-Broadcast (ADS-B) air traffic surveillance system [3]. For this research, MSDND is used to analyze whether a CPS with a hybrid monitor in place would deduce the occurrence of an attack.

Phan et al. introduce the idea of using information flow-metrics like Shannon entropy to measure information leakage in CPS programs [9]. Li uses Shannon Entropy to break down the physical dynamics of CPS and goes on to show the negative entropy that communication adds to the general entropy of a CPS [10]. In this text, Shannon Entropy is used to illustrate the decrease in the possibility of an attack in a CPS after the introduction of a hybrid monitor. This demonstrates the value of adding a hybrid monitor to CPS architectures as an extra security measure.



Shannon entropy is an information theory concept derived from the general idea of information entropy that was developed and introduced by Claude Shannon [8]. Entropy is basically a measure of uncertainty in a communication system where a low entropy value implies minimal uncertainty and a high entropy implies the contrary. Shannon entropy defines entropy ( $H$ ) as:

$$H[X] = E[I[X]] \quad (1)$$

Where  $I$  is the information content of the discrete random variable  $X$ . Therefore we can further define  $E[I(X)]$  in terms of the probability mass function of  $X$ : i.e.,

$$E[I[X] = E[-\log(P(X))] = - \sum_i P(x_i) \log(P(x_i)) \quad (2)$$

The generally definition of entropy  $H$  then comes to:

$$H[X] = - \sum_i P(x_i) \log(P(x_i)) \quad (3)$$

### 3 The Hybrid Monitor

To better protect cyber-physical systems, this research proposes the addition of a monitor. Many other researchers have explored the use of hybrid monitoring to ensure safety or/and security in CPSs. Li et al. propose extended hybrid automata modeling for vehicular CPSs as a safety and control measure [11]. Mao and Chen also introduce a runtime hybrid automaton monitoring framework for the Cooperative Adaptive Cruise Control Systems (CACC) [12]. This research borrows the idea of hybrid monitoring but with randomization as an additional feature. The addition of randomization in information flow paths' generation increases the system's entropy and in turn, reduces the chances of a successful attack in a generic CPS. That is because a higher number of information flow paths increases the number of points the attacker has to corrupt to remain undetected. Below is a detailed break down of this hybrid monitor's features;

- The monitor is hybrid, that is, both virtual and physical, central and decentralized. The monitor would have both a virtual component and physical component. The virtual components would be implemented as a hidden algorithm in every Supervisory control and data acquisition unit (SCADA) in the CPS. The physical component of the monitor would be a physical unit independent of the entire CPS and running a monitoring algorithm whose function is to oversee the operations of the monitor's virtual components. The physical component is, therefore, a central unit and the virtual components are the decentralized units. Pasqualetti et al. [13] shows that this kind of model is complete for attack detection.
- The monitor should be intelligent enough to generate physical invariants for every information flow path in the CPS. An invariant is simply a logical

assertion that should always be true throughout an execution cycle. Therefore physical invariants are logical properties of a CPS that cannot be transformed by cyber entities and should always be held true. Having physical invariants makes the CPS vastly more secure because they are secure from being corrupted by cyber attacks. With that in mind, the hybrid monitor uses generated physical invariants as a validator of the information received from other system modules. The automated generation of physical invariant using machine learning, deep learning or linear regression is also a viable research area. The automation of invariant generation is explored further by Cruz et al. [16] and Weimer et al. [6].

- The virtual components of the monitors would continuously generate a randomly increasing number of paths for the flow of information between any two CPS entities. The physical monitor should also generate a randomly increasing number of virtual paths as a compliment to a physical path for the flow of information between any two virtual components of the monitor.
- All paths generated by the monitor should be independent of each other. This ensures that all the randomly generated paths cannot be collectively corrupted by an attacker.
- To reduce the information flow overhead, information sent through the monitors should be sent through a randomly chosen path among the generated paths and then white noise should be transmitted on the rest of the paths.
- The monitor should have a routing algorithm that can be employed if the monitor detects a failure or corruption at any of the CPS' entities.
- Communication between the virtual and physical monitor should be done on an entirely different network than that used by the rest of the cyber-physical system.

Note that this hybrid monitor is only a theoretical idea but the exact physical realization should at the very least aim to implement the above mentioned features.

## 4 Methodology

This research uses two methods to highlight the significance of introducing a hybrid monitor to a CPS.

### 4.1 Method 1

The research employs attack scenarios to examine the security of a CPS with and without the hybrid monitor. There are three attack scenarios, that is, A purely cyber attack like a ransomware on a CPS, a completely physical attack like the attacker inflicting physical damage to the CPS by, for example, cutting wires and a cyber-physical attack like the Iran Stuxnet attack expounded upon in Kushner's [4] and Karnouskos' [5] work.

### 4.2 Method 2

In the second method, the research uses two proof models i.e Shannon entropy and MSDND to show that the addition of a hybrid monitor makes a CPS less susceptible to an undetected attack and much more effective at deducing attacks when they do occur.

## 5 Results

This section details the three attack scenarios, their respective results, the Shannon entropy proof, and MSDND proof.

### 5.1 Cyber Attack Scenario

As mentioned in the background, the FREEDM system is controlled by a distributed algorithm called the DGI. The DGI is set up to run on multiple nodes spread out over a network. It provides an interface for energy management applications to communicate with physical power devices.

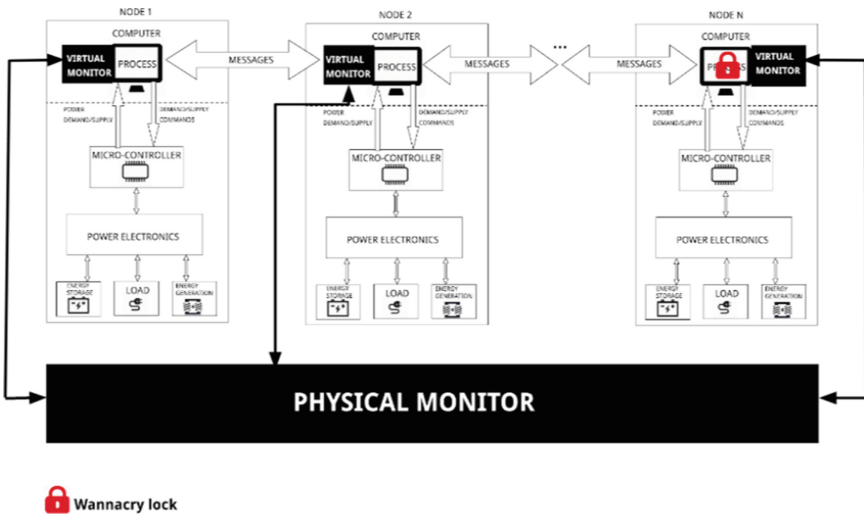


Fig. 1. The DGI under a WannaCry attack

Let us assume that a DGI node is being held hostage by the wannaCry ransomware (Fig. 1). This kind of attack is rather easy to detect because ransomware attacks normally make the user aware that the attack is in progress. Therefore, for this scenario, attack deductibility is not important. But because the attacker is holding a node hostage, all information flowing through this node could be infected by the attacker. This could give the attacker further access to other

nodes since all nodes of the DGI share state information. At this point, it's clear that the entire DGI could be held hostage. Since the DGI manages the entire FREEDM smart grid system, the entire CPS would be either rendered useless or could be left vulnerable to more damaging attacks.

Now let's consider a scenario where a hybrid monitor was in place with virtual units running alongside every DGI node and a physical unit to oversee the virtual units. Because all traffic that goes through a node is verified by the monitor and subjected to physical invariants generated by the monitor, it would be easy for the monitor to flag the presence of the ransomware. Since the monitor has information flow routing capabilities, all state information from other nodes would be safely rerouted through other nodes. While the infected node would not be saved, the rest of the DGI would continue to function without threat.

## 5.2 Physical Attack Scenario

For this scenario, we assume that an attacker has inflicted physical damage to the CPS without using cyber means. The damage could be as simple as cutting an Ethernet cord or breaking a sensor. The detection and solution for such an attack are also rather simple. However, if the CPS is a critical infrastructure like a nuclear reactor that needs to continuously keep some functions fully operational then even this simple attack could prove fatal. With a monitor in place, any failure in the CPS would quickly be detected. The monitor, through information flow rerouting, would go even further to keep critical functions running while the damage gets fixed.

## 5.3 Cyber-Physical Attack Scenario

The third and last scenario assumes that a microcontroller in FREEDM system is infected by Stuxnet (Fig. 2). Erroneous Information from this microcontroller could cause catastrophic damage to the smart grid. In this case, deducing the presence of the Stuxnet and reducing the damage to the smart grid are both necessary.

From the Iran attack, it's clear that the Stuxnet could go unnoticed for a long time if no extra security measure is put in place. Although, if the FREEDM system had a hybrid monitor, the Stuxnet would be detected because all information from the microcontroller would have to be verified by the monitor. Since the monitor has physical invariants to prove the correctness of information from this microcontroller, any discrepancies in the information generated by the Stuxnet would be caught. On detection, information flow would then be routed through other nodes and further infection would be avoided. The Stuxnet would have to infect all random paths used by the hybrid monitor to avoid detection. The Shannon Entropy proof below shows that there is a very small possibility of the Stuxnet or attacker infecting all of the hybrid monitor's random paths.

### 5.4 MSDND Proof

For this proof, let's look at the cyber-physical Stuxnet attack shown above. More specifically, the information path between the infected microcontroller and the DGI node process running on the computer without the monitor.

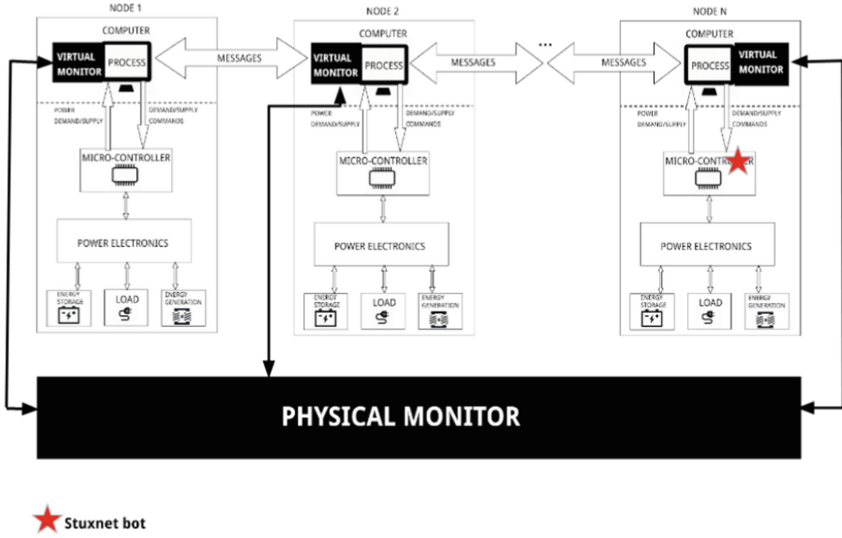


Fig. 2. The DGI under a Stuxnet attack

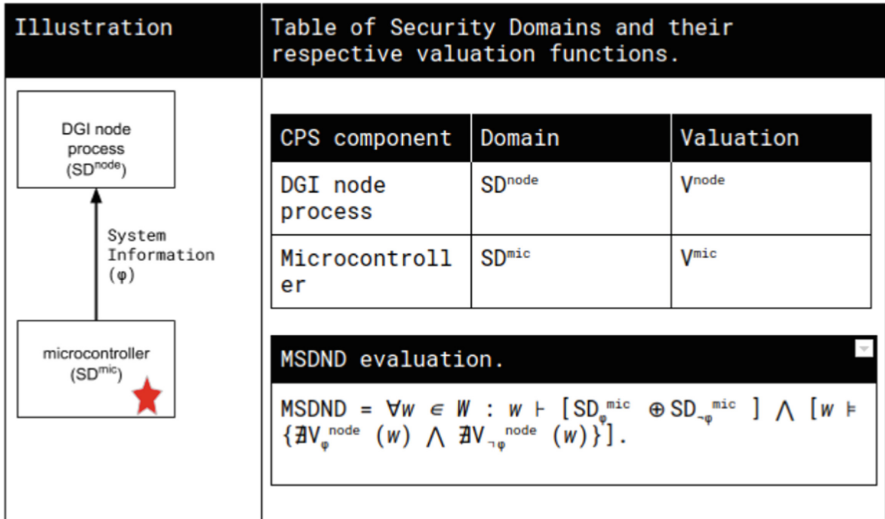


Fig. 3. MSDND evaluation for the DGI without a monitor

Let us define the two domains as  $SD_{node}$  for the DGI node and  $SD_{mic}$  for the microcontroller with valuation functions  $V_{node}$  and  $V_{mic}$  respectively. Then consider a scenario where arbitrary information ( $\varphi$ ) is sent from the infected microcontroller to the DGI node process as seen in Fig. 3. If the DGI node process and microcontroller are at the same level of security, then the DGI node process will trust that information from the infected microcontroller to be valid. Since the information can be either true or false, the first condition; i.e.,  $(SD_{\varphi}^{mic}, SD_{\neg\varphi}^{mic})$  for MSDND is met [1]. This is derived from the fact that if  $\varphi$  is true then  $SD_{\varphi}^{mic}$  is true or if  $\varphi$  is false then  $SD_{\neg\varphi}^{mic}$  is true hence the xor statement is always true.

The second condition is also satisfied from the assumption that the two domains are at the same security level [1]. Therefore, the DGI node process believes and trusts the infected microcontroller. This means the DGI node process has no valuation function to prove the validity of  $\varphi$  [1]. The absence of this valuation function ( $V_{\varphi}^{node}$ ) leaves the system in an MSDND secure system [1]. This is the MSDND secure evaluation shown in Fig. 4.

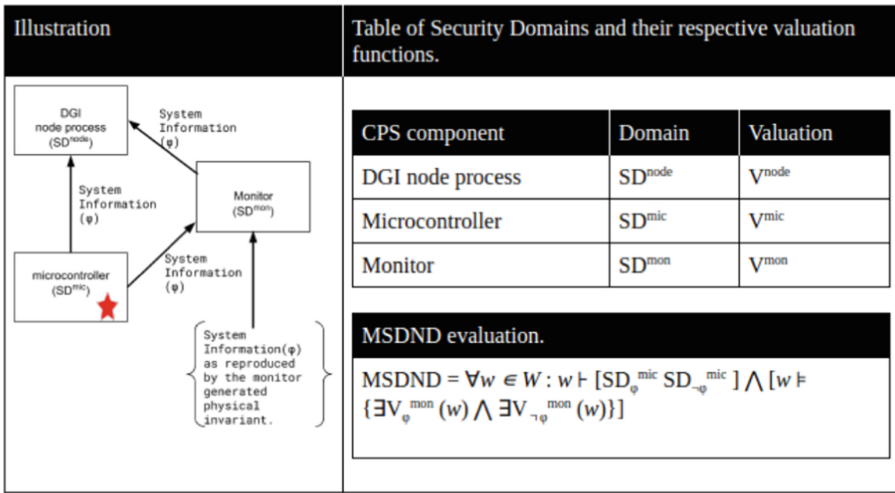


Fig. 4. MSDND evaluation for the DGI with a monitor

The implication of this MSDND evaluation is that if the infected microcontroller sent false information to the DGI node process, there would be no way of evaluating that the information is false. Therefore the Stuxnet would go undetected. Knowin this, let us take a look at a scenario with the monitor in place.

The difference in this scenario is the presence of a monitor that is equipped with physical invariants. Using a physical invariant, the monitor can evaluate the validity of  $\varphi$ . With this, the monitor can also determine the state of the microcontroller with respect to the validity of  $\varphi$ ; i.e., There exist a valuation  $V_{\varphi}^{mon}$

leaving the state  $SD_{\varphi}^{mic}$  deducible [1]. Hence the notMSDND secure evaluation shown in Fig. 4.

The proof shows that in the event of a cyber-physical attack like the Stuxnet attack, the presence of a monitor would render the attack deducible. For the attack to go undetected with a monitor in place, the attacker would have to infect every single monitor node, both virtual and physical. The next Shannon entropy proof will show that the possibility of compromising all the monitor nodes without being detected is rather minimal.

### 5.5 Shannon Entropy Proof

The proof considers two scenarios where the attacker is attempting to infect the information flow between the DGI node process and the microcontroller.

First, let us take a look at the entropy of the setup without the monitor. There are two possible information flow events  $x_1$  and  $x_2$  that the attack could target. With a sample space = 2, the probability of the attacker successfully infecting information flow between the DGI node process and microcontroller is 1/2. The entropy evaluation for this scenario is shown in Fig. 5.

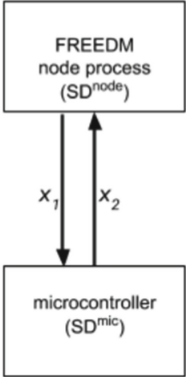
Illustration	Shannon Entropy Evaluation
 <p>The diagram shows two rectangular boxes. The top box is labeled 'FREEDM node process (SD<sup>node</sup>)' and the bottom box is labeled 'microcontroller (SD<sup>mic</sup>)'. Two vertical arrows connect them: a downward arrow labeled <math>x_1</math> and an upward arrow labeled <math>x_2</math>.</p>	<p>From; <math>H(X) = - \sum_{i=1}^n P(x_i) \log[P(x_i)]</math></p> <p><math>H(X) = - [P(x_1) \log P(x_1) + P(x_2) \log P(x_2)]</math></p> <p>Since we have two possible information flow paths, sample space = <math>\{x_1, x_2\}</math>.</p> <p>Therefore, <math>P(x_i) = 1/2</math></p> <p><math>H(X) = - [(\frac{1}{2}) \log_2(\frac{1}{2}) + (\frac{1}{2}) \log_2(\frac{1}{2})]</math></p> <p><math>H(X) = -(\frac{1}{2}) [\log_2(\frac{1}{2} * \frac{1}{2})]</math></p> <p><math>H(X) = -\log_2 \sqrt{1/4}</math></p> <p><math>H(X) = \log_2 2</math></p> <p><math>H(X) = 1</math></p>

Fig. 5. Shannon Entropy evaluation for the DGI without a monitor

With a monitor in place, the sample space grows to  $(2n + 2)$ , making the probability of successfully corrupting one path come to  $1/[2(n + 1)]$ . Here is the entropy evaluation;

The proof shows us that the entropy increases with the increase in the size of  $n$  paths (Fig. 6). From the attacker’s point of view, the uncertainty increases with increasing size of  $n$  paths. Therefore as the size of  $n$  increases, it becomes much harder for the attacker to launch a successful attack on the CPS. By adding the hybrid monitor, the system is not fully secure from an attack but the possibility of a successful attack is vastly smaller.

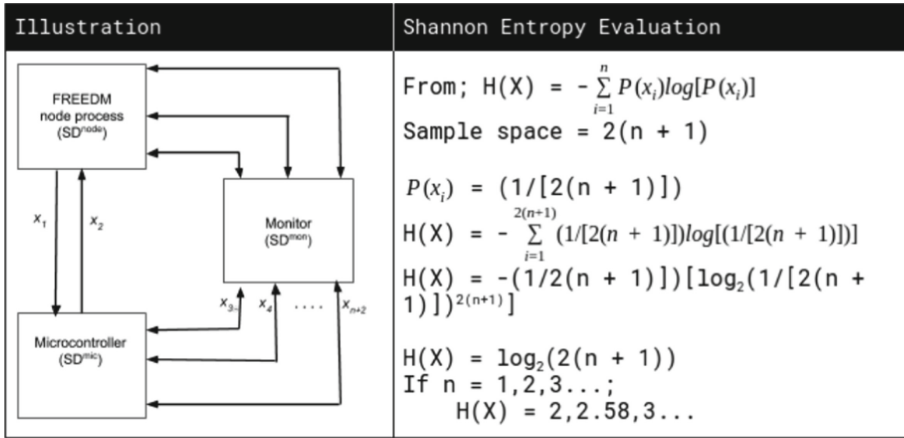


Fig. 6. Shannon Entropy evaluation for the DGI with a monitor

## 6 Conclusion

After 2017’s Ransomware attack [17], the world can not ignore the threat posed by the possibility of using attacks on cyber-physical systems as a tool for terrorism and cyber warfare. The increased occurrence of cyber-physical systems attacks is surely an indicator that traditional cybersecurity measures are insufficient at prevention and detection of these attacks. The world needs to start considering alternative or improved security measures. The combination of an intelligent, randomized physical monitor with existing virtual cyber measures to create a hybrid monitor is a good place to start. While, the hybrid monitor is not a foolproof solution to cyber-physical attacks, it could well be the best solution yet. Future research in this area should focus on design and physical implementation of the hybrid monitor and prevention of attacks targeting the hybrid monitor itself. This hybrid monitor could be the great leap towards fully securing an important and nonexpendable entity of smart living that is cyber-physical systems.

**Acknowledgments.** This research was sponsored by the United States National Science Foundation (NFS). The following colleagues made notable contribution to the research over the span of the project; Dr. Patrick Taylor (Associate Professor, Missouri S&T’s Department of Computer Science), Manish Jaisinghani (Graduate Student, Missouri S&T’s Department of Computer Science), Anusha Thudmilla (Graduate Student, Missouri S&T’s Department of Computer Science), Joshua Hermann (Graduate Student, Missouri S&T’s Department of Computer Science).



## References

1. Howser, G., McMillin, B.: Using information-flow methods to analyze the security of cyber-physical systems. *Computer* **50**(4), 17–26 (2017). <https://doi.org/10.1109/MC.2017.112>
2. Crow, M.L., McMillin, B., Wang, W., Bhattacharyya, S.: Intelligent energy management of the FREEDM system. In: IEEE PES General Meeting, Providence, RI, pp. 1–4 (2010). <https://doi.org/10.1109/PES.2010.5589992>
3. Thudimilla, A., McMillin, B.: Multiple security domain nondeducibility air traffic surveillance systems. In: IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), Singapore 2017, pp. 136–139 (2017). <https://doi.org/10.1109/HASE.2017.29>
4. Kushner, D.: The real story of stuxnet. *IEEE Spectr.* **50**(3), 48–53 (2013). <https://doi.org/10.1109/MSPEC.2013.6471059>
5. Karnouskos, S.: Stuxnet worm impact on industrial cyber-physical system security. In: IECON 2011–37th Annual Conference of the IEEE Industrial Electronics Society, Melbourne, VIC, pp. 4490–4494 (2011). <https://doi.org/10.1109/IECON.2011.6120048>
6. Weimer, J., Ivanov, R., Chen, S., Roederer, A., Sokolsky, O., Lee, I.: Parameter invariant monitor design for cyber-physical systems. *Proc. IEEE* **106**(1), 71–92 (2018). <https://doi.org/10.1109/JPROC.2017.2723847>
7. Ehrenfeld, J.M.: *WannaCry, Cybersecurity and Health Information Technology: A Time to Act*. Springer, New York, 24 May 2017
8. Shannon, C.E.: A mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 379–423 (1948). <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
9. Phan, Q.-S., Malacaria, P., Păsăreanu, C.S., D’Amorim, M.: Quantifying information leaks using reliability analysis. In: Proceedings of the 2014 International SPIN Symposium on Model Checking of Software (SPIN 2014), pp. 105–108. ACM, New York (2014). <https://doi.org/10.1145/2632362.2632367>
10. Li, H.: Information efficiency of communications for networked control in cyber physical systems: when carnot meets shannon. In: 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, pp. 1865–1870 (2016). <https://doi.org/10.1109/CDC.2016.7798536>
11. Li, Y., Chen, M., Zhang, G., Shao, Y., Feng, F., Hou, X.: A model for vehicular cyber-physical system based on extended hybrid automaton. In: 2013 8th International Conference on Computer Science & Education, Colombo, pp. 1305–1308 (2013). <https://doi.org/10.1109/ICCSE.2013.6554123>
12. Mao, J., Chen, L.: Runtime monitoring for cyber-physical systems: a case study of cooperative adaptive cruise control. In: Proceedings of the 2012 Second International Conference on Intelligent System Design and Engineering Application (ISDEA 2012), pp. 509–515. IEEE Computer Society, Washington, DC (2012). <https://doi.org/10.1109/ISdea.2012.592>
13. Pasqualetti, F., Dörfler, F., Bullo, F.: Attack detection and identification in cyber-physical systems. *IEEE Trans. Autom. Control* **58**(11), 2715–2729 (2013). <https://doi.org/10.1109/TAC.2013.2266831>
14. McParland, C., Peisert, S., Scaglione, A.: Monitoring security of networked control systems: it’s the physics. *IEEE Secur. Privacy* **12**(6), 32–39 (2014). <https://doi.org/10.1109/MSP.2014.122>

15. Pal, K., Adepu, S., Goh, J.: Effectiveness of association rules mining for invariants generation in cyber-physical systems. In: IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), Singapore, pp. 124–127 (2017). <https://doi.org/10.1109/HASE.2017.21>
16. Cruz, T., et al.: Improving network security monitoring for industrial control systems. In: IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, pp. 878–881 (2015). <https://doi.org/10.1109/INM.2015.7140399>
17. Mattei, T.A.: Privacy, confidentiality, and security of health care information: lessons from the recent WannaCry cyberattack. *World Neurosurg.* **104**, 972–974 (2017). <https://doi.org/10.1016/j.wneu.2017.06.104>. ISSN 1878-8750
18. Center for strategic and international studies, May 2019. <https://www.csis.org/programs/technology-policy-program/significant-cyberincidents>
19. Ding, D., Han, Q.-L., Xiang, Y., Ge, X., Zhang, X.M.: A survey on security control and attack detection for industrial cyberphysical systems. *Neurocomputing* **275**, 1674–1683 (2018)
20. Lee, R.M., Assante, M.J., Conway, T.: Analysis of the cyber attack on the Ukrainian power grid. Defense Use Case, E-ISAC, 18 March 2016
21. Giraldo, J., Urbina, D., Cardenas, A., Valente, J., Faisal, M., Ruths, J., Tiphpenhauer, N.O., Sandberg, H., Candell, R.: A survey of physics-based attack detection in cyber-physical systems. *ACM Comput. Surv.* **51**(4), 36, Article no. 76 (2018). <https://doi.org/10.1145/3203245>
22. Gharaibeh, A., et al.: Smart cities: a survey on data management, security, and enabling technologies. *IEEE Commun. Surv. Tutorials* **19**(4), 2456–2501 (2017). <https://doi.org/10.1109/COMST.2017.2736886>



# Software Implementation of a SRAM PUF-Based Password Manager

Sareh Assiri<sup>(✉)</sup>, Bertrand Cambou, D. Duane Booher,  
and Mohammad Mohammadinodoushan

Northern Arizona University, Flagstaff, AZ 86011, USA  
{sa2363, Bertrand.Cambou, duane.booher, mm3845}@nau.edu

**Abstract.** The main goal of narrating the password-management protocol is to reduce the prevalent attacks on cyber-physical systems such as the hacking of databases of User-ID-Password pairs and side-channel analysis. The architecture uses a hash function to hash the password and user ID has weakness can help to crack the password. So, the architecture utilizes both hash function and the Addressable Physical unclonable function (PUF) Generator (APG) to authenticate clients on the network without keeping the real format of passwords in the database. The hash function and APG together are more difficult to attack because they are unclonable, have a high level of randomness, and do not depend on storing information. This paper shows a simulation prototype for how the password manager protocol can work depending on the SHA-3-512 and SRAM PUF. Furthermore, the paper shows how to encrypt the database content of password manager by using the SRAM PUF and provides a software solution of the noise of SRAM PUF to reduce the rate of false rejections for the real user and false acceptance for the not existing user.

**Keywords:** Password management · Physical unclonable function · Hash functions · SRAM PUF with password manager · Authentication · New user · Exist user

## 1 Introduction

A password is an authentication mechanism that provides the ability to access systems, applications, or accounts online. In general, a password is a string of characters used to verify the identity of a user during the authentication process where most passwords are used with a username (USER ID). By design, only the user knows the password (PW) and USER ID (UID) needed to gain access to a device, application, or website. The simplest way to store passwords is in a database (DB) and create a table that contains the USERID and PW. The DB table keeps all of the UIDs and PWs in ‘plain text’ human-readable format [1]. For example, the user may set the UID and PW to BoB2019 and assEDA123/!, respectively. Subsequently, the UID and the PW will be saved in the DB table [1–3]. Should an intruder be able to hack the DB, then they will easily read all of the UID and DB content. For this reason, the UID and PW data that has been stored in plain

text formatted is vulnerable to be compromised. Consequently, in security terms, one of the worst possible methods used by some websites and applications, is to store a UID and PW in the original form plaintext format [1–3]. The vulnerability of DBs containing user ids and passwords is of major concern for information technology developers. It prompts investigation of a solution that will help make the content of DBs unreadable from hackers understanding the DB content [1, 2].

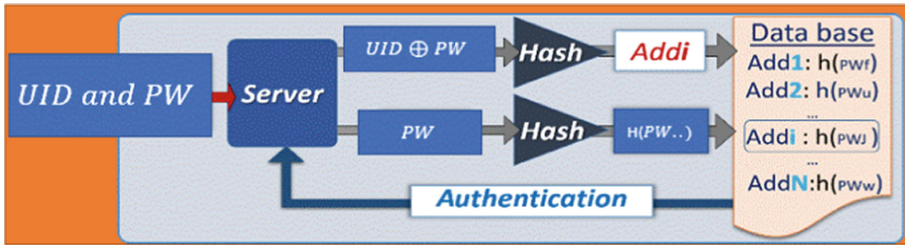
One solution is to encrypt the content of the DB by using the hash function to obtain the message digest (MD) [2–4]. Where in this solution, the real plaintext UID and PW are replaced by the MD. The hash is well known as a one-way encryption function which makes the input data inaccessible. Although the hash function shows great performance, it still has weakness because hackers can try many different passwords until one matches the hash output. In [4], both the hash function and PUFs suggested to be used in the password management scheme. The PUF is small hardware devices that provides a unique image per device which analogous to human fingerprints.

The solution presented in this paper is the development of additional lines of defense that replace the database of passwords by use of the one-time hash output and SRAM PUF challenges, thereby mitigating the risks related to most insider attacks. The architecture presented here demonstrates how the hash function and addressable PUF generators (APG) [9] are implemented in a password manager scheme. This scheme uses the SRAM as the PUF. In this protocol, the password is hashed and then the output of the hash is fed to the APG and the output of PUF (challenge) will be stored in the database, instead of the output of hash function MD [3–7]. This paper will focus on building a prototype to show how the content of the DB will become effectively unreadable content. Additionally, this paper gives a software solution to minimize the rate of false rejections and false acceptance.

## 2 Background

### 2.1 The Hash Function and Hash the Password

Many cryptography applications use a hash function such as SHA-1, SHA-2, or SHA-3, and by hashing the input, then you will not be able to obtain the value again. Similar to encryption, the hash function turns the password into a long binary value to keep it hidden. For example, if we used SHA-1 to hash “the password”, then after hashing it, the output may be like this hexadecimal string “e38ad214 943daad1 d64c102f aec29de4 afe9da3d”. A hash function can protect and replace the DB which contains the UIDs with their corresponding PW, as shown in the block diagram of Fig. 1. The hashing of passwords results in the first message digest of  $h(\text{PW})$ ; then, the UID and PW are exclusive or (XOR). Finally, the hashing of  $\text{UID} \oplus \text{PW}$  results in the second message digest of  $h(\text{UID} \oplus \text{PW})$ . Then named  $h(\text{UID} \oplus \text{PW})$  as the address in the database. Both  $h(\text{UID} \oplus \text{PW})$  and  $h(\text{PW})$  will be stored in the database (DB) [4–7]. The weakness of using a hash function is that hash function types, such as SHA-3, SHA-2, and SHA-1, have become known and popular. Thus, if the database in which the address and MD have been stored are exposed to the enemy, the information could be stolen by exploiting password-guessing methods, by use of big data analysis or brute force attempts of commonly used passwords [2, 3].



**Fig. 1.** Block diagram describing the data flow for authentication. On the left, PWs and UIDs are converted into MD, named as addresses ( $h(\text{UID} \oplus \text{PW})$ ) and  $h(\text{PW})$ . On the right, the database stores the addresses and  $h(\text{PW})$  [5].

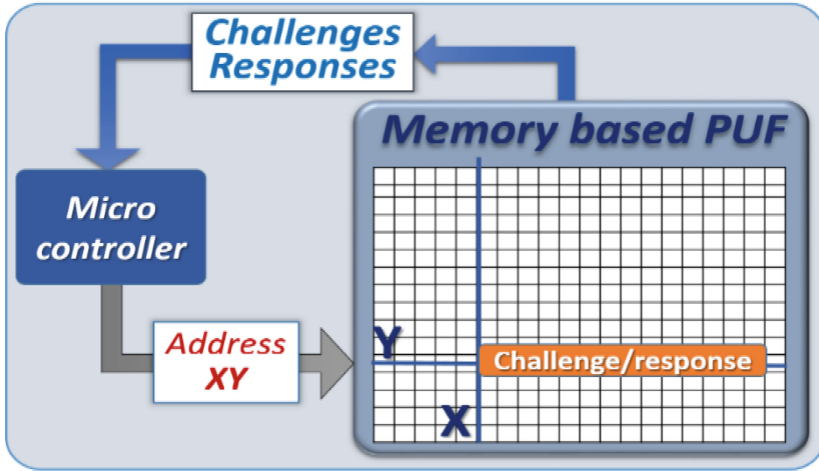
## 2.2 Physically Unclonable Functions (PUFs)

The PUF hardware components work like human fingerprints. The output of PUF is unclonable and random. PUFs strengthen the level of security. Therefore, authentication and several fields of security use PUFs. Several types of PUFs are excellent elements to generate strong PUFs: ring oscillators, Memory structures, SRAM, DRAM, Flash, ReRAM, and MRAM [4]. Furthermore, the output of PUF has two types, which are binary or ternary. The outputs of PUFs are called challenges or responses. The challenges are generated upfront from the PUF, whereas the responses are generated during access control rounds or authentication rounds. Each time the PUF is read, it is slightly mismatched, so it complicates authentication. However, as noted in [5–7], during authentication when we read the PUF to generate the responses, we must calculate the rate of matching challenge-response-pairs (CRP). If the matching CRP is high enough with a low error rate, then we can accept; otherwise, we must reject the user [5]. According to [5–7], In the password generation scheme, the APG is given the inputs and then receives the challenge or the response. The block diagram in Fig. 2(a) shows that the CRP generation varies with the relative value of the parameters within the multiple cells that are selected at a particular address; this means that if we give the value of ‘x’ and ‘y,’ then we will find a specific address in the PUF (see Fig. 2(b)). Therefore, a specific cell could be a “0” when part of one group of cells, and a “1” when part of a different group or when read with different instructions.

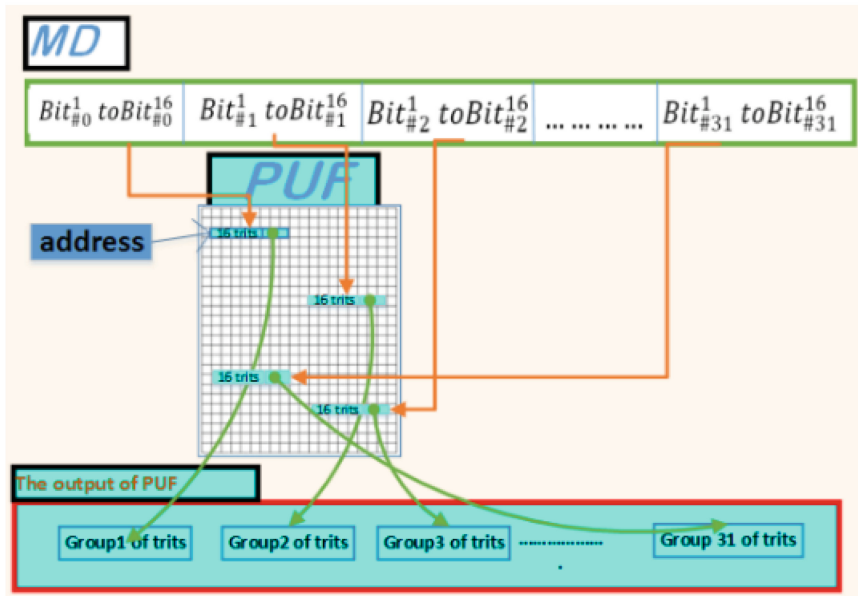
## 2.3 Differences Between Password Manager Research Efforts

The research presented in this paper is an extension with improvements to our previous research efforts as described in [7]. Both research efforts use a hash function and APG to build the protocol for the password manager. Where the hash function will produce the MD and is used to determine the address in the APG to produce challenges and responses for the UID and PW elements.

However, the previous research used methods when masking the data from the PUF, then saved to the MCU in that APG implementation. That increased the need for memory and the computation time for authentication of each UID and password. There is also a problem when the hacker can hack the database and access the information in the database [7], which is shown in Fig. 3. For this case, the hacker can test different common



(a) CRP using Addressable PUF Generation (APG) implementation [5].

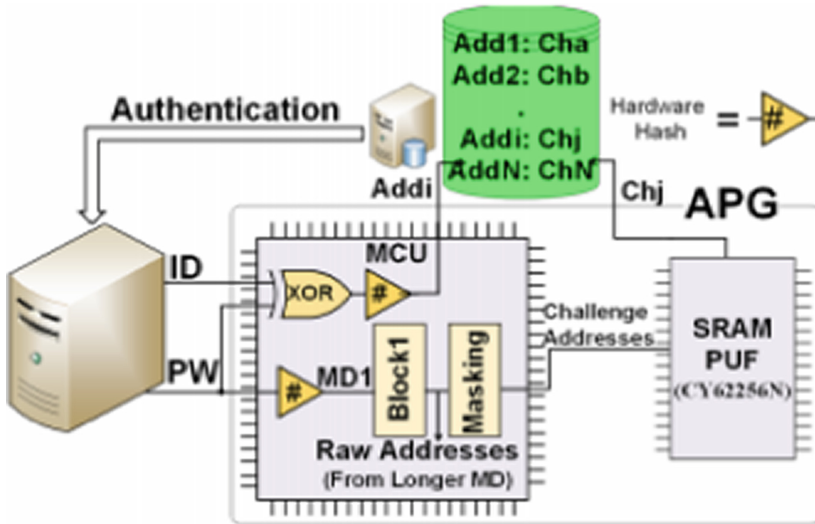


(b) Different addresses from the hash digest to extract the challenge from PUF [8].

**Fig. 2.** APG sub-diagrams with CRP interaction, (a) shows how a specific address is found according to the value of X and Y. (b) shows how are the values of X and Y assigned from the output of hash function; and X and Y point to a specific address in PUF; after that, the following bits of a specific address are read to obtain the challenges.

passwords, and XOR them with UID of the user and hash the result, then check to determine if it matches the same address which exists in the database. This could lead to a compromised system.

In this new research, alternate methods have been implemented to resolve the above problems. Where instead, as shown in Fig. 4, the PUF is used to extract the address of the database in which the challenges will be saved. Additionally, improvements are implemented which results in more efficient and much faster execution. The new methods are fully described in Sect. 3 Methods. Additionally, Sect. 4 implementation and Sect. 5 Results have all new information based upon the methods presented in this paper.



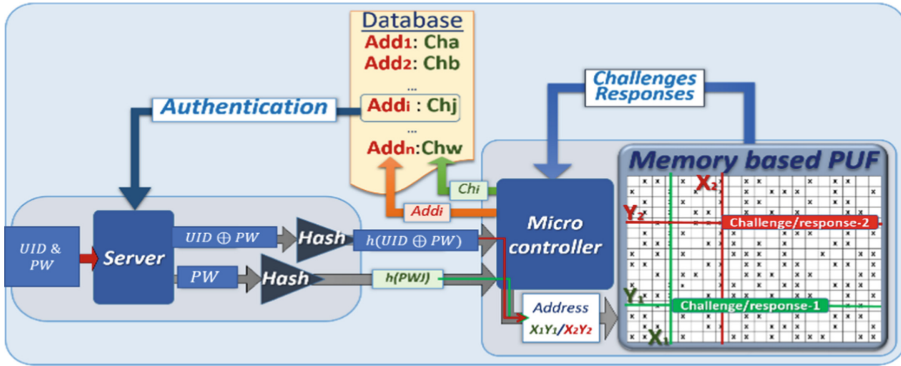
**Fig. 3.** The Architecture of protocol of the password management with ternary APG.  $h(\text{ID})$  is stored as address in DB, whereas the  $h(\text{PW})$  is fed to PUF, and challenge of PW is stored in DB. [7].

### 3 Methodology

Figure 4 shows the protocol approach of the password manager with the APG. This approach receives the UID and PW from the client, and the rest of the protocol procedures are done on the server side. The main purpose of this protocol is to store the UID and PW in an encrypted form in the DB by using the APG outputs. If hackers access the DB, they will not be able to read the content or understand the information in the DB.

After the UID and PW are received, they are XORed together. Both the XOR output and the PW will be fed separately to the hash function to obtain two MDs (MD1, MD2). Both MD1 and MD2 will be fed to the APG to get two different challenges as shown in Fig. 2(b) [8], which is different from [7]. The challenges that come from MD1 will point to the position (Add) in the DB. Whereas, the challenges (Ch) that come from MD2 will be stored in the database corresponding to its address (Addi) (as shown in Fig. 4).

To implement this approach, we use the C++ to code all protocol steps. Also, the SHA-3 is used as the hash function and the SRAM is used as PUF. To illustrate the result of the protocol, the graph user interface (GUI) has been built using HTML, java Script, and JSON. We use JSON to read the output from the C++ code environment and display the output in GUI.



**Fig. 4.** Block diagram showing a password manager for both the hash of PW and H(UID⊕PW) fed to the APG [5].

### 3.1 The Protocol Steps

As in [7], we considered two modes for the password manager: new users and existing users. The new user must be used the first time in order to register the new user’s existence in the system. Whereas, existing users must verify their existence.

#### The New User Section

The protocol for new users has nine steps to be implemented.

- The user will enter UID and PW
- The system will XOR UID with PW
- The system will hash the output of XORing operation to get MD1.
- The system will hash the PW to get MD2.
- In steps 3 and 4, SHA-3-512 has been used as hash function, and the SHA-3-512 produced 512 bits MD long. The security level of SHA-3-512 mentioned in Sect. 3. B (see Table 2).
- Both of MD1 and MD2 will extract the challenge from the SRAM. This step is different from the protocol in [7].
- The SRAM PUF is read and then the output of the SRAM is saved in a one-dimension array. This array is 32 k bytes because the SRAM size used a PUF equal to 32 k bytes.
- From MD, the first and second bytes hold the value of “X”, and “Y,” which point to the location in the PUF. After we find the location, the n bits will be read and stored in a new array. If the MD is 64 bytes long, then the number of the locations that will



be found in the PUF is 32 locations (as shown in Fig. 2b). At the end, the challenge length will be extracted from the SRAM PUF if the  $n$  bits equal 16 is 512 bits (see Fig. 2(b)) [6, 8].

- After the two challenges are extracted, the challenge received from MD1 points to the address, and the challenge received from MD2 considers the content of the password. This step is different from the protocol in [7].
- The content of the password that was obtained with its corresponding address will be stored in the DB. However, some of the users will get the same address and different content of the password. The result of getting the same address and different content of a password will cause collisions in the DB.

### *Solving the Collision*

A collision occurs when some users get the same address because the challenges of these users are stored in the same address as previous users in the memory. To address storing the challenges of addresses and challenges of passwords, a linked list scheme has been used to avoid collision between similar addresses. The link list solves this issue by storing the challenges of the password with the same address in nodes; the address points to these nodes. Using the linked list algorithm is one of main differences from the [7] protocol.

### **The Existing User Section**

For existing users, all steps are the same as the new user steps except step number 9. Step 9 will only be for comparing responses that have been extracted from the PUF with challenges that already exist in the DB. If the new response is similar to the existing challenge of the user in DB, it will accept the user; otherwise it will reject the user. In comparing operations in step 9, one issue has been found. The issue is that the SRAM BUF has cells constant with “0” and “1,” but it has also some cells that change to sometimes be zeros and sometimes ones. This means that both responses for the address and the password content will have flaky bits, which leads to mismatch with the challenges that have already been stored in the DB.

### *Solving the Flaky Bits Issue in the Response*

The first issue with flaky bits involves password content, which has been solved by measuring the error rate between the challenge and response. This solution is slightly similar to the protocol in [7] which is if the error rate is less than six percent, the response will be accepted; otherwise it will be rejected. For instance, if the responses are 512 bits, and the number of flaky bits is 30 bits, then the percentage will be  $30/512 = 0.058$  which is approximately 6%. We can say that, having 30 flaky bits in one response leads to accepting the user as a real existing user in the system.

The second issue with flaky bits involves the responses that point to the location in the DB. If flaky bits happen in responses that point to the address, the location for the correct password will not be found. This problem of flaky bits in responses pointing to the address has been solved by checking all bits that might have gotten flaky bits on it. For example, if the challenge points to location number 20 in hexadecimal, as we know  $20 = 0010\ 0000$  binary, the first bit is flipped in response; so, the value of byte

becomes 1010 0000 = A0, and then we will receive different numbers, which will point to different locations. For this reason, we have suggested building a function that can check all possible occurrences in which one or two bits have flipped amongst the byte's bits (see Fig. 5). If all possible occurrences of flaky bits have been checked, and the bits that have been flipped are corrected, then one value will match the correct address. When bits are been flipped among the byte's bits, where (1) shows n chosen r, where n number of bits and r number of chosen. When one bit is flipped among the byte's bits, then there are 8 possibilities to find the flipped bit. Whereas, with two bits are flipped then there are 36 possibilities (2).

$$c^R(n, r) = \frac{(n + r - 1)!}{r!(n - 1)!} \quad (1)$$

which will be in our example equal is:

$$c^R(8, 2) = \frac{(8 + 2 - 1)!}{2!(8 - 1)!} = 36 \quad (2)$$

Importantly, we decided the optimal length of the address is one byte. If the length is more than one byte, the possibility of getting flaky bits is high. We can say that check all the possibilities of one or two flipped bits in one byte is considered as new solution for the address issue in this paper. Table 1 shows the possibilities of a byte's values if we have two flipped bits in 1, 2, 3, 4, and 5 bytes.

### 3.2 Security Levels

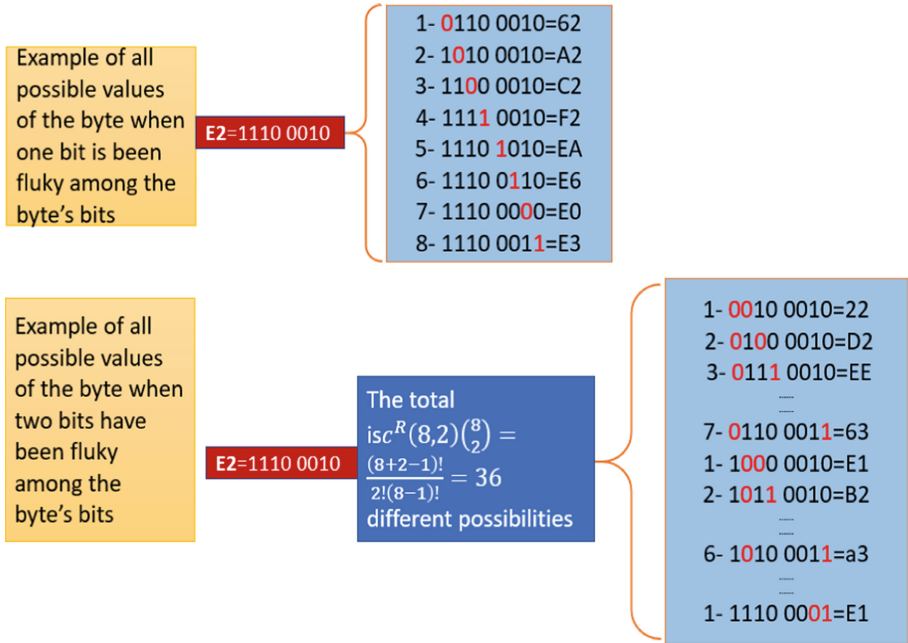
There are several levels of security in this protocol. As mentioned in Sects. 2.1 and 2.2, the hash function and the APG will be used in this protocol. Both APG and hash function have levels of security as explained in the next sections.

#### Entropy Enhancement in the APG

The example given is the one of an APG based on an array of 256 x 256 cells. The first 8 bits are used to find the X coordinate, and the next 8 bits are used to find the Y coordinate. The n-cells located after that address are used to generate PUF challenges (or responses) that consist of n-bits. Considering that message digests contain long streams of bits, typically 512, k addresses in the APG can be selected from each message digest, and m cells can be used by the address to generate the n-bit challenges (or responses), with n = km. For example, if n = 512, and k = 32 addresses are selected from the message digests, m = 16 bits are generated at each address. This largely increases the randomness of the protocol. For example, the number of combinations to select 16 bits each time from different positions in the PUF has 256 × 256 cells and the result of the combination C(256 \* 256, 16) the result of that is a huge number, which means that using the brute force attack to find the 16 bits that have been selected is impossible.

#### Security for Hash Function

For the security level in general, the hash generates fixed-sized data streams that have fixed output lengths no matter of the size of the input. They are "image resistant,"



**Fig. 5.** Showing an example when 1 or 2 values of bits, their values have changed in one byte. When one bit is flipped among the byte's bits, then there are 8 possibilities to find the flipped bit. Whereas with two bits are flipped, then there are 36 possibilities.

**Table 1.** The number of different combinations for two bits unstable(fluky) in 1, 2, and 3 bytes

Number of bytes	The combination replacement $C^R(n, r)$
1 byte ( $C^R(8, 2)$ )	036 different byte values
2 bytes ( $C^R(16, 2)$ )	136 different byte values
3 bytes ( $C^R(24, 2)$ )	300 different byte values
4 bytes ( $C^R(32, 2)$ )	528 different byte values
5 bytes ( $C^R(40, 2)$ )	820 different byte values

which means any small change in the input creates a new hash message digest that is totally different from the original message digest, as well as “collision-resistant,” which means the probability that two different inputs will create the same output is extremely low. The Secure Hashing Algorithm (SHA) is a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS).

Currently, there are three defined algorithms, which are SHA-1 SHA-2 and SHA-3. According to [1, 2, 12–17], passwords should be hashed with either PBKDF2, bcrypt or scrypt, MD-5 and SHA-3. We decided to use SHA- 3 512 bits because of the security

level strengths according to [11]. According to FIPS PUB 202 Table 2 summarizing the security strengths of the SHA-3 functions.

**Table 2.** Security strengths of the SHA-1, SHA-2, and SHA-3 functions (provided by FIPS PUB 202 [11])

Function	Output size	Security strengths in bits		
		Collision	Preimage	2nd preimage
SAH-1	160	<80	160	160-L(M)
SAH-224	224	112	224	min (224, 256-L(M))
SAH-256	256	128	256	256-L(M)
SAH-384	384	192	384	384
SAH-512	512	256	512	512-L(M)
SHA3-224	224	112	224	224
SAH3-256	256	128	256	256
SAH3-384	384	192	384	384
SAH3-512	512	256	512	512

## 4 The Implementation

The implementation of the new password manager protocol uses the hash function and PUF with several stages. This includes the GUI, the hash function, PUF, two modes and DB.

### 4.1 Building the GUI

For implementation, we built a graphical user interface (GUI) (as shown in Fig. 6) to simplified the protocol steps. The software that has been used to build the GUI are java script, Node, and WebStorm. First, the users must indicate if they are a new user or already existing user. After that, the user will enter the UID and the PW, then press the login button. When the user completes the login, all inputs will be sent to the main C++ code through the JSON tool. The inputs will be processed, then the outputs will be returned to the GUI. The outputs will be illustrated in all GUI fields as shown in Fig. 6 for new users, and existing users. To connect the C++ code with GUI, we used the JSON library to divide the output of the code as several objects. After that each object will connect to its part in the GUI by using java script and HTML.

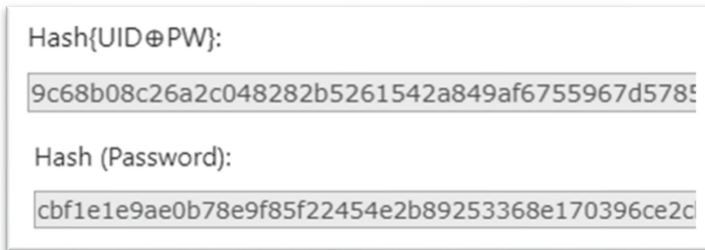
### 4.2 Building the Hash Function

In this work, it has been built a function to do the XOR for the UID with the PW ( $UID \oplus PW$ ). Both the output of the XORing and the PW itself will be fed to the hash



**Fig. 6.** GUI for new and existing user mode

function separately. The type of hash function that has been used in this work is SHA-3 512 bits. In this simulation, a function has been built that can take two parameters; one parameter is an array of streams of bits that come from the input string; whereas the second parameter is to return the result of hashing to the input in both modes (i.e. new user and existing user), the protocol requires the function that hashes two times, which produces two outputs. The two outputs of hash function are named MD1 and MD2. MD1 comes from the hash ( $UID \oplus PW$ ), whereas MD2 comes from the hash of the PW (see Fig. 7).



**Fig. 7.** The output of hash function  $H(UID \oplus PW) = MD1$ , and  $H(PW) = MD2$

### 4.3 Building the PUF

The SRAM PUF has been used for the PUF, which was developed by a cybersecurity lab at Northern Arizona University (NAU) (as shown in Fig. 9). According to the Ternary Addressable Public Key Infrastructure (TA-PKI) described in [10], there are “generatePubPriKeys” functions and “getPriKey” functions that retrieve associated public and private keys from the SRAM PUF. In the code, a function has been built to send a

“command read” to read the SRAM. The function must be called each time is needed to complete the registration or authentication. After reading the SRAM, the output of reading the SRAM will be saved in a one-dimension array to be used as the PUF.

Each two consecutive bytes in both MD1 and MD2 point to a position in the SRAM. To obtain the challenges, each two consecutive bytes from both MD1 and MD2 are fed to the SRAM PUF. By getting the position in the SRAM, the following 16 bits will be read and stored in new array of bits. If the MD is 64 byte long, we can say that there are  $64/2 = 32$  different positions from the SRAM that will be read. From each position, 16 bits will be obtained, which means that  $32 * 16 = 512$  bits can be obtained from the one MD. The new 512 bits are named the challenge or response. The challenges that comes out of MD1 will point to the address column in the DB whereas the challenges that come out of MD2 will be stored in the password column corresponding to its address (see Fig. 8). The first byte from the challenge (UID $\oplus$ PW) is 3f in hexadecimal form, which equals 63 in decimal; this means that the challenge of the PW is stored in the DB corresponding to block number 63.

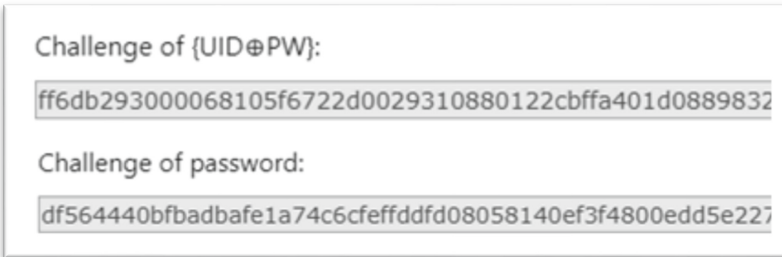


Fig. 8. The output of PUF (challenge (UID $\oplus$ PW), and challenge (PW)).



Fig. 9. SRAM PUF is developed by NAU cyber security Lab

#### 4.4 Building the Modes (New User and Exist User)

As mentioned in Sect. 3.1, the new user mode (shown in Fig. 6), will require several steps to register the user. Those steps are as following:

- Chose mode number one.
- Enter the UID and PW.
- Send the mode, UID, and PW to the C++ code.
- Complete the XOR.
- Feed the result of XOR to the hash.
- Feed the PW to the hash.
- Receive MD1, and MD2.
- Feed MD1 and MD2 to the PUF.
- Store the output in the database.

The existing user (shown in Fig. 6), first chooses the existing user mode from the GUI. After that, the same steps will be repeated as the new users from step number 2 until step number 8. Then, the output of PUF will be compared with existing information in the database; if the output of PUF matches with the existing user, then it will accept the user, otherwise, it will reject the user.

#### 4.5 Building the DB

The database table of the password manager commonly has two columns. One is for the user-ID, and the second is for the password content. Table 3 shows three ways the password manager schemes can be built on. The first scheme is to store the original format of UID and PW as shown in Table 3 first row. This scheme is not secure at all. In Table 3 s row, the hash function (SHA-3 512) has been used to hash both the UID XORing with PW and the PW. The result of the hash function is shown in row 2, but this scheme is also not recommended because the UID is public and the hash function is public too; this scheme can be exposed by using a password guessing engine. Whereas, row 3 in Table 3 shows the main contributions of our work. It shows the output of the SRAM - PUF for the UID and PW. The output of PUF in row 3 Table 3 affirms that the original format of the data will change to a different format. From the PUF output, we can affirm no one will be able to understand the content of the database, and also, no one can guess the password unless the person has the same PUF.

For building the database we used the linked list, which is a data structure algorithm. As we mentioned in Sect. 3.1, the reason for using the linked list is to avoid the collision of PW content when the users have similar addresses. The database output is a linked list algorithm that avoids the collision by creating a new node for new PW content.

**Table 3.** Three ways for data representation in DB used for Password Manager schemes

	User ID = Hello2020	Password = Password2020	The problem
1-Original form (UID and PW)	48 65 6c 6c 6f 32 30 32 30	50 61 73 73 77 6f 72 64 32 30 32 30	Exposed by insider attacks
2-The hash function UID = h(UID⊕PW) PW = h(PW)	0B53315A7468159E709A947 184D73C370EC043A5473ADE 455294C31A61883F27470E4 6B71A665ADDC71D56DBA879EA4AD3291 ECC558BD8D2240E8BE279FE71AD	ECF070F96BEC2A96507F1B229 2467866D0A07B7F09EB48C6 79B5F6726D02D0942F6F4E 0EB75DB1BF689E79980759BC1C 9CACD2FE5EE529BEF24D327F7444776A	Exposed by “password guessing”
3-The PUF UID = PUF(h(UID⊕PW)) PW = PUF(h(PW))	BDAE029A4001250A08191EDC3 1607F7FFF9F6820028443BBC 0F44CA021803DBF40804 122B76495F2860C0140FFFFB 29769042B3E0816EFBE8020B71F1DFFE2299	FC9E00019FEBFF6BC44A6F5 90041F3EF3008EFFB3FFF8F755 67FA024AD774C43F7FDD FF722005840F6F7EBED4440FFFFFEE 7DA8FFFFDFDFFF79B5075FF023F	Protected by the PUF



## 5 Results

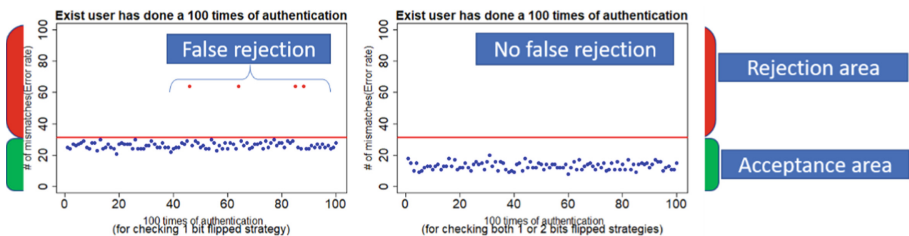
There are several experiments available to test how many mismatches occur when the user logs into the system, as well as how many times the error rate value will be higher or less than 6%. In this work, we have suggested three experiments. The experiment number one is to authenticate an existing user 100 times, whereas experiment number two will authenticate the same existing user 1000 times. The experiment number three is to test both false acceptance and false rejection while running the system.

### 5.1 Experiment One and Experiment Two

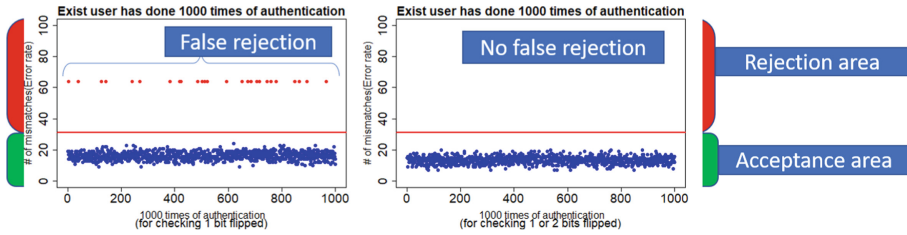
First, we will register the new user, and then the new user will be authenticated 100 times. As mentioned in Sect. 3.1, we will test the address when the address has a possibility of one bit being flipped, and we will test when the address has the possibility of two bits being flipped. Experiment two is same as experiment one except the new user will be authenticated 1000 times.

After completing experiment one, the result (in Fig. 10, left graph) indicates that for testing one flipped bit in the address, there are 96-times the user was considered as an existing user; whereas there are four times the user was not considered an existing user in the system. Those four rejections are considered false rejection. However, when both possibilities are checked for one or two flipped bits, the false rejection has been eliminated (as shown in Fig. 10, right graph).

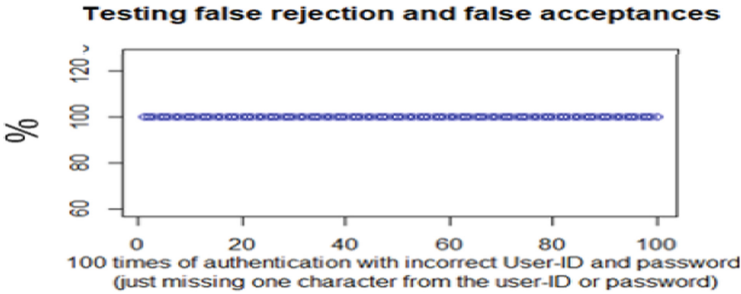
In addition, Fig. 10 shows that the existing user has been accepted 96 times when we checked for one flipped bit among the address bytes; the number of mismatches between the existing challenge in the DB and the new response from the PUF is less than 30 mismatches each time. The number of mismatches never goes higher than 30 mismatches. For this reason, the result of the experiment supports the protocol hypothesis. It is mentioned in Sect. 3.1 that if the response lengths are 512 bits and the number of flaky bits is 30 bits, then the percentage will be  $30/512 = 0.058$  which is approximately 6%. Having 30 flaky bits or less in one response leads to accepting the user as a real existing user in the system. From the results of experiment one (Fig. 10) and experiment 2 (Fig. 11), if the error rate is 6% or less when using the SPAM PUF in the password manager protocol, the user will be authorized to enter the system.



**Fig. 10.** Experiment 1 - for 100 times of authentication we show the number of mismatches and associated false rejections for the existing user when the address has 1 or 2 bits flipped.



**Fig. 11.** Experiment 2 - for 1000 times of authentication we show the number of mismatches and associated false rejections for the existing user when the address has 1 or 2 bits flipped. With checking 1-bit flipped strategy, still, there is false rejection. Whereas, 2-bit flipped strategy shows no false rejection occurred.



**Fig. 12.** Experiment 3 - the percentage of false acceptance is 100%, which means the number of mismatches is higher than 30 mismatches in each authentication time. Which means that the incorrect user has been rejected all times.

**5.2 Experiment Three**

False acceptance and false rejection need to be tested. Experiment one and experiment two test for false rejections. Experiment three is needed to test for false acceptance. The experiment steps are completed with registration of a new user. For authentication, we enter the UID with one missing character and the PW without any missing characters and repeat that 100 times. We perform the opposite operation for the UID without any missing characters whereas the PW is missing one character; this is also repeated 100 times.

For false acceptance, both situations have given positive results. Basically, the expected result from the experiment is that not allowed the user to be enter the system at all, because there is a missing character in one of UID or PW. The result of experiment did not show any false acceptance; all attempts for login were rejected (as shown in Fig. 12). As mentioned in Sect. 3.3, any mismatches higher than 30 mismatches between the challenge and response will be rejected from the system. The result of experiment three in Fig. 12 affirms the protocol hypothesis. The result shows all the login attempts higher than 30 mismatches led the system to reject the user. Logically, Fig. 12 shows that 100% of wrong login attempts were rejected, which is the desirable result. On the other hand, for false rejection (as shown in first part of Fig. 10 and Fig. 11), we tested the

address with the possibility of one flipped bit, there was a four percent false rejection. The second part of Fig. 10 and Fig. 11 shows that when we tested the address when it has the possibility of two flipped bits, the result shows there is no false rejection.

## 6 Conclusion and Future Work

In this research we have implemented a password manager that utilizes the SRAM-PUF and hash functions. Generally, the research presented is an extension with improvements to our previous research efforts as described in [7]. In this work we just focused on the software part to implement the password manager protocol; this works has included the GUI, the hash function, the SRAM PUF, the two modes (new user and existing user) with evaluating the result, and also it shows the result of the database how it looks like. Besides, the result of reducing the rate of the false rejection and false acceptance has shown a positive result in this research. Storing the original format of the user-ID and password has a security issue, which is if the database is hacked, all information will not be secret anymore. So, for password protection, the hash function has been used to change the original format of data to be encrypted by one-way encryption [18–20]. However, some attacks such as password guessing can be exploited against using hash function. By using the PUF, it will give an additional level of security to be helped to eliminate several kinds of attacking the password.

In this work both the PUF technology and the hash function are used. The output of the hash function, which is named message digest, is going to be fed to the PUF, and the output of PUF, which is named challenge/response, will be stored in the database. So, what has been stored in the database is become totally different from the original data; it is the output of PUF. Therefor without the same PUF, no one can retrieve the password or guessing the password. To get the same PUF, it will be hard because of that the PUF is hardware. The PUF that has been used in this study is the SRAM, and the hash function that has been used is the SHA-3 512. The SRAM PUF cells values have three statuses “0” for some cells “1” for some cells and some cells sometimes give “0” and sometimes give “1”. The fuzzy status will cause sometime false rejection for the real user during the authenticate the user. So, to solve the issues of the fuzzy status, we used two different patterns. The first pattern is that when the fuzzy bit in password content; then we have calculated the error rate between the challenge and response for the PW content. The second pattern is that when the flaky bit has occurred in the address; here we must check all combinations which might have one or two flaky bits among the address bits. So, depending on the error rate values for PW content and all combinations of flaky bits in address the validation of the user can be made. In conclusion, the password manager protocol using the hash function and SRAM PUF shows good result for changing the content of the database. Since the hash function is a one-way encryption, it is impossible for the hackers to infer the input of the hash function by looking at the SRAM PUF challenge that has been stored in the DB.

For the future work, prototypes should include additional password process models, such as the password expiration, replacement, verification of their strength, and firewall protection. Applying different data structure algorithms should also be considered to improve the system efficiency. Moving forward, the suggested algorithms should be

further improved the strength of PUF with different kinds of PUFs, such as the DRAM, Flash, MRAM, MRAM, and ReRAM based low power PUFs [21–24]. Additionally, the size of address in this work one byte. Thus, it should be bigger than one byte for the coming work.

**Acknowledgments.** The author is thanking the contribution of several graduate students at the cyber-security lab at Northern Arizona University, in particular, Christopher Philabaum, Vince Rodriguez, Ian Burke, and Dina Ghanaimiandoab. Also, the author is thanking the contribution of Jazan University.

## References

1. Coates, M.: “darkreading.com,” Safely Storing User Passwords: Hashing vs. Encrypting, 4 June 2014. <https://www.darkreading.com/safely-storing-user-passwords-hashing-vs-encrypting/a/d-id/1269374>. Accessed 20 Dec 2018
2. Gordon, W.: “Life hacker,” How Your Passwords Are Stored on the Internet (and When Your Password Strength Doesn’t Matter), 20 June 2012. <https://lifelhacker.com/how-your-passwords-are-stored-on-the-internet-and-when-5919918>. Accessed 28 Aug 2018
3. Higgins, K.J.: Dark reading, 8 May 2008. <https://www.darkreading.com/risk/hackers-choice-top-six-database-attacks/d/d-id/1129481>. Accessed 25 Oct 2018
4. Cambou, B.: Physically unclonable function based password generation scheme. United States of America Patent D2016-011, September 2016
5. Cambou, B.: Addressable PUF generators for database-free password management system. In: *Advances in Intelligent Systems and Computing*, Flagstaff (2018)
6. Cambou, B.: Password manager combining hashing functions and ternary PUFs. In: *Intelligent Computing-Proceedings of the Computing Conference*. Springer, Cham (2019)
7. Mohammadinodoushan, M., Cambou, B., Philabaum, C., Hely, D., Booher, D.: Implementation of password management system using ternary addressable PUF generator. In: *IEEE SECON 2019: IEEE STP-CPS Workshop*, June 2019, to appear
8. Assiri, S., Cambou, B., Booher, D.D., Miandoab, D.G., Mohammadinodoushan, M.: Key exchange using ternary system to enhance security. In: *IEEE 9th Annual Computing systems and Conference (CCWC)*, Las Vegas (2019)
9. Cambou, B.F.: Encoding ternary data for PUF environments. USA Patent US20180131529A1, 09 November 2016
10. Booher, D.D., Cambou, B., Carlson, A.H., Philabaum, C.: Dynamic key generation for polymorphic encryption. In: *IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, pp. 0482–0487 (2019)
11. Technology, Information Technology Laboratory National Institute of Standards and, “Team Keccak,” August 2015. <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf>. Accessed 12 Nov 2018
12. IT security community Blog: IT security community Blog, 13 September 2013. <https://security.blogoverflow.com/2013/09/about-secure-password-hashing/>. Accessed 22 Feb 2019
13. Arias, D.: auth0.com hashing passwords: one-way road to security, 25 April 2018. <https://auth0.com/blog/hashing-passwords-one-way-road-to-security/>. Accessed 4 Feb 2019
14. Keane, J.: Security researcher dumps 427 million hacked myspace passwords online, July 2016. <https://www.digitaltrends.com/social-media/myspace-hack-password-dump/>
15. Blocki, J., Harsha, B., Zhou, S.: On the economics of offline password cracking. In: *IEEE Symposium on Security and Privacy (SP)* (2018)

16. Zhang, Z., Yang, K., Hu, X., Wang, Y.: Practical anonymous password authentication and TLS with anonymous client authentication. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1179–1191. ACM (2016)
17. Tsai, J.-L.: Efficient multi-server authentication scheme based on one-way hash function without verification table. *Comput. Secur.* **27**(3–4), 115–121 (2008)
18. Balakrishnan, H., Popa, R.A., Zeldovich, N.: Methods and apparatus for securing a database. USA Patent US13/357,988, 25 January 2012
19. Gabel, D., Liard, B., Orzechowski, D.: Cyber risk: why cyber security is important, 1 July 2015. <https://www.whitecase.com/publications/insight/cyber-risk-why-cyber-security-important>
20. Bob, R., Phil, H., Walt, S., Bill, W.: “OUCH!,” SANS, October 2013. [https://www.phas.ubc.ca/sites/default/files/shared/it-service-catalogue/ouch/ouch-201310\\_en.pdf](https://www.phas.ubc.ca/sites/default/files/shared/it-service-catalogue/ouch/ouch-201310_en.pdf). Accessed 5 Oct 2018
21. Cambou, B., Afghah, F., Sonderegger, D., Taggart, J., Barnaby, H., Kozicki, A.M.: Ag conductive bridge RAMs for physical unclonable functions. In: 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), McLean (2017)
22. Korenda, A., Afghah, F., Cambou, B.: A secret key generation scheme for internet of things using ternary-states ReRAM-based physical unclonable functions. In: Submitted to International Wireless Communications and Mobile Computing Conference (IWCMC 2018) (2018)
23. Cambou, B., Orlowski, M.: Design of PUFs with ReRAM and ternary states. In: CISR 2016, April 2016
24. Cambou, B., Afghah, F.: Physically unclonable functions with multi-states and machine learning. In: 14th International Workshop on Cryptographic Architectures Embedded in Logic Devices (CryptArchi), France (2016)



# Contactless Palm Vein Authentication Security Technique for Better Adoption of e-Commerce in Developing Countries

Sunday Alabi<sup>(✉)</sup>, Martin White, and Natalia Beloff

University of Sussex, Falmer, Brighton, UK  
sa405@sussex.ac.uk

**Abstract.** e-Commerce has contributed immensely to the economies of developed countries and a factor in its success can be attributed to the adoption of e-commerce by their citizens. As such, it is perceived that e-commerce can also be an economic driver for developing countries. However, security has been identified as a major barrier that prevents citizens from adopting e-commerce in developing countries. Therefore, this paper examines Security Authentication Techniques (SAT), particularly Digital Signature (DS) and Digital Fingerprint Systems (DFS), including the limitations of these two security techniques, and then proposes Contactless Palm Vein Authentication (CPVA) as a potentially much better solution to increase adoption of e-commerce in developing countries. The architecture of this new CPVA technique is discussed in relation to Security, Privacy, Trust and Reliability. Participants are treated to a Design Fiction Documentary (DFD) and Design Fiction Simulation Experiment (DFSE) in our experimental design method to measure the potential Technology Acceptance (adoption) of the proposed CPVA technique over DS and DFS authentication techniques. The result of our pilot study indicates that citizens may be willing to adopt the proposed CPVA technique, which may increase their trust and likely adoption of more e-commerce applications. A larger main study is planned in the field in Nigeria starting January 2020.

**Keywords:** Palm vein · Design fiction · Reliability

## 1 Introduction

Nigeria is the largest African country [1, 2] with a population of 180 million, which has grown rapidly in the last 20 years. There is plenty of good fertile land for agriculture and other natural resources. All countries in Africa are still either underdeveloped or developing countries. Nigeria is located on the African continent and is one of the countries of the sub-Sahara region [1, 3]. Most of these countries have unstable economies and this is affecting the economic development of that continent. In actual fact, their economic problems can also be attributed to a lack of infrastructural facilities and poor governance [4]. Non-adoption of e-commerce by citizens also contributes to stunted growth of their economies [1].

e-Commerce is an online transaction processing (OLTP) technology in which the system responds immediately to user requests [5, 6]. This technology has made the world a global village comprised of many opportunities. Online transactions are ways of carrying out transactions via the Internet and it has been described as the new driver for economic growth especially for developing countries [4]. This provides a great opportunity for organizations, individuals and nations at large.

However, issues of security have been identified as a factor limiting e-commerce development and its total adoption in developing countries [7, 8]. Crime associated with theft and data manipulations are often detected [9]. One of the reasons why identify theft is so widespread is due to ineffective security measures. The most common method, at this time, of identity verification is based on digital identity and signature. These methods of security use: code, e.g. passwords and other behavioural features, to identify the person. However, all of these features are relatively easy to steal or forge, therefore, they are not effective identity verification or authentication methods [10].

Another security measure is fingerprints authentication. Okechukwu and Majesty argued in [11, 12] that it is necessary to introduce forensic methods of security into the e-commerce of developing countries so as to uplift the adoption rate of citizenry and benefits of e-commerce. Introduction of fingerprint identification into e-commerce applications make the system more secure and alleviate citizens' fear to a certain extent, that is, until the weakness of Digital Fingerprint Systems (DFS) becomes more apparent [13]. Due to the nature of many people's work, particularly those are who are involved in manual labour, damage to the finger tips are sustained and this leads to the DFS failing to recognize the user fingerprint. This means that at the moment, this method is not effective for many people. Therefore, introduction of a new method called Palm Vein Authentication becomes important.

Palm Vein Authentication (PVA), is a digital security technique that uses an individual vein pattern as personal authentication and identification data [14]. The research work outlined in this paper investigates the possibility of introducing Palm Vein Authentication to enhance e-commerce applications in developing countries focusing on Nigeria. Palm Vein Identification uses the unique internal vein pattern of the palm as a transactional authentication method. Its benefits are uniqueness, difficult to forge, secure and reliable [14, 15]. This method will help citizens of Nigeria and other developing countries since vein of the palm cannot be easily damaged due to dryness and by citizens engaging in hard labour using the hands.

The rest of this research paper is organised as follows: Sect. 2 discusses the research background, while Sect. 3 explains the research questions. Section 4 discusses the methodological approach used, Sect. 5 explains the experimental approach, Sect. 6 briefly presents the pilot study results, which lead the way to conduct a more extensive full study in the field in Nigeria (discussed in future work). Section 7 discusses our results so far in more detail. Finally, Sect. 8 draws some conclusions and states the future work.

## 2 Research Background

This research will consider the security aspects of user identification transaction systems, e.g. payment in e-commerce application, election authentication, access to e-government

and e-health services, building access, etc. in developing countries, specifically focusing on Nigeria as a case study.

A Design Fiction [16] approach will be used to educate users on security issues associated with user identification transactions. In particular, the Design Fiction will illustrate the use of Digital Signature (DS, e.g. pin, password) [17, 18], DFS [19] and Contactless Palm Vein Authentication (CPVA) [20] to authenticate (authorise) access to user identification transaction systems (UITS). The research project builds a simulation of an e-commerce application that will accept DS, DFS and CPVA access to a UITS, where in this case the UITS will be a simulated payment system, i.e. a simulation model that deploys various technology prototypes, and existing ICTs to implement the scenarios depicted in a Design Fiction [21]. For the purposes of the study, the Design Fiction will encompass an e-commerce shopping application that will be developed to facilitate measurement of the users acceptance of transactional risk based on the user identification method, i.e. the intervention (DS, DFS or CPVA) with a focus on palm vein authentication [22, 23].

The main aim of this research is to investigate the advantages of CPVA over DS and DFS technologies, and to determine how factors such as: security, awareness, trust, privacy, cost, digital identity theft, etc. (see Fig. 2) might affect the adoption of CPVA in Nigeria's existing IT infrastructure [24]. It is well reported in the literature that Nigerian Citizens do not have a high level of trust in Nigeria's current unsecure e-commerce platforms, thereby making them uncomfortable in engaging with e-commerce applications that require them to divulge personal and financial information [25]. This research aims to investigate how Nigerian e-commerce users may change their risk perception when using more secure technologies, such as CPVA. This study examines how the current use of DS security techniques has led to pervasive digital identity theft that has resulted in extensive fraudulent activities. This has led to a large degree of mistrust of e-commerce applications in Nigerian society similar to that reflected in other African countries [3, 4]. This level of distrust has to be improved with new technologies, such as CPVA, that are perceived to be able to deliver a high-level security.

### 3 Research Questions

The following research questions considered for the purpose of this research work:

1. Do biometric authentication techniques such as fingerprints, palm vein, iris, retina, Voice recognition system, facial image and digital signature overcome the 'fear and distrust' associated with e-commerce applications?
2. How can we convince the Nigerian citizen that new digital security methods, such as CPVA, can provide adequate protection against different fraudulent acts for typical e-commerce applications?
3. Will the level of risk perceived by the Nigerian citizen be adequately reduced to facilitate wider adoption of e-commerce applications if such applications implement CPVA for their user identification transaction system?

The literature survey provides ample evidence for the efficacy of iris, retina and facial image scanning, and so is not considered in this study [12, 26]. A Design Fiction



has been developed to educate the Nigerian citizen on the level of security established in different e-commerce applications using DS, DFS, and CPVA to facilitate answers to these research questions.

## 4 Methodology

This Design Fiction based experimental research method will measure the perceived level of risk that Nigerian citizens will adopt when using new security technologies, such as CPVA. The method will examine the impact of changing an independent variable (i.e. the factors: security, risk, fear, Web Assurance Seals Services (WASS), which relates to Trust, and usability, of an e-commerce application) to measure its effect on the dependent variable (Intention to Adopt) and therefore provide insight into the effects and consequences of distrust associated with the users' involvement, engagement and interactivity in their online experience of e-commerce using the Design Fiction (Documentary and Simulation Experiment) and their associated treatments.

The DF treatment is composed of a Design Fiction Documentary (DFD) shown to all participants, and a prototype development of an e-commerce application, i.e. a simulation incorporating CPVA, called the Design Fiction Simulation Experiment (DFSE) treatment. The overall DF treatment (i.e. DFD + DFSE) is followed by a post treatment survey, which is a detailed questionnaire, designed to elicit information around 'Intention to Adopt' from the subject participants based on the hypotheses (coded as H2d, H4a, and so on, see Fig. 1 and Fig. 2) that link the independent variables to the dependent variable. Figure 1 and 2 illustrate the e-commerce trust model entity relational diagram linking independent variable to the dependent variable.

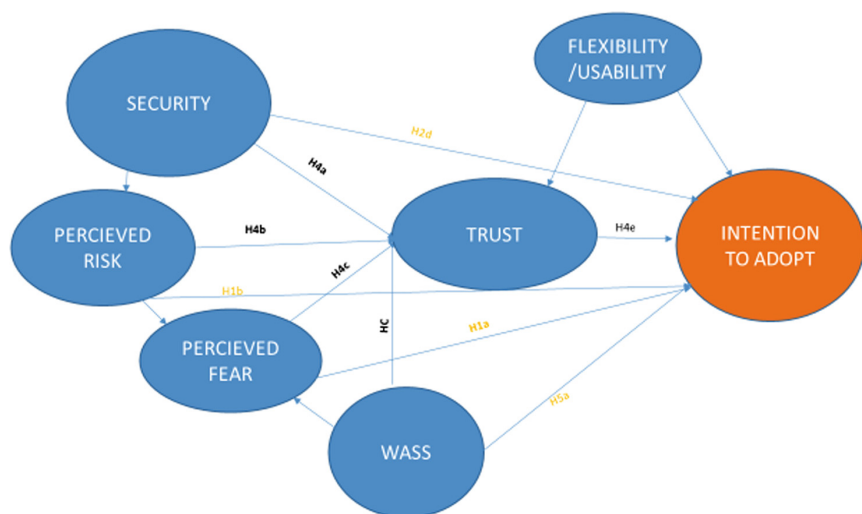
The associated hypothesis and questionnaire are too long and detailed to include in this paper, however they can be accessed in the GitHub archive for this project<sup>1</sup>.

We can see from Fig. 2 that many other factors affect users' perceived fear, risk, security, usability and so on, hence intention to adopt e-commerce. For example, technology factors around security such as use of CPVA, DS, DFS, a user's perception of whether their finger print will work, fear of digital identify theft, privacy issues all impact trust and intention to adopt. How aware the user is concerning security of e-commerce, what previous experience they have had, and specifics of that awareness also affect their perceived risk, hence trust of e-commerce and their subsequent intention to adopt.

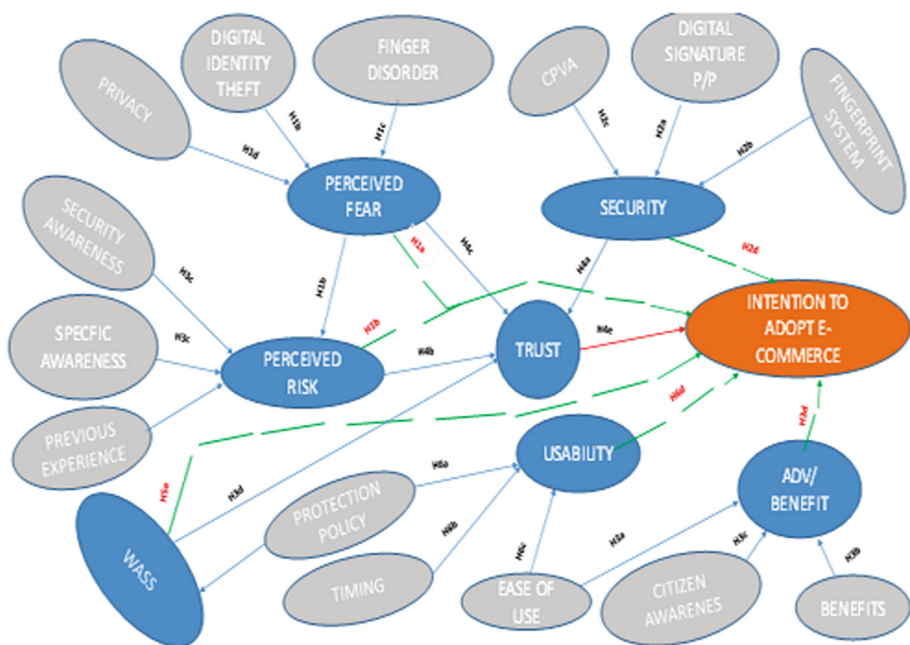
## 5 Experimentation

The experiment focuses on the use of a Design Fiction based treatment to first explain and educate the user on the benefits of CPVA, in terms its advantages over DS and DFS, with respect to securing and building trust worthy e-commerce transactional applications—this is the DFD. Then the developed DFSE task (i.e. a CPVA based e-commerce simulation) task will be taken by each participant using Within Subjects Design—here, each participant is exposed to every factor in the DFSE treatment (i.e. manipulation of all the different levels of the independent variables) to measure their Intention to Adopt e-commerce in their daily lives.

<sup>1</sup> <https://github.com/sundayalabi/>.



**Fig. 1.** Overview of the e-commerce trust model illustrating the experimental variables relationship



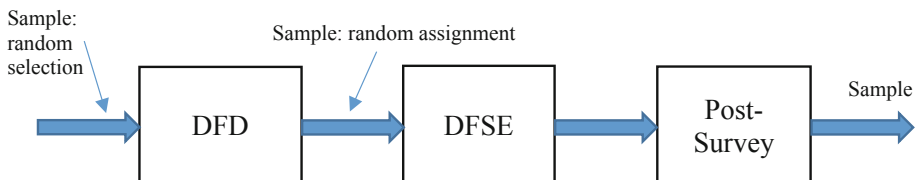
**Fig. 2.** Detailed research model entity relation linking independent and dependent variables

After completing the DFSE (CPVA based) experimental tasks, each participant will complete the post-experimental survey. The DF based treatment and survey will be conducted in the chosen area Centre (e.g. community hall in a Nigerian town). Thus, the design of each experiment includes a structured questionnaire (post-experimental survey), DFD (an explanation of the issues concerning e-commerce transactions) and the DFSE based on a CPVA paradigm (the e-commerce simulation). Appropriate subjective measures in the form of post-experiment questionnaires are selected for analysis.

The population of this research includes chosen categories of citizens of Nigeria. The volunteers (samples) are eligible to partake in the study whether they are experienced with e-commerce applications or not. Participants will be invited to participate as an individual. Participants will be randomly assigned to the treatment, (i.e. all participants will be selected for the DFD and randomly assigned the DFSE factors in the Within Subjects Design). Generally, with random assignment, participants have an equal chance of being assigned to a specific treatment (one of the factors or independent variables) to eliminate or reduce bias in the research [27]. In this case, with our Within Subjects Design, participants are randomly assigned to the order of being exposed to the independent variables (factors). Within Subjects Design has advantages besides cost efficiencies (half as many participants needed) such as providing control of extraneous participant variables because all participants have the same characteristics for each factor being tested—each participant(s) has the same mean IQ, socio-economic background, and so on because they are the same people [28]. Further, carry over effects are considered well<sup>2</sup>.

Randomly selected and randomly assigned participants also increases the external and internal validity respectively. The experiment so far involves a small pilot of 50 participants that are Nigerian students in the University of Sussex, to test the experimental method and fine tune the process. The result of the pilot study was used to validate the designed-questionnaires, refer to the GitHub archive, and the final experimental methodology. The structured questionnaires will be administered after the DFSE (CPVA based) experimental task is completed by the participants.

Figure 3 indicates the experimental method whereby the sample users are randomly selected to the e-commerce education using a Design Fiction Documentary (DFD), and randomly assigned to the Design Fiction Simulation Experiment (DFSE)—order of treatments is randomised per participant, i.e. the CPVA e-commerce simulation—using a Within Subjects Design for economical and statistical efficiency [16, 21]. DFD and DFSE is then followed by a Post-Survey.



**Fig. 3.** The ‘within subjects design’ experimental model

<sup>2</sup> <https://www.students4bestevidence.net/blog/2018/08/23/carryover-effects-what-are-they-why-are-they-problematic-and-what-can-you-do-about-them/>.

## 6 Pilot Study Result

The Design Fiction Documentary (DFD) and Design Fiction Simulation Experiment (DFSE) were tested using a minimum number of participants that are citizens of developing countries who were already accustomed with the experience of e-commerce in developing countries. After the DFD, a DFSE was completed followed by the questionnaire survey administration. A questionnaire that comprises of sixteen sections was used to take feedback data from the participants after the conduct of the experiment. The results gathered includes the previous experience of computer usage and Internet and e-commerce transaction of participants. How the dependent variable (Intention to Adopt) are being affected by independent variables are also tested.

Note, from Fig. 1, we can also think of Trust as an Independent variable that is manipulated to see the effects on the dependent variable Intention to Adopt, or we can think of Trust as a dependent variable that we measure after manipulating independent variables such as fear and risk, etc.

The pilot results significantly indicates that citizens of developing countries want a better security authentication technique, which may increase the citizen's trust in e-commerce. Again, the pilot result shows that, citizens may better accept the proposed CPVA architecture compared to existing DS and DFS based e-commerce transactions.

In our pilot study, DS based on Pin and Password is actually rejected due to its vulnerability that leads to citizens Identity theft, where citizens Identity theft reduces citizen's trust towards e-commerce. The pilot study result also shows that, DFS provides a more secure platform than the DS architecture. Also, further inference is shown that DFS based identification has denied several citizens from their rightful authentication thereby increasing citizen's fears over e-commerce security and this probably leads to e-commerce rejection. The result of citizens perspectives towards these DS and DFS security authentication techniques suggest that citizens may prefer the proposed security architecture based on Contactless Palm Vein Authentication (CPVA) when carrying out their day to day e-commerce transactions. Raw data results can be found in the GitHub archive previously mentioned.

## 7 Discussion

Building acceptable trusted e-commerce systems is desperately needed in developing countries to ensure the survival of their economies [3]. Research has shown that specific infrastructural barriers are limiting and also negatively affecting e-commerce growth in developing countries [1]. P. Japhet et al. have argued that framework barriers are a hindrance [8]. Other evidence suggests that the causes of e-commerce non-development among developing countries varies [29]. Billewar and Babu argue that it is unbelievable that many developing countries did not have any policy for protecting customers for online transactions [30]. Therefore, customers are not secure and are not able to resolve their problems, which reduces customer trust in e-commerce [31]. This research work investigates many salient barriers affecting e-commerce development and uptake in these areas.

K. E. Corey et al. [5] show that e-commerce is advancing on a daily basis and benefiting individuals, organizations and nations at large to the extent of it being an economic

catalyst in this present day's economy, and this prevalence of e-commerce makes it vulnerable to attacks. e-Commerce rapid growth makes an e-commerce platform attack to be more prone and frequent [8]. Further, attacks are now becoming more advanced in nature, as such these attacks are now becoming a major barrier to e-commerce growth in developing countries [10]. Therefore, e-commerce security issues are very important and pertinent, as such this research considers the security aspect of electronic commerce using a Contactless Palm Vein Authentication (CPVA) as a superior method for securing e-commerce transactions. Other research work discusses I-based passwords, finger print, finger vein and palm vein security technologies used for e-commerce transaction authentications [23, 32, 33].

Digital Signature involves the use of codes as a password for identity representation on an e-commerce platform [32]. In many cases, users have experienced losses due to stolen pin and password at alarming rates. This identity theft has impacted negatively on the development of e-commerce in developing countries [8]. However, the success of e-commerce may also impact positively on the economic situation of developing countries if security issues and identified barriers to uptake are successfully resolved [1].

The incorporation of biometrics to the PIN system has increased the security system in online transactions [29]. Although, despite security awareness customers are still often careless with their Pins and Password information, which increases the rate at which fraudsters are able to succeed in guessing their Pin or Password credentials. Evidence shows that several cases of identity theft have shown the need for new methods to improve the security aspect of e-commerce. However, more secure authentication is achieved using biometric techniques in which an individual unique identifier is used for authentication [17, 19, 34]. Physiological and behavioural features are proving to be more reliable in digital security identification [35].

Evidence shows that the fingerprint technique is not very effective for many citizens of developing countries, often due to nature of manual employment that degrades a person's physical fingerprints. Whenever the outer layer of the finger is subjected to damage, then the DFS begins to experience High False Rejection Rates (HFRR) [13], you can see this if you do some simple DIY at home, often you can't access your own mobile phone afterwards using DFS. Therefore, DFS at the moment may need to be replaced in online transactions in developing countries. However, the palm vein technique is unique for an individual, and this vein pattern under the palm can be captured with the use of an infrared camera [36]. Veins are tissues through which the blood flows in the body and the vein at the region of the palm is referring to as a Palm Vein [37]. Hand vein geometry is still at an early research stage [38]. Therefore, existing IT infrastructure may not be adequate for new CPVA security mechanisms?

However, there are examples of CPVA currently being used in a hospital to authenticate patients [39]. In such cases, sample palm vein images are being acquired from incoming patients using an infrared camera. The vein pattern of a particular patient is then used to compare with the already processed pattern in the database of the patient's medical records [40]. Laadjel M. et al. argues in [26] and M. Preethi in [22] argued that it will not be easy to steal palm vein patterns due to its complex structure and the authentication requires live blood flow through the veins. Also, Jain argues that the palm vein pattern is very likely to be more secure than a DFS [41].

Krishneswari and Arumugam show [42] show that the positive characteristics of CPVA that make this authentication method superior to DFS include:

1. CPVA systems are capable of using a pre-registered image of an individual's identity and comparing with the newly acquired image using blood veins palm pattern.
2. CPVA systems are likely to be acceptable by the user because of its non-invasiveness and the technique of using live blood veins makes the method very reliable.
3. CPVA images are difficult to replicate, this makes the technique highly dependable.

## 8 Conclusion

Security authentication methods for e-commerce transactions is attracting many researchers' attention, especially in developing countries. Their focus is on how new improved authentication techniques could be developed to increase e-commerce adoption in developing countries. The privacy, security and trust aspects have been attracting research interest since they are considered as critical issues and challenges for e-commerce adoption in developing countries. This paper reviewed a number of architectures for security authentication technique issues related to the privacy and security of e-commerce transactions.

This paper discussed the DS (Pin/Password) concept and challenges of identity theft. At the same time DFS technology architecture was reviewed, its effectiveness and the problem of High False Rejection Rate (HFRR) is discussed. In addition, this paper examined the problems associated with current security techniques. Furthermore, we proposed a new security architecture, focused on CPVA, to be used in the authentication of e-commerce in developing countries. This may overcome the issues apparent in DS and DFS architectures, particularly in relation to security and privacy since these are important in providing the adequate trust needed by the citizens. The proposed CPVA technology has a property that supports liveliness (i.e. the palm veins must have blood flowing to work), integrity, privacy and reliability.

## References

1. Oluyinka, S., Shamsuddin, A., Ajabe, M.A., Enegbuma, W.I.: A study of electronic commerce adoption factors in Nigeria. *Int. J. Inf. Syst. Change Manag.* **6**(4), 293 (2013)
2. Emmanuel, A.-A.O.: Adoption of E-commerce in Nigerian Businesses: a change from traditional to e-commerce business model in Richbol Environmental Services Limited. Thesis dissertation, Business School, Seinajoki University of Applied Sciences, Finland, p. 110 (2012)
3. Lawrence, J.E.: The growth of E-commerce in developing countries: an exploratory study of opportunities and challenges for SMEs. *Int. J. ICT Res. Dev. Afr. (IJCTRDA)* **2**(1), 11 (2011)
4. Kanyaru, P.M., Kyalo, J.K.: Factors affecting the online transactions in the developing countries: a case of e-commerce businesses in Nairobi County, Kenya. *J. Educ. Policy Entrep. Res.* **2**(3), 1–7 (2015)
5. Corey, K.E., Wilson, M.I., Lansing, E.: *e-Business and e-Commerce*, pp. 285–290. Elsevier Inc. (2009)

6. Al-najjar, S.M., Jawad, M.K.: Measuring customers' perceptions and readiness to accept e-commerce in Iraq: an empirical study. *J. Mark. Manag.* **4**(1), 151–162 (2016)
7. Shouk, M.A., Eraqi, M.I.: Perceived barriers to e-commerce adoption in SMEs in developing countries: the case of travel agents in Egypt. *Int. J. Serv. Oper. Manag.* **21**(3), 332 (2015)
8. Lawrence, P.J.E., Tar, U.A.: Persistent barriers to e-commerce in developing countries. *J. Glob. Inf. Manag.* **19**(3), 30–44 (2011)
9. Nikitkov, A.N., Bay, D.: Online auction fraud: an empirical analysis of shill-bidding practice. *J. Forensic Investig. Account.* **2**(3) (2010)
10. Ayo, C.: A framework for e-commerce implementation: Nigeria a case study. *J. Internet Bank. Commer.* **13**(2), 1–12 (2008)
11. Okwuchukwu, O.G.: Access to and pattern of ICT use among undergraduate students of Nnamdi Azikiwe University, Awka-Nigeria. *Int. J. Hum. Soc. Sci. (IJHSS)* **2**(1), 1–11 (2015)
12. Okechukwu, M., Majesty, I.: ATM security using fingerprint biometric identifier: an investigative study. *Int. J. Adv. Comput. Sci. Appl.* **3**(4), 68–72 (2012)
13. Yoon, S., Feng, J., Jain, A.K.: On latent fingerprint enhancement. In: *Conference Proceeding, Biometric Technology for Human Identification*, vol. 2 (2010)
14. Sayed, M.: Palm vein authentication based on the coset decomposition method. *J. Inf. Secur.* **6**(3), 197–205 (2015)
15. Kumari, P.A., Suma, G.J.: A novel multimodal biometric scheme for personal authentication. *Int. J. Res. Eng. Technol.* **2**(2), 55–65 (2014)
16. Coultron, P., Lindley, J., Akmal, H.A.: Design fiction: does the search for plausibility lead to deception? In: *Proceedings of DRS 2016, Design Research Society 50th Anniversary Conference*, pp. 1–16 (2016)
17. Kumar, D., Engineer, S., Solutions, I., Limited, P.: A review in various approaches of feature extraction and feature fusion in multimodal biometric system. *IJSRSET* **3**(3), 734–739 (2017)
18. Hemamalini, M., Jagadeesan, D.: Two step verification for withdraw the amount from ATM machine. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **4**(9), 698–702 (2014)
19. Dakhil, I.G., Ibrahim, A.A.: Design and implementation of fingerprint identification system based on KNN neural network. *J. Comput. Commun.* **06**(03), 1–18 (2018)
20. Elnasir, S., Shamsuddin, S.M., Farokhi, S.: Accurate palm vein recognition based on wavelet scattering and spectral regression kernel discriminant analysis. *J. Electron. Imaging* **24**(1), 1–24 (2015)
21. Grand, S., Wiedmer, M.: Design fiction: a method toolbox for design research in a complex world. *Designresearchsociety.Org*, pp. 1–25 (2006)
22. Preethi, M., Vaidya, D., Kar, S., Sapkal, A.M., Joshi, M.A.: Person authentication using face and palm vein: a survey of recognition and fusion techniques. *Int. J. Technol. Enhanc. Emerg. Eng. Res.* **3**(03), 55–69 (2015)
23. Sarkar, I., Alisherov, F., Kim, T.H., Bhattacharyya, D.: Palm vein authentication system: a review. *Int. J. Control Autom.* **3**(1), 27–34 (2010)
24. Alqahtani, M.A., Al-Badi, A.H., Mayhew, P.J.: The enablers and disablers of e-commerce: consumers' perspectives. *Electron. J. Inf. Syst. Dev. Ctries.* **54**(1), 1–24 (2012)
25. Kim, K.K., Prabhakar, B.: Initial trust, perceived risk, and the adoption of internet banking. In: *ICIS*, pp. 537–543 (2000)
26. Laadjel, M., Bouridane, A., Nibouche, O., Kurugollu, F., Al-Maadeed, S.: An improved palmprint recognition system using iris features. *J. Real-Time Image Process.* **8**(3), 253–263 (2013)
27. Bryman, A.: Barriers to integrating quantitative and qualitative research. *J. Mix. Methods Res.* **1**(1), 8–22 (2007)
28. Gray, N.S., Snowden, R.J., Peoples, M., Hemsley, D.R., Gray, J.A.: A demonstration of within-subjects latent inhibition in the human: limitations and advantages. *Behav. Brain Res.* **138**(1), 1–8 (2003)

29. Kavitha, K., Kuppasamy, K.: A hybrid biometric authentication algorithm. *Int. J. Eng. Trends Technol.* **3**(3), 311–319 (2012)
30. Billewar, S.R., Babu, D.H.: Approach to improve quality of e-commerce. *Int. J. Recent Technol. Eng.* **1**(5), 36–39 (2012)
31. Osho, O., Onuoha, C.I., Ugwu, J.N., Falaye, A.A.: E-commerce in Nigeria: a survey of security awareness of customers and factors that influence acceptance. In: *CEUR Workshop Proceedings*, vol. 1755, pp. 169–176 (2016)
32. Hwang, M.S., Chong, S.K., Chen, T.Y.: DoS-resistant ID-based password authentication scheme using smart cards. *J. Syst. Softw.* **83**(1), 163–172 (2010)
33. Hong, L., Wan, Y., Jain, A.: Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(8), 777–789 (1998)
34. Matyáš, V., Ríha, Z.: Biometric authentication systems. *FIMU Report Series*, vol. 2, pp. 1–46, November 2000
35. Abrazhevich, D.: *Electronic payment systems: a user-centered perspective and interaction design*. Proefschrift ISBN 90-386-1948-0 (2004)
36. Arakala, A., Hao, H., Davis, S., Horadam, K.J.: The palm vein graph - feature extraction and matching. In: *Proceedings of the 1st International Conference on Information Systems Security and Privacy*, no. 2015, pp. 295–303 (2015)
37. Crisan, S., Tebrean, B.: Low cost, high quality vein pattern recognition device with liveness detection. *Workflow and implementations. Meas. J. Int. Meas. Confed.* **108**, 207–216 (2017)
38. Huang, D., Zhu, X., Wang, Y., Zhang, D.: Dorsal hand vein recognition via hierarchical combination of texture and shape clues. *Neurocomputing* **214**, 815–828 (2016)
39. Kaur, N.: Vein pattern recognition: a secured way of authentication. *Int. J. Eng. Comput. Sci.* **5**(10), 18377–18383 (2016)
40. Han, W.Y., Lee, J.C.: Palm vein recognition using adaptive Gabor filter. *Expert Syst. Appl.* **39**(18), 13225–13234 (2012)
41. Jain, V.K.: A technique to ROI of palmpriint for palmline matching. *Int. J. Eng. Res. Appl. (IJERA)* **2**(6), 1007–1009 (2012)
42. Krishneswari, K., Arumugam, S.: A review on palm print verification system. *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.* **2**, 113–120 (2010)





# LightGBM Algorithm for Malware Detection

Mouhammd Al-kasassbeh<sup>1</sup> (✉), Mohammad A. Abbadi<sup>2</sup>, and Ahmed M. Al-Bustanji<sup>2</sup>

<sup>1</sup> Princess Sumaya University for Technology, Amman, Jordan

m.alkasassbeh@psut.edu.jo

<sup>2</sup> Mutah University, Karak, Jordan

abbadi@mutah.edu.jo, Bustanji@hotmail.com

**Abstract.** In Zero-Day malware challenges, attackers take advantage of every second that the anti-malware vendor delays identifying the attacking malware signature and provide the updates. Furthermore, the longer the detection phase delayed, the greater the damage to the host device. In other words, the inability to early detection of attacks complicates the problem and increases damage. Therefore, this study aims to develop an intelligent anti-malware system capable to instantly detect and terminate malware activities instead of waiting for anti-malware updates. In its scope, the study focuses on the Internet of Things (IoT) malware detection based on Machine Learning (ML) techniques. A recent open-source ML algorithm called Light Gradient Boosting Algorithm (LightGBM) is used to develop our instant anti-malware approach at both host and network layers without the need for any human intervention. The results show a promising approach for detecting and classifying malware with high accuracy reaches almost (100%) at both the network and host levels based on the cross-validation Holdout method. Furthermore, the results show the ability of the proposed approach to early detect IoT botnet attacks, which is an essential feature for terminating the botnet activity before propagating to a new network device.

**Keywords:** Malware · Machine learning · Botnet · Internet of Things · Gradient boosting · LightGBM

## 1 Introduction

Competition between attacks and security defenses will never end. With each security enhancement, new attacking tools are developed to overcome security defense. Malware or malicious software is the most common type of cybersecurity threats that can perform either active attacks, passive attacks or both together. Traditional virus scanning solutions rely on manually created malware signatures and statistics analysis, which never be able to practically satisfy the increasing demand for security defense solutions against malware. Off-the-shelf antivirus software products require to be updated frequently with the newly detected malware signatures. Therefore, traditional virus software unable to detect malware in real-time of the zero-day attack. However, after new malware's first attack and classified as wild, companies analysis the malware and create their signature then release definition updates to their products so it can recognize the new malware.

Before the release of definition updates, several terabytes of data may be lost or stolen, and millions of dollars might get lost because of these attacks. Governments, companies, and individuals are potential victims of malware attacks. With every zero-day malware attacks, there will be a massive and unrecoverable financial and data loss. The number of the victims grow as well as the loss until vendors of anti-malware update the client's software. Malware challenge is continually evolving along with the dramatic increase in the number of victims due to the increased number of cyberspace users. In 2015, Panda Security Company announced that 230,000 new malware attacks produced in a daily base [1, 2]. Thus, mitigating its impact has raised the demand to find a new approach for real-time detection and identification of malware attacks. Researchers who dealt with malware detection used various machine learning algorithms for classification; some achieved better results than others. On the other hand, the literature focused on malicious software for computer operating systems, while others focused on IoT malware. The LightGBM algorithm used in two types of studies, one type was not related to malware, such as Fonseca et al. [12] which dealt with the classification of the acoustic scene classification. Though it achieved significant results. The other type is closed to our study, such as Su et al. [13] who used LightGBM based on image recognition to classify IoT botnets. However, the results of the study were not good enough compared to other algorithms.

This research used machine learning to solve malware detection problem by applying malware classification using LightGBM on IoT botnet. However, in this research, Machine Learning (ML) techniques and its applications to manage malware attacks are exploited based on LightGBM, which is one of the most influential and high-performance machine learning algorithms recently developed by Microsoft [3] on IoT botnet. A pre-defined dataset [4] related to IoT heterogeneous devices connected to a network used for evaluating our proposed approach. The target dataset includes (115) features obtained from different IoT devices. Considering benign traffic and botnet traffic collected from Distributed Denial of Service (DDoS) attack initiated using two families of IoT malware (Mirai, and Gafgyt) [3–5].

- Our solution classifies the IoT botnets in both network and host layers, which provide more accurate results that maintained the same precision for classifying both IoT botnets tested in the experiment.
- Our solution can detect IoT botnet in the early phase of attack without losing its efficiency, regardless of the various devices and their operating systems.

This paper organized into six sections. The next two sections discusses some of the most recent and related works to malware classification. Precisely, we discuss the used methods and algorithms for malware detection and classification, besides the achieved results for each literature. The review includes studies for computers malware classification but mainly focused on IoT botnets. The fourth section discusses the model design and the methodology for our experiment. Then, it describes experiment settings regarding both data, classifier, and experiment phases.

The fifth section discusses the evaluation criteria, then analyze the experiment results and findings to evaluate them properly. At the end of the first part, we compare the output results based on the experiment phases, as well as the devices and botnets to understand

their relation. Then it compares our method with the most related works. Finally, the last section concludes this study, along with our recommendations for future work.

## 2 Botnet Attacks

The botnet term is an abbreviation for “robots network,” a network that connects IoT devices that hosting a (robots) together with victims’ computers. Attackers are taking advantage of IoT vulnerabilities to inject botnet malware into the IoT devices to initiate a wide range of attacks to one or several machines. A botnet can remotely control IoT devices as a group using unauthorized remote access [6, 7]. Thus, IoT devices are suitable to host robots to commit a cybercrime while the criminal is safe and sound somewhere else in the world. Figure 1 demonstrates how the IoT botnet attack works.

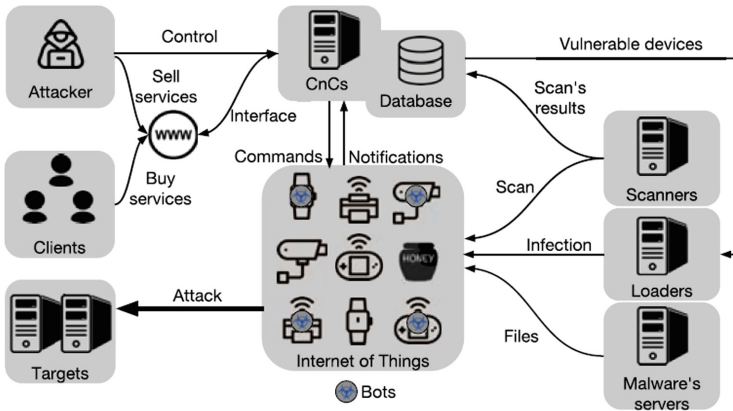


Fig. 1. Overview of an IoT botnet [8].

There is a long list of IoT malware capable of making active or/and passive attacks. The DDoS attack is the most famous attack executed by IoT botnets related to many families besides credential theft, phishing and other types of attacks [9, 10]. The most common families Among IoT botnets are:

- Mirai: This is one of the most used IoT botnets for DDoS attacks; it can infect in one hour about 4000 IoT devices [4, 9, 11].
- Bashlite: which also called Gafgyt, Torlus, or Liz kebab, targets IoT devices that have firmware based on Linux OS [4, 9, 11].

## 3 Related Works

Gandotra et al. [6] presented an extensive literature review on the related works (before 2013). They concluded that statistical analysis alone was not sufficient because it would not instantly detect the zero-day malware. Dynamic analysis is flawed as well because it

requires and consumes time and resources. Their recommendation is to adopt a hybrid technique combining static and dynamic analysis. Su et al. [3] claimed that their study was the first classification that tested on real IoT botnet samples. Also, they claimed that the introduced classification system could be easily deployed to any real IoT device. Based on the detection schema for the proposed light-weight solution they argued that the lightweight classification system using CNN does not need training data based on compared to all other studies that used SVM, or  $k$ -nearest neighbors. Based on Neural Network (NN), they implemented their experiment for image recognition. The dataset has the most samples from the following three families (Mirai, Gafgyt, and Linux.Fgt) the rest of the samples from other different families. However, the best result they could accomplish using CNN is 94% accuracy with 94.67% True Positive Rate (TPR) and 93.33% True Negative Rate (TNR).

Meidan et al. [4] proposed a network-based technique using deep learning to perform anomaly detection. They extracted behavior static features using IoT benign traffic from devices infected with real botnets (Mirai, and Bashlite which known as well as Gafgyt) and used it to train Deep AutoEncoder (DAE), which is a deep learning neural network architecture. The results of their experiment were promising. Although, the training time for benign traffic of each device was relatively high, for example, training on 19,528 benign instances with 172kB size took 190 s. Moreover, their research was limited to the node layer. The experiment tested each device in separate of other devices excluding the network layer, while including network layer could introduce a wider solution for malware detection.

Costin et al. [12] introduced an open-source framework for the analysis of IoT malware. They claimed that their work would fill the gap between studies in the IoT malware field and their framework would help researchers in the future to better understand IoT malware and better defense them. The study estimated 90 days of the advantage of automated IoT malware detection before samples analyzed for its signature. They found that almost 60% of IoT malware families had two instances, which could be an interesting finding that explains a lack of accuracy in static analysis and can be an essential error factor. In the other hand, Although the paper published in Aug 2018, it was not published in Research Gate and has zero citations in google scholar to the time when this thesis wrote. They recommended the researchers to improve cyber-security by improving classification performance and quality. Alejandro et al. [10] emphasized the importance of detecting botnet during the early phase of its life cycle. In their study, they focused on the detection process during the C&C phase according to Leonard et al. [13]. Authors proposed Genetic Algorithm (GA) integrated with the C4.5 algorithm for classification and evaluating the generated features. For the experiment, the authors developed a Java program to implement GA and applied based on two datasets that represent centralized botnets in one dataset and decentralized botnets in the other one. The best result they could achieve was 99.58% after ten iterations. They recommended testing the proposed solution on massive datasets.

Fonseca et al. [14] combined LightGBM and Convolutional Neural Network (CNN) for acoustic scene classification. Although, this study subject is not related to cyber-security, it worth to review the classification algorithm used. The study experiment proved that LightGBM algorithm is more accurate and performs classification training

faster than its predecessor eXtreme Gradient Boosting (XGBoost). Furthermore, the experiment results were better than the previous related study which used the same dataset. The experiment results divided into two stages. In the first stage, LightGBM achieved 80.8% accuracy improving the Multi-Layer Perceptron (MLP) of the previous study by 6% using the same dataset. While in the second stage, LightGBM combined with CNN to achieve 83% of accuracy, which means 8.2% improvement on the MLP of the previous study. Islam et al. [15] investigated the efficiency of classification using data mining and machine learning. They argued that after making the classification public, the attacker would obfuscate the extracted features for that classification, which substantially reduce its accuracy. They added that the classification based on a given set of malware would fail or at least will not perform well with zero-day malware. They extracted static features required for their experiment, then used extracted features with their classification system. The authors concluded that ML using Random Forests is the best classifier for their dataset. The accuracy of the experiment was 97% with the conclusion that the age of malware impacts the results.

Meng et al. [16] extracted features from behavioral analysis using API calls called Static malware Gene Sequences (SGS). They defined software genes as a fragment of the code extracted from programs that have functional information. In their experiment, they used a recursive descent algorithm using Interactive Disassembler Python (IDA Python) to extract the genes arranged in a two-dimensional matrix. They proposed a neural network module called “Static Malware Gene Sequences-Convolution Neural Network” (SMGS\_CNN) for classification. In the experiment, they applied CNN on a dataset chosen from “VX-Heavens” web site [17]. The accuracy increased slightly in each iteration to achieve 98% accuracy after more than 3500 iterations. They concluded that using CNN is better than traditional SVM, which achieved 94.7% accuracy using the same dataset. Alejandro et al. [10] emphasized the importance of detecting the botnet during the early phase of its life cycle. In their study, they focused on detection process during the C&C phase, according to Leonard et al. [13]. Authors proposed Genetic Algorithm (GA) integrated with the C4.5 algorithm for classification and evaluating the generated features. For the experiment, the authors developed a Java program to implement GA and applied based on two datasets that represent centralized botnets in one dataset and decentralized botnets in the other one. The best result they could achieve was 99.58% after ten iterations. They recommended testing the proposed solution on massive datasets.

## 4 Design and Methodology

### 4.1 LightGBM Classifier

Gradient boosting is one of machine learning algorithms used for classification and regression. It combines models from different algorithms to produce new iterative one. Gradient boosting is one of the most widely used machine learning algorithms due to its accuracy and efficiency [18]. It started with the Adaptive Boosting (AdaBoost) then developed into many algorithms and techniques such as GBM and Model-Based Boosting (MBoost), then to CatBoost, XGBoost and LightGBM [19]. IoT networks made of heterogeneous devices that have limited resources. Such that the Lightweight

classification algorithm is the best choice for any good performance security system for IoT networks. LightGBM machine learning algorithm inspired by Su et al. [3] and Fonseca et al. [14] who used it in similar studies.

LightGBM is a new gradient boosting framework based on decision tree algorithms, which introduced by Microsoft. It supports many algorithms like GBM, GBDT, Gradient Boosted Regression Tree (GBRT), and Multiple Additive Regression Tree (MART); as a result, it is scalable, accurate, and efficient [20]. The decision tree, in this algorithm, grows in leaf-wise [21], which optimize the loss which generates branches, Hence, this algorithm faster and less complexity than level-wise growth which extend the tree depth [18]. According to Meidan et al. [4], the time complexity for the lightGBM calculated as  $O(\#Data \times \#Features)$ .

XGBoost is a simple solution that uses presort based algorithm for decision tree learning. Although it is not smooth and a little bit complex to optimize. However, LightGBM increases training performance and reduce the usage of memory by using algorithms based on the histogram.

## 4.2 Datasets

IoT devices use the same internet protocols, which is the main common thing that can describe IoT devices' similarities. Traffic analysis is the best choice in the IoT network to detect and classify cyber-attacks. In any experiment, to get accurate results, accurate data must be provided as the experiment input. Hence, a dataset collected from real IoT devices' traffic is better to develop an applicable and reliable system in the real world IoT devices. Most previous experiments used datasets collected using a sandbox, which is not accurate as it would be in the case of a real-world environment. In this approach, we adopted real data introduced in [4]. In their study; they set up their lab using real IoT devices for DDoS attacks initiated with two botnet families and nine IoT devices [4]. The files in a ".csv" format for selected datasets, uploaded by the authors to the repository system in the University of California, Irvine. The datasets are freely accessed online source, and consist of (115) features described in [4].

## 4.3 Proposed Solution

The proposed approach aims to develop a real-time system for detecting and classifying IoT botnets based on LightGBM ML techniques. The approach architecture and the model of the experiment illustrated in Fig. 2. First of all, we carefully selected the dataset which collected from a real IoT network, and then we implemented machine learning classifier to classify botnets attacks. Later in this chapter, architecture and experiment settings will clarify our approach.

## 4.4 Architecture

Our approach implemented as a distributed architecture that covers multi-layers, as shown in Fig. 3. Our approach based on implementing the selected algorithm for training and classification on the host and the network layers, such that, each layer has two stages.

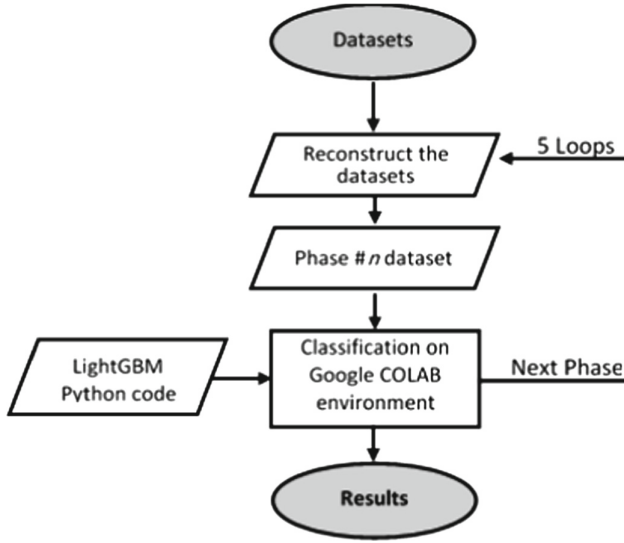


Fig. 2. Approach Architecture.

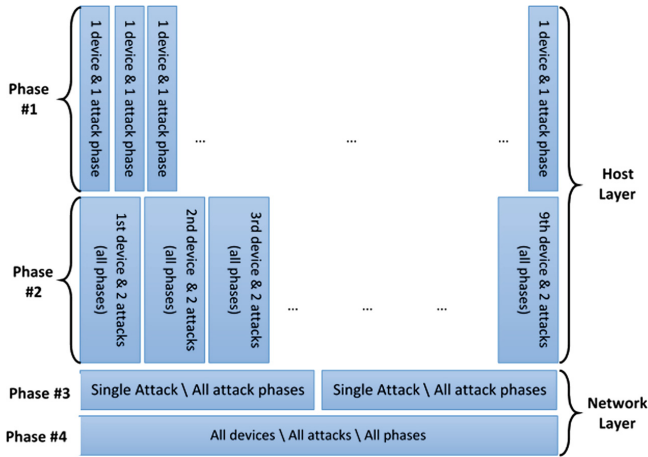


Fig. 3. Experiment's architecture.

The experiment is carried out in four stages; while results organized into five categories. The following list connects each results category to the related phase of the experiment:

1. **Phases of attack\device:** *The most detailed results for the 1<sup>st</sup> phase.*
2. **Attack phases:** *It has driven out of the first category to evaluate the detection during an early phase of the attack.*
3. **Host-related results:** *for the host layer in the 2<sup>nd</sup> phase.*

4. **Malware:** *The results from the 3<sup>rd</sup> phase were based on a whole single IoT botnet repeated for the two tested IoT botnets.*
5. **Network layer:** *The results form the 4<sup>th</sup> phase of the experiment.*

## 4.5 Experimental Settings

First of all, we reconstructed the selected dataset to align with the designed architecture. Our experiment implemented using Google COLAB ordered as the following:

- **Phase #1: Phase of attack\device:** We separately performed the training and classification for each IoT device's benign traffic combined by one botnet attack (Mirai and Gafgyt) separately. Botnet attack selected from one attack phase on the same IoT device. Similarly, the training repeated for each phase for each botnet per device using 80 datasets, contains 100% of original datasets.
- **Phase #2: Host layer:** The same procedures in the 1<sup>st</sup> phase applied; however, we involved both botnet attacks (Mirai and Gafgyt). Botnet attack selected from all attack phases on the same IoT device. Similarly, the training repeated for all botnets attacks for each device using nine datasets, 50% of original datasets.
- **Phase #3: Network layer per malware:** Again, we separately performed the training and classification for benign traffic form all IoT devices combined by single botnet attack (Mirai and Gafgyt). Botnet attack selected from all attack phases on all IoT devices. The training repeated twice for attacks for all devices using two datasets contains 25% of original datasets.
- **Phase #4: Network layer:** The same procedures in phase #3 applied; however, we involved both botnet attacks (Mirai and Gafgyt). The botnet attack selected from all attack phases on all IoT devices and contains 12.5% of original datasets. In the first stage of this phase, we implemented the classifier on the binary-class dataset using  $k$ -fold cross-validation. While in the second stage, we used the Holdout method to handle multi-class dataset in the second stage.

Furthermore, we developed two codes for LightGBM algorithm using Python. One code used Holdout method to handle multi-class dataset in the second stage of the last phase. The other code designed to use  $k$ -fold, where  $k = 10$ , to handle all other datasets used for training in all phases.

## 5 Findings and Discussions

### 5.1 Evaluation Criteria

$k$ -Fold cross-validation is a statistical procedure to evaluate the skill of supervised machine learning models. Cross-validation delivers predictions accuracy and avoiding the overfitting where the model repeats the labels to get a perfect score but fails to get predictions [22]. The  $k$ -fold in cross-validation splits the datasets to  $k$  equal sets, then uses  $(k - 1)$  sets for training. The validation results in each loop come from testings the last set. Eventually, it calculates the average of results collected from all  $k$  loops (folds).



To evaluate the experiment results, we considered a set of classification metrics such as (Accuracy, Area Under Curve (Auc), True Positive Rate (TPR), False Positive Rate (FPR), Mean Squared Error (MSE), Matthews Correlation Coefficient (MCC), Logarithmic Loss, and F1 Score). Also, to compare the results with related works, we used precision and recall as well.

## 5.2 Results Evaluation

Recalling the experiment setting, the proposed approach implemented in different four layers. The 1<sup>st</sup> phase used datasets that represent each device vs one phase of a botnet attack. The metrics were very much similar for all devices, where the accuracy percentage is ranging between (0.9999) and (1.0000), with standard deviation range between ( $\pm 0.0000$ ) to ( $\pm 0.0004$ ).

Then, detailed results reanalyzed to conclude the results for each phase for both botnet attacks, which summarized in Table 1. The results in Table 1 are the averages of each metric for each phase.

**Table 1.** Metrics for each attack phase.

Attack phase	Acc.	TPR	FPR	MSE	MCC	Log. loss	F1 score
Phase #1	99.99%	100%	0.01%	0.28%	99.99%	0.18%	99.99%
Phase #2	100%	100%	0.01%	0.24%	99.99%	0.15%	100%
Phase #3	100%	99.99%	0.00%	0.04%	99.99%	0.13%	99.99%
Phase #4	100%	99.99%	0.00%	0.06%	99.99%	0.14%	99.99%
Phase #5	100%	100%	0.00%	0.24%	99.99%	0.11%	100%

In the second phase which represents the host layer, which used nine balanced datasets that represent the nine devices for all phases of both botnet attacks on the same device. All metrics measurements in this layer are better than the previous one. Their averages are ranging between (0.9997) and (1.0000), with standard deviation range between ( $\pm 0.0003$ ) and ( $\pm 0.0006$ ). Malware phase uses two balanced datasets, one dataset for each botnet. The accuracy for both botnets is 100%, and the standard deviation for both is 0%. Similarly, all other metrics indicate that the detection precision for both is the same. The last phase (the network layer), used only one dataset for all phases of both attacks on all devices, along with an even combination of the benign traffic from all devices. Table 2 presents the metrics measurements in this phase.

## 5.3 Comparison of the Experiment Results

### 1) Attack phases

The comparison result between each attack phase shows a slight difference between them.

**Table 2.** Metrics for the network layer.

Metric	Value
Accuracy	100%
AUC	100%
TPR fold	100%
FPR fold	0.00%
MSE	0.00%
MCC fold	100%
Log. loss	0.06%
F1 score	100%

*2) Layers*

In this test, the average values of classification accuracy, AUC and F1 score are considered for each layer to be compared with averages for the host layer. Table 3 emphasizes the superiority of the network layer, which has 100% accuracy compared with the host layer with minor differences. The results are undeniable as values for MSE and Loss dropped for the network layer from 0.0003 to 0.0000 for the MSE, and from 0.0038 to 0.0000 for the Log. Loss.

**Table 3.** Comparison of results for layers

Layer	Acc.	AUC	F1 score	MSE	Log. loss
Host layer	99.99%	99.98%	99.78%	0.03%	0.38%
Network layer	100%	100%	100%	0%	0%

*3) Methods*

Both k-fold cross-validation and Holdout methods used in the last experiment on the network layer. Table 4 shows the results. These results indicate the same accuracy level for both method despite the class. This similarity indicates no differences in classification and detection accuracy as well as predictions of both used botnets (Mirai, and Gafgyt). On the other hand, We found that precision is (100% and 100%) and recall is (100% and 100%), respectively.

**5.4 Comparison with Related Works**

Table 5 compares our method with three similar approaches for IoT malware classification. The results for all other approaches maintain a lower accuracy and TPR than our approach does in the network layer. Moreover, none of the previous approaches achieves similar overall results in such a way that our approach does.

**Table 4.** Comparison of network layer methods

		Mean			Std. dev.		
Method	Class	Accuracy	F1 score	MSE	Accuracy	MSE	F1 score
K-fold	Binary	100%	100%	0%	0%	0%	0%
Holdout	Multi	100%	100%	0%	0%	0%	0%

**Table 5.** Comparison with results from related works.

	Our method	Meidan et al. [4]	Su et al. [3]	Alejandre et al. [10]
Devices	IoT	IoT	IoT	IoT
Algorithm	ML	DL	DL	ML
	LightGBM	DAE	CNN	GA
Accuracy	100%	–	94.00%	–
F1 Score	100%	–	–	–
MSE	0.00%	–	–	–
Precision	100%	–	93.42%	–
Recall	100%	–	94.67%	–
TPR	100%	100%	94.67%	99.46%
FPR	0.00%	0.70%	5.33%	0.57%

## 6 Conclusion and Future Work

In this study, the results prove that the advanced ML algorithms and DL does not necessarily lead to better solutions. In contrast, it may increase the complexity of the problem. For instance, LightGBM algorithm achieved almost 100% accuracy, which proves the efficiency of this ML algorithm over DL strategies. Besides, the improved accuracy using classical ML alternates anti-malware producers to use deep learning. The results demonstrate the ability of our approach to detect botnets attacks with the same high accuracy regardless of its family. Also, our approach provides an instant, accurate and straightforward method to early detect IoT botnets in the network level before infecting any more IoT devices. Network-level means that anti-malware can analyze and detect malware out of sniffed traffic from the network despite traffic source, destination host, device type, device brand, or the device's OS. LightGBM, in particular, achieves promising results in the network layer, which is more accurate than in the host layer context. Therefore, the proposed algorithm won the race to be the light, efficient and precise malware machine learning classifier in both host or network layers. For the best of our knowledge, vendors of anti-malware systems can implement the presented system on all real IoT devices. Utilizing these results in the anti-malware system will make it possible to identify the threat in real-time and terminate it before the damage occurred. Based on the promising results of the proposed IoT botnets classification algorithm, researchers

are recommended to pay more attention to involve computer malware as well. Despite the high-performance levels achieved by the proposed approach, the metrics discussed in the previous chapter did not consider time performance. The proposed system will have more potentials if more studies improve its time performance. Finally, we recommend to perform the same experiment using computer malware dataset and consider improving the time performance.

## References

1. Al-Kasassbeh, M., Mohammed, S., Alauthman, M., Almomani, A.: Feature selection using a machine learning to classify a malware. In: *Handbook of Computer Networks and Cyber Security*, pp. 889–904. Springer, Cham (2020)
2. Al-kasassbeh, M., Almseidin, M., Alrfou, K., Kovacs, S.: Detection of IoT-botnet attacks using fuzzy rule interpolation. *J. Intell. Fuzzy Syst.* **38**(1) (2020)
3. Su, J., Vargas, D.V., Prasad, S., Sgandurra, D., Feng, Y., Sakurai, K.: Lightweight classification of IoT malware based on image recognition. In: *Proceedings - International Computer Software and Applications Conference*, vol. 2, pp. 664–669, 11 February 2018
4. Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y.: N-BaIoT-network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Comput.* **17**(3), 12–22 (2018)
5. Alauthman, M., Aslam, N., Al-kasassbeh, M., Khan, S., Choo, K.-K.R.: An efficient reinforcement learning-based Botnet detection approach. *J. Netw. Comput. Appl.* **150**(15), 102479 (2020)
6. Gandotra, E., Bansal, D., Sofat, S.: Malware analysis and classification: a survey. *J. Inf. Secur.* **5**(2), 9 (2014)
7. Chawathe, S.S.: Monitoring IoT networks for botnet activity. In: *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)* (2018)
8. Marzano, A., Alexander, D., Fonseca, O., Fazzion, E., Hoepers, C., Steding-Jessen, K., Chaves, M.H., Cunha, Í., Guedes, D., Meira, W.: The evolution of Bashlite and Mirai IoT botnets. In: *2018 IEEE Symposium on Computers and Communications (ISCC)*, Natal, Brazil (2018)
9. Angrishi, K.: Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT Botnets, vol. 1, 13 February 2017
10. Alejandre, F.V., Cortés, N.C., Anaya, E.A.: Feature selection to detect botnets using machine learning algorithms. In: *International Conference on Electronics, Communications and Computers*, Cholula, Mexico (2017)
11. De Donno, M., Dragoni, N., Giaretta, A., Spognardi, A.: DDoS-capable IoT malwares: comparative analysis and Mirai investigation. *Security and Communication Networks* **2018**, 30 (2018)
12. Costin, A., Zaddach, J.: IoT malware: comprehensive survey, analysis framework and case studies. In: *Black Hat Conference*, Las Vegas (2018)
13. Leonard, J., Xu, S., Sandhu, R.: A framework for understanding botnets. In: *International Conference on Availability, Reliability and Security*, Fukuoka, Japan (2009)
14. Fonseca, E., Bogdanov, D., Gong, R., Gomez, E., Slizovskaia, O., Serra, X.: Acoustic scene classification by ensembling gradient boosting machine and convolutional neural networks. In: *Workshop on Detection and Classification of Acoustic Scenes and Events*, Munich, Germany (2017)
15. Islam, R., Tian, R., Batten, L.M., Versteeg, S.: Classification of malware based on integrated static and dynamic features. *J. Netw. Comput. Appl.* **36**(2), 646–656 (2013)

16. Meng, X., Shan, Z., Liu, F., Zhao, B., Han, J., Wang, H., Wang, J.: MCSMGS: malware classification model based on deep learning. In: 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Nanjing, China (2018)
17. VX Heaven: vxheaven.org (2016). <http://83.133.184.251/virensimulation.org/>
18. Read the Docs, Inc. & contributors: LightGBM's documentation! Read the Docs, Inc. & contributors, 7 February 2019. <https://media.readthedocs.org/pdf/lightgbm/latest/lightgbm.pdf>
19. Khandelwal, P.: Which algorithm takes the crown: light GBM vs XGBOOST? Analytics Vidhya, 12 June 2017. <https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/>. Accessed 2018
20. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
21. xgboost: Introduction to Boosted Trees. xgboost.readthedocs.io. <https://xgboost.readthedocs.io/en/latest/tutorials/model.html>
22. Scikit-learn: Machine Learning in Python, INRIA and others. [https://scikit-learn.org/stable/modules/cross\\_validation.html#cross-validation](https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation)



# Exploiting Linearity in White-Box AES with Differential Computation Analysis

Jakub Klemsa<sup>(✉)</sup> and Martin Novotný

Czech Technical University in Prague, Prague, Czech Republic  
jakub.klemsa@fel.cvut.cz, novotnym@fit.cvut.cz

**Abstract.** Not only have all current scientific white-box AES schemes been mathematically broken, they also face a family of attacks derived from traditional Side Channel Attacks, e.g., Differential Computation Analysis (DCA) introduced by Bos et al. Such attacks are very universal and easy-to-mount – they require neither knowledge of the implementation, nor use of reverse engineering. In this paper, we particularly focus on DCA against white-box AES by Chow et al. which shows lower than 100% success rate as opposed to other schemes studied by Bos et al. We provide an explanation of this phenomenon while unraveling another weakness in the design of white-box AES by Chow et al. Based on our theoretical results, we propose an extension of the original DCA attack which has a higher chance of key recovery and might be adapted for other schemes.

**Keywords:** White-box AES · Differential Computation Analysis · Linear cryptanalysis

## 1 Introduction

Standard ciphers like AES (*Advanced Encryption Standard*, [30]) were designed with respect to so-called *black-box model*. In this model, an adversary is only allowed to observe ciphertexts of chosen plaintexts while she does not gain *any* other information about the encryption algorithm execution – neither intermediate values, nor timing. I.e., the adversary has an access to an *encryption oracle* while her goal is to recover the key or employ the oracle for effective decryption.

However, real-world hardware implementations like smart cards *do* leak certain portion of internal information through various side-channels, e.g., power consumption or electromagnetic radiation. This attack scenario is referred to as the *gray-box model*.

Later, there has emerged a need for the most extreme scenario where the adversary has a full control over the execution environment. Such a model is called the *white-box model*. Note that in this model, the adversary is free to

---

This work was supported by the Grant Agency of CTU in Prague, grant No. SGS19/109/OHK3/2T/13.

observe or alter all intermediate values as well as instructions. It follows that the original cipher’s intermediates—which typically allow for key recovery—must be somehow hidden or masked. In the wild, several techniques and layers of protection are being put in place, ranging from software obfuscation to mathematical approaches. In our paper, we will particularly focus on the mathematical point of view, however, our results will turn out to be highly practical.

## 1.1 White-Box Cryptography

In 2002, Chow et al. proposed white-box implementations of AES and DES [11, 12] (WBAES, WBDES). These implementations aim at protecting the keying material from an adversary who is in possession of the implementation which includes the (masked) key. Even though many years have passed, all scientific white-box AES schemes got eventually broken (to the best of our knowledge), especially since the usage of side-channel attack techniques like Differential Computation Analysis (DCA) [9], Differential Fault Analysis (DFA) [14, 17] and/or their recent enhanced variants [2, 5, 7, 31].

However, the business need is stronger, hence this field is still very active, despite relying on software obfuscation techniques and secret design, i.e., violating the Kerckhoffs’ principle [18]. Applications of white-box cryptography include—but not limited to—*Digital Rights Management* (DRM) for protected content distribution, *Host Card Emulation* (HCE) on mobile devices for mobile payments, or memory-leakage resilient software; see Bogdanov et al. [6] for a detailed description of each. For an extensive literature research regarding white-box cryptography, we recommend a recent work by Goubin et al. [16].

## 1.2 Our Contributions

In this paper, we point out the atypically low success rate of the DCA attack against Chow’s WBAES presented by Bos et al. [9]. For this phenomenon, we propose a theoretical explanation which identifies a vulnerability of Chow’s WBAES to the DCA attack. Based on our results, we further generalize and extend the set of targets that were employed by Bos et al. in their original attack. We also motivate to use our novel targets for a DCA attack against other implementations that use (semi-)linear masking of intermediates.

In the experimental part, we provide a description of our attack toolkit and employed algorithms, and we provide detailed numerical results including timing. Notably, we confirm the vulnerability that was identified during the theoretical analysis. Finally, we study the behavior of false positives in case of a blind attack and derive an optimal number of traces in terms of computational effort.

## 1.3 Paper Organization

The paper is organized as follows: In Sect. 2, we give a brief description of Chow’s WBAES, we provide a short introduction to side-channel attacks and highlight

the usage of DCA in the white-box attack context. We analyze the DCA attack against Chow’s WBAES in Sect. 3. In Sect. 4, we describe the practical attack in detail, we support our explanation by an experiment and propose a methodology for practical usage based on a comprehensive testing set. We conclude our work in Sect. 5.

## 2 Preliminaries

### 2.1 Construction of White-Box AES by Chow et al.

One of now classical mathematical approaches how to hide an AES key—in fact all intermediate values as well—in a software implementation is to turn all AES operations into somehow masked lookup tables. Such an approach was introduced in 2002 in a seminal paper by Chow et al. [12]. In their construction, there are four types of lookup tables while the intermediate values are masked using both linear and non-linear random bijections. However, this particular design was mathematically broken two years later by Billet et al. [4].

In the following, we give a high-level description of tables Type II of Chow’s WBAES because this is where the attack of our interest will show to be operating. Note that we will be using plain AES without input and output encodings which is technically just an obfuscation layer—we need a plain AES encryption oracle anyways. For further details, we refer to Muir’s tutorial [28] which we highly recommend over the original paper by Chow et al.

Lookup tables Type II combine several AES operations together with both linear and non-linear masking, see (1). Description of each operation follows.

$$\text{plaintext} \rightarrow \text{AddKey} \rightarrow \text{SBox} \rightarrow \text{MB} \circ \text{MC} \rightarrow \text{Enc}^{-1} \rightarrow 1^{\text{st}} \text{ intermediate} \rightarrow \dots$$

in table Type II

(1)

**plaintext:** The table inputs 1 byte of an AES plaintext block, i.e., the table contains 256 entries.

**AddKey:** This operation XORs respective byte of the (unknown) AES key with the plaintext byte.

**SBox:** This operation is a standard AES SBox, i.e., a 1-byte non-linear bijection.

**MB  $\circ$  MC:** This operation is a composition of two 4-byte linear bijections: MC, which stands for standard AES MixColumns, and MB, which is a random linear bijection (hence unknown). Their 4-byte input is split into four 1-byte values, which are handled in separate tables and XORed together in subsequent tables using linearity. Hence this operation inputs 1 byte and outputs 4 bytes (32 bits).

**Enc<sup>-1</sup>:** This operation is a random 4-bit non-linear bijection, which is applied to each of the eight 4-bit nibbles of the 32-bit input value. Note that it is re-randomized for each nibble and each table while its correct counterpart Enc must be applied at the input to the subsequent lookup table.



**1<sup>st</sup> intermediate:** The output value. It can be found by the adversary in the lookup table.

*Note 1.* Enc bijection is only 4 bits wide, because two such 4-bit nibbles are later XORed together, hence making the input for the subsequent table 8 bits wide. If Enc were 8 bits wide, the subsequent table would need to input 16-bit values, which would make the table very large, however, this approach is used in some white-box implementations.

## 2.2 Side-Channel Attacks in White-Box Cryptography

Let us briefly recall the principle of side-channel attack and particularly one of its variants upon which Bos et al. [9] built their attack in the white-box context.

*Side-Channel Attack* (SCA) is a large family of attacks that exploit any weakness of a real-world implementation of a cryptographic algorithm to recover the key (i.e., SCA assumes the gray-box context). SCA was pioneered by Kocher in 1996 in [24] where he focuses on public key cryptography. However, the general idea can be ported to symmetric cryptography as well.

On the one hand, SCA may exploit passively observable measures coming from different sources of information leakage, e.g., power consumption [25], electromagnetic radiation [15], or timing [24]. On the other hand, there exist also active attacks that attempt to alter the computation data or flow and observe corrupted results. The phenomenon of faults in cryptographic algorithms was first addressed by Boneh et al. [8]. For a comprehensive reading we refer to Koç [23, Chapters 13–18].

**Differential Power Analysis.** There are several types of passive SCA's against AES depending on type of the leakage, among them, we will particularly focus on a specific case of *Differential Power Analysis* (DPA). Let us consider that we can measure voltage on a system bus where we expect to capture transfers of AES intermediates. Such records will be referred to as the *traces*. Given a set of plaintexts and respective traces, there exists a moment in time  $t_0$  when certain intermediate value is being transferred over the bus. The goal is to guess a small portion of the key and precompute the expected intermediate value. If the guess is correct, we will find a big correlation between the precomputed intermediates and values across the traces at  $t_0$ . Otherwise, no significant correlation shall occur at any position within the traces.

Specifically, we will consider individual bits of the first SBox output as the intermediates, i.e.,  $t = \text{SBox}(PT[i] \oplus k)[b]$  for  $i$ -th byte of a plaintext  $PT$ , a key guess  $k$  and  $b$ -th bit of the SBox output. Such values will be referred to as the *targets* or *hypotheses*, i.e., values that we expect to occur across the traces.

The attack proceeds as follows: we loop all 16 key byte positions  $i$ , all 256 guesses on  $i$ -th key byte  $k$  and all 8 target bit positions  $b$ . For each trace, indexed by  $j$ , we compute the expected target value as

$$t_j = \text{SBox}(PT_j[i] \oplus k)[b]. \quad (2)$$

Based on the value of  $t_j \in \{0, 1\}$ , we split the traces into two sets  $S_0$  and  $S_1$ , respectively. Note that for the correct key guess, the traces in  $S_0$  are expected to have a low value at  $t_0$  and a high value in  $S_1$ , respectively. Therefore we compute absolute difference of means of the two sets  $D = |\bar{S}_1 - \bar{S}_0|$  where  $\bar{S}$  denotes a mean trace, i.e.,  $\bar{S} = \frac{1}{|S|} \sum_{t \in S} t$  using point-wise trace addition. Then, for the correct key guess, there shall emerge a clear peak at time  $t_0$ , otherwise the differences of means shall be small and blurry. For each key candidate, we refer to the magnitude of the peak as the *rank* of the candidate. A pseudocode for a derived attack will be given later.

*Note 2.* In case of a noisy measurement, the peak might be unclear. For this reason, we rank the candidates – the higher rank, the more likely the guess is correct. This might be later used also for brute force key recovery if the initial key guess is incorrect – first we search the candidate bytes with lowest rank.

### 2.3 Adaptation of SCA to White-Box Attack Context

As introduced by Bos et al. [9], the powerful tools of SCA can be advantageously used for an attack against white-box implementations of cryptographic algorithms. Regarding white-box challenge implementations—de facto encryption oracles—the main benefit of such attacks is that they do not need knowledge of particular implementation, often neither use of reverse engineering, which makes them very universal and easy-to-mount. In this paper, we focus on passive attacks, however, active attacks like *Differential Fault Analysis* (DFA, first introduced against DES [3], later also against obfuscated implementations [17] and in particular against AES [14]) might be adapted as well while making it probably the most powerful attack in the white-box attack context.

In their paper, Bos et al. adapted DPA (as introduced in Sect. 2.2) for an attack against several white-box implementations; they call this adaptation the *Differential Computation Analysis* (DCA). Instead of physical measurements, they employed instrumentation tools like Valgrind [29] or PIN [26] to capture program-memory interactions, i.e., addresses and/or contents of memory reads and/or writes, referred to as the *memory traces* or *memtraces*.

For all challenges but one attacked by Bos et al., the results showed 100% success rate while using only a couple of memtraces. Neither of these challenges used any form of mathematical obfuscation of intermediates, i.e., the intermediates were directly observable in the memtraces; these challenges relied solely on software obfuscation techniques.

In one particular challenge, Klinec [22] implemented Chow’s WBAES, hence the AES intermediates were not directly observable in the memtraces. However, even this implementation got, maybe surprisingly, broken. For their attack, Bos et al. used an augmented set of targets:

$$T_1 = \text{SBox}(PT[i] \oplus k), \quad (3)$$

i.e., the output of the first SBox—the classical targets, cf. (2)—and

$$T_2 = (PT[i] \oplus k)' \quad (4)$$

where  $(\cdot)'$  stands for *Rijndael inverse* – a multiplicative inverse in *Rijndael field*  $\text{GF}(2^8)$  modulo  $x^8 + x^4 + x^3 + x + 1$ . The idea behind was motivated by the construction of the original AES SBox:

$$\text{SBox}(X) = A(X') + B, \quad (5)$$

where  $A$  is a linear mapping and  $B$  a constant byte.

*Note 3.*  $T_1$  targets are affine mappings of  $T_2$  targets and vice versa, cf. (3), (4) and (5).

*Note 4.* In the rest of this paper, we will neglect constant bits, i.e., all affine mappings will be considered as linear. Indeed, flipping the target bit only swaps the sets  $S_0$  and  $S_1$  as defined for SCA, hence has no effect on the final result – we are only interested in absolute difference of their means.

Bos et al. employed 500 memtraces with  $T_1$  targets and 2000 memtraces with  $T_2$  targets. In both cases, they achieved similar success rate – a key byte leaked in about 30% of cases.

### 3 Analysis of DCA Against Chow’s White-Box AES

First of all, recall that all of the target bits in  $T_1$  and  $T_2$  can be obtained by a linear mapping of the first SBox output, cf. Note 3, and let us refresh the operations within the first lookup table:

$$\text{plaintext} \rightarrow \underbrace{\text{AddKey} \rightarrow \text{SBox} \rightarrow \text{MB} \circ \text{MC}}_{\text{in table Type II}} \rightarrow \text{Enc}^{-1} \rightarrow 1^{\text{st}} \text{ intermediate} \rightarrow \dots$$

Let us assume that we can get the intermediate value before the final  $\text{Enc}^{-1}$ , i.e., right after  $\text{MB} \circ \text{MC}$ . Such a value consists of 32 bits while each bit  $t'$  can be computed as a linear mapping of the first SBox output, i.e.,  $t' = R^T \cdot \text{SBox}(PT[i] \oplus K[i])$  for some vector  $R$ . Since  $\text{MB}$  is a random linear bijection, then  $R$  is a random-like non-zero vector. Therefore, in some cases,  $R$  might happen to be a standard basis vector, e.g.,  $(0, 1, 0, \dots, 0)$ , or it might be equal to a row of  $A^{-1}$ , cf. (5). Note that in such cases, a target from  $T_1$  or  $T_2$ , respectively, would perfectly fit  $t'$ . However, there are another  $255 - 16 = 239$  cases which are not covered by  $T_{1,2}$  – let us define a complete set of such targets.

**Definition 1.** Let  $P$  and  $K$  be a plaintext and key byte, respectively. We define the set of all linear AES-DCA targets as

$$T_{lin} = \{R^T \cdot \text{SBox}(P \oplus K) \mid R \in \text{GF}(2)^8 \setminus (0, 0, \dots, 0)\}. \quad (6)$$

It follows that the set of targets  $T_{lin}$  fully covers the intermediates before the final  $\text{Enc}^{-1}$ , however, in the real implementation,  $\text{Enc}^{-1}$  is employed as well. It follows that  $\text{Enc}$  actually poses the only protection against our linear AES-DCA

targets. As per the results of Bos et al., some targets leak the key byte anyways, hence let us focus on the (non-)linearity properties of Enc.

In the design of Chow’s WBAES, Enc is defined to be a random 4-bit bijection while posing the only non-linear (confusion) element. They provide the following argumentation: “Ideally for security, we would explicitly avoid linear transformations. But randomly choosing bijections, essentially all will be non-linear: . . . less than 0.000 002% are affine.” Hence they do not encourage for any non-linearity check although it is a widely studied topic for regular ciphers by methods of *linear cryptanalysis*, introduced by Matsui et al. [27].

On the one hand, the ratio of fully linear mappings is indeed extremely low, on the other hand, DCA can exploit the intermediates even when there occurs only one bit in Enc output that is linear in its input. Note that there are lot more such 4-bit bijections – indeed, there are  $2 \cdot 4 \cdot (2^4 - 1) \cdot 8! \cdot 8!$  of them among  $16!$  bijections which is almost 1%. Since there are several Enc instantiations for each key byte, 1% chance is very much non-negligible. Furthermore, DCA is based on a physical SCA, i.e., it is designed to handle errors, cf. Note 2. For this reason, even such Enc bijections that are linear in single output bit on majority of inputs pose a weakness. There are obviously much more than 1% of such bijections making the protection vulnerable to DCA with our  $T_{lin}$  targets.

Since our linear AES-DCA targets address *all* linear transformations of the first-round AES intermediates, they can be advantageously applied to other schemes that employ linear protection (or semilinear, like Chow’s WBAES). Note that a random linear bijection is a handy masking technique since it can easily combine several bytes together – thanks to its linearity. See, e.g., a recent report by Goubin et al. [16] recovering the hardest challenge submitted to WhibOx 2017 Contest Workshop [13] – the Adoring Poitras challenge<sup>1</sup>. In their work, they introduce *linear decoding analysis* which also correctly assumes linear encoding of intermediates.

## 4 Practical Attack and Results

First, we describe the DCA bitwise attack in detail and provide an overview of the whole attacking procedure. Next, as the main goal of our experiments, we confirm our hypothesis about leakage as introduced in Sect. 3, i.e., leakage from the first set of tables Type II. Then we focus on a scenario with unknown key while inspecting properties of false positives. Finally, we suggest a methodology to estimate an optimal number of traces for this type of attack and evaluate the optimum for Chow’s WBAES. Note that we performed all experiments on a single core of Intel Core i5-7600K processor @ 4.1 GHz, i.e., all execution times are with respect to this hardware.

---

<sup>1</sup> Available at <https://whibox-contest.github.io/show/candidate/777>. Accessed: August, 2019.

## 4.1 Bitwise DCA

The most demanding part of our attack is the bitwise DCA/DPA attack as outlined in Sect. 2.2 – lots of memtrace-, i.e., vector-, additions are performed. We implemented that part of the attack in C++ [19]. We summarize this attack in Algorithm 1 where  $P$ ,  $Trc$ ,  $trg$  and  $B$  denote the arrays of plaintexts, respective traces represented in bits, target tables as per Definition 1, and attacked byte number (i.e.,  $1 \dots 16$ ), respectively. The output array  $dif$  is a list of key candidates sorted by their rank, for each target bit.

---

### Algorithm 1. Bitwise DCA/DPA attack.

---

```

1: function BITWISEDCA( $P$ ,  $Trc$ ,  $trg$ ,  $B$ )
2:   // a 256-tuple of 8-tuples of triples: key guess, rank and leakage position
3:    $dif = ((0x00, 0.0, 0), \dots, (0x00, 0.0, 0)), \dots, ((0xff, 0.0, 0), \dots, (0xff, 0.0, 0))$ 
4:   for  $kg = 0x00 \dots 0xff$  do // key guess
5:      $absdif = ((0.0, \dots, 0.0), \dots)$  // an 8-tuple of vectors of trace bit-size
6:      $mean_{0,1} = ((0.0, \dots, 0.0), \dots)$  // both an 8-tuple of vectors of trace bit-size
7:      $num_{0,1} = (0, 0, 0, 0, 0, 0, 0, 0)$  // both an 8-tuple
8:     for  $i = 1 \dots |P|$  do
9:        $p = P[i]$ ,  $trc = Trc[i]$ 
10:       $hyp = trg[p[B] \oplus kg]$  // hypothesis, i.e., “SBox” output, cf. (2)
11:      for  $b = 1 \dots 8$  do // target bit
12:        if  $hyp[b] == 0$  then
13:           $mean_0[b] += trc$  // most demanding
14:           $num_0[b] += 1$ 
15:        else
16:           $mean_1[b] += trc$  // most demanding
17:           $num_1[b] += 1$ 
18:      for  $b = 1 \dots 8$  do
19:        if  $num_0[b] \neq 0$  then  $mean_0[b] /= num_0[b]$ 
20:        if  $num_1[b] \neq 0$  then  $mean_1[b] /= num_1[b]$ 
21:         $absdif[b] = |mean_1[b] - mean_0[b]|$ 
22:        // maximal absolute difference and its position is found & saved
23:         $dif[kg][b] = (kg, \max(absdif[b]), \arg \max(absdif[b]))$ 
24:        sort  $dif[\cdot][b]$  by rank
25:   return  $dif$ 

```

---

## 4.2 Steps of the Attack

The practical implementation of our attack consists of several tools and follows the steps described in Algorithm 2.

---

**Algorithm 2.** Steps of the practical attack.

---

- 1: acquire memtraces
  - 2: filter constant values from memtraces
  - 3: generate memtrace preview & identify leakage range
  - 4: **if** found leakage range **then then go to 6**
  - 5: attack some byte (possibly first and/or last) & identify leakage range
  - 6: crop traces to leakage range
  - 7: run full attack, process & display results
- 

All the tools are written in Ruby and published in our `White-Box-DPA-Processing` toolkit [21]. Next we describe each step and/or component of the toolkit.

**Trace Acquisition.** Our acquisition tool generates random AES plaintexts, feeds them to the target WBAES implementation while acquiring the memtrace. For this purpose, we employ Intel PIN [1] with our custom memory tracing tools [20]. There are total four tools that enable to acquire contents or addresses of memory reads or writes, respectively. As a reasonable initial number of traces, for an unprotected implementation, even 25 is sufficient, for an implementation with a semi-linear protection similar to Chow’s WBAES, lower hundreds of traces are needed<sup>2</sup>. Acquisition of 200 traces of Klinec’s implementation took us roughly 4 min. If the number of traces shows to be insufficient during the attack, our acquisition tool enables to acquire additional traces. Note that we acquired contents of memory reads, i.e., we expect that there occur values from those aforementioned white-box lookup tables.

Last but not least, it is highly important to have the traces well aligned, hence it is recommended to turn off Address Space Layout Randomization (ASLR). On Unix-like systems, this can be done by the command

```
$ setarch 'uname -m' -R /bin/bash
```

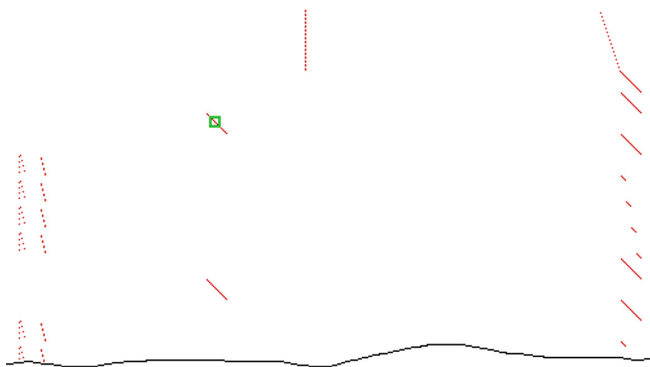
**Filtering Constant Regions.** In the memtraces, there occur several regions which are identical across all traces, hence carry no information for the attack. Our acquisition tool automatically creates a filtering mask based on a couple of traces (by default 30 traces) and filters these regions out. As a result, there remains no constant value across traces while the traces remain aligned. For 200 of Klinec’s traces, this step took us about 15 s.

**Identify the Leakage Range from a Memtrace Preview.** Next, our tool creates a memtrace preview: the  $x$ -axis represents the address space (partial and usually “zoomed out” to fit reasonable dimensions), the  $y$ -axis represents the execution time (top-bottom), memory writes are represented by a red pixel; see Fig. 1 (the green marker will be explained later). If we can clearly recognize

---

<sup>2</sup> Later we will discuss optimal number of traces for this type of WBAES and recommend 200 traces.

where SBoxes of the first AES round take place, we can skip the next step and continue to cropping the traces to the leakage range.



**Fig. 1.** Partial memtrace of memory writes of a naive AES implementation with 7<sup>th</sup> byte leakage position emphasized in green. The memtrace is cropped within the 2<sup>nd</sup> AES round.

**Initial Attack to Identify the Leakage Range.** In case we are not sure where exactly the leakage takes place, we recommend to attack single key byte (the first and the last byte could possibly show the beginning and the end of that range, respectively) and use our marker tool to emphasize the exact place within the memory trace; cf. Fig. 1. Details to the attacking procedure are given in Sect. 4.2. For 200 of Klinec’s traces, attacking single key byte with full traces took us less than 2 min.

**Crop Traces to the Leakage Range.** Once we identified the leakage range, we can further crop the traces by specifying the address and row intervals in our cropping tool. This step is the most important one for the overall attack acceleration. For Klinec’s traces, we cropped the traces from originally 2456 entries to 197 entries, i.e., we reduced the traces as well as the attack complexity by a factor of 12.

Once we decide to repeat the attack on the same implementation, only with a different key, we can make use of the exact leakage position, hence making the attack yet faster and ready for use with automated tools [10].

**Full Attack.** At this point, everything is ready for the full attack. First, the bitwise DCA attack is performed as per Algorithm 1 and detailed results are saved, i.e., for each key byte, each attack target, each target bit, key candidates are sorted by their rank together with the position of the maximum (i.e., the leakage index). Second, these results are processed: for each such a piece of result,

relative gap between the rank of the two top candidates is computed and used as a measure of candidate quality. With 200 of filtered Klinec’s traces, the first step took us roughly 2 min for all 16 key bytes, the second step cca 20 s. In Table 1, we show the results of the attack with Rijndael inverse taken as the target.

*Note 5.* In our results, we recognize two kinds of best candidates based on the gap: if the gap is greater than 10%, it is referred to as the *strong candidate*, otherwise it is referred to as the *weak candidate*. Since we know the key, we can identify the position of the correct key byte within all guesses. In order to recognize a successful attack, we emphasize it in case it occupies the top position: in black ■ if it is a strong candidate, and in gray ■ for a weak candidate.

### 4.3 Confirmation of the Leakage Hypothesis About Chow’s WBAES

In order to confirm our hypothesis on the leakage point in Chow’s WBAES as introduced in Sect. 3, we decided to perform two experiments:

1. reproduce the attack by Bos et al. which takes the values from memtraces,
2. modify Klinec’s implementation to dump the intermediates coming from the first set of tables Type II—this is where leakage in the original attack is expected to take place—and use them directly instead of memtraces.

We performed both attacks with identical setup, the only difference was in the trace data origin – it either came from memtraces, or from a direct manual dump. To our satisfaction, both results were *perfectly identical*. This confirms our hypothesis that the vulnerable intermediates are those identified in Sect. 3, i.e., the output of the first set of tables Type II. In the following experiments, we used direct dumps from the modified implementation instead of memtraces for performance reasons.

### 4.4 Blind Attack on Chow’s WBAES

In a real-world scenario where the key is unknown, we do not know at which position the correct key byte is within the list of candidates. In order to suggest a methodology to recognize the correct candidate, we need further observations about how both correct and incorrect candidates behave. For this purpose, we ran a set of attacks: we created 8 instantiations of the white-box tables, captured 500 traces and used all 255 targets as per Definition 1 – this makes altogether 32 640 attacks on individual key byte which took us almost 50 min.

The most significant problem is that there often occurs a strong, yet incorrect top candidate (i.e., a *false positive*), cf. Table 1 (e.g., 10<sup>th</sup> key byte and 5<sup>th</sup> target bit with almost 20% gap). In our overall results, 22% of top candidates were strong and correct with an average and maximal gap of 38% and 76%, respectively. However, there were also 8% of (strong) false positives with an average and maximal gap of 14% and 35%, respectively. It follows that a simple rule using single gap threshold would work bad. On the other hand, we observed



**Table 1.** DCA using 200 memtraces and 8 bits of Rijndael inverse as targets. For each key byte and each target bit, percentual gap of the best candidate and position of the correct key byte are given. Note that the position ranges from 0 to 255 while 0 is replaced with ■ or ■ for a strong or a weak candidate, respectively; cf. Note 5.

Key byte	Target bit															
	1. bit	2. bit	3. bit	4. bit	5. bit	6. bit	7. bit	8. bit								
1.	14.5	148	19.9	■	7.6	■	0.3	241	0.5	81	4.7	226	3.7	3	17.0	■
2.	4.5	164	12.9	218	3.3	187	33.7	■	7.3	54	0.3	117	0.2	167	30.4	■
3.	0.7	183	0.2	205	0.2	4	18.8	■	2.5	192	4.2	115	1.4	184	8.5	72
4.	2.3	91	1.5	■	12.9	163	2.2	59	2.5	68	0.5	152	0.2	219	2.6	162
5.	1.9	15	2.8	68	5.7	60	1.1	153	0.2	42	5.1	161	0.0	35	0.3	127
6.	30.2	■	9.7	210	7.6	101	6.5	135	1.0	2	35.6	■	28.0	■	0.2	58
7.	2.7	7	6.0	57	0.7	179	6.6	241	1.8	137	5.2	1	2.4	123	6.5	198
8.	50.8	■	3.3	211	1.4	198	2.1	251	1.7	155	2.4	255	35.2	■	4.6	176
9.	18.5	181	5.9	111	0.6	52	0.3	235	3.9	86	5.0	154	33.2	■	1.3	121
10.	0.8	38	6.2	152	20.5	■	26.0	■	19.3	111	0.9	137	27.8	■	34.5	■
11.	4.5	141	17.3	■	6.8	35	10.2	176	2.9	137	8.8	66	3.2	79	1.3	136
12.	24.1	■	4.0	206	5.6	113	2.5	213	5.6	69	2.3	210	34.7	■	2.1	77
13.	4.2	24	5.9	246	2.8	244	0.2	15	30.4	■	3.3	■	9.1	125	11.1	34
14.	49.7	■	1.7	248	4.9	33	20.5	■	7.6	98	16.7	252	14.7	■	29.9	■
15.	6.5	139	0.6	126	6.3	16	5.4	37	2.3	64	3.6	1	3.6	■	5.8	100
16.	17.5	157	40.7	■	5.8	105	0.1	37	23.9	■	14.2	184	0.1	211	2.1	17

that the same false positive does not appear to repeat very much across the 255 targets for given key byte: the average number of repetitions of the best false positive (for given key byte) was 1.75, the global maximum was only 3. A summary of results will be given after we introduce another quantity in Sect. 4.5.

**Suggested Strategy.** We suggest the following strategy: for each key byte, keep looping the 255 targets until any strong candidate exceeds a cumulative bound of 50% with its gaps. Note that if such a candidate were a false positive, it would need about 4 average gaps of a false positive to exceed the bound, which is still more than ever observed number of repetitions of a false positive. Even if bad things happen in a rare case, we can possibly increase the cumulative bound or brute-force the least confident key byte(s). Note that a similar strategy can be derived to other schemes than Chow's WBAES.

#### 4.5 Optimal Number of Traces

In their attacks, Bos et al. used 500 and 2000 traces to attack Chow's WBAES, let us now have a look at results of the attack with much less traces, namely 100, 200, 300 and 500 traces. For each number of traces, we attacked all of our 8 instantiations and observed ratios of strong candidates (both correct and incorrect) together with their average gap; see results in Table 2. Note that the number of repetitions of false positives remained up to three.

With less traces, the number of correct candidates and their average gap decrease, i.e., we need to use more targets in order to reach the cumulative bound, and vice versa. Hence our goal is to give a reasonable estimate on the optimal number of traces in terms of computational effort. For this purpose, we introduce the *reduced cost of gap* as

$$C(n, s, g) = \frac{n}{s \cdot g}, \quad (7)$$

where  $n$  stands for the number of traces,  $s$  for the average success rate and  $g$  for the average gap of a strong candidate. Note that this quantity corresponds with the average computational effort: indeed, the more traces, the more effort,

**Table 2.** Ratios and average gaps of correct and incorrect strong candidates, respectively, and reduced cost of gap, for different numbers of traces.

Traces	100	200	300	500
Correct candidates	6.5%	17%	19%	22%
Average gap of correct candidates	22%	29%	34%	38%
False positives	2.3%	7.5%	8.2%	8.3%
Average gap of false positives	9.8%	14%	14%	14%
Reduced cost of gap	7 000	4 100	4 600	6 000

the better success rate or the bigger average gap, the less effort. According to Table 2, the lowest value of reduced cost of gap was achieved for 200 traces, therefore we suggest to use 200 traces in this scenario.

## 5 Conclusions

After a brief overview of white-box cryptography and Chow's WBAES, we recalled the idea of SCA usage in the white-box context, pioneered by Bos et al. We highlighted the abnormal behavior of their attack against Chow's WBAES, for which we proposed a theoretical explanation. The problem of Chow's WBAES has shown to be linearity of the Enc bijection which was intended to be non-linear. Although Chow et al. provided a reasoning about its non-linearity, it is not sufficient against DCA anymore, in particular when using our extended set of linear AES-DCA targets. We motivated the use of our targets against other implementations that use (semi-)linear masking of intermediates.

In the experimental part, we described our tools and, in particular, we confirmed our hypothesis by a comparison of two differently obtained sets of detailed results. Next, we focused on the behavior of false positives in case of a blind attack and suggested a strategy for this purpose. Finally, we derived an optimal number of traces for this kind of attack in terms of average computational cost to make the attack effective. With resulting 200 of filtered traces of Klinec's implementation, we ran the attack in less than two and a half minutes on our hardware.

## References

1. Pin 3.11 User Guide. <https://software.intel.com/sites/landingpage/pintool/docs/97998/Pin/html/>. Accessed Aug 2019
2. Banik, S., Bogdanov, A., Isobe, T., Jepsen, M.: Analysis of software countermeasures for whitebox encryption. *IACR Trans. Symmetric Cryptol.* **2017**, 307–328 (2017)
3. Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: *Annual International Cryptology Conference*, pp. 513–525. Springer (1997)
4. Billet, O., Gilbert, H., Ech-Chatbi, C.: Cryptanalysis of a white box AES implementation. In: Handschuh, H., Hasan, M.A. (eds.) *Selected Areas in Cryptography*, pp. 227–240. Springer, Heidelberg (2004)
5. Bock, E.A., Brzuska, C., Michiels, W., Treff, A.: On the ineffectiveness of internal encodings-revisiting the DCA attack on white-box cryptography. In: *International Conference on Applied Cryptography and Network Security*, pp. 103–120. Springer (2018)
6. Bogdanov, A., Isobe, T., Tischhauser, E.: Towards practical whitebox cryptography: optimizing efficiency and space hardness. In: *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 126–158. Springer (2016)
7. Bogdanov, A., Wang, J.M., Vejre, S.: Higher-order DCA against standard side-channel countermeasures. In: *Constructive Side-Channel Analysis and Secure Design: 10th International Workshop*, vol. 11421, p. 118. Springer (2019)

8. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of checking cryptographic protocols for faults. In: International conference on the theory and applications of cryptographic techniques, pp. 37–51. Springer (1997)
9. Bos, J., Hubain, C., Michiels, W., Teuwen, P.: Differential computation analysis: hiding your white-box designs is not enough. In: International Conference on Cryptographic Hardware and Embedded Systems, pp. 215–236. Springer (2016)
10. Breunese, C.B., Kizhvatov, I., Muijrs, R., Spruyt, A.: Towards fully automated analysis of whiteboxes: perfect dimensionality reduction for perfect leakage. IACR Cryptology ePrint Archive 2018, 95 (2018)
11. Chow, S., Eisen, P., Johnson, H., Van Oorschot, P.: A white-box DES implementation for DRM applications. In: Feigenbaum, J. (ed.) Digital Rights Management, pp. 1–15. Springer, Heidelberg (2002)
12. Chow, S., Eisen, P., Johnson, H., Van Oorschot, P.: White-box cryptography and an AES implementation. In: Nyberg, K., Heys, H. (eds.) Selected Areas in Cryptography, pp. 250–270. Springer, Heidelberg (2002)
13. CryptoExperts: WhibOx 2017 (2017). <https://whibox-contest.github.io/2017/>. Accessed Aug 2019
14. Dusart, P., Letourneux, G., Vivolo, O.: Differential fault analysis on AES. In: International Conference on Applied Cryptography and Network Security, pp. 293–306. Springer (2003)
15. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: concrete results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems-CHES 2001, pp. 251–261. Springer, Heidelberg (2001)
16. Goubin, L., Paillier, P., Rivain, M., Wang, J.: How to reveal the secrets of an obscure white-box implementation. Technical report, Cryptology ePrint Archive, Report 2018/098 (2018). <https://eprint.iacr.org/2018/098>
17. Jacob, M., Boneh, D., Felten, E.: Attacking an obfuscated cipher by injecting faults. In: Feigenbaum, J. (ed.) Digital Rights Management, pp. 16–31. Springer, Heidelberg (2002)
18. Kerckhoffs, A.: La Cryptographie Militaire. Journal des sciences militaires **9**, 538 (1883)
19. Klemsa, J.: Bitwise DPA. Git repository. <https://github.com/fakub/BitwiseDPA>
20. Klemsa, J.: Memory Tracing Tools for Intel PIN. Git repository. <https://github.com/fakub/MemoryTracingTools>
21. Klemsa, J.: White-Box-DPA-Processing toolkit. Git repository. <https://github.com/fakub/White-Box-DPA-Processing>
22. Klinec, D.: White-box attack resistant cryptography (2013)
23. Koç, Ç.: Cryptographic Engineering. Springer, Boston (2008)
24. Kocher, P.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Annual International Cryptology Conference, pp. 104–113. Springer (1996)
25. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Annual International Cryptology Conference, pp. 388–397. Springer (1999)
26. Luk, C.K., Cohn, R., Muth, R., Patil, H., Klauser, A., Lowney, G., Wallace, S., Reddi, V.J., Hazelwood, K.: Pin: building customized program analysis tools with dynamic instrumentation. In: ACM SIGPLAN Notices, vol. 40, pp. 190–200. ACM (2005)
27. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Hellesteth, T. (ed.) Advances in Cryptology-EUROCRYPT '93, pp. 386–397. Springer, Heidelberg (1993)

28. Muir, J.A.: A tutorial on white-box AES. Technical report, Cryptology ePrint Archive, Report 2013/104 (2013). <http://eprint.iacr.org/2013/104>
29. Nethercote, N., Seward, J.: Valgrind: A framework for heavyweight dynamic binary instrumentation. In: ACM SIGPLAN Notices, vol. 42, pp. 89–100. ACM (2007)
30. PUB, NIST FIPS: 197: Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197, 441–0311 (2001)
31. Rivain, M., Wang, J.: Analysis and improvement of differential computation attacks against internally-encoded white-box implementations. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**, 225–255 (2019)



# Immune-Based Network Dynamic Risk Control Strategy Knowledge Ontology Construction

Meng Huang<sup>1,2</sup>, Tao Li<sup>1</sup>(✉), Hui Zhao<sup>1</sup>, Xiaojie Liu<sup>1</sup>, and Zhan Gao<sup>3</sup>

<sup>1</sup> College of Cybersecurity, Sichuan University, Chengdu, Sichuan, China  
jshuangm@163.com, {litao, zhaohui, liuxiaojie}@scu.edu.cn  
<sup>2</sup> College of Computer Science and Engineering, Chongqing Three Gorges University,  
Wanzhou, Chongqing, China  
<sup>3</sup> College of Computer, Sichuan University, Chengdu, Sichuan, China  
m18980933897@163.com

**Abstract.** Knowledge base of dynamic risk control strategy based on immunity is a significant effect on effective analysis and defense against illegal network intrusion. How to realize the automatic understanding and processing of computers with control strategy knowledge is of great significance for quickly responding to network security risks. As a kind of knowledge representation tool, ontology can provide support for knowledge sharing, reuse and automatic computer understanding in specific fields, and has been widely used in various fields. This paper first introduces the immune-based network dynamic risk control model and network dynamic risk quantitative evaluation. And then, according to the ontology modeling method of network dynamic risk control strategy knowledge, this paper extracts domain knowledge concepts, attributes, relationships, instances, etc., and constructs domain ontology model, application ontology model, and atom ontology model for the network dynamic risk control strategy knowledge. These ontology models are represented using semantic Web ontology expression languages PDF and OWL, and are constructed using the protégé ontology editing tool. Finally, the important concepts in the knowledge of network dynamic risk control strategy and the relationship between concepts are expressed in the form of graph, so as to help the network security analysts and decision makers to effectively control and make decisions.

**Keywords:** Artificial immunity · Network dynamic risk control · Knowledge base · Ontology

## 1 Introduction

With the rapid development of modern information technologies such as cloud computing, internet of things, and 5G, network security has emerged in a new form, which puts higher demands on network security risk detection and control. The detection and control of network security risks is a systematic project, which requires each subsystem in the system to work together efficiently to ensure high robustness. There are striking similarities between cybersecurity risk detection and control security issues and the

problems encountered with biological immune systems, both of which can maintain system stability in a constantly changing environment [1–3]. Therefore, the introduction of artificial immune related theory into the research of network security risk detection and control is a very important and meaningful research direction.

Knowledge base is a knowledge set that can be organized, stored, managed, used and shared by adopting the corresponding knowledge representation method according to the needs of solving problems in specific fields. Constructing a comprehensive knowledge base of network dynamic risk control strategies will help to exchange and share network security knowledge, which will help to better analyze network intrusion behavior and play an important role in network defense. However, the scientific establishment of knowledge and the scientific establishment of the knowledge base model are hot topics.

Ontology is an effective tool for realizing knowledge sharing, and it is a formal specification of the conceptual model of domain knowledge sharing. Ontology technology can unify the knowledge concept of network dynamic risk strategy and the relationship between concepts, and enhance the normalization, consistency and extensibility of knowledge representation of network dynamic risk control strategy, and realize dynamic risk control strategy of knowledge sharing, reuse, and the computer automatic analysis and processing.

It is of great significance to establish a good network dynamic risk control strategy knowledge ontology. Based on the analysis of the general methods of ontology modeling, this paper proposes a method for constructing the ontology of network dynamic risk control strategy. According to this method, the network dynamic risk control strategy knowledge ontology is constructed. According to the hierarchical structure of the domain concept, the network dynamic risk control strategy knowledge domain ontology, application ontology and atomic ontology are constructed. The Protégé ontology modeling tool is used to construct the model.

The paper is organized as follows. The second section introduces related work, mainly based on immune network dynamic risk control model and dynamic risk quantitative assessment based on immune network. The third section presents the construction process of network dynamic risk control strategy knowledge ontology. The fourth section introduces the network Ontology implementation of dynamic risk control strategy knowledge. The fifth section summarizes the paper.

## 2 Related Work

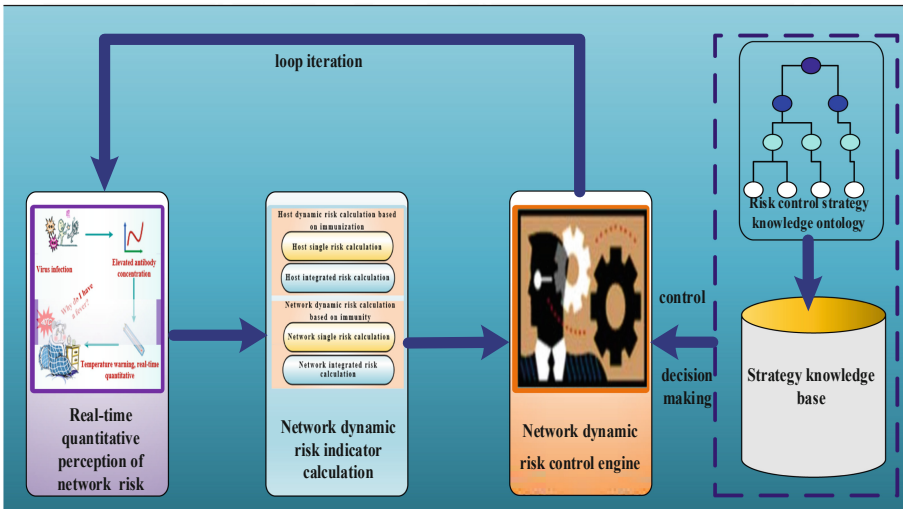
### 2.1 Network Security Domain Ontology Construction

Wu et al. [4] studied the network intrusion knowledge base, constructed the network intrusion ontology, and formalized the various types of network intrusion behaviors, and presented a multi-level and multi-dimensional network intrusion knowledge base classification system. Obrst et al. [5] from MITRE Corporation used ontology technology to build a knowledge base in the field of network security. Iannacone et al. [6] proposed an ontology-based network security knowledge base, which merges data from different data sources into the network security domain through an iterative design process. The ontology contains 15 entity types and 115 attributes. Jia et al. [7] use machine learning methods to extract cyber security domain entities, and then build ontology to build a

network security knowledge base. Falk [8] builds a network security ontology based on Lockheed Martin’s kill chain model to support cyber threat intelligence to help threat intelligence analysts effectively organize and search open source intelligence and threat metrics to effectively address cyber threats. Mozzaquatro et al. [9] proposed an ontology-based IoT network security framework to solve the security problems of IoT devices and IoT business processes. It can be seen that ontology technology has been widely applied and developed in the field of network security, and has become a powerful tool for knowledge representation in the field of network security.

**2.2 Immune-Based Network Dynamic Risk Control Model**

Figure 1 shows an immune-based network dynamic risk control model. Firstly, obtain the risk indicators of the current network environment threat change through immune-based cyber security threat change perception (dynamic risk calculation), and then, according to the risk indicator, select a targeted defense strategy from the strategy intelligence, including logs, warnings, traffic control, limited services, etc. implement active and proactive defense strategies, and provide targeted control over different categories and levels of risk to prevent the spread of attacks and improve the viability of the system in complex application environments.



**Fig. 1.** Immune-based network dynamic risk control model

**2.3 Quantitative Assessment of Network Dynamic Risk Based on Immune**

How to quantitatively calculate the risk value of network systems under attack in real time. According to the theory of artificial immunity [10], by simulating the human body temperature risk warning mechanism, the risk level of network threat change is divided.



The risk indicator is modeled as a real number between 0 and 1, with a larger value indicating a higher risk, 1 means extreme risk and 0 means no risk. These real-time and quantitative calculation formulas for network dynamic risk is as follow. Let  $Ac$  is the antibody concentration value,  $\theta$  is the decay step size of the antibody concentration,  $\lambda$  is the decay period of the antibody concentration,  $\alpha$  is the initial antibody concentration value, and  $\beta$  is the reward parameter,  $\omega_i$  is the risk weight of the  $i$  th attack, and  $\mu_m$  indicates the asset weight of the host  $m$ ,  $Ac_i$  represents the antibody concentration value of the host  $m$  subjected to the  $i$ th attack,  $I$  indicates the number of attack types,  $M$  represents the number of hosts in network  $n$ .

The concentration of antibody was calculated as Eq. 1 and Eq. 2:

When a network attack is detected, the antibody concentration increases, and the formula is as follows:

$$Ac(t) = \alpha + \beta Ac(t - 1) \tag{1}$$

When no network attack is detected, the antibody concentration is reduced and the formula is as follows:

$$Ac(t) = \begin{cases} Ac(t - 1) - \frac{Ac(t-1)}{\lambda - \theta(t-1)}, & \theta(t - 1) < \lambda \\ 0, & \theta(t - 1) \geq \lambda \end{cases} \tag{2}$$

The risk calculation is expressed as Eq. 3 to Eq. 6:

The risk value of the  $i$ th attack that host  $m$  receives at time  $t$ :

$$r_{m,i}(t) = \frac{2}{1 + e^{-\omega_i \bullet Ac_i}} - 1 \tag{3}$$

The risk value of the type  $I$  attack that host  $m$  receives at time  $t$ :

$$r_m(t) = \frac{2}{1 + e^{\left(-\sum_{i=1}^I \omega_i \bullet Ac_i\right)}} - 1 \tag{4}$$

The risk value of the  $i$  th attack that network  $n$  receives at time  $t$ :

$$R_{n,i}(t) = \frac{2}{1 + e^{\left(-\omega_i \bullet \sum_{m=1}^M \mu_m \bullet Ac_i\right)}} - 1 \tag{5}$$

Network overall risk value:

$$R(t) = \frac{2}{1 + e^{\left(-\sum_{i=1}^I \left(-\omega_i \bullet \left(\sum_{m=1}^M \mu_m \bullet Ac_i\right)\right)\right)}} - 1 \tag{6}$$

### **3 Construction of Network Dynamic Risk Control Strategy Knowledge Ontology**

#### **3.1 Ontology Review**

Ontologies have different definitions in different subject areas. In the field of philosophy, it reflects the systematic description of the objective beings in the world. In the field of industrial intelligence, one of Ontology's more accepted definitions is defined by Studer et al. [11]. They consider Ontology to be a clear formal specification of a shared conceptual model. In the field of knowledge engineering, the goal of building domain ontology is to acquire knowledge of a certain domain, determine the concept of common understanding in the domain, and give a clear expression of concepts and the relationship between concepts. The ultimate goal is to realize the sharing of knowledge within the domain, and to provide services for decision making.

In practical applications, ontologies can be represented by natural language, semantic network and logical language. With the application research of ontology on the Web, currently the most popular ontology presentation languages are RDF [12] and OWL [13] recommended by the W3C.

The following sections in this chapter are organized as follows. Section 3.2 introduces the method of constructing the knowledge ontology of network dynamic risk strategy based on software engineering ideas. Section 3.3 first introduces ontology analysis of network dynamic risk strategy knowledge, and then introduces domain ontology, application ontology, and atomic ontology for network dynamic risk control strategy.

#### **3.2 Construction Method of Network Dynamic Risk Strategy Knowledge Ontology**

The concept of ontology is introduced in the field of network dynamic risk control. The ultimate goal is to enable computers to fully understand semantic information, so as to be more intelligent for network dynamic risk control decision-making services. Therefore, in the process of ontology construction in this field, the experience of software engineering is fully borrowed to form the following ontology development process. The process is shown in Fig. 2:

Firstly, ontology construction should understand the requirements of domain ontology construction, and determine the scope and use of ontology coverage. The domain knowledge is then collected and the key concepts of the domain of interest are abstractly defined, along with the relationships and hierarchies between concepts. Then, the ontology logic reasoner is used for detection to determine whether the extracted concepts and relationships meet the requirements. Finally, ontology instantiation. The process of ontology construction is also the process of iterative evolution of ontology. Through this iterative evolution, an ontology model that meets the user's needs can be constructed.

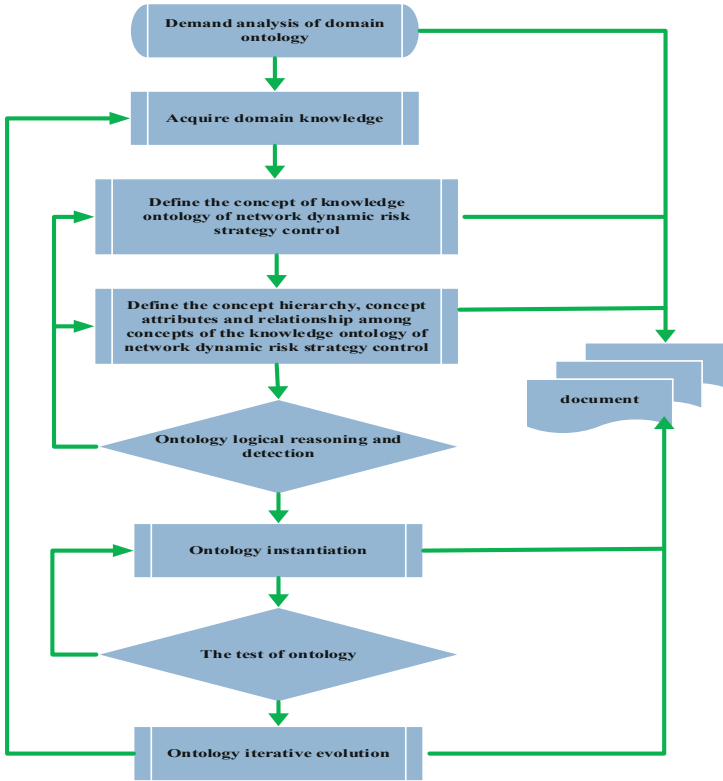


Fig. 2. Ontology development process

### 3.3 Ontology Model of Network Dynamic Risk Control Strategy Knowledge

Firstly, this section introduces an example of network dynamic risk control strategy ontology, and then introduces domain ontology, application ontology, and atomic ontology for network dynamic risk control strategy.

**(1) Ontology Analysis of Network Dynamic Risk Strategy Knowledge.** In the construction process of the network dynamic risk control strategy knowledge ontology, the knowledge of the domain is classified into categories, multi-level and multi-dimensional. The concept of knowledge of the domain, the relationship between concepts, and examples are clearly defined. The ontology of network dynamic risk control strategy is divided into domain ontology, application ontology and atomic ontology. The domain ontology is the top-level ontology of the domain, reflecting the top-level concept of the domain; the application ontology further refines the domain ontology according to the different applications to be taken in different stages; the atomic ontology is an entity element that can be directly applied according to different applications. It is the lowest level of the ontology.

Figure 3 shows a portion of the network dynamic risk strategy ontology. The network dynamic risk strategy knowledge ontology can be represented as a graph  $G$ . In the

semantic web, the relationship between entities and entities can be represented by RDF triples. Where, the Rdfs: subClassOf tag indicates that one concept is a subclass of another. The relationship between the schema graph and the data graph is represented by rdf: type tag, that is, the data graph is an instance of the class in the schema graph. For example, in the figure below, <Digital UNIX network service buffer overflow attack, control mode, stop service> means that the control method for Digital UNIX network service buffer overflow attack is to stop the service.

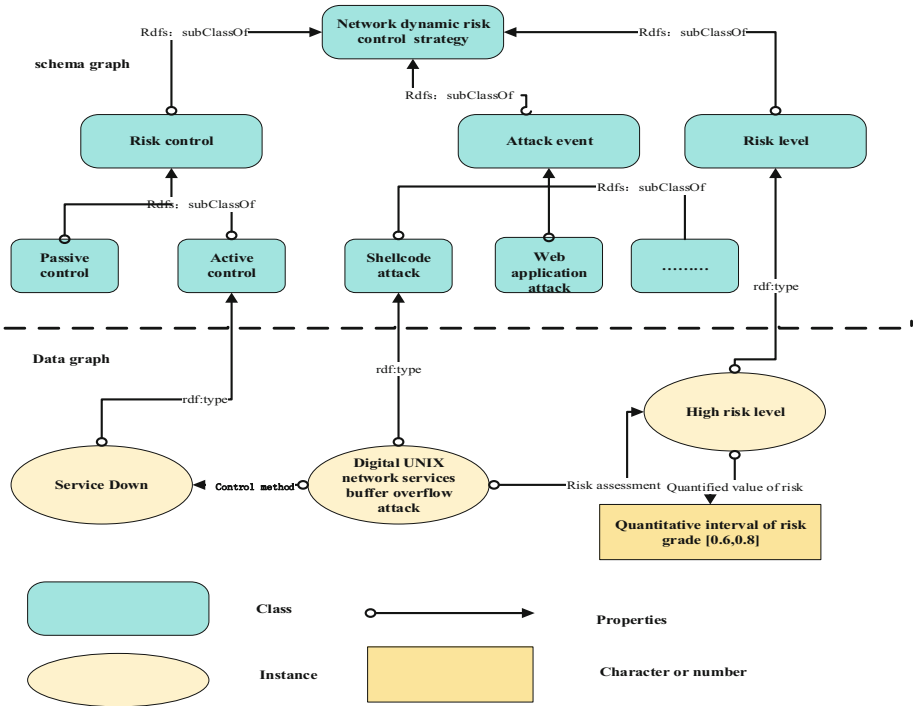


Fig. 3. Analysis of the ontology of network dynamic risk control strategy

**(2) Domain Ontology of Network Dynamic Risk Control Strategy Knowledge.** The domain ontology of network dynamic risk control strategy knowledge is the topmost ontology in this domain, which contains the basic concepts of the domain of network dynamic risk control strategy and the relationships between them. The domain ontology of network dynamic risk control strategy knowledge is shown in Fig. 4, which is composed of six categories, including attack events, attack targets, attackers, risk indicators, vulnerabilities, and dynamic network risk control measures. The rounded rectangular box represents the concept, and the arrow represents the relationship between the concepts.

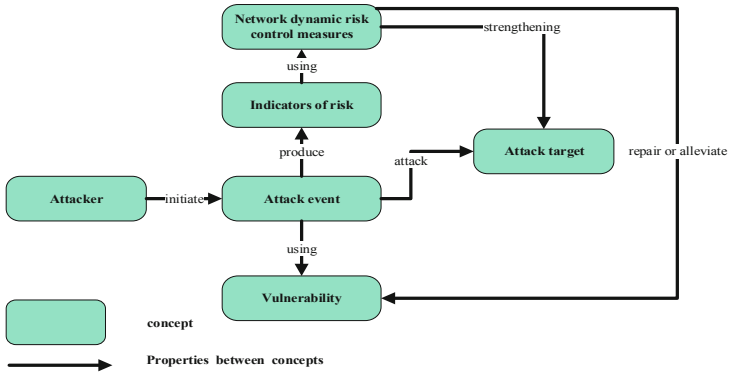


Fig. 4. Network dynamic risk control strategy knowledge domain ontology

**(3) Application Ontology of Network Dynamic Risk Control Strategy Knowledge.**

Network dynamic risk control strategy knowledge application ontology is a further subdivision of the domain top-level ontology. The six categories of attack events, attack targets, attackers, risk indicators, vulnerabilities, and dynamic network risk control measures are further subdivided. The ontology hierarchy diagram is shown in Fig. 5.

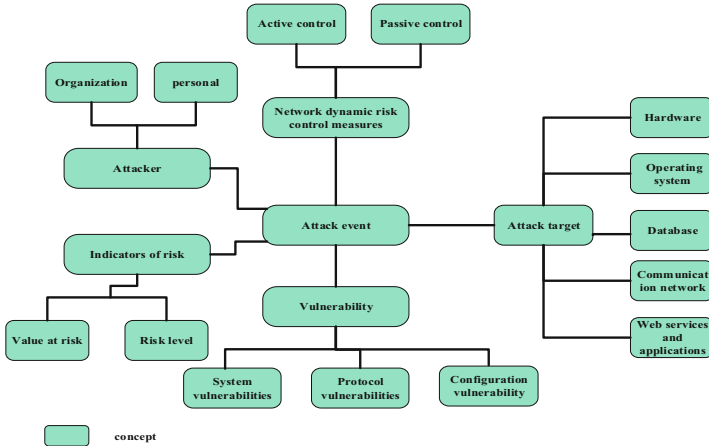


Fig. 5. Network dynamic risk control strategy knowledge application ontology

The attacker is the subject that initiates the attack, which is divided into organization and individual in the application ontology. Network dynamic risk control measures refer to the corrective strategies adopted to detect network security risks in the network. They are divided into active control strategies and passive control strategies in the application ontology. An attack target is an object that an attacker initiates an attack event and divides into five sub-classes: hardware, operating system, database, communication network, Web service, and application. Vulnerabilities are defects in the hardware, software, protocol implementation or system security policy, which can enable an attacker to access or destroy the system without authorization, and it can be divided into three

sub-categories: system vulnerability, protocol vulnerability and configuration vulnerability. The risk indicator refers to the quantified value of the current network attack risk obtained by the immune-based dynamic risk calculation, and is divided into two sub-categories: risk level and quantized value.

**(4) Atom Ontology of Network Dynamic Risk Control Strategy Knowledge.** The atomic ontology represents the smallest indivisible concept in the ontology, and the relationship between the atomic ontology and the application ontology is the relationship between the class and the instance. In the network dynamic risk control strategy knowledge ontology, we enumerate some atomic ontology.

(1) *Risk indicator atomic ontology.* In practical applications, we grade the current network attack risk indicators obtained through dynamic risk calculation. Specifically, we divide the interval [0, 1] into several different disjoint intervals, as shown in Fig. 6. For example, 5 intervals (which may vary slightly in actual application) to indicate different levels of network risk.

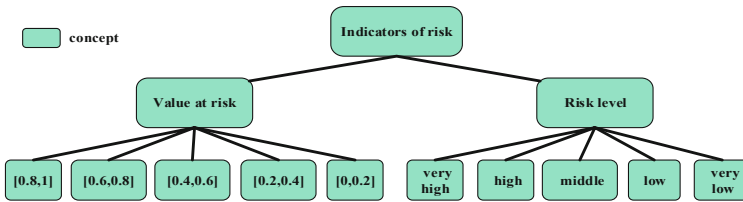


Fig. 6. Risk indicator atomic ontology

(2) *Risk control atomic ontology.* The immune-based dynamic risk control system formulates different risk control strategies for different types of attacks and the level of risk generated by the attacks. Extended active and passive control methods increase the system’s ability to handle different types of attacks and risk levels, and control strategies are more flexible and dynamically tuned. According to the real-time risk level of each attack category and the principle and feature description of the attack, the risk control mode is dynamically adjusted to control the system risk flexibly and ensure the security of the network. As shown in Fig. 7, the control strategy is divided into active and passive total 14 specific control methods.

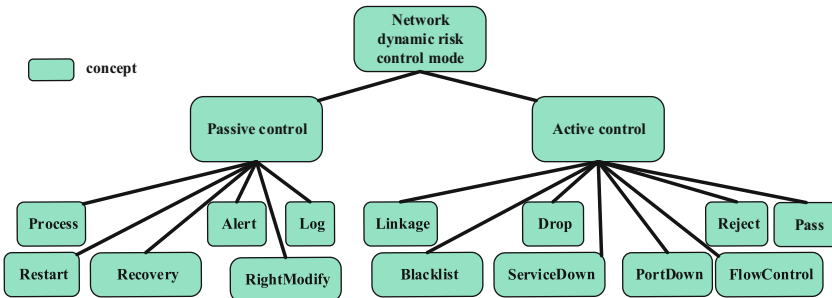


Fig. 7. Risk control atom ontology

## 4 Realization of Domain Knowledge Ontology of Network Dynamic Risk Control Strategy

Protégé [14] is a free and open source software platform that builds domain ontology models and develops knowledge-based applications. The ontology built by Protégé can output in various forms, support output in the form of RDF(S), OWL and XML Schema, and save the ontology in relational database. Protégé provides a rich set of plug-ins for knowledge modeling that can be used to create, visualize, manipulate, and manage ontology, and supports a variety of ways to express ontology. Figure 8 represents part of the knowledge ontology of the constructed network dynamic risk control strategy.

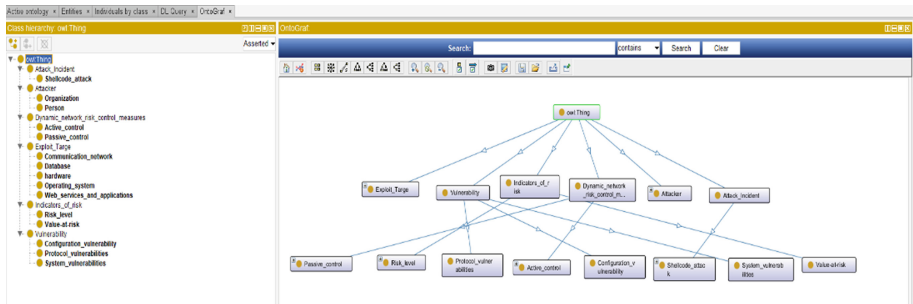


Fig. 8. Part of ontology visualization diagram

## 5 Conclusion and Future Work

This paper proposes a method to construct the ontology of network dynamic risk control strategy, and constructs an ontology based on immune dynamic network risk control strategy. Ontology modeling was performed using Protégé software. The most important task in the future is to enrich the network dynamic risk control strategy knowledge base and inference rules and apply them to related research on network intrusion detection and threat intelligence analysis.

**Acknowledgments.** This work was supported in part by the Natural Science Foundation of China (Grant No. U1736212, No. 61572334, No. 61872255), in part by the Sichuan Province Key Research & Development Project of China (Grant No. 2018GZ0183), in part by the Fundamental Research Funds for the Central Universities, and in part by the National key research and development program of China (Grant No. 2016YFB0800600).

## References

1. De Castro, L.N., Timmis, J.I.: Artificial immune systems as a novel soft computing paradigm. *Soft Comput. J.* 7(8), 526–544 (2003)

2. Jiao, L., Du, H.: Development and prospect of the artificial immune system. *Electron. J.* **31**(10), 1540–1548 (2003)
3. Xiao, R., Wang, L.: Artificial immune system: principle, models, analysis and perspectives. *J. Comput. Sci.* **25**(12), 1281–1293 (2002)
4. Wu, L.J., Wu, D.Y., Liu, S.L., Liu, L.: Research on network intrusion knowledge base model based on ontology. *Comput. Sci.* **40**(9), 120–129 (2013)
5. Obrst, L., Chase, P., Markeloff, R.: Developing an ontology of the cybersecurity domain. In: *CEUR Workshop Proceedings*, vol. 966, pp. 49–56 (2012)
6. Iannacone, M., Bohn, S., Nakamura, G., Gerth, J., Huffer, K., Bridges, R., et al.: Developing an ontology for cybersecurity knowledge graphs. In: *Proceedings of the 10th Annual Cyber and Information Security Research Conference*. ACM, New York (2015)
7. Jia, Y., Qi, Y., Shang, H., et al.: A practical approach to constructing a knowledge graph for cybersecurity. *Engineering* **4**(1), 53–60 (2018)
8. Falk, C.: An ontology for threat intelligence. In: *European Conference on Cyber Warfare and Security*. Academic Conferences International Limited (2016)
9. Mozzaquatro, B., Agostinho, C., Goncalves, D., et al.: An ontology-based cybersecurity framework for the Internet of Things. *Sensors* **18**(9), 3053 (2018)
10. Li, T.: An immunity based network security risk estimation. *Sci. China Ser. F: Inf. Sci.* **48**(5), 557–578 (2005)
11. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. *Data Knowl. Eng.* **25**(1–2), 161–197 (1998)
12. RDF. <https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/>. Accessed 5 Oct 2019
13. OWL. <https://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/>. Accessed 7 Oct 2019
14. Protégé. <https://protege.stanford.edu/>. Accessed 20 Sept 2019





# Windows 10 Hibernation File Forensics

Ahmad Ghafarian<sup>(✉)</sup> and Deniz Keskin

University of North Georgia, Dahlonega, GA 30597, USA

{ahmad.ghafarian,dkesk9340}@ung.edu

**Abstract.** Memory forensics is an essential part of any computer forensics investigation. Main memory provides valuable evidences, which may otherwise not be retrievable from hard drive. In cases when capturing main memory image is not possible, hibernation files are good source of information. The aim of this research is to show the importance of hibernation file forensics in a computer forensics investigation. Specifically, we focus on retrieving evidential information related to the use of Facebook and Instagram. Firstly, we develop a process that can simplify the task of hibernation file forensics. The proposed process explores concepts, tools, techniques and methodologies most suitable for Windows 10 hibernation file acquisition and analysis. Subsequently, we use the proposed process to experimentally demonstrate the extraction of critical personal and confidential information related to Facebook and Instagram activities, from hibernation file. The extracted data can be used to establish a link between the suspect and the evidences.

**Keywords:** Windows 10 · Hibernation · Hiberfil.sys · Facebook · Instagram · Social media · Forensics

## 1 Introduction

In recent years, the use of memory forensics has become an essential part of any investigation. The data contained in main memory is considered volatile because it is lost when a machine is powered down or temporarily put to sleep. Main memory provides valuable information which may otherwise not be retrievable from hard drive [1]. Capturing evidence from memory image is only possible if the system powered on and logged in. In practice, there are times when capturing memory image file is not possible due to the machine status, i.e. powered off. In these cases, hibernation file may be used as an alternative to memory image file [2].

Windows operating systems has an energy saving feature called hibernation [3]. When a machine is inactive for a set period or if the laptop lid is closed, the content of memory of the machine is copied and saved to hard drive in a file called *hiberfil.sys*. When the system is turn back on, the content of this file is copied back to main memory and the system continues [4]. The *hiberfil.sys* is overwritten each time the system hibernates so that just one hibernation file will be present on the system [5]. System backups such as restore points, volume shadow copies and other external backups can include archived copies of the hibernation file.

The purpose of this research is to contribute to the existing Windows 10 hibernation file forensics research. During our initial study, we found that there exists no formal guidelines or structured approach for hibernation file forensics. To address this issue, we propose a process for hibernation file forensics that helps the practitioner in the investigation. The process explores concepts, tools, techniques and methodologies most suitable for Windows 10 hibernation file acquisition and analysis. Subsequently, we use the proposed process to experimentally demonstrate the value of hibernation file forensics. Specifically, we focus on retrieving artifacts related to Facebook and Instagram activities.

The remainder of this paper is structured as follows: Sect. 2 provides literature review and background. Section 3 presents scope and research methodology. Section 4 covers analysis of the results. Our contribution is summarized in Sect. 5. The limitations of this research is briefly discussed in Sect. 6. The paper concludes in Sect. 7.

## 2 Background and Literature Review

### 2.1 Background

The hibernation file, i.e. *hiberfil.sys*, is created by Windows 10 operating systems when a system goes into sleep or a laptop lid is closed and is in the root directory of the hard drive [11]. The *hiberfil.sys* file is a Microsoft proprietary compression, which is hard to process. In circumstances when taking memory image file is not possible due to the system's status, the hibernation files can be used instead. Hibernation forensics can also be used in addition to memory forensics. The hibernation feature is complex and varies between operating systems versions and hardware configurations. Understanding these variations are not a trivial task because of the variations in tools and technologies that can parse and analyze these files. Recently, researchers and practitioners have developed unique tools that can reverse engineer the hibernation file format to the binary format which can then be processed by existing tools. Using these tools, we can retrieve forensics artifacts from the hibernation file such as, recent processes, list of open apps, information regarding open apps, internet history, videos, photos, user's credentials, geolocation information and timestamps.

### 2.2 Literature Review

Hibernation was mentioned in early stages of memory forensics, but due to its complexities it was never used [6]. The Windows XP hibernation file format was first reverse-engineered by Ruff and Suiche using a tool they developed for this purpose [7]. Windows XP hibernation forensics is also used by Mrdovic et al. [8]. In their research, the authors used both static media forensics and hibernation forensics. They used hibernation file on virtual environment and were able to retrieve various forensics artifacts from hibernation file. In 2012 Microsoft introduced Windows 8 and with this release Microsoft changed the format of the hibernation file. Consequently, the previously developed tools were not useable anymore for analyzing the hibernation file [4]. However, in 2016 Suiche introduced a new tool called Hibr2Bin. The Hibr2Bin converts Windows Hibernation files to binary format suitable for processing [9].

Prior research for Windows 10 hibernation file related to social media is very limited. In 2015 Murtuza conducted hibernation research on Windows 8.x and was able to acquire various social media artifacts from static and volatile memory. Murtuza's work mainly focused on the extraction of app-specific data [10]. In 2016 Singh complemented Murtuza's research. Singh's work mainly outlined the differences of hibernation header signatures and a possible method of hibernation file artifact extraction [11]. Also, in 2016, Silve et al. highlighted the main differences of Windows hibernation files over the years and the various methods of artifact extraction [4]. The purpose of this research is to develop a hibernation file forensics process that can simplify the task of investigators. The process explores concepts, tools, techniques and methodologies most suitable for Windows 10 hibernation file acquisition and analysis. Subsequently, we use the proposed process to experimentally demonstrate that using hibernation file forensics, it is possible to extract critical personal and confidential information related to Facebook and Instagram activities.

### 3 Scope and Research Methodology

This study aims to find out evidence of Facebook and Instagram activities in a hibernation file. We found no formal hibernation file forensics process in the literature. Therefore, we propose the following process of hibernation file forensics. The process is consisting of the following major phases, each with its own agenda of tasks and issues.

- Tools and testing platform
- Social media scenarios
- Client machine hibernation
- Hibernation file location and extraction
- Hibernation file conversion
- Acquisition of forensics artifacts

#### 3.1 Tools and Testing Platform

##### 3.1.1 Tools Used

This section lists the software tools and technologies needed to acquire, convert and analyze Windows 10 hibernation files. Upon examining several exiting forensics utilities, we selected some of the stable tools that are acceptable to the court of law. The selected tools are available on the Internet for free download. The tools are listed below in no particular order.

*Comae Tool Kit 3.0.2:* Comae Tools Kit (formerly known as MoonSols) is consisting of two tools, i.e. Hib2Bin and DumpIt. Hibr2Bin can be used to convert the compressed hibernation file into a binary file through reverse engineering. DumpIt can be used to generate a physical memory dump of Windows 10 machines [12]. It is compatible with both 32bit and 64bit architectures.

*HxD Binary Editor 2.0:* HxD is a hexadecimal editor for Windows dump files. It can open and edit the raw memory image file as well as displaying the memory used by running processes [13]. HxD can be used to search for specific string such as password.

*Foremost*: Foremost is a console program to carve files based on their headers, footers, and internal data structures. Foremost can work on image files or directly on a drive [14]. Foremost is mainly used for recovering images, videos and audios.

*Bulk Extractor (1.5.5)*: Bulk Extractor is a utility that scans a disk image, a file, or a directory of files to extract information without parsing the file system. The results can be inspected, parsed, or processed with automated tools [15].

*Others: Firefox 52.8, Instagram v38.1.0.5.30 and Facebook v152.0.1.37.362.*

### 3.1.2 Testing Platform

Our experiment have been carried out using the following physical machines

- One Linux Machine, for data transfer and image parsing: Intel i7-4810Q 4 Core Hyperthreading @ 2.80 GHz and 16 GB of RAM.
- One Windows 10 Forensic Machine, for forensics investigation activities including data analysis and data parsing: Intel i7-4810Q 4 Core Hyperthreading @ 2.80 GHz and 16 GB of RAM
- One Windows 10 Client Machine, for testing and carrying out the experiments. Intel Core Duo T2400 2 Core @ 1.83 GHz and 4 GB of RAM.

### 3.2 Social Media Scenario

In this study, we will not use real account information for privacy purpose. Instead, we created several fictitious Facebook and Instagram accounts for this project and performed all the activities on those accounts. We accessed Facebook and Instagram via web browser and performed the scenarios shown in Table 1.

**Table 1.** Instagram and Facebook Activities Scenarios.

Sequence Number	Instagram	Facebook
1	Login to Instagram act	Login to Facebook act
2	Check user profile	Check user profile
3	Update Instagram bio	Update Facebook bio
4	Set user profile picture	Check news feed
5	Post a picture	Create a post on the timeline
6	Tag a friend	Like
7	Set Location	Comment
8	Like	Upload pictures
9	Comment	Send a chat/message
10	Check friends list	Make a video call
11	Check friends' profile	Brows Facebook town hall

(continued)

**Table 1.** (continued)

Sequence Number	Instagram	Facebook
12	Receive notification	Browse friends profile
13	Receive personal message	
14	Reply to personal message	

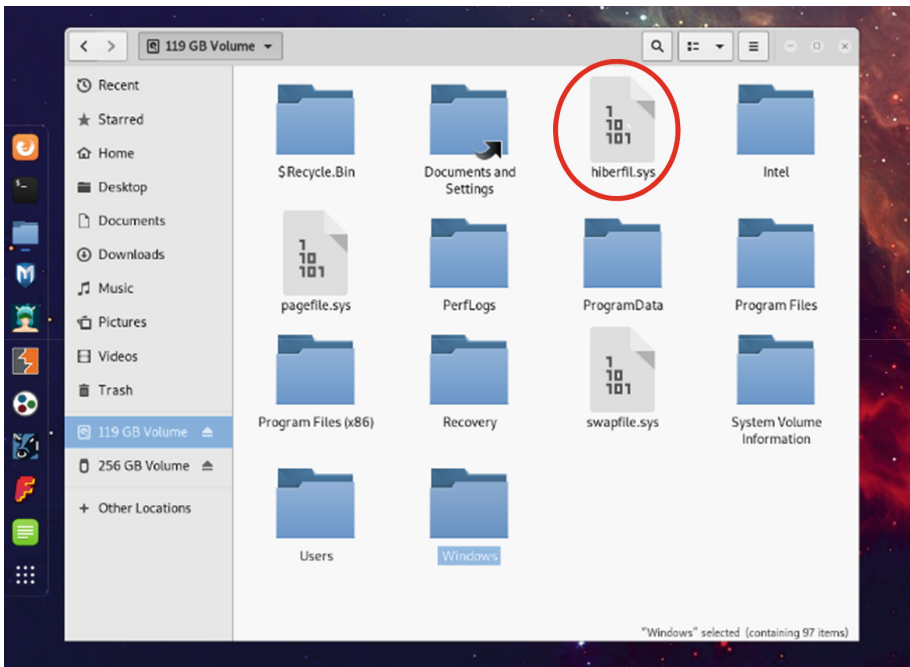
### 3.3 Client Machine Hibernation

After performing the scenarios listed in Table 1, we immediately put the system in the controlled hibernation mode by issuing `shutdown/h` from the command line. This process saves the content of main memory into the *hiberfil.sys* file.

### 3.4 Hibernation File Location and Extraction

Currently, Windows 10 does not allow copying *hiberfil.sys* file without altering hibernation file permissions. To ensure file integrity, we performed the following steps to bypass Windows user permissions for extracting the hibernation file.

- Remove hard drive from the client machine.
- Connect the client machine's hard drive to the Linux machine (read-only)
- On the Linux machine, locate the Windows hibernation file, shown in Fig. 1
- Copy the hibernation file to a forensically cleaned USB storage device
- Attach the USB storage device to the Windows forensic machine for processing.



**Fig. 1.** Location of the hibernation file on the Linux machine

### 3.5 Hibernation File Conversion

Currently, Windows 10 uses a proprietary hibernation compression that is not easy to process. Therefore, we used Hibr2Bin [12] tool to convert it to a binary file. The conversion process is listed below.

- Copy the *hiber.sys* file from the USB flash drive into the folder that contains Hibr2Bin.exe.
- Open the command prompt and navigate to the folder containing both *hiberfil.sys* file and Hibr2Bin.exe.
- Type “Hibr2Bin” to see the available options as shown in Fig. 2.

```
Microsoft Windows [Version 10.0.17763.316]
(c) 2018 Microsoft Corporation. All rights reserved.

D:\Comae-Toolkit-3.0.20190124.1\x64>hibr2bin

Hibr2Bin 3.0.20190124.1
Copyright (C) 2007 - 2017, Matthieu Suiche <http://www.msuiche.net>
Copyright (C) 2012 - 2014, MoonSols Limited <http://www.moonsols.com>
Copyright (C) 2015 - 2017, Comae Technologies FZE <http://www.comae.io>
Copyright (C) 2017 - 2018, Comae Technologies DMCC <http://www.comae.io>

Usage: Hibr2Bin [Options] /INPUT <FILENAME> /OUTPUT <FILENAME>

Description:
  Enables users to uncompress Windows hibernation file.

Options:
  /PLATFORM, /P      Select platform (X64 or X86)
  /MAJOR, /V         Select major version (e.g. 6 for NT 6.1)
  /MINOR, /M         Select minor version (e.g. 1 for NT 6.1)
  /OFFSET, /L        Data offset in hexadecimal (optional)
  /INPUT, /I         Input hiberfil.sys file.
  /OUTPUT, /O        Output hiberfil.sys file.

Versions:
  /MAJOR 5 /MINOR 1  Windows XP
  /MAJOR 5 /MINOR 2  Windows XP x64, Windows 2003 R2
  /MAJOR 6 /MINOR 0  Windows Vista, Windows Server 2008
  /MAJOR 6 /MINOR 1  Windows 7, Windows Server 2008 R2
  /MAJOR 6 /MINOR 2  Windows 8, Windows Server 2012
  /MAJOR 6 /MINOR 3  Windows 8.1, Windows Server 2012 R2
  /MAJOR 10 /MINOR 0 Windows 10, Windows Server 2017

Examples:
  Uncompress a Windows 7 (NT 6.1) x64 hibernation file:
  Hibr2Bin /PLATFORM X64 /MAJOR 6 /MINOR 1 /INPUT hiberfil.sys /OUTPUT uncompressed.bin
```

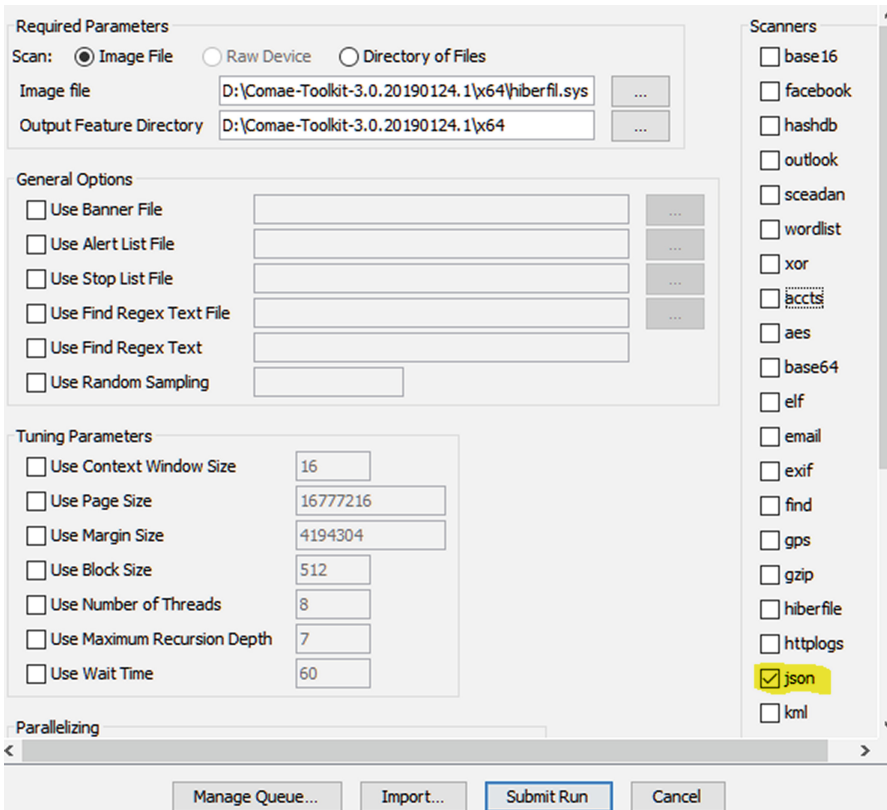
Fig. 2. Hibr2Bin.exe and file conversion options ready for conversion

- Since the machines we used were Windows 10 × 64 architecture, select the following option which produces *uncompressed.bin* file ready for analysis. Hibr2Bin /PLATFORM X64 /MAJOR 10 /MINOR 0 /INPUT hiberfil.sys /OUTPUT uncompressed.bin.”

### 3.6 Acquisition of Forensics Artifacts

In Facebook and Instagram and other social media, user activity data is usually presented in JSON (JavaScript Object Notification) format, which is a standard data interchange format. It is primarily used for transmitting data between a web application and a server.

We used Bulk Extractor [15] to parse JSON file for both Facebook and Instagram (see Fig. 3).



**Fig. 3.** Various options in Bulk Extractor for parsing JSON file

Figure 3, shows location of the binary image file, destination of output parsing data and JSON box have been checked. This action creates a text file called JSON.txt. We can use a text editor like Notepad to do string search for evidential information. For example, Fig. 4 shows searching for Instagram string in the parsed JSON.txt file.

For image, audio, and video parsing, we were not able to find a suitable tool in Windows environment. Therefore, we used Foremost [14] as it is described here. We copied/pasted the binary hibernation file into the root directory of the Linux forensic machine. Then, we opened a Linux terminal and typed foremost -t all -i uncompressed.bin, also shown in Fig. 5. Issuing this command allows us to parse the binary hibernation file for image, audio, and video files (see Fig. 6).



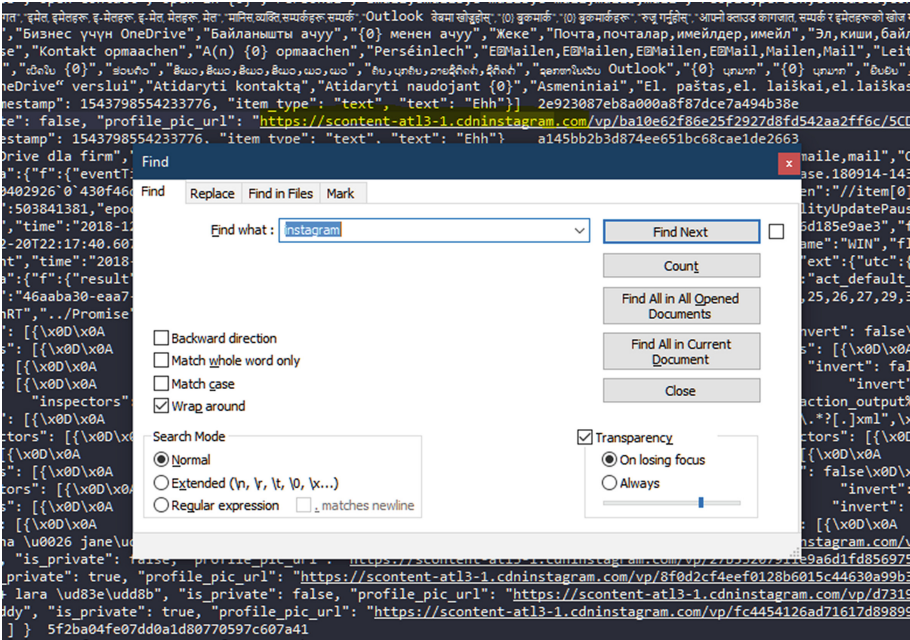


Fig. 4. String search for Instagram in JSON.txt.

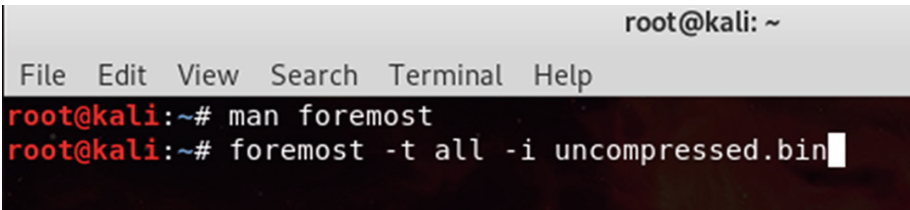


Fig. 5. Command issued to parse for images, videos and gif files.

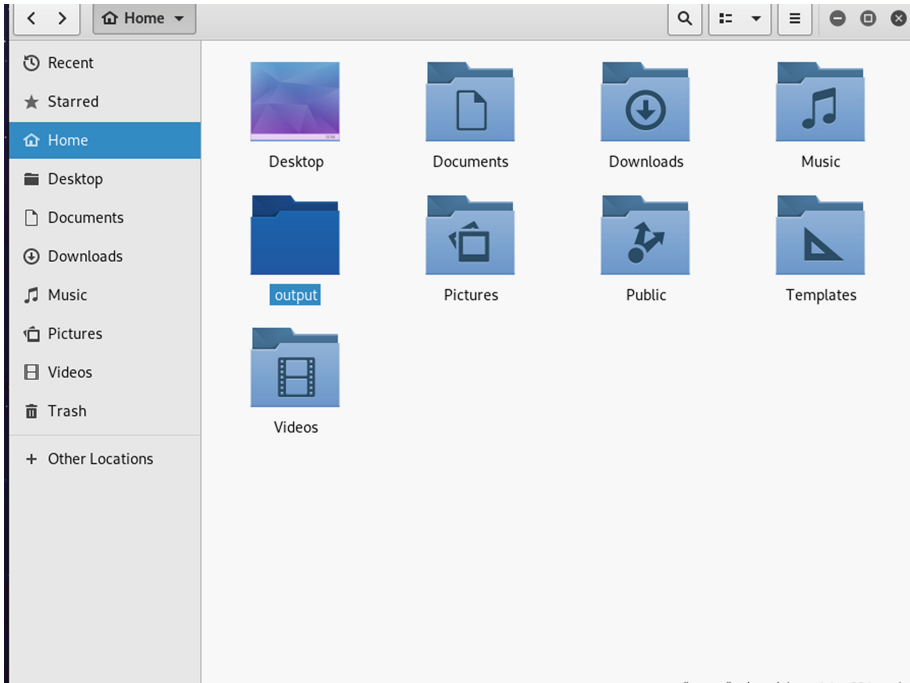
## 4 Results

This section provides the results of Facebook and Instagram hibernation file forensics experiment.

### 4.1 Instagram

Our experiment shows that forensic investigators can gather tremendous amounts of information regarding Instagram users and their friends from the Windows hibernation file. We were able to recover personal and account activity information as described below. To the best of our knowledge, our study is the only published work on recovering Instagram artifacts from the hibernation file. Below we discuss the artifacts.





**Fig. 6.** Outcome of Foremost parsing.

#### 4.1.1 User Profile

We were able to extract personal information from hibernation file. For example, Fig. 7 shows user login credentials, “hiberfil” extracted from JSON file.” Figure 8 shows user full name and profile picture URL to their Instagram profile picture, i.e. Hiber Fil”.

```
"username": "hiberfil", "pk": "9397768451",
```

**Fig. 7.** Instagram user login Id

```
"username": "hiberfil", "full_name": "Hiber Fil", "is_private": false, "profile_pic_url": "https://instagram.fc4i2-2.fna.fbcdn.net"
```

**Fig. 8.** User full name and the URL to the user picture

#### 4.1.2 Messages

In Instagram we can send and receive messages from friends. Figure 9 shows Instagram personal message received by the user. An important piece of information is that we were able to carve the picture of the sender of the message which is shown on the left side of the message, shown in Fig. 9.



**Fig. 9.** Instagram message received by the user

Figure 10 shows the parsed JSON API data for the message in Fig. 9. It shows all the attributes of the message including badge number, alert and the title of the message.

```
{"badge":1,"alert":"ege_aj_kes: The first law, also known as Law of Conservation of Energy, states that energy"
```

**Fig. 10.** Instagram parsed JSON API reveals all attributes of the message.

### 4.1.3 Notifications

In Instagram, you can choose to get push notifications when someone likes or comments on your post. If you have notifications turned on, you can also choose accounts that you want to receive notifications about. We turned on notification for the accounts we created for this experiment. Figure 11 shows JSON entry for notification. In this case, “sophie\_\_carla started following you.”

```
{"text": "sophie__carla started following you.", "links": [{"s
```

**Fig. 11.** JSON entry for Instagram notifications

### 4.1.4 Feed

Instagram Feed is a place where users can share and connect with the people. When a user opens Instagram or refresh his/her feed, the photos and videos will appear towards the top of the Feed. All posts from accounts a user follows on Instagram will appear in the Feed. Figure 12 shows the news Feed such as what the user liked, commented and shared, i.e. “love you guys happy thanksgiving”.

### 4.1.5 List of Instagram User’s Friend

Instagram Close Friends is a new feature that allows users to share their more personal Stories with just a select few friends. Figure 13 shows the friend’s list of suspect’s personal user account in the form of a JSON file.

```

22 { "AppId": "U:Microsoft.Windows.ShellExperienceHost_10.0.16299.637_neutral_neutral_cw5n1h2txyewy!App",
23 { "ver": "2.1", "name": "Win32kTraceLogging.AppInteractivitySummary", "time": "2016-11-23T01:21:34.8078758",
24 { "text": "tameramowrytwo liked itsdorian's comment: Love you guys\u201c\u201d happy thanksgiving",
25 { "type": 1, "story_type": 60, "args": { "text": "shrutzhaasan liked rohanshestha's post.", "links":
26 { "type": 2, "story_type": 60, "args": { "text": "ege_aj_kes liked 3 posts.", "links": [{ "start": 0,
27 { "action": "send_item", "client_context": "7573BCCB-E221-4E5A-A87C-1D6EBD5ABC7D", "item_type": "text", "te
28 { "utc": { "flags": 469762661, "app": { "ver": "1.1", "asId": 3152, "os": { "ver": "1.1", "bootId": 17, "device": {
29 { "ReportId": "20ac25d5-00d1-4f48-b1d0-794e07db3732", "InterpreterReportId": "", "EventName": "WindowsUpdat

```

Fig. 12. The news Feed API data.

```

{"pk": 3595804349, "username": "fibergourmet", "full_name": "Fiber Gourmet", "is_private": false, "pr
{"pk": 7635106931, "username": "womenwithfibromyalgia", "full_name": "Women With Fibromyalgia", "is_p
{"pk": 5879773518, "username": "eurobasket", "full_name": "FIBA EuroBasket", "is_private": false, "pr
{"pk": 4800174905, "username": "macrameanddriftwood", "full_name": "MACRAME+DRIFTWOOD | fibre art",
{"pk": 9794692531, "username": "jasonmccloskeydc", "full_name": "Jason McCloskey FIBFN DC", "is_priv
{"pk": 5557794060, "username": "hand and fiber", "full_name": "Amaa Aman-Tran", "is_private": false,
{"pk": 4526106435, "username": "onyxfiberarts", "full_name": "Aria", "is_private": false, "profile_p
{"pk": 1241084037, "username": "cf_foundation", "full_name": "Cystic Fibrosis Foundation", "is_privat
{"pk": 1947877974, "username": "echoviewfibermill", "full_name": "Echoview Fiber Mill", "is_private":
{"pk": 455103367, "username": "fiberartnow", "full_name": "Fiber Art Now", "is_private": false, "prof
{"pk": 22881148, "username": "niromastudio", "full_name": "Fiber Art \u201c\u201d Cindy",
{"pk": 4097186889, "username": "karmannghiafibra", "full_name": "Karmann Ghia de Fibra", "is_private
{"pk": 5516785858, "username": "fibersandfringe", "full_name": "Macrame Art + Accessories", "is_priv
{"pk": 2706772799, "username": "fibulaairtravelmk", "full_name": "Fibula Air Travel Macedonia", "is_p
{"pk": 769126570, "username": "paradisefibers", "full_name": "Paradise Fibers", "is_private": false,
{"pk": 3021300046, "username": "valkyrie_fibers", "full_name": "Lauren Brien-Wooster", "is_private":

```

Fig. 13. List of Instagram user’s friends’ full name

## 4.2 Facebook

To process JSON file for Facebook, we used both string search in JSON.txt as well as using HxD for JSON binary file. Some example of the information we were able to retrieve from both files is shown below.

### 4.2.1 Messages

Like Instagram, in Facebook users can also send messages and exchange photos, videos, stickers, audio, and files, as well as react to other users’ messages. Figure 14 shows the details of a conversation we obtained from Facebook messages using HxD tool, i.e. “Hey how are you today? How was farmers market?”

```

050AA2CC0 48 65 79 20 68 6F 77 20 61 72 65 20 79 6F 75 20 Hey how are you
050AA2CD0 74 6F 64 61 79 3F 20 48 6F 77 20 77 61 73 20 66 today? How was f
050AA2CE0 61 72 6D 65 72 73 20 6D 61 72 6B 65 74 3F B3 01 armers market?'.
050AA2CF0 63 6F 6D 70 6F 73 65 64 41 74 74 61 63 68 6D 65 composedAttachme
050AA2D00 6E 74 C0 DA 00 31 01 46 42 4D 65 73 73 61 67 65 nt\u0001.FBMessage

```

Fig. 14. Facebook Message

### 4.2.2 Posts

A Facebook post is a message in a special delivery cyber-bottle. It is a comment, picture or other media that is posted on the user’s Facebook page or “wall”. Figure 15 show users post and Fig. 16 shows a picture that was attached to that post. We were able to recover message and picture from both API data we gathered from the JSON.txt using HxD and Foremost tools, respectively.

```

69 74 69 65 73 00 73 04 00 00 00 74 65 78 74 01 titles.s...text.
01 1E 00 00 00 00 00 00 43 61 6E 27 74 20 47 .....Can't G
65 74 20 45 6E 6F 75 67 68 20 6F 66 20 54 68 69 et Enough of Thi
73 20 50 6C 61 63 65 00 72 0D 00 00 00 70 72 69 s Place.r...pri
76 61 63 79 5F 73 63 6F 70 65 01 00 00 00 00 00 vacy_scope.....
    
```

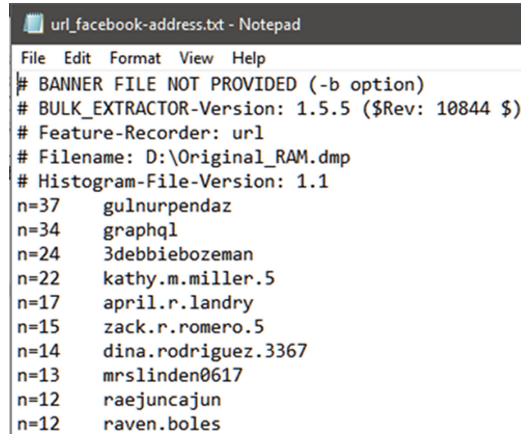
Fig. 15. A user’s post on Facebook



Fig. 16. The picture Associated with Fig. 15 post

### 4.2.3 Facebook Friends

In Facebook, when you add someone as a friend, you automatically follow that person, and he/she automatically follows you. This implies that you may see each other’s posts in News Feed. Figure 17 shows using Bulk Extractor tool, we recovered full friends list of a Facebook user.



```
url_facebook-address.bt - Notepad
File Edit Format View Help
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 1.5.5 ($Rev: 10844 $)
# Feature-Recorder: url
# Filename: D:\Original_RAM.dmp
# Histogram-File-Version: 1.1
n=37 gulnurpendaz
n=34 graphql
n=24 3debbiebozeman
n=22 kathy.m.miller.5
n=17 april.r.landry
n=15 zack.r.romero.5
n=14 dina.rodriguez.3367
n=13 mrslinden0617
n=12 raejuncajun
n=12 raven.boles
```

Fig. 17. List of Facebook friends recovered from hibernation file using Bulk Extractor

## 5 Contribution

To the best of our knowledge, our Windows 10 Hibernation forensic research is the first research to collect critical personal and confidential information from hibernation file, related to the use of Instagram and Facebook. With the rise of popularity of Instagram and Facebook this research will set the baseline for what researches are expected to gather from the Windows 10 Hibernation file.

In this research, we developed a hibernation file forensics process that can simplify the task of investigators. With this new process, we were able to explore concepts, tools, techniques and methodologies most suitable for Windows 10 hibernation file acquisition and analysis. We experimentally demonstrated the application of the hibernation file forensics process. Using our developed hibernation file forensics process, it is possible to extract critical personal and confidential information related to Facebook and Instagram activities reliably from Windows 10 machine.

## 6 Limitation of the Study

The limitation of this study was due to a lack of research, support and availability of the tools for Windows 10 hibernation file and extraction methods. In 2012 Microsoft introduced Windows 8 and with this release, Microsoft changed the format of the hibernation file. Consequently, the previously developed tools were not useable anymore for analyzing the hibernation file [4]. While this limitation affected the range and availability of the tools, we were able to find other tools and recreate results reliably.

## 7 Conclusions

The hibernation file forensics is a complicated task. This is due to two main reasons. First, there are no systematic and structured published guidelines for this task. Second,

Windows 10 hibernation format keeps changing from one version of the operating systems to another. As a result, the existing tools and methodologies may not be applicable anymore. This research has provided a more detailed information related to the hibernation file forensics. We provided a process that investigators and researchers can use for hibernation file forensics. Our proposed process is consisting of several major phases each of which with its own issues and tasks.

In the experiment part of the project, we used our hibernation file forensics process to demonstrate the effectiveness of hibernation file forensics. Our experiment focused on retrieving forensics artifacts related to Facebook and Instagram activities. We used several software tools to extract specific Facebook and Instagram information such as pictures, audio and video files. For Facebook, we were able to retrieve user profile information, friend list, profile pictures, videos, reaction icons, login credentials, feed, news, comment, notifications, likes and chats. For Instagram, we were able to retrieve feed, profile information, friends list, user bio, friends list, personal messages, comments, URL of friends' profile pictures, friends' user names, the full name of the friends' name, profile status, user profile picture URL. The experiment were performed in a forensically sound manner. The results demonstrate that using hibernation file forensics, the investigators can retrieve significant forensically valuable information related to social media usages. In addition, information such as login credentials, pictures etc. can be sued to establish a link between the suspect and the activities in a manner that is acceptable to the court of law.

There are many ways that this research can be extended. Repeating the experiment on different platforms, repeating this process at least twice to see discrepancies if any, and develop similar hibernation file forensics process for mobile devices as well as for virtual machine.

## References

1. Cai, L., Sha, J., Qian, W.: Study on forensic analysis of physical memory. In: 2nd International Symposium on Computer, Communication, Control and Automation (3CA 2013), pp 221–224 (2013)
2. Ayers, A.L.: Windows Hibernation and Memory Forensics, A Capstone Project Submitted to the Faculty of Utica College, April 2015, UMI Number: 1586690
3. Singh, A., Sharma, P., Nath, R.: Role of hibernation file in memory forensics of windows 10. *Int. J. Sci. Eng. Res.* **7**(12), 42–47 (2017), ISSN 2229-5518
4. Sylve, J.T., Marziale, V., Richard, G.G.: Modern windows hibernation file analysis. *Dig. Invest.* **20**, 16–22 (2016). <https://doi.org/10.1016/j.diin.2016.12.003>
5. Carvey, H.: Windows Forensic Analysis DVD Toolkit. Syngress Publishing Inc., Burlington (2007)
6. Carrier, B.D., Grand, J.: A hardware-based memory acquisition procedure for digital investigations. *Dig. Invest.* **1**, 50–60 (2004)
7. Ruff, N., Suiche, M.: Enter Sandman (why you should never go to sleep). In: PacSec Applied Security Conference (2007)
8. Mrdovic, S., Huseinovic, A., Zajko, E.: Combining static and live digital forensic analysis in virtual environment. In: 2009 XXII International Symposium on Information, Communication and Automation Technologies (2009). <https://doi.org/10.1109/icat.2009.5348415>

9. Suiche, M.: Your favorite Memory Toolkit is back... FOR FREE! – Comae Technologies. <https://blog.comae.io/your-favorite-memory-toolkit-is-back-f97072d33d5c>
10. Shariq, M., et al.: A tool for extracting static and volatile forensics artifacts of Windows 8.x Apps. IFIP Adv. Inf. Commun. Technol. Adv. Dig. Forensics XI, **462**, 305–320 (2015). [https://doi.org/10.1007/978-3-319-24123-4\\_18](https://doi.org/10.1007/978-3-319-24123-4_18)
11. Singh, A., Sharma, P., Nat, R.: Role of hibernation file in memory forensics of windows 10. Int. J. Sci. Eng. Res. **7**(12), 42–47 (2016)
12. Malin, C.H., Casey, E., Aquilina, J.M.: Memory Forensics. Malware Forensics Field Guide for Windows Systems, 93–154 (2012). <https://doi.org/10.1016/b978-1-59749-472-4.00002-0>. <http://index-of.es/Varios-2/Malware%20Forensics%20Field%20Guide%20for%20Windows%20Systems.pdf>
13. Hörz, M. (n.d.): HxD - Freeware Hex Editor and Disk Editor. <https://mh-nexus.de/en/hxd/>
14. SourceForge. (n.d.). Foremost. <http://foremost.sourceforge.net/>
15. Bulk Extractor. [https://www.forensicswiki.org/wiki/Bulk\\_extractor](https://www.forensicswiki.org/wiki/Bulk_extractor)



# Behavior and Biometrics Based Masquerade Detection Mobile Application

Pranieth Chandrasekara, Hasini Abeywardana<sup>(✉)</sup>, Sammani Rajapaksha<sup>(✉)</sup>,  
Sanjeevan Parameshwaran, and Kavinga Yapa Abeywardana

Department of Information Systems, Sri Lanka Institute of Information Technology,  
Colombo, Sri Lanka

praniethkumara@gmail.com, hasi.abeywardana@gmail.com,  
sammani.rajapaksha5@gmail.com, sanjeevancyca@gmail.com,  
kavinga.y@sliit.lk

**Abstract.** Mobile phone has become an important asset when it comes to information security since it has become a virtual safe. However, to protect the information inside the mobile, the manufacturers use the technologies as password protection, face recognition or fingerprint protection. Nevertheless, it is clear that these security methods can be bypassed. That is when the urge of a post-authentication is coming to the surface. In order to protect the phone from an unauthorized or illegitimate user this method is proposed as a solution. The aim of the proposed solution is to detect the illegitimate user by monitoring the behavior of the user by four main parameters. They are: 1) Keystroke dynamics with a customized keyboard; 2) location detection; 3) voice recognition; 4) Application usage. In the initial state machine learning is used to train this mobile application with the authentic user's behavior and they are stored in a central database. After the initial training period the application is monitoring the usage and comparing it with the already saved data of the user. Another unique feature of this is the prevention mechanism it executes when an illegitimate user is detected. Furthermore, this application is proposed as an inbuilt application in order to avoid the deletion of app or uninstallation of the app by the intruder. With this Application which is introduced as "AuthDNA" will help you to protect the sensitive information of your mobile device in a case of theft and bypassing of initial authentication.

**Keywords:** Authentication · Biometrics · Machine learning · Masquerade

## 1 Introduction

According to the "Statista", by the year 2019, the number of mobile phone users will surpass four billion. With the ability of multitasking almost as a computer, the individual users as well as enterprises, government agencies and the military are rapidly integrating mobile devices into their systems. The value of these devices varies from the sensitivity of stored data and the monetary value of the device itself. Even though there are both existing software and hardware security measures, the number of attacks on the device



increase by the day. There are some cases these security features such as fingerprints can be bypassed as well. But the central problem seems to be the inability for users to make better security choices. Most of the mobile user does not have a full understanding of the available security measures or to take full advantage and utilize the existing protection measures. The simple task of having a pin or a pattern to the phone has not accomplished by most of these users. The existing options tend to seek the guidance of the user which leads some users to eventually ignore the security measurements. While the consequences of compromise are severe the struggle to mitigate this risk is not yet been reduced but getting complexed rapidly. The need for constant monitoring for a device's legitimate user is still not fulfilled with these existing options. The users are more likely to be using a security measure that will not interfere with their use of the device and does not need frequent input of information. Since the security preferences vary from different components like user behavior, interests, a profession the market needs a security measure that is specific yet applicable to every person. Already existing options do not cover a wide range but focus on specific users only.

This study will propose a method to identify the user according to their behavioral patterns. Research is conducted on four main components: Keystroke dynamics, voice recognition, application usage, and Geolocation. The user is identified according to his/her previous behavior. The owner is able to give weight to each of these components according to their convenience. The solution that is proposed is user-specific and requires minimum user interactions. The main purpose of the final outcome is to detect an intruder even after the initial Authentication. Further, prevention mechanisms are implemented to minimize the effect of possible theft.

## 2 Background

There have been no current researches covering all the functionalities in this research. Most of the background information was collected by several papers, journals, articles, and online resources according to each component as well as the application of machine learning on mobile devices and other technologies.

Application of Machine Learning Algorithms to Keystroke dynamics is a moderately developed research dimension. As such, the use of Machine Learning can improve the accuracy and execution of authentication [5]. In the paper presented by Kang and Cho, it is declared by altering the edge of the test datasets the error rates were accomplished. By using four novelty algorithms they characterize the legitimate user's keystroke data rather than finding a decision limit between an owner and the imposter [5].

A voice recognition system was introduced in a paper presented by Ganesh K. Venayagamoorthy et al. Voice recognition using artificial neural networks which have given an acceptable level of success mainly with the self-organizing neural networks [8]. However, the application was developed for a desktop application.

There are comparatively much researches done for tracing the location of a mobile device. An application developed by Bhuvana Sekar et al. is able to increase the security of the device as well as the owner's safety [7]. The paper presents a solution to track a device in case of theft by the location detection system.

Another main part of this project is to select a suitable database that is compatible in the mobile environment and that can store all the data extracted from the keystroke,

app usage, and geolocation and voice recognition components. The most significant limitation of establishing a database in a mobile device is the memory and the source of energy [4]. The paper presented by T. Farzad et al. provides a wide comparison between four lightweight databases; SQLite (v3.6.18), Db4object (v7.1), H2 (v1.1) [4]. The comparison is done according to the ACID properties (Atomicity, consistency preservation, Isolation, Durability) and continues to platforms, interfaces, Full Unicode support, Footprints and boundary Limitations. This paper provided a ground for choosing a suitable database for the project.

A paper presented by Harrison John Bhatti discusses the pros and cons of embedding cloud technology regarding cloud databases, cloud computing, and databases. Further, it provides examples such as On-Demand Self-Service, Broad Network Access and rapid elasticity, etc. as the main advantages of using cloud storage [3]. The paper provides a detailed analysis of currently available cloud databases. It states StormDB is quick, adaptable and can be used along with any given programming dialect. MySQL is another open-source social database that is hearty, multi-strung, value-based DBMS [3]. PostgreSQL uses distributed computing to rearrange the procedure of provisioning numerous Postgres arrangements [3]. More suitable for little to medium-sized applications, Google Cloud SQL is stated as easy to utilize and doesn't need any product establishment or support [3].

Further background study was done to find information on the applicability of machine learning models on the android platform. P. Basavaraju and Aparna S. Varde paper on Supervised Learning Techniques in mobile Apps paper provided a comprehensive review of useful approaches and describe the application in current using mobile systems [2]. The decision tree method can be used in both classification and regression problems and is stated as an efficient nonparametric method. Training vectors are divided into two different classes by building decision planes or hyperplanes to classify the Support Vector Machines. Artificial Neural Networks provide a general, practical method for vector-valued, discrete-valued and learning real-valued functions.

The background study of the encryption portion for the research was done through cryptographic researches on the mobile platform. Sujithra, M. et al. have proposed a hybrid approach of encrypting the data in three-tier security using cloud architecture [6]. First-tier indicates MD5 encryption using a given key to the user followed by the second tier of symmetric encryption using AES. The last step is asymmetric encryption using RSA or ECC [6]. This method can be used in both local and remote environments. Since this research was conducted focusing on the remote environment this paper provided a basic idea.

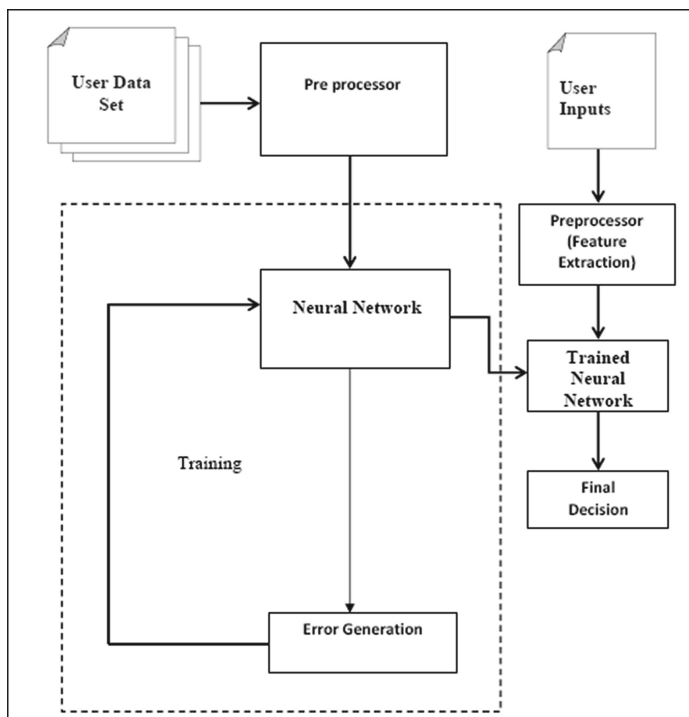
Ahirrao, S.A. et al. have presented an Android-based surveillance System that was informative regarding the prevention mechanisms component [1]. The authors state that a mobile camera capture can be shared through a Computer from a remote location [1].

### 3 Methodology

According to previous studies, there were four different machine learning models, 500–1000 samples of data were gathered for training purposes.

### 3.1 Keystroke Dynamics

#### 3.1.1 Overall Architecture



**Fig. 1.** Overall architecture

With the Change of mobile phone, typing behavior varies according to the typing environment. Therefore, a customized keyboard is developed to collect data and so the research is conducted in a controlled environment. The keystroke Process consists of two separate phases named the enrollment phase and the Identification phase (see Fig. 1). The enrollment phase is in charge of training the algorithm utilized and it is active in the underlying time frame as it were. Identification phase responsible for foreseeing the legitimacy of the user and is active only after the network is trained and is dynamic thereafter. Also with every single right and legitimate input the algorithm gets improved.

#### 3.1.2 Components of the Keystroke Dynamics

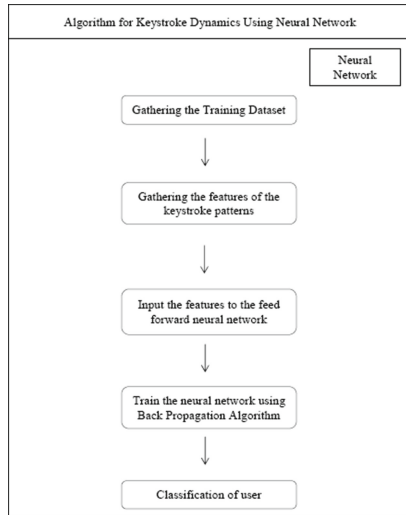
*Data Collector Module.* In this phase the features i.e., Finger touch, Touch Pressure, Dwell time, Flight time and Digraph are extracted to feed the neural network (see Table 1).

**Table 1.** Results (training inputs)

Password	Dwell time	Flight time	Finger pressure	Coordinates		Digraph time	Tri-graph time	Finger size
				X	Y			
Button 01	0.0018	0.001466	1.0	0.0390	0.00576	NA	NA	0.0262
Button 02	0.0015	0.002	1.0	0.0435	0.0571	0.00485	NA	0.0278
Button 03	0.0014	0.0025	1.0	0.0492	0.0580	0.005	0.0083	0.0142
Button 04	0.0014	0.00085	1.0	0.0499	0.0600	0.0054	0.0090	0.0207
Button 05	0.00103	0.0023	1.0	0.0552	0.0571	0.0033	0.0073	0.0241

*Pressure Analyzer Module.* For the extraction of this feature, we have utilized the inbuilt pressure analyzer segment in the android phone to recognize the pressure, the sensor of the touch screen to distinguish the co-ordinates of touch and size of finger contact, milliseconds exactness timer module to precisely recognize distinctive timing characteristics. These extracted features are additionally normalized utilizing unit-based normalization technique so as to get them the scope of [0, 1].

*Keystroke Recognition Module.* Preparation Phase.



**Fig. 2.** Preparation phase

As shown above Fig. 2, during the preparation stage, the preparation data-set is sustained as a contribution to the Feed Forward Network alongside the ideal output values. The Error generator module produces the error and proliferates the error in reverse to alter the weight vectors. After the total preparing stage, the weights are balanced as the network gets completely trained.

Testing Phase. During the testing stage, the contributions from the user are nourished into the network. The network as indicated by its settled weights and contributions from the user, characterizes the user as genuine or masquerader. The whole testing procedure is viably condensed in the figure (Fig. 3). On each legitimate attempt, the network is again trained with that arrangement of qualities, therefore adjusting to changing typing rhythm or pattern through its life-cycle.

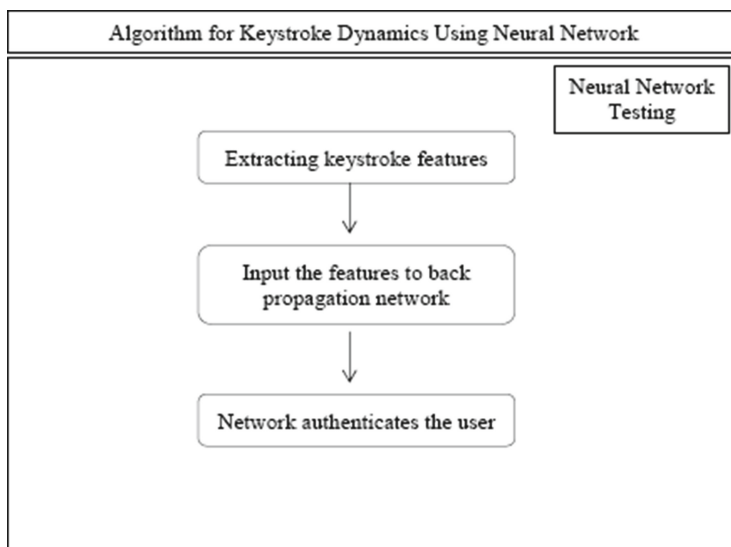


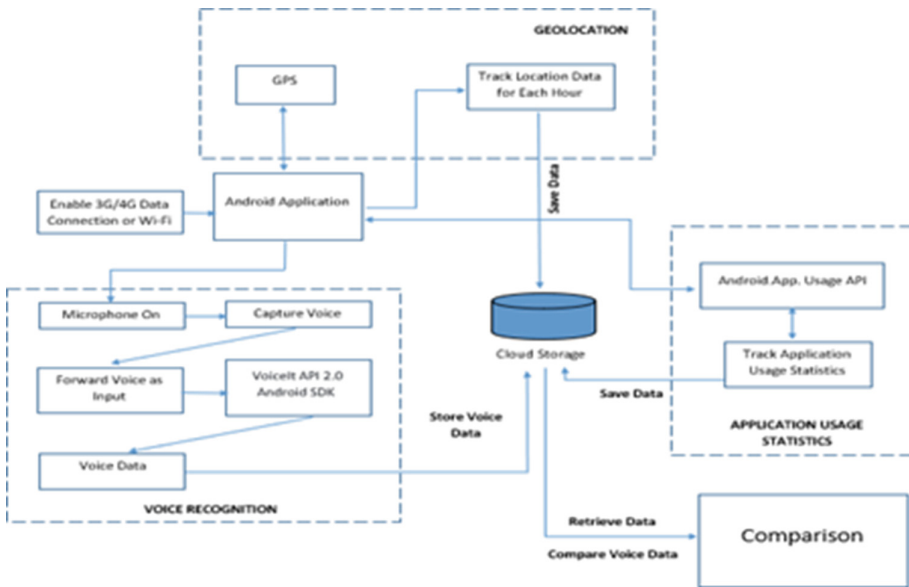
Fig. 3. Testing phase

Pattern Matching. We have utilized the Error Back Propagation Training Algorithm to prepare the neural network. This algorithm makes a neural network with three layers - the input layer, hidden layer, and output layer. The input layer comprises of 45 input hubs which relate to the 45 trademark features unique to a user. One hidden layer with 16 hidden nodes has been taken which gives the best efficiency. Two output nodes correspond to the two classes to which classification is done i.e. Legitimate, Illegitimate.

### 3.2 Identification of User Behavior with the Aid of Geolocation, App Usage Statistics and Voice Recognition

In order to obtain the behavioral data of the user, the behavioral patterns such as geolocation, app usage statistics and voice data of the user are examined, and the data was gathered appropriately. For the purpose of data gathering, the aforementioned behavioral patterns were integrated into a single application and were set to run in the background to retrieve the behavioral data. Once the necessary data were collected and stored in the central database, it was utilized to create the user profile accordingly.

Below attached Fig. 4 delineates the architecture of data gathering, storage, and comparison.



**Fig. 4.** System architecture for geolocation, app usage statistics, and voice recognition systems

Initially, separate systems were implemented for gathering geolocation data, app usage statistics data and voice data. Once each and every single system was developed and tested, they were integrated together.

System to retrieve the geolocation data was implemented in a way such that, it requires user’s permission to access the location data during the training period. If the user is willing to share his/her location details the system captures the latitude and longitude values according to the current location of the user. This process was set to run in the background for every hour. With the assistance of “LocationActivity” class, with respect to the GPS provider status, latitude longitude values were captured under the method “LocationService”. Furthermore, the locality address value was also captured from the above-mentioned method. Once the latitude, longitude and locality address values were captured and assigned to the respective variables, it was forwarded to the central database which has been hosted in cloud storage.

Statistics of the application usage was gathered by running a service in the background in order to obtain the package details of the services running in the foreground. Then, the details were listed in a graph by presenting each individual application’s usage with the time of use, the number of counts and the percentage of use. During the time of the initial setup, the user needs to enable the service to run in the background. Thereafter, the percentage of usage with respect to each application was set to be retrieved at the end of each day. Moreover, the application was developed in the way in which the user can view the usage of the application for the present day as well as for the previous day.

The voice recognition system was implemented with the assistance of “VoiceIt API 2.0 Android SDK”. During the initial setup, an API key and a token were generated. Once the API key and the token were generated, a reference to the SDK inside an activity was

initialized by passing the API credentials or the token. Then the encapsulation methods were used to enroll and verify the authentic user's voice. The voice recognition process consists of two segments namely "Register Activity" and "Login Activity". During the process of the registration, the user is required to provide a user name, first name and last name along with the voice. In order to capture the voice pattern and the frequency of the voice, the user must repeat a hardcoded phrase three times which will be displayed on the mobile screen during the run time. Once the voice is captured it will be enrolled with the user name. The above-mentioned process will occur during the training period.

### 3.3 Collecting, Analyzing and Concluding Data in the Central Database

The data that is sent from the keystroke, Geo-Location and Application usage is stored in the centralized database in the training period (see Fig. 5).

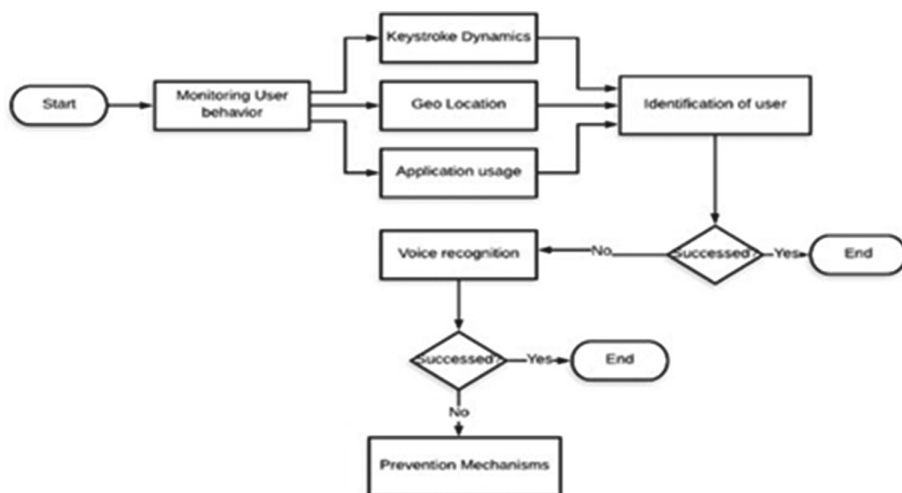


Fig. 5. Main architecture

**Database.** The data gathered from all four components tend to need to take storage space from the device. It was decided to use a cloud facility to store this data. "ThingSpeak" was chosen by the API to store and retrieve data using the HTTP protocol. The keystroke dynamics component provides an accuracy rate while the geolocation and app usage components send raw data that should be further examined. Further, the cloud will be used as a developing platform as well.

**Geo Location.** For geolocation data, since the data is provided according to the user the machine learning process will adopt the supervise learning on a Classification problem. The features of this classification are the above main data according to customer behavior. K nearest neighbor is chosen as the algorithm to detect the location anomalies compared to the training period locations. The data will be gathered in the initial training phase.

When the user is in an unfamiliar location the app will detect the location details and feed it into the model and will check whether there are nearest training period locations with a 5 km radius. If not, it will be count as an anomaly.

**App Usage.** Application usage will be measured by providing the user to choose four applications that he would frequently use. The average usage of these applications will be measured in the initial stage. After the training period, the application usage of the past 24 h will be measured and compared with the present usage. If the user changes within the usual range, it would not be counted as an anomaly. But a drastic difference in usage will be detected.

**Final Accuracy Rate and the Overall Process.** All three accuracy rates of geolocation, application usage, and keystroke dynamics will be calculated with the user assigned weights in the initial stage. The final result will depend on these weights and if the final accuracy rate is less than 75% Voice recognition mechanism will be triggered.

Here, if the user was identified as an intruder, it will be redirected to the login interface of the voice recognition application, where the particular intruder's voice will be examined. During the login step, if the voice data do not match and if the system concludes that the user as an intruder, code segments related to the prevention mechanism will be executed.

### 3.4 Prevention Mechanism and Data Protection

The Prevention mechanism is the way to protect the data inside the mobile phone when it finds out that there is unauthorized access. The main input data to the prevention mechanism is the alert that is coming from the voice authentication parameter. When the illegitimate user alert is triggered it automatically executes the prevention mechanism. This executes in two main methods (see Fig. 6):

1. A photo of the unauthorized person will be captured and sent to the cloud database along with the location.
2. The files in the phone will be encrypted using blowfish encryption method.

By these methods, the user can identify the location and the image of the person who has the mobile device. The input for the first step would be the captured photo and the location of the device. It will be sent to a predefined cloud located profile of the particular user or in this purpose mainly an e-mail address which is defined by the user during registration (in the training period). The second step ensures that the intruder won't take anything from the device and in order to achieve that feature, it encrypts the files inside a mobile device.

The capturing of a photo and location details to the cloud is coded by a Java-based platform. The technical input for this part is taken by referring to several webcams, phone lock, and encryption applications.



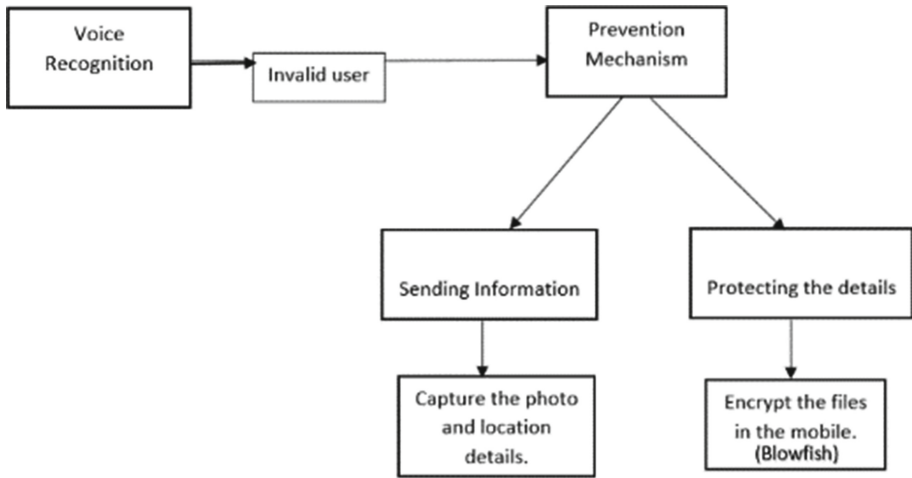


Fig. 6. Prevention mechanism.

### 4 Results and Discussion

The outcomes introduced in this fragment are the consequence of a user trial of a “199510” secret code, with more than 10 accentuations each. The fundamental arrangement of results was extracted during the preliminary stage, which demonstrates the features and timestamps of a user on the different significant focuses that can be utilized to recognize a legitimate user from masquerades (see Fig. 7 and Fig. 8).

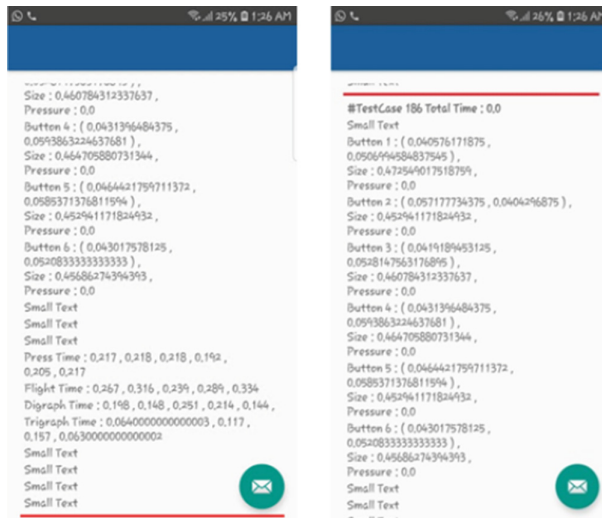
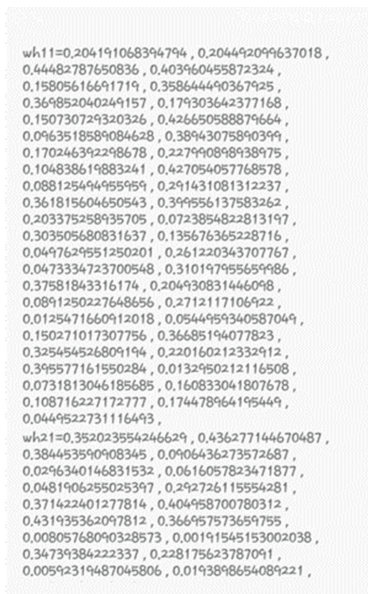


Fig. 7. Feature extraction of single passcode typing



**Fig. 8.** Weights assigned by Back propagation algorithm with respect to the model training

Furthermore, capturing user behavior such as geolocation, app usage statistics and voice recognition are working efficiently by providing error-free outputs. The application was working 95% accurately without any defects.

30 rows of datasets were acquired for the 30 day training period. After calculating the average use of each app the daily update of app usage is compared with the training period to gain the results. If the daily acquired data is not in the range calculated in the training period the output results will be 1 and will be considered as an anomaly.

Geo-location data was captured through the training period for the future use of the k nearest algorithm. Every half an hour user’s location was captured and fed into the algorithm to find anomalies which will result in 1.

As displayed in Table 2, these results will be calculated with weights assigned and the final result on the first layer is given as a percentage. If the percentage is more than 20% the voice recognition will be activated.

The mic will detect the user’s voice and confirm the user. If the user’s voice proven to be not from the device owner prevention mechanisms will be activated. The following table (see Table 2) provides a sample of test cases for the overall system.

Additionally, in the prevention mechanism, the capturing the photo of the intruder by automatically opening the front camera and exchanging of the encryption of the selected files are working at 90% accuracy, nevertheless, it was implicit that some parts of the research are hard to implement due to the fact that developers don’t have kernel access or ability to execute system commands. However, the App is introduced as an inbuilt app. But in implementation there were few limitations in the automation of the processes, enabling some features due to previously mentioned fact. Apart from these technical limitation, there were some theoretical changes that had to be considered while

**Table 2.** Sample of the test cases used

Test no.	Geo location	App usage	Keystroke	Weight of travelling %	Weight of app usage %	Layer 1 final results %	Voice recognition status	Voice recognition results	Prevention mechanism status
01	1	0	0	70	80	23.33	Activated	Positive	Unaffected
02	1	1	1	70	80	83.33	Activated	Negative	Activated
03	1	1	0	70	80	50	Activated	Negative	Activated
04	0	0	1	70	80	33.33	Activated	Positive	Unaffected
05	0	0	0	70	80	0	Unaffected	–	–

**Table 3.** Final statistics of the results

Type	Test cases	Results	True positive rate	True negative rate
Legitimate user	150	TN: 142 FP: 8	–	94.66%
Masquerader	150	TN: 137 FN: 8	91%	–

implementing the prevention mechanism for example due to the time the encrypting the whole phone was changed into encrypting only selected files (otherwise it takes a lot of time to encrypt the whole phone and that is not a good characteristic of a prevention mechanism) and secondly, uploading everything in the phone to the cloud has changed into sending the photo and location to the profile or email, due to the assumption that the data (internet package) of the mobile will not be enough to upload every single file into cloud and there is a limit to the cloud storage space. Therefore with this new features (encrypting only selected files and emailing the photo of the intruder to the valid user along with the location), in the training period there will be interface in the app to choose files that needs to be encrypted in case of emergency and to register the user's email address. During the testing period the encryption algorithm was changed from AES-128 to Blowfish in order to speed up the encryption process. Apart from the above mentioned limitations the implementation and processing of the prevention mechanism is a 90% success as shown in the Table 3.

## 5 Conclusion and Future Work

AuthDNA app protects your data and helps you to find the intruder or the thief. Additionally, it provides extra security for your mobile device. This mechanism can be further developed into other devices such as a computer, tablet, etc. where the authentication method is limited to only biometrics. Furthermore, this research is only done for an android based mobile device, the researches can give their attention to other platforms

as well (such as IOS). The researchers can increase the accuracy rates with novel technologies and trends and that would also be an interesting path for a research study since this is beneficial for every mobile user these days as well as in the future.

## References

1. Ahirrao, S.A., Ballal, S.S., Sawant, D.K.: Android based remote surveillance system and content sharing between PC and mobile. *Int. J. Comput. Appl. Technol. Res.* **4**, 153–156 (2015). <http://ijcat.com/archives/volume4/issue2/ijcatr04021013.pdf>
2. Basavaraju, P., Varde, A.S.: Supervised learning techniques in mobile device apps for androids. In: *School on Machine Learning for Data Mining and Search, SIGIR/SIGKDD. ACM SIGKDD Explorations Newsletter* (March 2017). <https://dl.acm.org/doi/10.1145/3068777.3068782>
3. Bhatti, H.J., Rad, B.B.: Databases in cloud computing. *Int. J. Inf. Technol. Comput. Sci.* **9**(4), 9–17 (2017). [https://www.researchgate.net/publication/315993485\\_Databases\\_in\\_Cloud\\_Computing](https://www.researchgate.net/publication/315993485_Databases_in_Cloud_Computing)
4. Farzad, T., Azam, A., Asadollah, S., Reza, E.A.: A comparison of lightweight databases in mobile systems. *J. Comput.* **3**(7), 147–152 (2011). [https://www.researchgate.net/publication/236969019\\_A\\_Comparison\\_of\\_Lightweight\\_Databases\\_in\\_Mobile\\_Systems](https://www.researchgate.net/publication/236969019_A_Comparison_of_Lightweight_Databases_in_Mobile_Systems)
5. Kim, J., Kim, H., Kang, P.: Keystroke dynamics-based user authentication using freely typed text based on user-adaptive feature extraction and novelty detection. *Appl. Soft Comput. J.* (2018). <https://doi.org/10.1016/j.asoc.2017.09.045>
6. Sujithra, M., Padmavathi, G., Narayanan, S.: Mobile device data security: a cryptographic approach by outsourcing mobile data to cloud. *Procedia Comput. Sci.* **47**, 480–485 (2015)
7. Sekar, B., Liu, J.B.: Location based mobile apps development on android platform. *IEEE* (2014). <https://ieeexplore.ieee.org/document/6931527>
8. Venayagamoorthy, G.K., Moonasar, V., Sandrasegaran, K.: Voice recognition using neural networks. In: *South African Symposium on Communications and Signal Processing*, 8 September, 1998. IEEE, Rondebosch (1998). <https://ieeexplore.ieee.org/document/736916>



# Spoofer/Unintentional Fingerprint Detection Using Behavioral Biometric Features

Ammar S. Salman<sup>1</sup>(✉) and Odai S. Salman<sup>2</sup>

<sup>1</sup> Syracuse University, Syracuse, NY 13244, USA  
assalman@syr.edu

<sup>2</sup> Carleton University, Ottawa, ON K1S-5B6, Canada  
odaisalman@cmail.carleton.ca

**Abstract.** Fingerprints are common biometrics in smartphones as they are used for access to the device itself, or for authentication in applications. While fingerprints provide many benefits, they are vulnerable to spoofing attacks. This paper investigates countermeasures to spoofing attacks that use live fingerprints without consent either by force or by theft. We used behavioral biometrics to differentiate between intentional and forced fingerprint authorization attempts. Data was collected from several sensors and the most discriminating one was the accelerometer. A total of six data subsets, each with about 100 instances were collected, four for testing and two for calibration. A corresponding six tests were made on the subsets, in addition to one test on the combination of feature vectors from all sensors before and after using Correlation-based Feature Selection (CFS) to reduce the number of combined features. We used Naïve Bayes, Linear-Kernel and Cubic-Kernel Support Vector Machines (SVMs), and Deep Neural Network (DNN) classifiers. For the accelerometer-combined data, the classifiers scored 61%, 81%, 88% and 94%, respectively showing the DNN as the most powerful classifier, and for individual runs, performance was higher. The investigation was successful in differentiating between intentional and forced uses of fingerprint authentication systems.

**Keywords:** Fingerprint · Biometrics · Spoofing · Liveness detection · Anti-spoofing protection · Security

## 1 Introduction

Prehistoric picture writing of a hand with ridge patterns was discovered in Nova Scotia. In ancient Babylon, and Egypt, fingerprints were used on clay tablets for business transactions. In ancient China, thumb prints were found on clay seals. In 14th century Persia, various official government papers had fingerprints, and it was observed that no two fingerprints are identical.

After the Industrial Revolution, the French police used anthropometrics to identify subjects. Due to lack of awareness of its potential, it did not see rapid increase until late 20<sup>th</sup> century with the fast development of computer systems. Smartphones now have fingerprint authentication systems.

Fingerprint authentication in mobile devices provides fast access compared to PIN codes or passwords; user identification; and data protection. A major drawback is the vulnerability to spoofing [1].

There are two main methods of spoofing fingerprints: 1) using fake fingerprints and 2) using valid fingerprints without consent. Most research focuses on the former. However, in daily use, a person's own fingerprint may be used to gain unauthorized access. This can happen during sleep, or by force.

Multiple biometrics can be used to mitigate the risk of compromised biometrics. Some systems require face and fingerprint authentication. Multimodal biometrics can provide new layers of authentication, but this can make it impractical, as users have to identify multiple times.

We present motivation and objectives in Sect. 2; related work in Sect. 3; methodology, data collection procedure, and features selection in Sect. 4; results and analysis in Sect. 5; and finally, the conclusion in Sect. 6.

## 2 Motivation and Objectives

Fingerprint spoofing has been a big issue for many years [1] discussed spoofing and countermeasures back in 2002, when fingerprint applications were not utilized in mass public domain. The analysis was limited to immobile fingerprint sensors in official facilities. The paper mentioned the liveness detection and predicted a flourishing research to use it for counter spoofing attacks. True to that prediction, liveness detection is widely researched, and many algorithms contributed.

Liveness detection methods do not cover unintentional spoofing. In the rise of smart-phone fingerprint authorization systems, it is relatively easily to spoof authorized users live fingerprints to gain access to their devices and operations. These kinds of spoofing show the difficulty of developing reliable countermeasures to unauthorized use of fingerprints considering the scope and ease of utilization.

Our research targets unintentional/forced spoofing attacks in smartphone fingerprint authorization systems. The work intends to provide a transparent protection which relies on users' behavior prior to unlocking their devices and aims to work on existing mobile devices. In this paper, we develop a new experimental setup and feature extraction method to enhance spoofed fingerprint detection to discriminate between intentional and not intentional or forced authenticated and valid fingerprints.

## 3 Related Work

Schuckers [1] reviews spoofing attacks that target fingerprint applications at the sensor level, and proposes countermeasures. The work discusses scenarios where users are compelled to identify (or verify) themselves using fingerprints. He suggests that there is no biometric countermeasure for these attacks. The article discusses gummy fingerprints, or fooling the sensors' nature of work. Temperature-based sensors can be breathed upon, and optical sensors can be dusted and re-focused with intense halogen light.

The article introduces a new method of spoofing that relies on dental impression materials and casts made from clay and Play-Doh to create molds. The attack was

performed on multiple fingerprint scanners including capacitive (AC and DC), optical, and opto-electronic based scanners, yielding varied success rates.

Furthermore, the author tested cadaver fingers by enrolling dismembered fingerprints onto the scanners. Success rates varied from 40–94% depending on the scanner. Possible anti-spoofing techniques include additional password protection, enrolling several samples per person, supervising the identification/verification process, multi-modal biometric approaches, and liveness detection.

Yuan et al. [2] targets liveness detection in fingerprint images using Convolutional Neural Network (CNN). The tests were performed using LivDet competition datasets from the years 2009 and 2011. The error rate was significantly lower than the winners of the two mentioned competitions, and other methods developed later. The novel idea is the reliance on the CNN to form semantic features from fingerprint images, which can be used to discriminate between fake and real fingerprints. The authors also discuss the required preprocessing of fingerprint images using ROI. Their reasoning is that empty areas of images can cause issues with classification. Furthermore, they have used PCA in pooling layers to reduce over-fitting and the number of relevant features.

Marcialis et al. [3] reviews fingerprint recognition systems vulnerability to spoof attacks, like molds made of silicone, gelatin or Play-Doh. Liveness detection, of the vitality information from the biometric signature itself, was proposed to defeat these spoof attacks. LivDet 2009 competition compared different methodologies for software-based fingerprint liveness detection with a common experimental protocol and large dataset of spoof and live images. Four submissions resulted in successful completion: Dermalog, ATVS, and two anonymous participants (one industrial and one academic). Each participant submitted an algorithm as a Win32 console application. The performance was evaluated for three datasets, from three different optical scanners, each with over 1500 images of “fake” and over 1500 images of “live” fingerprints. The best results were from the algorithm submitted by Dermalog with a performance of 2.7% FRR and 2.8% FAR for the Identix (L-1) dataset.

## 4 Methodology

### 4.1 Solution

Maintaining convenience, and compatibility requires utilization of existing smartphones components for the countermeasures. We have investigated sensors in smartphones to build user-specific profiles for the counter-spoof. Data was collected from nine different sensors as described in the data section. Two types of sensors were used; sensors that measure the forces acting on the system including the touch force detected mainly by the accelerometer, and with less sensitivity are the gyroscope, gravity, and linear acceleration. The second type are positioning sensors that detect the device orientation, including rotation, proximity, magnetic parameters and pressure. In total, we have used seven of the nine sensors. Pressure data is not significant if operations are local. Proximity should show surroundings, but it was unable to collect data continuously in a signal fashion.

The Accelerometer is highly useful to measure the touch force and duration using instantaneous frames of reference. In essence, it measures the touch strength. Considering that touches are highly personal based, it is unlikely for different people to emulate identical touches, especially if one person is trying to force the owner to make the touch and the same applies if the owner is asleep. The other important sensors are rotation and orientation which can indicate the device orientation.

Data from named sensors is collected prior to device unlocking attempt using fingerprint. If the input fingerprint matches any enrolled template, the attempt is authenticated, and data collection is stopped to store the data. This data can be used to determine if the attempt was forced/unintentional or valid. This, in turn, can be used to reject the attempt if it is not valid, or can be used to investigate fraud operations after the fact.

The collected data is classified into two classes: 1) intentional attempt where the user purposely unlocks the device; and 2) unintentional or forced attempt where the owner is forced to make the unlocking operation.

We have used four classifiers: Naïve Bayes, Linear Kernel Support Vector Machine (SVM), Cubic Kernel SVM and Deep Neural Network. We compared their performances and accuracy levels as discussed in Sect. 5. We have also used Correlation-based Feature Selection (CFS) to reduce the number of features. WEKA [4] was used to perform CFS and all classification problems. Other works compared the three classifiers for a variety of applications [5–7].

## 4.2 Constraints

The solution to forced/unintentional spoofing should remain:

1. Convenient to the users. Otherwise, they may disable the application.
2. Transparent and simple.
3. Compatible with existing smartphones, because adding a new hardware is not a good option.
4. Lightweight performance as mobile devices have limited power usage.

## 4.3 Sensors and Data Collection

We have developed an application using Android Operating System (OS) to collect data from the sensors [16]. It can be extended to serve on other operating systems. The following sensors were utilized.

1. Accelerometer: measures acceleration w.r.t an instantaneous rest frame, including gravity.
2. Gyroscope: measures orientation and angular velocity.
3. Gravity: measures gravity force in all directions.
4. Linear acceleration: measures the force applies to the device in all directions excluding gravity.
5. Magnetic field: measures the ambient geomagnetic field for all three physical axes.
6. Orientation: measures degrees of rotation that a device makes around all three physical axes.



7. Rotation vector: measures device orientation by providing the three elements of the rotation vector.

Our application starts recording the data when attempting to unlock the device and stops when unlocked. The attempt type is marked in relation to the person, and the nature of the unlocking attempt (intentional or unintentional/forced).

#### 4.4 Correlation-Based Feature Selection (CFS)

A relevant work by Mark Hall [8] reviews identifying a representative set of features from which to construct a classification model. He addresses the problem of feature selection through a correlation-based approach. The central hypothesis is that good features are highly correlated with the class, and preferably uncorrelated with each other. An evaluation formula, based on test theory, provides a definition of this hypothesis. CFS algorithm couples this evaluation formula with an appropriate correlation measure, and a heuristic search strategy. CFS was evaluated by experiments on artificial and natural datasets.

#### 4.5 Classification Algorithms

**Naïve Bayes.** NB is a family of probabilistic classifiers that use Bayes' theorem with strong (naive) independence assumptions between features. It has been studied since the 1950s and was introduced under the name text retrieval with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with support vector machines. It is also useful in medical auto diagnosis. NB classifiers are highly scalable, requiring several parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression which takes linear time. NB models are known under a variety of names, including simple and independence Bayes. Rish [9] reviews the performance of the NB classifiers, assuming features independence in a given class.

**Support Vector Machines.** SVM is a supervised machine learning algorithm which can be used for classification problems [10]. In this algorithm, we plot each data item as a point in  $n$ -dimensional space, where  $n$  is the number of features, with each feature representing a particular coordinate. Figure 1 shows a simplified SVM visualization of two clearly separated classes between features  $F_1$  and  $F_2$ . In our work we have features well over two making the task in hand a bit more difficult for the classifier.

**Deep Neural Network.** DNNs are normally preferred for multiclass classification problems. Due to the large number of features extracted from all sensors as DNNs, tend to perform well on large sets of features when there is enough data. We used MultiLayer Perceptron (MLP) [11, 12] with a learning rate of 0.2, momentum of 0.2, and a batch size of one. Figure 2 shows a schematic DNN structure similar to the one we used in this work. WEKA's default implementation doubles the number of features for each hidden layer [13]. It's worth noting that the DNN used for the final test combined features from all sensors, but the structure follows the same rules.

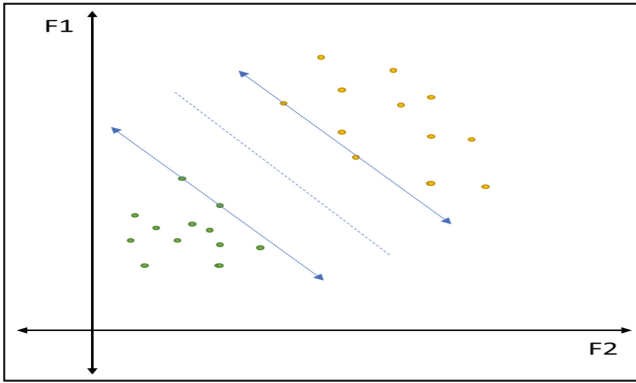


Fig. 1. Hyper-plane that clearly separates the two classes

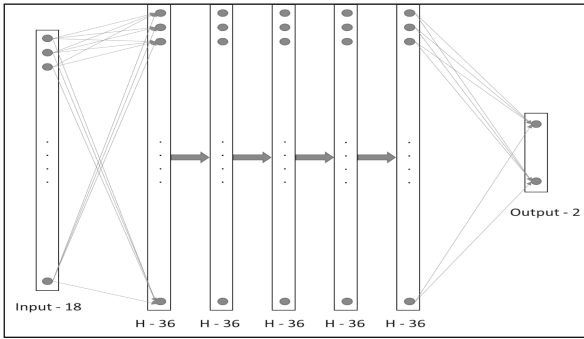


Fig. 2. DNN structure. Input layer consists of 18 neurons for the 18 features. Each hidden layer consists of 36 neurons. Finally, the output layer consists of 2 neurons for the two classes.

## 5 Results and Analysis

### 5.1 Datasets and Feature Selection

The data sets represent variable real time conditions as explained in the Table 1. Six subsets with nearly 100 instances each represent a reasonable range of values and statistical samples for training and testing the classifiers.

The application we developed runs on Android devices and collects the sensors data, when a finger print push is made on the scanner. The data is collected over the time span when the system is processing the finger print authenticity. It usually takes a few seconds to collect one instance.

The data for each subset was labeled 50% *intentional* representing authenticated instances (class = 1), and the other 50% representing *forced* or unauthorized (class = 0). The main measurable difference is due to the force magnitude and direction applied to the scanner.

**Table 1.** Biometric dataset: runs codes and settings. Total number instances from runs 1–4 is 426, with 18 features per sensor. Total number of features is 126. Class = 1 intentional; class = 0 forced. Pressing force ratio  $F01 = F0/F1$ ;  $F_x =$  force for class  $x$ . Composition for all runs is 50% class = 0, and 50% class = 1.

Run code	Type	F0/F1	Configuration
R1	Classification	2.5/1	CP1: two different fixed pushes for the two classes to optimize contrast between the two cases for most sensors
R2	Reproduction of R1	2.5/1	CP1
R3	classification	2/1	CP2: two different pushes for the two classes; the push changes slightly during the application for each class The push ratio is slightly less than CP1
R4	Classification	1.5/1	CP3: two different pushes for the two classes. Push changes during the application for each class. The push ratio is a little less than the R3
R01	Calibration	1/1	CP4: Push and positioning are fixed for all instances but labeling 50% class = 1 and 50% class = 0
R02	Calibration	2/1	CP5 slightly different positioning and different pushes for the two classes

## 5.2 Feature Selection

We have used A rich Frequency and Amplitude based Series Timed signals with 18 features extraction Algorithm (FAST18) developed by one of the authors [14]. For each sensor's data, the feature vector consists of 18 total features. One is the total signal time in milliseconds, the second is the root mean square deviation of each reading time within the whole sensor signal, and the third is the position angle. In addition, there are

**Table 2.** Comprehensive list of features.

No.	Feature name	No.	Feature name
1	signalTime	10	oscPerRelAngleSign
2	rmsdPSignal	11	rmsdPosVal-X
3	oscPerSignal-X	12	rmsdPosVal-Y
4	oscPerSignal-Y	13	rmsdPosVal-Z
5	oscPerSignal-Z	14	rmsdRelPosVal-X
6	oscPerRelSignal-X	15	rmsdRelPosVal-Y
7	oscPerRelSignal-Y	16	rmsdRelPosVal-Z
8	oscPerRelSignal-Z	17	rmsdAngVal
9	oscPerAngleSign	18	rmsdRelAngVal

three fluctuations per signal in all three dimensions, and three for their time components. Furthermore, we have three root mean square deviations for three dimensional sensor readings, and finally, their three time components. Table 2 shows these features. For the combined sensors we have established 126 component features vector that was reduced after applying the CFS to 16 powerful features. Figure 3 shows the reduced set of combined features. The combined selections have improved the rate for all classifiers except the naïve bays which is expected.

### 5.3 Classifications

Table 3 shows the results for the various runs under Naive-Bayes Classification. The success rate is high for the standard runs with a minimum number of oscillating variables within each setting, and they are expected to get the best results. The two calibration runs show as expected.

With R01 the data is labeled as different while in fact it is identical, the expected success-rate should be a random walk problem yielding 50% rate. For the R02 the sets are actually different and just similar to the R1 but with only one difference that the CP5 is used, meaning the orientations did not have a large difference as the acceleration, and

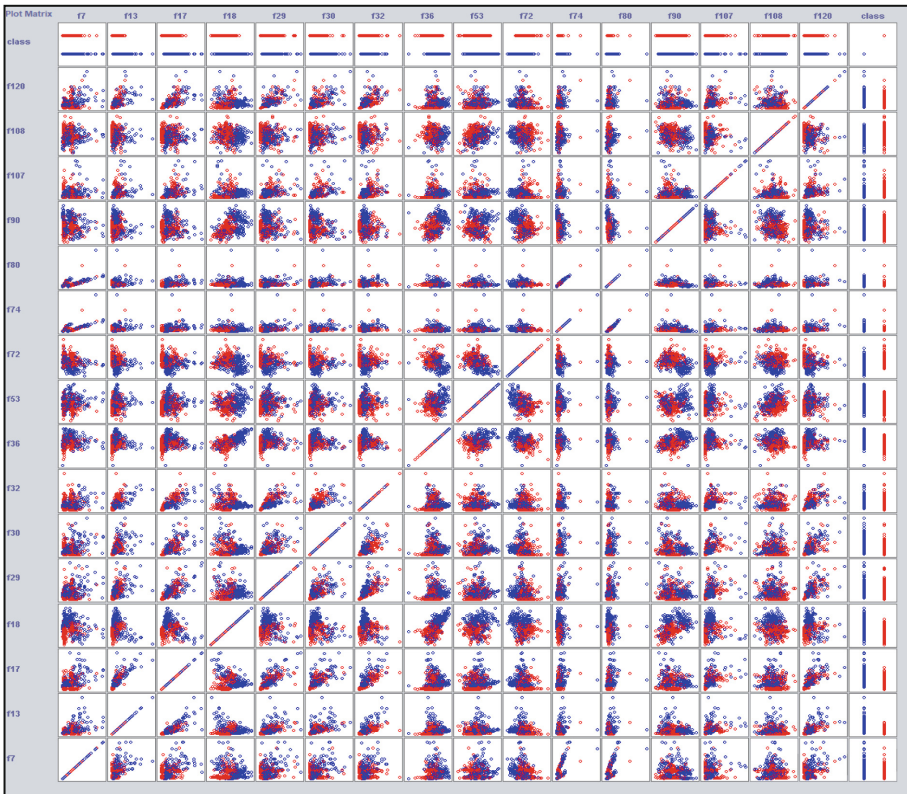


Fig. 3. Correlation matrix between the 16 selected power features, and the class, using CFS

that would yield a lower success for position classification. The two main runs R3–R4 yield an almost real time data with variation as it would be anticipated from an owner versus an outsider push and timing. The results show a little lower success, but not that much, because the Accelerometer is still the main contributor. The other position sensors also give lower success rates.

Table 4 shows the result for the Linear Kernel SVM classifier, and the success is better as expected. The SVM can benefit from correlated features, and hence should generally give better results.

Table 5 shows the data for the Cubic Kernel SVM classifier, and that also shows some improvements albeit small. Table 6 shows the DNN results and again the results are as expected. Finally, Table 7 shows the results obtained after concatenating all sensors feature vectors into one, then training the system on overall with 10-CV test, without and with CFS.

**Table 3.** Naive-Bayes classification accuracy percentage, for the various sensor measurements binary classification intentional/forced

Function	R1	R2	R3	R4	All	R01	R02
Accelerometer	97.06	97.03	92.68	81.00	60.80	44.00	97.00
Gravity	97.06	97.03	64.23	77.00	66.20	38.00	90.00
Gyroscope	95.10	98.02	65.85	77.00	61.97	58.00	72.00
Linear Acc.	96.08	93.07	67.48	78.00	63.85	38.00	68.00
Magnetic	95.10	97.03	69.92	79.00	63.85	50.00	64.00
Orientation	95.10	97.03	68.29	79.00	62.21	56.00	80.00
Rotation	96.08	96.04	63.41	76.00	60.56	46.00	79.00

**Table 4.** Linear Kernel SVM classification accuracy prediction %, for the various sensor measurements binary classification intentional/forced

Function	R1	R2	R3	R4	All	R01	R02
Accelerometer	97.06	97.03	98.37	85.00	81.00	36.00	97.00
Gravity	97.06	98.02	85.37	84.00	77.93	42.00	90.00
Gyroscope	97.06	98.02	81.30	77.00	68.54	42.00	88.00
Linear Acc.	97.06	96.04	84.55	81.00	71.60	38.00	80.00
Magnetic	98.04	96.04	89.43	86.00	77.00	46.00	64.00
Orientation	96.08	98.02	80.49	78.00	77.23	42.00	83.00
Rotation	97.06	98.02	93.50	74.00	71.13	50.00	82.00

Feature reduction helps Naïve Bayes classifier because it selects features which are the most independent, while it harms the DNN, and SVM classifiers, because they lose

**Table 5.** Cubic Kernel SVM classification accuracy prediction percentage, for the various sensor measurements binary classification intentional/forced

Function	R1	R2	R3	R4	All	R01	R02
Accelerometer	97.06	97.03	96.75	86.00	88.03	36.00	97.00
Gravity	98.04	97.03	81.30	87.00	79.43	48.00	91.00
Gyroscope	95.10	97.03	78.86	78.00	66.67	54.00	86.00
Linear Acc.	97.06	96.04	91.06	78.00	67.37	46.00	80.00
Magnetic	97.06	97.03	87.80	82.00	75.82	48.00	56.00
Orientation	90.20	97.03	78.05	78.00	71.83	46.00	83.00
Rotation	98.04	98.02	83.74	77.00	67.14	38.00	84.00

**Table 6.** DNN classification accuracy prediction percentage, for the various sensor measurements binary classification intentional/forced

Function	R1	R2	R3	R4	All	R01	R02
Accelerometer	97.06	97.03	97.56	93.00	93.66	38.00	98.00
Gravity	98.04	98.02	87.80	91.00	84.51	46.00	88.00
Gyroscope	97.06	97.03	89.43	81.00	83.80	50.00	84.00
Linear Acc.	98.04	98.02	88.62	76.00	83.80	46.00	78.00
Magnetic	98.04	97.03	95.12	85.00	85.92	54.00	66.00
Orientation	97.06	98.02	89.43	85.00	89.44	44.00	87.00
Rotation	97.06	98.02	93.50	80.00	90.14	38.00	79.00

**Table 7.** Accuracy rate from concatenating all sensors feature vectors into one, then training the system on overall with 10-CV test, with and without CFS

Classifier	Without CSF (126 features)	With CSF (18 features)	Wo/w CFS
DNN	95.07%	92.02%	1.04/0.96
Cubic poly kernel SVM	94.37%	92.72%	1.02/0.98
Linear poly kernel SVM	91.31%	86.62%	1.05/0.95
Naive-Bayes	63.62%	78.64%	0.81/1.24

some correlations in the omitted features. In fact, the table shows they have lost some accuracy after the reduction. In contrast, Naive-Bayes benefited significantly, because it assumes independent features. The results are consistent, and as expected.

## 6 Conclusions

The main point here is not testing the various classifiers, but the feasibility to distinguish between ordinary and unauthorized touches. The results show very promising prospects. Considering that this data is an exercise to construct real life data conditions, a thorough and detailed data collection under various conditions would be pursued.

A successful discrimination between these two types of operations can have a significant impact on the protection of the devices, and owners from brute forceful actions, or stealing while asleep or unaware. A follow-up application can provide online protection needs, with some considerations to meet the boundary conditions of simplicity, and reversibility without harming performance. Even if that task is not achieved with full efficiency, many high risk or vulnerable users would rather use it, to provide a critical protection against high odds. A third good utility is to provide some means, to check the authenticity of the operations after the fact. The service can extend to other applications. The FAST16 extractor worked very well for all classifiers, and reported better results when using CNN with Feature to Image Transformation [FIT] [15].

**Acknowledgments.** This work was thoroughly and critically evaluated; and manuscript corrected by Professor Salman M Salman from Alquds University. We also thank Professors Garrett Katz and Vir Phoha both from Syracuse University for their critical insights and recommendations.

## References

1. Schuckers, S.A.C.: Spoofing and anti-spoofing measures. *Inf. Sec. Tech. Rep.* 7(4), 56–62 (2002). [https://doi.org/10.1016/S1363-4127\(02\)00407-7](https://doi.org/10.1016/S1363-4127(02)00407-7)
2. Yuan, C., Sun, X., Lv, R.: Fingerprint liveness detection based on multi-scale LPQ and PCA. *China Commun.* 13(7), 60–65 (2016). <https://doi.org/10.1109/CC.2016.7559076>
3. Marcialis, G., Lewicke, A., Tan, B., Coli, P., Grimberg, D., Congiu, A., Tidu, A., Roli, F., Schuckers, S.: First international fingerprint liveness detection competition-LivDet 2009. In: *Proceedings of the LNCS International Conference on Image Analysis and Processing (ICIAP), Vietri sul Mare, Italy. Lecture Notes in Computer Science*, vol. 5716 (2009). [https://doi.org/10.1007/978-3-642-04146-4\\_4](https://doi.org/10.1007/978-3-642-04146-4_4)
4. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: *Data Mining: Practical Machine Learning Tools and Techniques*, 4th edn. Morgan Kaufmann Publishers Inc., San Francisco (2016)
5. Collobert, R., Bengio, S.: Links between perceptrons, MLPs and SVMs. In: *Proceedings of the 21 International Conference on Machine Learning, ICML 2004, Banff, Alberta, Canada (2004)*. <https://doi.org/10.1145/1015330.1015415>
6. Sharma, A.K., Prajapat, S.K., Aslam, M.: A comparative study between Naïve Bayes and neural network (MLP) classifier for spam email detection. In: *International Journal of Computer Applications (IJCA), National Seminar on Recent Advances in Wireless Networks and Communications, NWN(2)*, pp. 12–16, (2014)
7. Shi, H., Liu, Y.: Naïve Bayes vs. support vector machine: resilience to missing data. In: *Proceedings of the AICI 3 International Conference, Taiyuan, China. Artificial Intelligence and Computational Intelligence (2001)*. [https://doi.org/10.1007/978-3-642-23887-1\\_86](https://doi.org/10.1007/978-3-642-23887-1_86)
8. Hall, M.A.: Correlation-based feature selection for machine learning. Ph.D. thesis, Department of Computer Science, University of Waikato Hamilton, New Zealand (1999)

9. Rish, I.: An empirical study of the Naive Bayes classifier. T.J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532 (2001)
10. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995). <https://doi.org/10.1007/BF00994018>
11. Schmidhuber, J.: Deep learning in neural networks: an overview. *Neural Netw.* **61**, 85–117 (2015). <https://doi.org/10.1016/j.neunet.2014.09.003>
12. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer, Heidelberg (1995)
13. Frank, E., Mark, A., Hall, M.A., Witten, I.H.: “The WEKA Workbench” Online Appendix for “Data Mining: Practical Machine Learning Tools and Techniques”. Morgan Kaufmann, Burlington (2016)
14. Salman, O., Jary, C.: Frequency and Amplitude based Series Timed signals 18 features extraction Algorithm (FAST18), pattern classification project report, SCE Carleton University, Spring 2018
15. Salman, A.S., Salman, O.S: Extending CNN classification capabilities using a novel Feature to Image Transformation (FIT) algorithm. In: SAI Computing Conference, London (2020)
16. Android Developers: Sensors Overview. [https://developer.android.com/guide/topics/sensors/sensors\\_overview](https://developer.android.com/guide/topics/sensors/sensors_overview)





# Enabling Paratransit and TNC Services with Blockchain Based Smart Contracts

Amari N. Lewis<sup>(✉)</sup> and Amelia C. Regan

Department of Computer Science, University of California Irvine, Irvine, CA, USA  
{amaril,aregan}@uci.edu  
<http://sites.uci.edu/AmariLewis>, <https://faculty.sites.uci.edu/ARegan>

**Abstract.** Paratransit services provide mobility solutions for disabled travelers and older adults. In the United States, the requirements for efficient and affordable provision of these services significantly increased in 1990 with the passage of the Americans with Disabilities Act. These transportation services are essential to the well-being of these populations, however, nearly thirty years later, they remain notoriously expensive to provide and inconvenient for its passengers. The issues related to paratransit are apparent from passenger feedback and complaint forms. In this work, we explore a potential solution for improving paratransit services under consideration by transit agencies around the world, the integration of TNC's (Transportation Network Companies) such as Uber, Lyft, Didi or Grab and taxi services with paratransit. The contribution of this work is to develop privacy preserving secure smart contracts to enable these extended paratransit systems. We examine the use of blockchain and simple IoT devices to host these contracts. Through proof of concept prototype development using open source blockchain resources, we examined the proposed architecture and system design.

**Keywords:** Blockchain · Secure contracts · Paratransit · Internet of Things

## 1 Introduction

In North America we use the term Paratransit for transportation services that complement fixed-route transit services to meet the needs of disabled travelers and older adults. In the United States, the requirements for efficient and affordable provision of these services increased significantly in 1990 with the passage of the Americans with Disabilities Act [1]. Similar legislation governs Canadian services for people with disabilities. These transportation services are essential to the well-being of these populations, but they remain expensive to provide and inconvenient for users. A potential solution for improving paratransit services under consideration by transit agencies around the world, is the integration of ride-hailing services (also known as Transportation Network Companies, or TNCs) such as Uber, Lyft, Didi or Grab and taxi services with paratransit.

This integration is justified, as many paratransit users are mobility or vision impaired but not wheelchair bound, so they do not require special vehicles. And, some wheelchair accessible vans are inconvenient or even unsafe for vision impaired travelers with guide dogs. Enabling these services at the same time as ensuring user privacy, data security and safety will require considerations that are not present in standard ride-hailing and taxi services.

Our work examines ways to develop IoT based blockchain enabled smart contracts to enable these extended paratransit systems. In the U.S., because of HIPAA<sup>1</sup> [2], there are implied regulations which ensure the privacy of the disclosure of patron's personal health data. This established the first national standards in the United States to protect patients' personal or protected health information. The U.S. department of Health and Human Services issued the rule to limit the use and disclosure of sensitive personal or protected health information in 1966. In theory, these systems must be privacy preserving and secure. The use of blockchain methods provides a way to securely create, store, and transfer digital assets in a distributed, decentralized environment. We strongly believe that this is important when considering the incorporation of TNCs and taxis with paratransit operations.

Synchronization in public blockchains typically requires significant computational power and extensive, amounts of storage, which makes their use inefficient or infeasible for memory-limited IoT devices. Thus, in this paper, we examine the use of a private blockchain for a paratransit system. Private blockchains are permission based environments in which only the approved entities are able to access and add blocks. This initial work focuses on the design and development of working prototypes, while the next phase will involve direct input from relevant public agencies.

The remainder of the paper is organized as follows, in the next section we discuss the background and motivation for the research. In Sect. 3, we provide an overview of blockchain in supply chain management. In Sect. 4, we explore the related research that influenced our work. In Sect. 5, we outline the pilot studies from across the U.S. that have incorporated TNCs in their paratransit services. In Sect. 6, we list and define the various cyber-attacks that make IoT systems vulnerable. In Sect. 7, the prototype design is presented. In Sect. 8, the use of smart contracts is described. In Sect. 9 discusses the work and present limitations of the work and lastly, we provide a concluding section.

## 2 Background and Motivation

Through the use of blockchains we are able to achieve a secure method of processing, storing, and maintaining paratransit trip data including its transactions. The considerable vulnerabilities in cyberspace make security an essential feature of data sharing. The Internet of Things (IoT) is a microcosm of interconnected

---

<sup>1</sup> Health Insurance Portability and Accountability Act was established in 1966 in the United States of America, this Act mandated the data privacy and security provisions safeguarding medical information.

devices. There are many common cyber-attacks, but, in IoT environments the risks are increased due to the number of connected devices. Another benefit of blockchain is that it operates in a fully decentralized environment. In a decentralized environment, there is no single point of failure. Rather than relying on a central authority to manage secure transactions, blockchain uses consensus protocols across the entire network of nodes to validate transactions and record data in an incorruptible manner [3].

The differences between centralized and decentralized IoT systems is explored in detail in [4]. Figure 1 presents the differences between centralized and distributed (decentralized) IoT systems. Blockchains and Smart Contracts for the Internet of Things can lead to more distributed applications [5]. Lastly, immutability, which is the ability to remain unchanged, is another major benefit of blockchain.

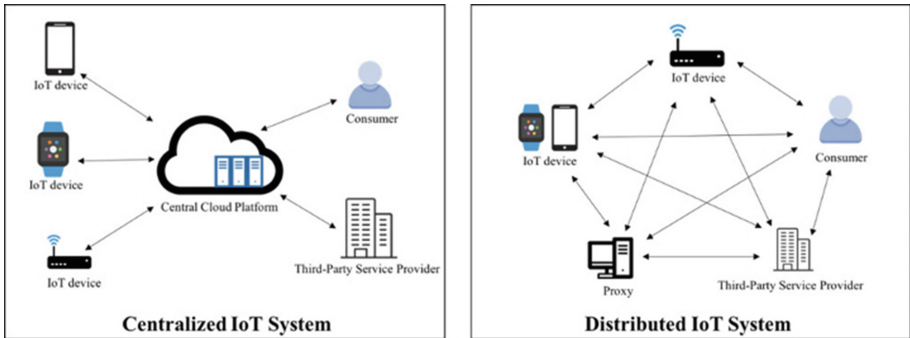


Fig. 1. Comparison of centralized and distributed IoT systems [4]

### 3 Blockchain in Supply Chain Management and Logistics

Blockchain has proven itself useful in many business applications, and has the potential to improve processes and enhance business models in logistics and Supply Chain Management (SCM). In [6] four use cases were addressed as areas of significant opportunities of improvement by incorporating blockchain technologies, including: ease of paperwork processing, identifying counterfeit products, facilitating origin tracking and, the operation of IoT.

Many new companies have emerged over the years and the pace of development is increasing. One such example is Fetch.ai, a company that is combining blockchain technology and machine learning applications with the goal of developing “a decentralized digital representation of the world in which autonomous software agents perform useful economic work” [7]. While the company is working in many domain areas, it seems clear that transportation will be an early application.

## 4 Related Work

In the past few years, there has been quite a bit of work in the area of privacy preserving transportation systems. In this section we highlight a few of the most relevant related work.

In [8], Kanza and Safra demonstrate how blockchain, cryptocurrency and pseudonymity technologies can enable a decentralized ride-hailing service that preserves location privacy and pseudonymity. Yuan et al. [9] present a vision for integrating ITS and blockchain systems. The researchers point out some early development, but those companies do not seem to have succeeded. Shiver et al. [10] presents a secure and decentralized blockchain-based ride-hailing platform for autonomous vehicles, but the findings could be applied in a pre-autonomous vehicle setting. In [11] the researchers propose the first privacy-preserving Blockchain-based incentive network in ad hoc vehicular networks (VANETs). The researchers propose to use an incentive mechanism called CreditCoin to motivate users to share network information via a vehicular announcement network. The researchers developed a novel privacy-preserving incentive announcement network VANET based on blockchain via an effective anonymous vehicular announcement aggregation protocol. In both Singh et al. [12] and Javaid et al. [13], the examination of the use of blockchain technologies to enable secure inter-vehicular communication is explored.

For more general tutorials on blockchain and smart contracts [5] provides an exemplary tutorial. Additionally, the Blockchain in Transportation Industry Alliance<sup>2</sup> (BiTA) provides information about standards for blockchain in transport.

Mo et al. created a ridesharing option for paratransit that was integrated with Dial-A-Ride [14]. Through this work, the researchers addressed the low vehicle utilization and high rejection rate of service requests in urban areas of Hong Kong, in collaboration with the largest community transportation organizations in the area. The ridesharing option in their algorithm was based on user tolerance of early pickup or late drop-off. The work focuses on the service design and community based operations research in the area of accessible services through their Dial-A-Ride (DAR). As a result of this study, the researchers conclude that the route optimization is heavily dependent on the user tolerance and decision, which makes the decision complexity very high. However, through experimentation, they were able to prove with the major transportation organization that it is possible to serve more people without increasing the number of vehicles being dispatched.

Yuan and Wang [9] provide a unique approach, that paper discusses the integration of blockchain technologies for Intelligent Transportation Systems (ITS). Within that work, a 7 layer architecture was developed. Systems (apps) in use that are modeled by this architecture include La'zooz (a decentralized

---

<sup>2</sup> <https://www.bitastudio.com>.

ride-sharing company located in Israel)<sup>3</sup>, Arcade city<sup>4</sup> and DACSEE<sup>5</sup>, both US companies. Lastly, Luo et al. [15] explored technical concerns related to Online Ride Hailing (ORH) services. The goal of their work was to achieve the privacy preserving yet practical ORH systems by keeping the driver and rider locations private by having the drivers encrypt their locations using ephemeral public key from potential riders and send the ciphertexts to the ORH server.

## 5 Transportation Network Company (TNC) Pilot Studies

These companies are generally mobile app-enabled ride hailing services. In over 63 countries and with a growing number of over 22,000 employees [16]. Since 2015, Uber has incorporated accessible transport for wheelchair passengers through its UberWAV (Wheelchair Accessible Vehicle) program. Additionally, Uber has since adopted Uber Assist as well, which is a program designed to provide additional assistance to passengers who are seniors and persons with disabilities. Both of these programs are only available in select cities. UberWAV is available in four cities in the U.S., while Uber Assist is available in 40, worldwide.

In 2017 Lyft, a competing app-enabled ride-hailing company began to incorporate a medical transportation program to provide rides for patients from their healthcare providers. The program is; Non-Emergency Medical Transportation (NEMT) currently deployed in parts of Arizona, Texas, and Florida.

Throughout different parts of the U.S., paratransit agencies have piloted collaborative programs with TNCs including; Uber, Lyft and curb. Below, we provide some descriptions of the pilot programs.

In the U.S., in Boston, the Massachusetts Bay Transportation Authority initiated a pilot program in 2018 that has been extended several times, lately until March 31, 2020. The drivers of the TNC services (Uber, Lyft, Curb) do not provide ADA complementary Paratransit Services, but they can accommodate ADA approved passengers who do not require wheelchair accessible vehicles.

In another example, California's Tri-delta transit launched a pilot program pairing TNCs with ADA Paratransit in 2018. Both Uber and Lyft participated in the pilot which was launched in 2018 to tackle the driver shortage and manage the high operational costs. As a result of this pilot, the agency cost per trip was significantly reduced. Costs were estimated to have been reduced from \$30–\$32 per ride to \$8 [17]. That agency recently launched another pilot offering \$2 ride-hailing services linking households with transit services for all users [18].

## 6 Cyber Attacks

The number of connected heterogeneous devices in IoT systems make them especially vulnerable to cyber-attacks. Here, we highlight five types of attacks.

<sup>3</sup> <http://lazooz.org>.

<sup>4</sup> <https://arcade.city>.

<sup>5</sup> <https://dacsee.com>.

The first is the sybil attack. A sybil attack is a well-known cyber-attack that involves the attacker creating multiple fake profiles to achieve an unreasonably high user rating.

Another attack is an eclipse attack. These attacks are prominent in decentralized networks. This attack involves the attacker isolating an entity to where the victim cannot participate in the network at large.

The next attack is a malicious ledger which is essentially important within our work as it will incorporate the use of cryptocurrencies and the management of a ledger. In this attack, the user may wish to inquire information about another entity by stealing the ledger.

Packet sniffing is another attack that compromises the system. The attacker tries to sniff the networks that transactions travel through to retrieve or infer private information about users.

Finally, a malicious client application is an attack where the operator may attempt to develop a client application that could gain access to avoid information.

Awareness of cyber-attacks is particularly important to this research because we are hoping to develop robust, secure systems.

## 7 Prototype Design

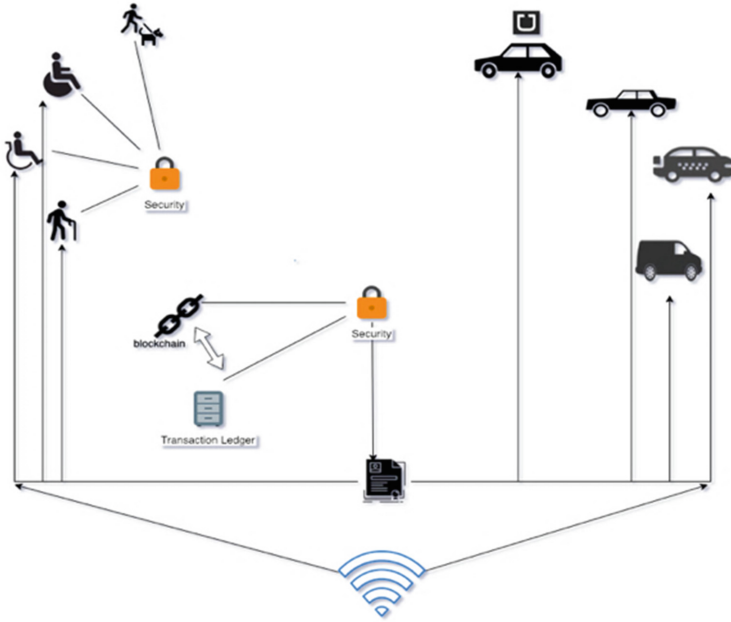
In order to simulate the decentralized blockchain-based paratransit environment, we use popular Ethereum blockchain tools. Ethereum blockchain is an open source, public blockchain-based distributed computing platform that runs on the Ethereum Virtual Machine (EVM) which executes opcodes.

### 7.1 Materials

The tools included in this project are Truffle suite and Ganache. Truffle is a development environment, testing framework and asset pipeline for blockchain using the EVM. Within Truffle, we use Ganache, a personal blockchain on Ethereum to deploy contracts, develop applications and run tests.

Using these open source blockchain tools; Truffle and Ganache, we are able to simulate the transactions between the nodes in the network. The nodes consist of the various TNCs such as Uber and Lyft, taxi services and paratransit services that we wish to incorporate on the network.

Figure 2 shows how the ecosystem will be managed. All of the entities on the network are securely connected to each other in a decentralized environment. The blockchain maintains the blocks and transaction ledger. The entities are representative of the IoT devices which will be used to requests, accept and pay for the transportation. These devices include; mobile phones and wearable devices such as smart watches.



**Fig. 2.** The ecosystem for blockchain-based IoT paratransit

## 7.2 Architecture

### 7.2.1 Frontend

The frontend of the prototype involves the use of HTML, CSS and JavaScript to connect to the blockchain. The blockchain testing framework that is used is in Ganache. Ganache is a test blockchain that allows us to access a personal Ethereum blockchain to run, execute commands and inspect the state of the blockchain. It can be used through the command line or the Graphical User Interface (GUI).

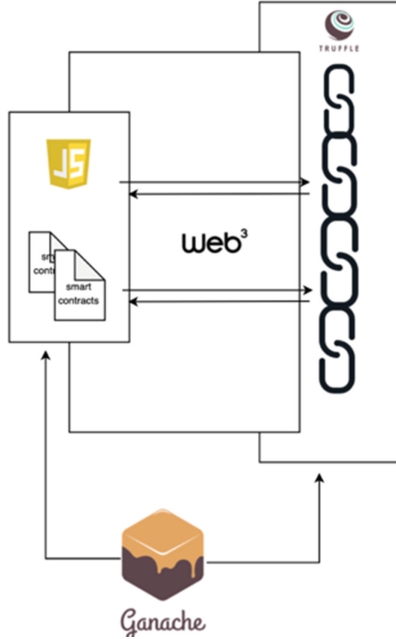
Ganache, provides users with an account and the resources (Ether) to deploy and process transactions between entities on the network. Ether is used to pay for the computational resources needed to run an application or program. Ganache connects via a Remote Procedure Call (RPC) server.

### 7.2.2 Backend

The backend of the implementation is in Ethereum. The transactions are collated into blocks; blocks are chained together using a cryptographic hash as a means of reference forming the blockchain. Each block has access to the information of the previous blocks as well. The cryptographic hash function provides an extra layer of security, assuring that each transaction is verified. Ethereum works on the backend of decentralized applications. Through mining, each block on the chain comes to a consensus to approve the newest block. This is a brute force

process. Proof of stake and the proof of work are the two main algorithms used in Ethereum blockchain.

Web3.js (see Fig. 3) is a collection of libraries which allow you to interact with a local or remote Ethereum node, using a HTTP or IPC connection.



**Fig. 3.** Frontend and backend architecture of the prototype design

## 8 Smart Contracts

Smart contracts are secure transactions between two or more entities in a trustless environment. There are several programming languages for writing smart contracts but, the most popular is Solidity. Other examples include: Golang, Vyper, JavaScript and Simplicity.

Since we are dealing with large scale environments with a large number of potential users, it is essential to incorporate smart contracts as the method of transactional execution. The transactions within this network consist of the requests, approval and payment of transportation service.

### 8.1 Nodes on the Network

#### 8.1.1 Nodes

Within the network, the nodes represent the various vehicles in the ecosystem. The vehicles will be paratransit vehicles, TNCs or taxi services.



### 8.1.2 Mining/Miner

Miners are the peers on the blockchain. The role of the miner is to verify the legitimacy of each block. The miners are responsible for the maintenance of the decentralized ledger. The miner will run the block's unique header metadata through a hash function only changing the nonce value. If the hash matches the target, then, the miner is awarded ether and broadcasts the block on the network for each node to validate and add their own copy of the ledger. When the proceeding miner finds the correct hash, the previous minor will stop the current block and repeat the process for the next blocks. This process is computationally intensive but, as a result, the miner will be rewarded.

### 8.1.3 Consensus Protocols

Consensus is best defined as a fault tolerant mechanism that is used in computer and blockchain systems to achieve the necessary argument on a single data value or a single state of the network among distributed processes or multi-agent systems [19]. Through the consensus protocols, the network remains secure and each transaction is verified. Below, the most common protocols are defined:

- a. Proof of Work (PoW) – Is the well-known protocol that is very time consuming, its execution time is about 15–30 transactions per second. Each node must store the entire blockchain to verify transactions. This protocol is dependent on computing power. Mining nodes have to complete a cryptographic puzzle before they can post new blocks on the blockchain. The miners have to predict the input of the cryptographic hash, such that the output is less than the difficulty number. PoW protocol is modeled by the famous computing problem, the byzantine general problem.
- b. Proof of Stake (PoS) – This protocol was established after PoW as a more energy efficient and secure consensus protocol. In contrast, PoS depends on a nodes' amount of ether (stake). Any node that wants to participate in the creation of a new block must put down a deposit and join the pool of miners. The miner with the largest stake has a greater chance of successfully mining a new block. Within this protocol, if any miner is malicious they would lose initial stake and privilege to be selected from the mining pool. There are two selection algorithms: randomized and coin age selection, to select the miner from the pool.

## 9 Discussion and Limitations

We believe that through the use of blockchain-based smart contracts, paratransit services could be improved in several ways. Enabling new technologies will allow for secure collaborations with TNC and taxi services, thereby proposing a cost effective solution for transit agencies. But, there are several limitations. These include: access to resources, initial costs associated with development and deployment of blockchain, and the real-world implementations at transit agencies.

## 10 Conclusions and Future Work

In this work, we provided a proof of concept prototype design for a blockchain based IoT paratransit system. We have outlined the importance of privacy when incorporating TNC services with paratransit. Blockchain methods are one possible solution.

In our on-going work we are conducting in-depth semi-structured interviews with representatives of Metropolitan Planning Organizations (MPOs) in Southern California, and conducting a survey of MPOs and transit agencies state-wide. The survey was sent to over 180 such agencies. Some of these conduct paratransit services, while others oversee these services in various ways.

The purpose of the surveys and interviews are to gain perspective on the agencies and organizations willingness to adopt new technologies and to cooperate with private transportation service providers.

In closing, paratransit operations pose challenges for users. These are chiefly, inconvenient scheduling, long waiting times for appointments and for service within those appointments, equipment miss-matches. The scope of our work is to use emerging technologies to improve paratransit. Shifting users who do not require wheelchair accessible vehicles to passenger cars, especially with same day service, would lead to a vase reduction in passenger wait times, an increase in passenger satisfaction, and reduction in agency costs with the caveat that increased service would certainly lead to increased demand, which would then raise agency costs. Through this work, we show how paratransit and TNC and taxi services can be enabled through blockchain based smart contracts.

## References

1. ADA.gov (2019). <https://www.ada.gov>
2. HHS.gov: Health Information Privacy (n.d.). <https://www.hhs.gov/hipaa/index.html>. Accessed 11 Aug 2019
3. Lisk.io: Blockchain Basics (n.d.). <https://lisk.io/academy/blockchain-basics>. Accessed 11 Aug 2019
4. Wei, L., Liu, S., Wu, J., Long, C., Ma, S., Li, B.: Enabling distributed and trusted IoT systems with blockchain technology. *IEEE Blockchain Technical Briefs* (2019)
5. Delmolino, K., Arnett, M., Kosba, A., Miller, A., Shi, E.: Step by step towards creating a safe smart contract: lessons and insights from a cryptocurrency lab. In *International Conference on Financial Cryptography and Data Security*, pp. 79–94. Springer, Heidelberg, February 2016
6. Hackius, N., Petersen, M.: Blockchain in logistics and supply chain: trick or treat? In *Proceedings of the Hamburg International Conference of Logistics (HICL)*, pp. 3–18 (2017)
7. Simpson, T., Sheikh, H., Hain, T., Rønnow, T., Ward, J.: Fetch: Technical Introduction (revision 2.0.3) (2019). <https://fetch.ai/uploads/technical-introduction.pdf>. Accessed 11 Aug 2019
8. Kanza, Y., Safra, E.: Cryptotransport: blockchain-powered ride hailing while preserving privacy, pseudonymity and trust. In: *ACM SIGSPATIAL* (2019)

9. Yuan, Y., Wang, F.Y.: Towards blockchain-based intelligent transportation systems. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 2663–2668. IEEE, 1 November 2016
10. Shivers, R.M.: Toward a secure and decentralized blockchain-based ride-hailing platform for autonomous vehicles. Doctoral dissertation, Tennessee Technological University (2019)
11. Li, L., Liu, J., Cheng, L., Qiu, S., Wang, W., Zhang, X., Zhang, Z.: CreditCoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles. *IEEE Trans. Intell. Transp. Syst.*, 2204–2220 (2018)
12. Singh, M., Kim, S.: Crypto trust point (cTp) for secure data sharing among intelligent vehicles. In: 2018 International Conference on Electronics, Information, and Communication (ICEIC), pp. 1–4. IEEE, January 2018
13. Javaid, U., Aman, M.N., Sikdar, B.: DrivMan: driving trust management and data sharing in VANETs with blockchain and smart contracts. In: 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), pp. 1–5. IEEE, April 2019
14. Mo, Y. D., Wang, Y., Lee, Y.C.E., Tseng, M.: Mass customization paratransit services with a ridesharing option. In: 2017 IEEE Transactions on Engineering Management, pp. 1–13 (2018)
15. Luo, Y., Jia, X., Fu, S., Xu, M.: pRide: privacy-preserving ride matching over road networks for online ride-hailing service. *IEEE Trans. Inf. Forensics Secur.*, 1791–1802 (2018)
16. Uber Technologies (2019). <https://www.uber.com/newsroom/company-info/>
17. Jordan, S.: Pairing TNCs and Paratransit, California Transit Association (2018). <https://caltransit.org/news-publications/publications/transit-california/transit-california-archives/2018-editions/october/pairing-tncs-and-paratransit/>. Accessed 11 Aug 2019
18. MBTA: On-demand Paratransit Pilot Program. <https://www.mbta.com/accessibility/the-ride/on-demand-pilot>. Accessed 17 Dec 2019
19. Frankenfield, J.: Consensus Mechanism (Cryptocurrency) (2019). <https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp>. Accessed 11 Aug 2019



# A Review of Cyber Security Issues in Hospitality Industry

Neda Shabani and Arslan Munir<sup>(✉)</sup>

Kansas State University, Manhattan, KS 66506, USA  
{nshabani, amunir}@ksu.edu

**Abstract.** The purpose of this study is to emphasize the importance of cyber security in hospitality industry. This study further identifies and analyzes several common network threats and recommends useful security practices and techniques to prevent cyber attacks in hotels. This study is a rich source of information for Information Technology (IT) directors and Chief Information Officers (CIO) to advance their policies and procedures for security of electronic information in hotels using the most recent and updated information available in the area of hospitality industry. The methodology of this study is a unique combination of qualitative method and review method for an in-depth understanding of real-life issues within the industry and the most recent technical and practical solutions that hotels use to handle and solve these issues. The findings of this study show that the techniques currently utilized by hotels to prevent cyber attacks are mostly rudimentary and outdated. Furthermore, study indicates that most of the hotel staff lacks the knowledge and expertise to handle potential threats and thus hospitality industry becomes even more vulnerable to cyber threats and attacks. Finally, the paper discusses some implications and recommendations to hotel's policy makers to help secure the hotels' and guests' information from security attacks.

**Keywords:** Cyber security · Hospitality industry · Information security

## 1 Introduction

Technology in hospitality industry is driven by the increasing transaction volumes, complex reporting requirement, e-marketing [14], and international communication needs. Information technology (IT) can improve almost all areas of hospitality industry, such as guest services, reservations, food and beverage management, sales, food service catering, maintenance, security, and hospitality accounting. More recently, Internet of things (IoT) is shaping the future of hospitality management industry by opening up new avenues for immediate, personalized, and localized services. For example, in-room IoT units like thermostats, motion sensors, and ambient light sensors can be utilized to control the temperature and lighting in hotel rooms based on room occupancy to minimize energy

costs. Moreover, edge/fog computing can be utilized to provide location-based services for the hospitality industry [10]. Although technology incorporation in hospitality industry over recent years has transformed the way services are provided and received and has helped in improving guest experiences, it has also given rise to various challenges among which ensuring the cyber security of these incorporated technologies in the hospitality industry is of paramount significance [8, 11].

The use of technology in hospitality industry often requires gathering of guest information and thus can lead to data breach and information loss. To prevent against losses, organizations monitor their computer networks for a multitude of security threats, such as computer-assisted fraud, espionage, sabotage, vandalism, hacking, system failures, fire, and flood, etc. Since hospitality industry is a consumer-centric business where consumer loyalty and trust directly translates to revenue, hence to retain the public trust and to prevent copycat hackers to hack into an organization's computer systems, most of the hospitality organizations try not to reveal the data breaches and cyber attacks against their computer systems [4]. Thus, this paper mainly focuses on the review of cyber security threats and risks faced by the hospitality industry, state-of-the-art tools and techniques that can be employed by the hospitality industry to defend against cyber attacks, and implications and recommendations for the hospitality industry to help secure the hotels' and guests' information from security attacks.

### 1.1 Research Purpose and Research Questions

The purpose of this study is to emphasize the significance of cyber security in hospitality industry by identifying and analyzing several common network threats and recommending useful security practices and techniques related to electronic information and network systems to prevent cyber attacks in hotels. The following research questions were created to be answered based on the unique methodology leveraged by this study:

1. What methods, tools and techniques are currently used in hotels regarding computer network and information protection?
2. What are the current threats to computer network security in hotels?
3. What are the ways of handling security attacks in the hotel's computer networks?
4. What is the importance of network security in hotels?
5. Which methods hotels leverage to secure their websites for data and financial transactions?
6. What criteria hotels consider in making a strong password for their computer networks and logins (computers and websites)?

The remainder of this paper is organized as follows. Section 2 provides an in-depth review of cyber security issues in hospitality industry. Section 3 outlines the methodology employed by this study to answer the research questions posed by this study. Findings and results of this study are presented in Sect. 4. Section 5 concludes this study and provides recommendations for hospitality industry to help secure hotels' and customers' data from potential security attacks.

## 2 Background and Literature Review

This section discusses background and literature review related to cyber security in hospitality industry. In particular, this section discusses common hardware/software used in hospitality industry, information security tools and techniques, cyber threats, risks, and challenges in hospitality industry, and cyber attack prevention methods in hospitality industry. Figure 1 depicts an overview of cyber security threats in hospitality industry and potential cyber attack prevention methods that are discussed in this paper.

### 2.1 Hardware/Software Used in Hospitality Industry

IT is the science and technology of using computers and other electronics to save and transmit information. Organizations that use IT need to tackle and administer electronic information safely and securely. The organization's administrative managers are responsible for the protection of the organization's assets and information [4]. Like other organizations, IT systems in hotels comprise of both software and hardware. The basic software in a hotel includes the property management system (PMS), point-of-sale system (POS), call accounting system (CAS), and hotel accounting system. The basic hardware in a hotel include front desk computers, POS terminals, back office computers, cameras, printers, routers, switches, network cables, sensors and other IoT devices. The front and back office computers, POS terminals, and printers are connected to routers and switches with network cables that enable communication between these devices. The hotel's local area network (LAN) typically consists of devices within the hotel's premises. The hotel LAN is connected to other networks and the Internet through routers. The firewalls protect the hotel network from outside attacks. The hospitality industry uses the POS and PMS to manage reservations while avoiding duplex reservations for the same date and time [5].

### 2.2 Information Security Tools and Techniques

Organizations using IT are vulnerable to various security threats and attacks. The most common threats include viruses, inside attackers for network access, laptop theft, spoofing, unauthorized insider access, unauthorized outside attack, and denial of service attacks. Information security aims at maximizing the revenue of organizations and investments by minimizing the damage that could be caused by security attacks [13]. Most of the information security systems aim at providing three main security services: confidentiality, integrity and availability. Information security systems strive to protect valuable assets from disclosure or damage. This protection can be attained through both technological and non-technological methods, such as physical security of assets, user identification and authentication, biometrics, and firewalls [4]. We define some of the information security terminology, tools and techniques in the following:

1. ***Digital Identifiers (IDs)*** are the electronic counterparts of driver's licenses, passports, and membership cards. Digital IDs often include a username

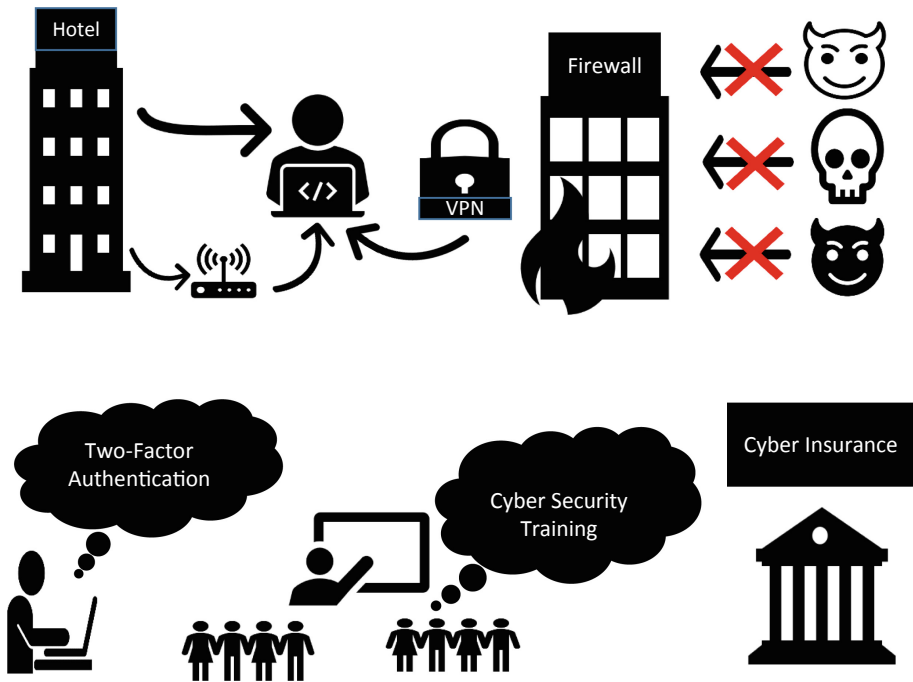


Fig. 1. Overview of cyber security threats for hospitality industry and potential cyber attack prevention methods.

and a password. In asymmetric cryptography (a type of information security system), a user/system possess a public key and a private key, which can serve as IDs. Digital *certificates* are used in asymmetric cryptography to authenticate public keys and IDs. A certificate binds the ID of a user/system to its public key by providing a digital signature over the public key and the ID of the user/system [12].

**2. *Intrusion Detection System*** is a system that analyzes the events happening in a computer system or a network to detect intrusions or attacks. An intrusion can be defined as an effort to circumvent security services employed by the system, such as confidentiality, integrity, and availability. Many times intrusions from malicious actors are aimed at carrying out a denial of service attack that makes the computer systems of an organization unavailable. Intrusions can be caused by various means: (i) attackers connecting to the systems from the Internet or the outside networks; (ii) authorized users of the systems who try to obtain additional privileges for which they are not authorized; and (iii) authorized users who misuse and abuse the privileges given to them.

**3. *Physical Security*** refers to keeping the networking and computing equipment of an organization in a secure physical environment [4].

**4. *Firewall*** can be a hardware, a software or a combination of hardware and software equipment to monitor the traffic between devices and/or two or more computer networks. A hardware firewall is a physical device that is attached to a network while software firewall is a software that is installed on devices (e.g., computers, tablets, phones, etc.) in a network to monitor the network traffic flow. The firewall can also block particular malicious packets trying to enter or leave a computer network.

**5. *Encryption*** is the process of hiding the information by making the information transformed in a way that is impossible or very hard to understand. Encryption mainly provides confidentiality security service. The aim of encryption is to keep the information secret from all but the authorized parties.

**6. *Biometrics*** is a technology of authenticating a user based on physical or behavioral characteristics, such as finger prints, voice recognition, gait, and retina or iris identification. Biometric technology is an effective method of identity verification. The biometric systems measure the physical characteristics of an individual, and compare them with the recorded characteristics to verify the user's identity [1].

**7. *Access Control*** are techniques of restricting usage of system resources to authorized users/processes. Access control typically comprises of authentication and authorization.

**8. *Vulnerability Assessment Scan*** is a software that examines the system for potential vulnerabilities and inform the system administrator of those vulnerabilities so that system can be safeguarded against those weaknesses [4].

### 2.3 Cyber Threats, Risks, and Challenges in Hospitality Industry

Cyber crimes have always been there since the introduction of computers, however, the nature of attacks and crimes varies as the technology evolves. Hacking, technology theft, and frauds are the most common security attacks whereas other security attacks are also possible [4]. Most of the hacking attacks are aimed at obtaining confidential information (e.g., financial information of banking accounts, user accounts information) without authorization. Technology theft occurs when an attacker consciously connects to a computer with intentions to steal technological information. Theft of trade secrets happen when a person or a business uses confidential trade information for (another) business without authorization. Fraud transpires when an attacker consciously connects to a computer with intentions of fraud or masquerades a legitimate user of the computer system.

The appraised cost of cyber crimes is approximately \$6 trillion per year on average through 2021 [6]. Consequently, organizations are increasing their cyber security budgets to mitigate potential data breaches. The average cost of a data breach for an organization is in millions, however, this cost only accounts for the direct cost of the data breach which is quantifiable. The true cost of data breach for a business is much higher than this when outlook for a business and collateral effects in the aftermath of a breach are considered.



Although hospitality industry aspires to provide trust and comfort to guests, achieving this is not easy anymore due to lots of potential threats that are aimed at destroying the reputation of hotels as well as destroying the customer trust by acquiring and abusing guests' personal and financial information. Hoteliers must know that these threats are always out there, and they need to take responsibility for any data loss as they can prevent this data loss and breach from happening by adopting effective preventing measures.

Data breaches in the world of business are consistent and remind the organizations the significance of incorporating cyber security tools and techniques in their businesses to help prevent such incidents. Besides implementing cyber security tools and techniques, it is imperative for hotel users and staff to have fundamental knowledge about cyber security and practices. One of the most significant factors that can prevent data from being breached or loss is to be careful and conscious about what sources of information to trust and have some knowledge about secure websites and emails versus not secure websites and emails. For example, opening an unsecure website or clicking on the link within an unsecure email can cause a big disaster for a hotel if the staff are not knowledgeable enough in this regard. In addition, hackers and attackers are not only able to abuse the computer system of the hotels by using different type of phishing email, viruses, etc., but also, they are able to attack and take advantage of Wi-Fi in the hotels [3]. Most of the hotels nowadays offer free Wi-Fi to their guests and the guests can have access to the same network all over the hotel such as lobby, convention center, dining room and other places within the hotel. If the hotel Wi-Fi is not secure, which is the case for most of the contemporary hotels, hackers can monitor the guests' traffic on the Wi-Fi and use that to steal the guests' private information. Furthermore, by taking advantage of hotels' Wi-Fi, hackers can offer malicious "updates" for famous software such as Adobe Reader or Flash Player so that the users would not hesitate to update their software and then those updates contain malware that criminals use to get all the usernames, passwords, or other important information from users' computers or smartphones. Interestingly, in most of the cases, the software programs that are used by hackers and cyber criminals, are not new programs and can be even decade old programs. However, due to negligence and ignorance of many hotels and lack of system updates, even these old software programs can be utilized by the hackers to acquire confidential information from the hotels [7].

Attackers can be from both inside and outside an organization. Especially in hospitality and tourism industry where turnover rate is very high, the possibility of inside attackers is higher in comparison to other industries. Consequently, some organizations, such as Burger King Corporation, take measurements to prevent inside attackers, and provide infrastructure to ensure only a single sign-on by an employee. In this case, only one record needs to be expunged from the system in case of an employee's termination or resignation so that the former employee will not have any access to the system [4].

There are so many ways that attackers can get into guests' information, however, one of the most common ways that attackers use specifically in hospitality

industry to breach data is “Fake Booking”, in which the attacker build and design a website with the exact look and features of the main hotel’s website and use the same name to pretend that it is the hotel’s legitimate website. Consequently, many potential guests visit that phishing website and probably some of them book their room through that fake website, thus revealing all their personal and financial information to the attacker.

## 2.4 Cyber Attack Prevention Methods in Hospitality Industry

It is to be noted that none of the security software, antiviruses, and other tools can 100% guarantee to prevent hotels and any other business from cyber attacks, however, hotels must implement the most effective and updated tools and techniques to secure their information as much as possible. In general, hotels can divide the process of securing their information into three phases: prepare and protect, defend and detect, respond and recover.

One of the tools to prevent data breach attacks that hotels can take advantage of is web application firewall (WAF). WAF is different from regular firewall (discussed in Sect. 2.2) in that a WAF is able to filter the content of specific web applications and thus help preventing attacks originating from web application security flaws, such as SQL injections, buffer overflow, and security misconfigurations. WAF solutions are also useful to detect and prevent data theft because in case of attackers targeting credit card database, the WAF solutions can detect and block the database.

Another way to secure data in hotels is by employing digital certificates (Sect. 2.2). Digital certificates bind a message to the owner/generator of the message and help provide non-repudiation security service. In hospitality industry, digital certificates can help prevent frauds from customers or hotel/restaurant owners as false claims from either can be legally challenged and the truth be established by using digital certificates. The use of digital certificates by hotels for their websites ensure the authenticity of their websites to the customers.

Cyber security insurance can provide hotels another way to secure themselves and customers’ information as well as cover their losses in the case of data breach. According to Butler [2], cyber security insurance must be a consideration for any hotel owners. Hotel owners must know that data breaches and cyber claims are not included in general liabilities policy, which makes it even more important for hotel owners to think about and acquire cyber insurance to cover any claims in case of a cyber attack. Cyber security insurers will cover both first and third party in the case of cyber attack and data losses. The third party can be both customers and government or any regulatory agencies.

## 3 Methodology

This paper takes advantage of two research methodologies to emphasize the importance of cyber security in hospitality industry. One of the methodologies is an in-depth review of all academic and professional articles available in the area

of hospitality cyber security. The authors have summarized in Sect. 2 the most important findings and issues related to cyber security and threats for hospitality industry in this paper. The other methodology was qualitative in nature and the authors interviewed thirty hospitality professionals, academicians and hotel guests. Among thirty interviewees, three of them were hospitality professors who teach “Hotel IT”, seven of them were hotel managers from different size hotels, ten of them were hotel staff (front desk clerks) and ten of them were hotel guests. Each interview took an average of ten minutes (about five minutes with guests and fifteen minutes with staff and managers). The questions that were asked from managers and professors were very similar to the research questions of this paper and the questions that were asked from hotel staff and guests were mostly basic questions about computer, email and website security.

## 4 Findings and Results

The findings and results of this study after interview with the front desk employees, guests, managers and professors indicate that many hotels use rudimentary tools/software such as antiviruses to prevent data breach and data loss. It was shocking to learn that many medium- to small-size hotels do not have an IT manager or dedicated computer security professional. Even some of those hotels do not have any contract with any IT company for handling cyber security issues. When computer-related problems are faced by these hotels, they call random IT professionals from different companies to fix their computers’ problems which can be a big security risk to the hotel as that random IT person can be a potential threat to the hotel computer system and network.

Two of the managers interviewed for this study confessed that they have experienced data breach in last five years and one of them mentioned that the attacker was likely a former employee who had some personal problem with the IT manager, and he wanted to take revenge in this way. Unfortunately, due to sensitivity of the topic, both managers refused to explain the extent of data breach. Furthermore, except a couple of hotel staff members, majority of the hotels’ staff mentioned that they have not received any suspicious or unsecure email in which the sender asked them to click on some random link to get their information. Indeed, most of the hotel staff and most of the hotels’ guest that were interviewed did not have much knowledge about cyber security. During the interview with one of the guests, the guest mentioned that she had an experience of getting her data breached through a hotel network system. The guest indicated that she was a loyalty-program member of that specific hotel and observed some abnormal transactions in her credit card and later on the data breach of the hotel went viral. One of the other hotels’ guest talked about his experience of “Fake Booking”. He said that he booked a hotel room online from a known hotel brand and when he went to the hotel, the front desk staff was not able to find his reservation. He then showed his printed reservation confirmation upon which the hotel staff recognized that the website name was misspelled and the website was not the legitimate hotel website. While asking about hotels’

Wi-Fi, unfortunately most of the guests mentioned that they use the hotels' Wi-Fi, which is often unsecure and is very vulnerable to security attacks. Hence, the sensitive information entered by guests, such as bank account details or user name and passwords for different accounts, over the hotels' unsecure Wi-Fi network is susceptible to theft by hackers. The authors suggest the guests to use a virtual private network (VPN) when using hotels' Wi-Fi and entering their personal information on websites they visit during their stay.

During interview with managers and staff, the focus of conversation was on training as managers informed that there is no formal training in place for staff regarding cyber security and data privacy. One of the managers complained about the high turnover rate in hospitality industry and he mentioned that due to this turnover issue, it might not be cost-effective to train the staff about these "marginal" issues. It was shocking to hear that word from a manager who should know best about the loss his hotel may experience due to data breach, and which can be many times more than the cost of training staff regarding data security and privacy. During our discussion with professors, professors pointed out theoretical aspects of cyber security as they did not have practical experience in the area, however, they provided useful suggestions that hoteliers can use to provide better security and privacy. We have covered some of these suggestions in this paper.

Overall, findings have shown that lack of knowledge and carelessness is the most observable issue of the hospitality industry staff (managers and front desk clerks) and majority of guests. Another finding was lack of training for employees and lack of usage of strong tools and techniques to secure hotel computer systems and network. Lack of IT experts in hotels was another finding of this paper which has a direct relationship with vulnerability of hotels for data breaches. Finally, failing to update the hotel software on regular basis, not changing the passwords periodically, and not creating strong passwords were some of the other important findings of this paper that indicated the vulnerability of hotels to security attacks.

## 5 Conclusion, Implications, and Recommendations

This study aims at emphasizing the importance of cyber security for hospitality industry. The study discusses the tools and techniques that can help prevent cyber attacks in hospitality industry. Findings and results from this study reveal some of the main causes that create security vulnerabilities for the hospitality industry, however, due to sensitivity and confidentiality of the topic, certainly the authors are not able to figure out many other factors during the interviews that may affect information security of hotels.

After reviewing many academic and professional resources, we summarize that there are five major risks and challenges that hotels have faced so far. As noted by Hiller [9], these five challenges are:

1. Identity theft leading to credit card fraud has caused many data breaches and information stealing from hotel's network systems.

2. Silent invasions are cyber-crime attacks that employ powerful tactics such as social engineering (e.g., phishing) and recently advanced persistence threats (APTs) that bypass the defenses that are in place by hotels.
3. Unfortunately majority of the hotels have either no security audit or longer security audit cycles that put the investors and the guests at high risk for security attacks.
4. Physical crimes like terrorism that put the hotels at risk.
5. Loss of competitive advantage and negative outlook that is experienced by hotels after cyber security attacks.

In general, cyber attacks can occur in any of the following three forms:

1. The intruder may obtain unauthorized access to the network.
2. The intruder may destroy, otherwise corrupt or alter the data.
3. The intruder may acquire fake permission for system user and then implement some malicious procedures to fail, hang, or reboot the system.

There are many implications and recommendations available for users in hospitality industry to take advantage of, however, in this paper, we present a few important ones.

Checking a website domain and secure socket layer (SSL) certification of websites plays a significant role in Internet era and users must be very careful in entering their personal and financial information on websites. We suggest a few tips and advices for hotel customers while traveling and especially during the hotel stay. We suggest customers to: (i) not use online banking on public computers and public Wi-Fi, (ii) not access email inbox when traveling and connected to an unsecure Wi-Fi, (iii) prevent computer or smartphone from automatic connection to unknown Wi-Fi networks, (iv) use remote desktop applications instead of saving sensitive information on the laptop or smartphone when traveling, and (v) utilize a VPN network for browsing and entering personal information on websites when connected to an unsecure Wi-Fi network.

Research has shown that hotels which have loyalty programs are more vulnerable to security attacks because attackers know that these hotels have access to more consumer data as compared to those hotels that do not have this program. Thus, the information of guests and customers of hospitality industry who are the member of these loyalty programs is more vulnerable. Managers of those hotels can take a few measures to protect the information and data of loyal customers:

1. Giving the customers information about the possibility of being hacked by cyber attackers and advise/notify them to regularly change their passwords. Further, managers can inform the customers to avoid using the same password for several websites. Also customers can be informed to check and monitor their account activity more often. Managers can also reward customers for being security cautious.
2. Sending the customers an automatic email and notification in case of password change or login to their account so that in case if they have not logged in to

their account or change the password, they can immediately be aware of potential abuse and report it.

3. Empowering the system to employ two-factor authentication so that for logging into accounts, in addition to providing user name and passwords, guests would also be required to submit the security code that they will receive on the same email address or phone number that they provided while signing up for the account. Hence, in case of attempted masquerade attacks, a cyber attacker will not be able to access the account if they are not in possession of the email account or the phone (number) that the account was registered with as the attacker will not be able to acquire the passcode sent by the authentication system.

There exist a variety of tools and techniques available to scan the vulnerability of the computer system and network. The hotels can utilize these tools and techniques depending on the affordability to protect data and personal information of guests. Furthermore, it is advised that each hotel should have a contract with an IT company or a dedicated IT manager whom the hotel trusts so that the hotel computer systems and networks are security audited on a regular basis. Additionally, hotels should dictate internal regulations and policies for the hotel's employees regarding cyber security and computer network usage. The hotels should also have a cyber security training program in place for those employees whose job is computer related and are tasked with handling emails and social media. Furthermore, hotels must have a secure and certified website that leverages extended validation or at least domain validation, so that the guests be able to book the rooms or amenities provided by the hotel online without having concern of being hacked or abused. Finally, the hotels should acquire a cyber insurance so that the insurance covers the loss and liabilities in case the hotel experiences a data breach or cyber attack.

## References

1. Bilgihan, A., Karadag, E., Cobanoglu, C., Okumus, F.: Research note: biometric technology applications and trends in hotels. *FIU Hospitality Rev.* **31**(2), 1–18 (2013)
2. Butler, J.: Not Just Heads In Beds – Cybersecurity for Hotel Owners (2016). <https://www.hospitalitynet.org/opinion/4073687.html>. Accessed 26 Dec 2019
3. Clark, C.: The Serious Cyber Security Threat That Could Hurt Hotels (2015). <http://www.pcma.org/news/news-landing/2015/04/13/the-serious-cyber-security-threat-that-could-hurt-hotels#.VqhHLGCZaJV>. 26 Feb 2016
4. Cobanoglu, C., Demicco, F.J.: To be secure or not to be: isn't this the question? a critical look at hotel's network security. *Int. J. Hospitality Tour. Administration* **8**(1), 43–59 (2007)
5. Collins, G.R., Cobanoglu, C., Bilgihan, A., Berezina, K.: *Hospitality Information Technology: Learning How to Use It*. Kendall Hunt Publishing, Dubuque (2017)
6. Eubanks, N.: The True Cost Of Cybercrime For Businesses, July 2017. <https://www.forbes.com/sites/theyec/2017/07/13/the-true-cost-of-cybercrime-for-businesses/#764083449476>. Accessed on 26 Dec 2019

7. Greenberg, A.: Cybercrime Checks Into Hotels (2010). <https://www.forbes.com/2010/02/01/cybersecurity-breaches-trustwave-technology-security-hotels.html#4f1684853c8c>. Accessed 26 Dec 2019
8. Hahn, D.A., Munir, A., Mohanty, S.P.: Security and privacy issues in contemporary consumer electronics. *IEEE Consumer Electron. Mag.* **8**(1), 95–99 (2019)
9. Hiller, S.: Top 5 Risks and Security Challenges for Hotels in 2015, January 2015. <https://insights.ehotelier.com/insights/2015/01/22/top-5-risks-and-security-challenges-for-hotels-in-2015/>. Accessed 26 Dec 2019
10. Kansakar, P., Munir, A., Shabani, N.: A fog-assisted architecture to support an evolving hospitality industry in smart cities. In: *Proceedings of the 16th International Conference on Frontiers of Information Technology (FIT)*. IEEE, Islamabad, Pakistan, December 2018
11. Kansakar, P., Munir, A., Shabani, N.: Technology in hospitality industry: prospects and challenges. *IEEE Consumer Electron. Mag.* **8**(3), 60–65 (2019)
12. Paar, C., Pelzl, J.: *Understanding Cryptography*. Springer, Heidelberg (2010)
13. Rusch, J.J.: Computer and internet fraud: a risk identification overview. *Elsevier Comput. Fraud Secur.* **2003**(6), 6–9 (2003)
14. Shabani, N., Munir, A., Hassan, A.: Revolutionizing e-Marketing via augmented reality: a case study in tourism and hospitality industry. *IEEE Potentials* **38**(1), 43–47 (2019)



# Extended Protocol Using Keyless Encryption Based on Memristors

Yuxuan Zhu<sup>1(✉)</sup>, Bertrand Cambou<sup>1(✉)</sup>, David Hely<sup>2(✉)</sup>, and Sareh Assiri<sup>1(✉)</sup>

<sup>1</sup> School of Informatics Computing and Cyber Systems,  
Northern Arizona University, Flagstaff, AZ, USA  
{yz298,bertrand.cambou,sa2363}@nau.edu

<sup>2</sup> Univ. Grenoble Alpes, Grenoble INP, LCIS, 26000 Valence, France  
David.hely@grenoble-inp.fr

**Abstract.** The growing interest for keyless encryption calls for new cryptographic solutions with real keyless schemes. This paper introduces an extended protocol using keyless encryption, which is hash-based and generic in cryptography. The sender side and the receiver side will be contained in the protocol. The sender will encrypt a plaintext and then send the cipher to the receiver side, and the cipher used in the protocol will be based on memristor arrays. We will use values of blocks of the plaintext to sort the cipher, which will improve the difficulty of being deciphered. Then, the receiver will receive the cipher and use it to decrypt the plaintext. The method of implementation is thoroughly detailed in this paper, and the security of the protocol is evaluated by testing random plaintexts thousands of times.

**Keywords:** Security and privacy · Keyless cryptography · Security protocol for encryption · Security evaluation

## 1 Introduction

For a long time, most of the traditional cryptography uses keys to encrypt the messages. But in recent years, the interest of encryption without using the keys is growing very fast. The reason for choosing keyless encryption topic is that the key generation, key distribution, and key storage in key cryptography are incredibly complex. Also, there are many issues with the keys that can help a hacker to extract the key such as the attack based on differential power analysis. This kind of attack is practical and non-invasive; the information will be leaked through hackers analyzing power consumption to extract secret keys from a wide range of devices [11]. Also, there is another reason that can motivate researchers to pay attention to the keyless encryption, which is to make protection for the network of internet of things (IoTs). Because the IoTs have a limitation of power and memory, the long-secret keys and some strong cryptographic schemes are hard to be implemented [1, 13]. The keyless encryption has been considered one of the best cryptographic methods for protecting IoTs [7].



A keyless protocol was invented at Northern Arizona University (NAU) cyber security lab and named as “Memristors to Design Keyless Encrypting Devices”. This protocol is based on keyless encrypting devices with arrays of memristors, which convert the message to encrypt into the modulation of the currents driving the cells at low power. It has used the multifactor of security such as the random number generator, password, hash functions and memristor arrays as shown in Fig. 1 [5].

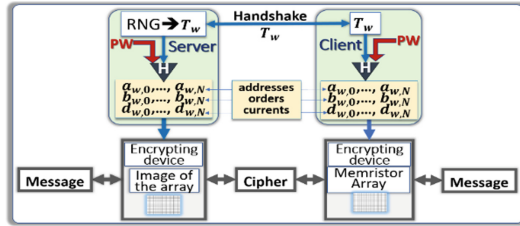


Fig. 1. Keyless encryption schemes with memristor [5]

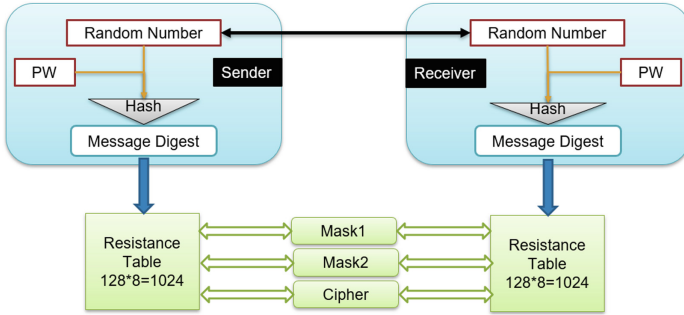
Figure 1 shows all the protocol steps, which are the following: first, the handshake is generated between the sender side and the receiver side. Second, each party will XOR the password (PW) with a random number (RN), and the result of the XOR will be fed to the hash function to get message digest (MD). The MD will be divided into three parts: addresses, orders, and currents. The third step is to do the encryption. The encryption step has two other steps: (a) the plaintext will be divided into several blocks; (b) each block will be combined with resistance value which comes from “images” of the memristors. Both address and current will point to the cell (resistance value), which will be picked to combine with the block that comes from the plaintext, whereas the orders will help to reorder the cipher.

NAU cybersecurity lab has developed Physical Unclonable Functions (PUFs) from ReRAM array memristors. The PUF is generated via the injection of low currents in cells of memristor arrays, and it will give new variable resistances each time different low currents are injected. These different values of resistances have been exploited to design PUFs. In this paper, the extended keyless protocol based on the protocol in “Memristors to Design Keyless Encrypting Devices” will be designed the same as the architectures of Fig. 1 except the orders [5]. Instead of getting the orders from the MD, they will be provided from the plaintext itself. This extended protocol is designed to perform encryption and decryption in a safe method. The paper explains how to design it from the software perspective.

## 2 Environmental Setup

The protocol contains the sender side and the receiver side. The sender side will encrypt a message and then send the cipher and masks to the receiver side.

Then, the receiver side will use the information received to restore the same message. It is a complicated process that our team has written it in C++ language.

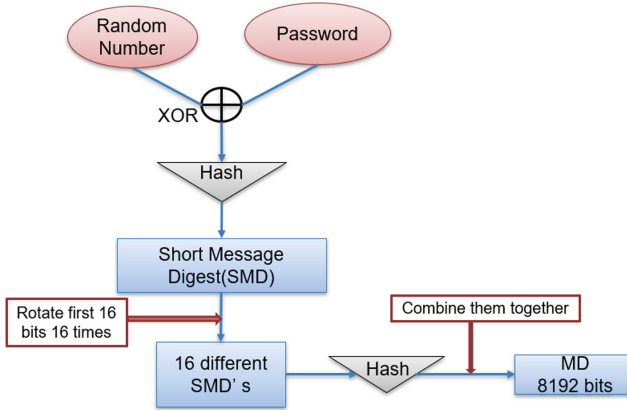


**Fig. 2.** Receiver side and sender side in protocol

First, we want to use Fig. 2 to introduce the basic environment for building this protocol. As shown in Fig. 2, both the sender side and the receiver side share the same random number (RN) which is a 64-byte long binary stream that came from our random number generator function [4, 6]. The RN is public which means everyone can access it. At the same time, the sender side also generates a 64-byte long password (PW) which is also in binary representation, and it will be sent to the receiver side in a secure environment. “Secure” means only two sides can access it. In network system, using https requests and encrypting and transmitting with RSA is a good option.

The cybersecurity lab at NAU provided a table of resistances, which contains experimental data generated from 128 ReRAM cells measured with negative bias between 100 nA and 800 nA in the 0 °C to 60 °C temperature range [3, 5]. This data has been saved as a CSV file and sent to our team, and both sides in the protocol will read the file and use this table as a two-dimensional array, which contains 128 rows and 8 columns. We will call it the resistance table for simplicity. In sum, the sender side and the receiver side will have the same RN, PW, and resistance table (RT). There are a few emerging themes between RN and PW. They are both 64 bytes long and in binary form. The only difference between them is that the RN is public, but the PW is private.

Then, the sender side will combine the RN and the PW using the Exclusive or (XOR) operation and will get a binary stream (64 bytes). The protocol chose XOR because other people cannot get any bit of the original message bytes. For example, every time the sender side sees a ‘1’ in the encrypted byte, that ‘1’ could have been generated from a ‘0’ or a ‘1’. The same thing with a ‘0’, it could come from both ‘0’ or ‘1’. Therefore, not a single bit is leaked from the original message byte after using XOR logical operation [8]; this will greatly improve the level of security. After that, the sender side will use the SHA3-512 (Secure Hash Algorithm 3) function to generate a short message digest (SMD), which is 64 bytes (512 bits) long [2]. This procedure is shown in Fig. 3.



**Fig. 3.** How to generate SMD and MD.

After getting the SMD, the sender side will try to extend it because the length of the message digest decides how many characters can be encrypted in the protocol. To do it, the first  $n$  bits of SMD will be rotated, each time the output of rotation is going to feed the SHA3-512 Hash Function to obtain a new SMD, and finally all SMDs will be combined to get a longer message digest. In software implementation, the sender side will rotate the first 16 bits and then obtain 16 different SMDs. Finally, those 16 SMDs will create the longest message digest (MD), which will be  $512 * 16 = 8192$  bits.

### 3 Encryption

After obtaining 8192-bits MD, it's time to do the encryption in sender side. The first step in encryption architecture is shown in Fig. 4.

At first, the MD will be divided into  $n$  blocks; each block contains “address” and “current;” the address size is 7 bits, and the current size is 3 bits. The decimal value of 7 bits will be from 0 to 127, whereas the decimal value of 3 bits will be from 0 to 7. As a result, the decimal values of address ( $A_i$ ) and decimal values of current ( $C_j$ ) are used to determine the position ( $RT_{ij}$ ) in the resistance table. As we mentioned earlier, the resistance table (RT) is a Two-Dimensional array and it has 128 rows and 8 columns; there are a total of 1024 data. The decimal value of address array (0–127)  $A_i$  will decide on the row in the resistance table, and the decimal value of current array (0–7)  $C_j$  will decide on the column in the table. These two indices will determine the exact resistance  $RT_{ij}$ .

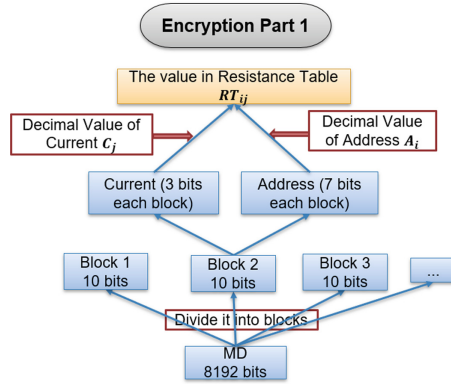


Fig. 4. Encryption part1

Since it is a protocol for encryption and decryption, it is natural to have information that needs to be encrypted. The operation for encrypting information is shown in Fig. 5.

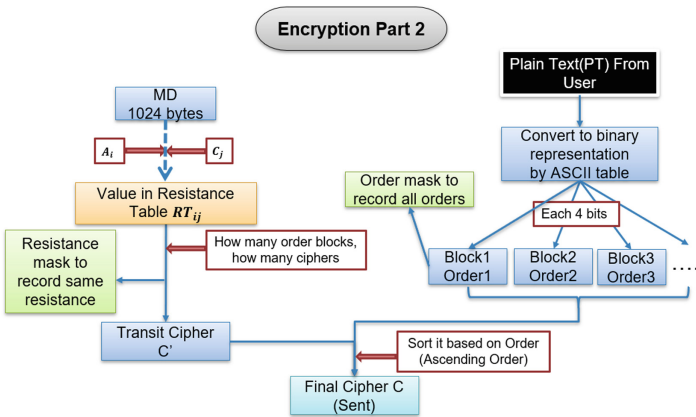


Fig. 5. Encryption part2

As shown, the sender side will be allowed to enter a plaintext (PT) that needs to be encrypted, and the PT will be divided into blocks by using the ASCII table [10]. This is shown on the right side of Fig. 5. For example, if a user enters “hello” as the PT, it will be converted into its hexadecimal representation at first which are 0x68 (h), 0x65 (e), 0x6c (l), 0x6c (l), 0x6f (o). Then it will be divided into  $5 \times 2 = 10$  blocks. Each block contains 4 bits. If the hexadecimal notation “0x6” represents 1 byte (8 bits), then there are two blocks for this character, and ‘6’ and ‘8’ will be decimal values for each block. After getting the corresponding blocks, decimal values of each block obtained from PT will be shown in Fig. 6:

The string you entered is:                    h        e        l        l        o  
 The hexadecimal notion of plaintext is:    06 08 06 05 06 0C 06 0C 06 0F  
 The decimal value for each block(4 bits) are: 6   8   6   5   6   12 6   12   6   15

**Fig. 6.** Decimal values of each block extracted from PT

These values are also considered to be orders in the protocol, they will be used to sort the cipher. The sender side will also use an “order mask” (OM) to record these values because they will be used in the decryption process. The sender side will create an array with 16 positions (the index is from 0 to 15) to record how many times each value appears. In the above example, number ‘5’ appears one time, number ‘6’ appears five times, number ‘8’ appears one time, number ‘12’ appears two times, and number ‘15’ appears one time. The result of this array should look like in Table 1. The values in the right column will be contained in the OM.

**Table 1.** Order mask, values are on the right

0	=>	0
1	=>	0
2	=>	0
3	=>	0
4	=>	0
5	=>	1
6	=>	5
7	=>	0
8	=>	1
9	=>	0
10	=>	0
11	=>	0
12	=>	2
13	=>	0
14	=>	0
15	=>	1

In short, we have introduced how to get resistances from RT by using  $A_i$  and  $C_j$ . These resistances will be seen as ciphers directly, and decimal value of PT blocks is known as order, which will be used for sorting these ciphers. Also the number of blocks extracted from PT will decide the number of ciphers. Such as in the example whose PT is “hello;” it has a total of 10 blocks, so the sender side will extract ten values from RT as well.

Then the sender side will decide which ten values in RT will be used as ciphers. Figure 5 explains this process. First, the sender side will extract ten values from the RT by using decimal values of “address” and “current”, which are extracted from blocks of MD.  $A_0$  and  $C_0$  will give us  $RT_{0,0}$ ,  $A_1$  and  $C_1$  will provide  $RT_{1,1}$ , and so on, until the  $A_9$  and  $C_9$  will give us  $RT_{9,9}$ . The subscripts of A and C represent which block they are using from MD; in this step, the sender reads from the first block of MD (subscript is 0). From now on, we will use  $R[i]$  to represent  $RT_{i,i}$  for simplicity. The ten values  $R[0]$  to  $R[9]$  are candidates for the cipher.

In the process of getting  $R[i]$  from  $A_i$  and  $C_i$ , the protocol will create another mask which is the “resistance mask” (RM) that will be responsible for recording which position in  $R[i]$  array contains the same resistance. The RM will only consist of either 0’s or 1’s. ‘0’ means the value in  $R[i]$  array corresponding to this position (0’s position in RM) is unique, and ‘1’ means the value in  $R[i]$  corresponding to this position (1’s position in RM) is repeated. The protocol will avoid repeated resistances as ciphers. So, the sender side will check and delete the same value in  $R[i]$  array and push ‘1’ to RM to label this position. To illustrate, let’s use the example. If the characters in PT are “hello;” and from  $R[0]$  to  $R[9]$ , the value of  $R[8]$  is the same as the value of  $R[0]$ , the sender side will use ‘1’ to represent this repeated condition in RM. Table 2 below illustrates this step.

**Table 2.** Resistance mask, and  $R[8]$  is the same as  $R[0]$

<b>R[i]</b>	R[0]	R[1]	R[2]	R[3]	R[4]
<b>RM</b>	0	0	0	0	0
<b>R[i]</b>	R[5]	R[6]	R[7]	R[8]	R[9]
<b>RM</b>	0	0	0	1	0

When sender side finds that  $R[8]$  is equal to  $R[0]$ , it will use ‘1’ for this position in RM and will not use  $R[8]$  as a candidate cipher. After discarding  $R[8]$ , the sender side will have a total of 9 candidates whose values are all unique; these nine values are the cipher to be sent by the sender side. But that is not enough; the protocol needs ten ciphers since the PT has been divided into ten blocks. As a result, the sender side will read one more value  $R[10]$  from the RT, which is got through  $A_{10}$  and  $C_{10}$ . Then the sender will check if  $R[10]$  is different from  $R[0]$  to  $R[9]$ . If so, the sender side will use  $R[10]$  as the 10<sup>th</sup> cipher and use ‘0’ to label this position in RM. The result after this operation is shown in Table 3.

In addition, the length of the RM is not fixed. In Table 3, only  $R[8]$  and  $R[0]$  are repeated, so the sender side used ‘1’ to represent that and then read a new value from the RM as a new cipher. Then the 10 values  $R[0]$  to  $R[10]$  ( $R[8]$  is dropped) will be included in our transit cipher ( $C'$ ). Table 4 illustrates this example:

**Table 3.** Read a new value R[10] from RT

	R[0]	R[1]	R[2]	R[3]	R[4]	
<b>RM</b>	0	0	0	0	0	
	R[5]	R[6]	R[7]	R[8]	R[9]	<b>R[10]</b>
<b>RM</b>	0	0	0	1	0	<b>0</b>

**Table 4.** Build transit cipher  $C'$  from R[0] to R[10]

	R[0]	R[1]	R[2]	R[3]	R[4]
<b>Transit Cipher <math>C'</math></b>	$C'_0$	$C'_1$	$C'_2$	$C'_3$	$C'_4$
	R[5]	R[6]	R[7]	R[9]	R[10]
<b>Transit Cipher <math>C'</math></b>	$C'_5$	$C'_6$	$C'_7$	$C'_8$	$C'_9$

But there is the possibility that the new value R[10] is also repeated with other values from R[0] to R[9] which needs to use another ‘1’ in the RM, and then read a new value R[11] from the RT. In this case, the length of the RM will be 12. The length of the cipher will be still 10, which means the length of the RM may change, but the length of the cipher is fixed, only depending on the number of blocks of PT. Also, the sender side has got 10 decimal values from blocks of PT; the sender side will call these values “order” (Od) because it will use these values to sort  $C'$ . The Od and  $C'$  are shown in Table 5. The values of Od used here are from Fig. 6.

**Table 5.** Table contains order and transit cipher

<b>Od</b>	6	8	6	5	6
<b>Transit Cipher <math>C'</math></b>	$C'_0$	$C'_1$	$C'_2$	$C'_3$	$C'_4$
<b>Od</b>	12	6	12	6	15
<b>Transit Cipher <math>C'</math></b>	$C'_5$	$C'_6$	$C'_7$	$C'_8$	$C'_9$

The next step is to sort  $C'$  according to Od and then get the final cipher (FC). Figure 7 illustrates it.

As shown, the protocol will sort the transit cipher in the ascending order according to the values in Od array. For example, number ‘5’ is the smallest in Od array, and the transit cipher corresponding to ‘5’ is  $C'_3$ . So  $C'_3$  will be in the first position in the final cipher FC. Then, number ‘15’ in Od array is the largest value, so the value  $C'_9$  corresponding to it should be in the last position of FC. Finally, after sorting the  $C'$  based on Od array, the result will be shown in Table 6.

So far, we have got the final cipher FC, the resistance mask RM, and the order mask OM. To implement decryption on another side, they all need to be

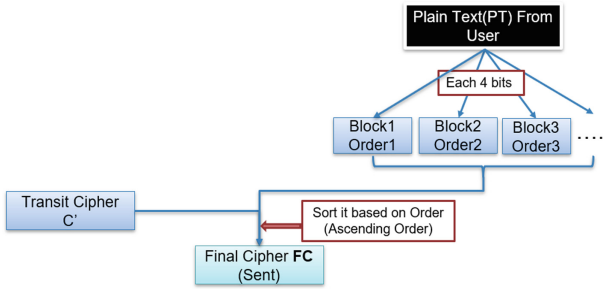


Fig. 7. How to get final cipher (FC)

Table 6. Final Cipher in correct order

<b>Od</b>	5	6	6	6	6
<b>Final Cipher FC</b>	$FC_0 = C'_3$	$FC_1 = C'_0$	$FC_2 = C'_2$	$FC_3 = C'_4$	$FC_4 = C'_6$
<b>Od</b>	6	8	12	12	15
<b>Final Cipher FC</b>	$FC_5 = C'_8$	$FC_6 = C'_1$	$FC_7 = C'_5$	$FC_8 = C'_7$	$FC_9 = C'_9$

sent to that side. FC can be sent directly and safely, but it is not safe to send two masks in the same way without any protection.

The protocol will implement XOR (Exclusive or) operation on these two masks and half of MD separately in order to ensure the safe transmission of information. This step is shown in Fig. 8.

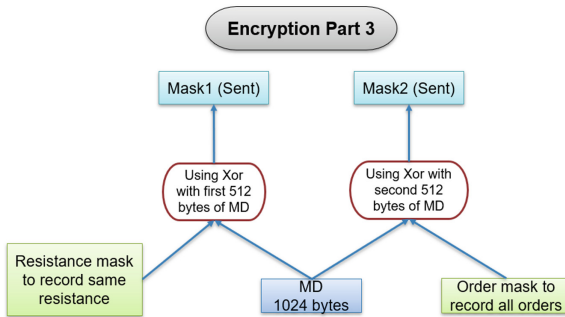


Fig. 8. Encryption part3

The protocol uses XOR function here because XOR is an involutory function, which means if the protocol applies XOR twice, it can get the original RM and OM back during decryption [12]. For RM, the protocol will use it to perform XOR operation with the first half of MD, and OM uses the second half of MD to perform XOR operation. After that, “Mask1” and “Mask2” will be generated,



which are both 512 bytes long. “Mask1” and “Mask2” are different from the two initial masks OM and RM. But RM and OM are shorter than half of MD, so when doing XOR operation, the system will automatically fill in some bytes at the end of these two masks to let them have 512 bytes. Finally, “Mask1” and “Mask2” will be sent to another side. That is all for encryption.

### 4 Decryption

Upon receiving “Mask1”, “Mask2”, and FC, the receiver side will use this information to restore the same PT generated in encryption. In environmental setup section, we have mentioned that the RN and PW will be known by both sides. So, the receiver side can get the same MD by doing the same operation that the sender side have done with encryption. And the “Mask1” and “Mask2” are both obtained by doing XOR operation with half of MD. XOR is an involutory function. If the protocol applies XOR twice, it will get the original thing back. Figure 9 illustrates that the receiver side can retrieve the RM and OM back by implementing XOR operation with MD again.

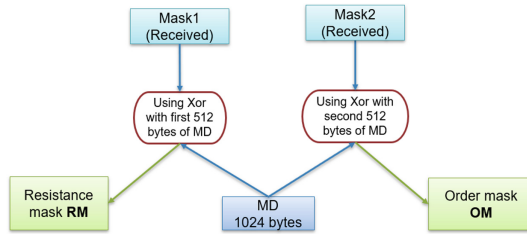


Fig. 9. Retrieving OM and RM

In Fig. 9, after the XOR operation is performed on the “Mask1” and the first half of MD, the receiver side will get the RM back. The problem here is that the receiver side does not know which part in the RM is for regulating ciphers. Taking the previous example whose PT is “hello,” the actual length of RM is 11, but after the XOR operation in encryption, its length becomes 512 bytes. So, the receiver side gets a 512-byte RM, and does not know which part is useful.

The receiver side will use another method to solve this problem. Because both sides have the same MD and share the same  $A_i$  and  $C_j$ ,  $A_0$  and  $C_0$  will provide  $R[0]$  on both sides,  $A_1$  and  $C_1$  will provide  $R[1]$  on both sides and so on. As some unique  $R[i]$ ’s will be used as  $C'$  directly, then FC will be generated based on sorting  $C'$ . Table 7 shows the relationship between  $C'$  and FC.

As a result, if the  $R[i]$  is contained in  $C'$ , it must also be found in FC. The receiver side will loop from  $A_0$  and  $C_0$  and get its corresponding  $R[0]$  from the RT, and then check if  $R[0]$  appears in FC. If so, it will continue to search until an  $R[m]$  value is not found in FC, it will stop searching, and elements from  $R[0]$

**Table 7.** Relationship between FC and  $C'$

$C'$	$C'_3$	$C'_0$	$C'_2$	$C'_4$	$C'_6$
<b>FC</b>	$FC_0 = C'_3$	$FC_1 = C'_0$	$FC_2 = C'_2$	$FC_3 = C'_4$	$FC_4 = C'_6$
$C'$	$C'_8$	$C'_1$	$C'_5$	$C'_7$	$C'_9$
<b>FC</b>	$FC_5 = C'_8$	$FC_6 = C'_1$	$FC_7 = C'_5$	$FC_8 = C'_7$	$FC_9 = C'_9$

to  $R[m-1]$  must be  $C'$ 's elements. Next, the receiver side will count how many elements are there from  $R[0]$  to  $R[m-1]$ , and the result will give the receiver side the useful part in the RM. This useful part will be named ERM. The receiver will use ERM to get  $C'$ . Using the same example whose PT is “hello”, the ERM obtained above is shown in Table 8.

**Table 8.** ERM Table

0	0	0	0	0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---

And  $R[i]$  got from the RT will be shown in Table 9:

**Table 9.** Resistances from RT

$A_0, C_0 \rightarrow R[0]$	$A_1, C_1 \rightarrow R[1]$	$A_2, C_2 \rightarrow R[2]$	$A_3, C_3 \rightarrow R[3]$	$A_4, C_4 \rightarrow R[4]$
$A_5, C_5 \rightarrow R[5]$	$A_6, C_6 \rightarrow R[6]$	$A_7, C_7 \rightarrow R[7]$	$A_8, C_8 \rightarrow R[8]$	$A_9, C_9 \rightarrow R[9]$
$A_{10}, C_{10} \rightarrow R[10]$	$A_{11}, C_{11} \rightarrow R[11]$	$A_{12}, C_{12} \rightarrow R[12]$	.....	

There will be a lot of resistances extracted from the RT, but since the length of the ERM is 11, the receiver side will only use 11 values from  $R[0]$  to  $R[10]$ . The value in the ERM determines which R in Table 9 can be used as  $C'$ . For example, the corresponding value of  $R[0]$  is ‘0’ in Table 8, so  $R[0]$  will be pushed to  $C'$ . But if the corresponding value is ‘1’ that value will be discarded such as  $R[8]$ . Then other values from  $R[0]$  to  $R[10]$  will become elements of  $C'$ . The  $C'$  the receiver side gets here is the same as the previous one when encrypting. Figure 10 provides a summary of the above steps.

After getting FC and  $C'$ , the receiver side will get the OM by doing XOR operation on “Mask2” and second half of MD. The first 16 values in the output are useful and should be same as the right side in Table 10.

In Table 10, see following page, the numbers on the left side are indices, and the numbers on the right side are the content really included in the OM. For example, “5 => 1” means number ‘5’ appears one time, “6 => 5” means number ‘6’ appears 5 times, until “15 => 1” means number ‘15’ appears one time. The result in Table 11 turns the number of occurrences into actual numbers. Because number ‘6’ appears five times, there will be five 6’s in the table.

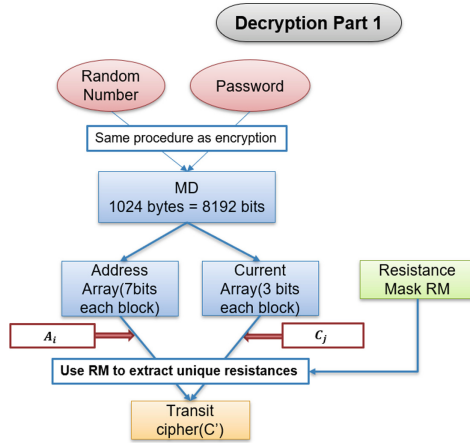


Fig. 10. Decryption part1

Table 10. Order mask, values are on the right

0	=>	0
1	=>	0
2	=>	0
3	=>	0
4	=>	0
5	=>	1
6	=>	5
7	=>	0
8	=>	1
9	=>	0
10	=>	0
11	=>	0
12	=>	2
13	=>	0
14	=>	0
15	=>	1

Table 11. Random order array RO

5	6	6	6	6
6	8	12	12	15

The receiver side thinks of it as an array with 10 elements, and we will call it RO, which means “random order”. The reason is, compared with Table 5, the elements in RO array are same as the elements in Od array but in a different order. If the receiver side wants to get a correct PT, it has to turn it into the correct order. The plan implemented is shown in Fig. 11.

The receiver side will make a pair of  $C'$  and RO as described in Fig. 11. And then it will find each value of  $C'$  in this pair and get its corresponding  $RO_i$ . The example is shown in Table 12. Numbers in the “Corresponding RO in the *Pair*” row are the same as the numbers in Od array, and they are in the same order.

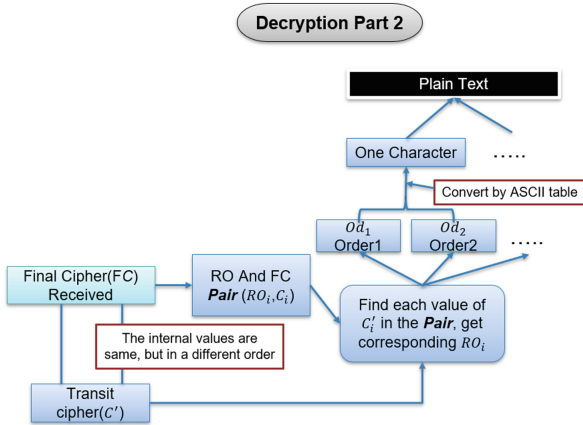


Fig. 11. Decryption part2

Table 12. Get correct Od array by searching  $C'$  in the *Pair*

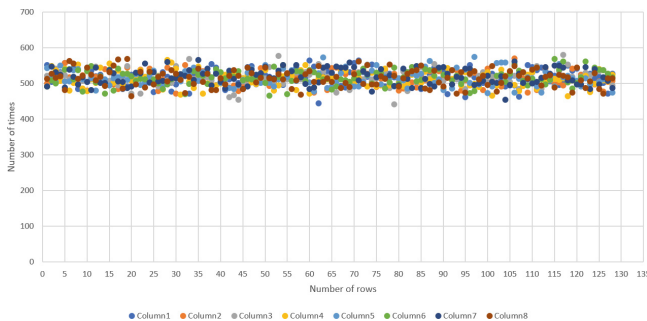
<b>(FC, RO) Pair</b>	$(FC_0, 5)$	$(FC_1, 6)$	$(FC_2, 6)$	$(FC_3, 6)$	$(FC_4, 6)$
<b>Relationship between FC and <math>C'</math></b>	$FC_0 = C'_3$	$FC_1 = C'_0$	$FC_2 = C'_2$	$FC_3 = C'_4$	$FC_4 = C'_6$
<b>(FC, RO) Pair</b>	$(FC_5, 6)$	$(FC_6, 8)$	$(FC_7, 12)$	$(FC_8, 12)$	$(FC_9, 15)$
<b>Relationship between FC and <math>C'</math></b>	$FC_5 = C'_8$	$FC_6 = C'_1$	$FC_7 = C'_5$	$FC_8 = C'_7$	$FC_9 = C'_9$
<b>Transit Cipher <math>C'</math></b>	$C'_0$	$C'_1$	$C'_2$	$C'_3$	$C'_4$
<b>Corresponding RO in the <i>Pair</i></b>	6	8	6	5	6
<b>Transit Cipher <math>C'</math></b>	$C'_5$	$C'_6$	$C'_7$	$C'_8$	$C'_9$
<b>Corresponding RO in the <i>Pair</i></b>	12	6	12	6	15

So far, the receiver side has got Od array back. Every two values in Od array represent one character and can be converted into their corresponding character through ASCII table. For example, the first two numbers in Od are ‘6’ and ‘8’, then the receiver will combine them into hexadecimal notation, which is “0x68”. The character corresponding to “0x68” in ASCII table is ‘h’. Finally, the receiver side will convert the ten numbers in RO array into five characters and get the original PT - “hello”. That is all for decryption.

## 5 Security Evaluation

It is important that other people cannot retrieve cipher values by observing cell usage in RT. So, in this section, we discuss and verify the security of the protocol by measuring cell usage. We have introduced that ciphers came from the RT directly, where the RT is a two-dimensional array with 128 rows and 8 columns, a total of 1,024 cells. The protocol uses addresses and currents which are obtained from MD to determine specific cell from RT. Then these cell values will be treated as cipher after sorting. The problem is that if we encrypt different messages, the values extracted from the RT each time are the same, which means that ciphers used are almost the same every time. Hackers can decipher information by observing cipher usage. But if every data in the RT has the opportunity to be used, the ciphers generated during encryption will be different, which will increase the difficulty of cracking our message. So, we will measure the cell usage in the form of a statistical chart below [9].

Figure 12 is the Scatter graph after doing encryption and decryption 1,000 times with 140 random characters. 140 characters is the maximum value that the protocol can implement encryption and decryption in the case where the MD length is 8,192 bits. We want to test the performance of the protocol in extreme states. There are 128 rows and 8 columns in the RT, which constitute a total of 1,024 cells. In Fig. 12, the horizontal axis represents the row number, the vertical axis represents the number of times used, and the points with eight different colors represent the column number. So, Fig. 12 shows how many times each cell is used. As Fig. 12 demonstrates, each cell has been called approximately 520 times, which means each cell has been used a similar number of times without the extremely unsafe situation where some cells have been used tens of thousands of times, and some have been used only a few times.



**Fig. 12.** Scatter graph for each cell in RT, 1,000 times, 140 characters

Figure 13 is similar to Fig. 12, but it does not show the usage for each cell. It shows the usage of each row. From Fig. 12, we can see that each cell is used about 520 times, and there are 8 cells in a row, so it is expected that the number of uses per row should be around 4,160. The data in Fig. 13 confirms our expectations.

In order to get more accurate results, we performed more tests. As shown in Fig. 14 and 15 below, we used 140 random letters for encryption and decryption 10,000 times.

The data in Fig. 14 has the similar meaning as in Fig. 12, the only difference is the number of tests. Under 10,000 tests, we can see that most cells are used around 5,000 times, and the smallest value in the graph is still greater than 4,900. There is no case where the cell is not used or is rarely used.

Then, in Fig. 15, the data reflects the number of times each row is called. Again, all of the values in the column chart are larger than 40,000. The largest value in the chart is about 41,800. The gap is within acceptable limits.

By verifying the encryption and decryption procedure of 140 characters 1,000 and 10,000 times, we found that our method has very high and similar utilization rates for different cells in the RT; even under 10,000 tests, there are no special cases. In general, we can conclude that the protocol is safe, and it is not easy for hackers to decipher the message by discovering cell usage.

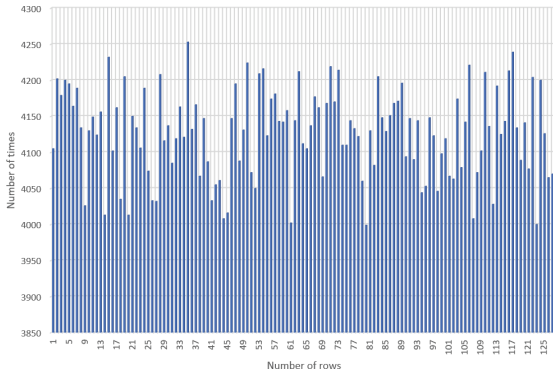


Fig. 13. Column chart for each row in RT, 1,000 times, 140 characters

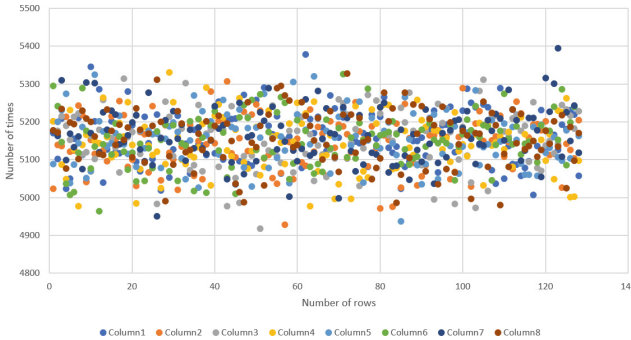
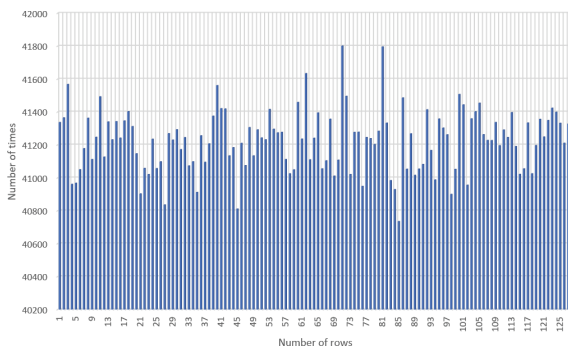


Fig. 14. Scatter graph for each cell in RT, 10,000 times, 140 characters



**Fig. 15.** Column chart for each row in RT, 10,000 times, 140 characters

## 6 Conclusion

This paper proposed an extended protocol based on memristor arrays to perform encryption and decryption without using any keys. First, an MD is generated on the sender side and the receiver side using the hash function. Then, both sides import the resistance table, which is obtained from the memristor arrays. Next, the plaintext to be encrypted is divided into small blocks. Finally, the protocol uses decimal values of these blocks to sort the ciphers extracted from resistance table. In the evaluation section, this paper also provided security proof for this protocol in the case of multiple tests. Going forward, we want to implement the protocol into hardware because hardware design was not involved with our study. For example, we can build two communicating devices based on memristor arrays [5,14]. If so, ciphers generated can only be decrypted by the same memristors stored in separate devices.

## References

1. Baracaldo, N., Bathen, L.A.D., Ozugha, R.O., Engel, R., Tata, S., Ludwig, H.: Securing data provenance in Internet of Things (IoT) systems. In: International Conference on Service-Oriented Computing, pp. 92–98. Springer (2016)
2. Boutin, C.: NIST releases SHA-3 cryptographic hash standard. NIST information technology laboratory (2015)
3. Cambou, B., Orlowski, M.: Design of PUFs with RERAM and ternary states. In: Proceedings of the 11th Annual Cyber and Information Security Research Conference, Oak Ridge, TN, USA, pp. 5–7 (2016)
4. Cambou, B.: A XOR data compiler: combined with physical unclonable function for true random number generation, pp. 819–827, July 2017
5. Cambou, B., Assiri, S., Hely, D.: Memristors to design keyless encrypting devices. ACM J. Emerg. Technol. Comput. Syst. (JETC) (2018, submitted)
6. Cambou, B.F.: Design of true random numbers generators with ternary physical unclonable functions. Adv. Sci. Technol. Eng. Syst. J. **3**, 15–29 (2018)

7. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): a vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **29**(7), 1645–1660 (2013)
8. Han, J.-W., Park, C.-S., Ryu, D.-H., Kim, E.-S.: Optical image encryption based on XOR operations. *Opt. Eng.* **38**, 47–54 (1999)
9. Kahate, A.: *Cryptography and Network Security*. Tata McGraw-Hill Education, New York (2013)
10. Kaushik, A., Kumar, A., Barnela, M.: Block encryption standard for transfer of data. In: 2010 International Conference on Networking and Information Technology, pp. 381–385. IEEE (2010)
11. Kocher, P., Jaffe, J., Jun, B., Rohatgi, P.: Introduction to differential power analysis. *J. Cryptograph. Eng.* **1**(1), 5–27 (2011)
12. Li, C., Li, S., Alvarez, G., Chen, G., Lo, K.-T.: Cryptanalysis of two chaotic encryption schemes based on circular bit shift and XOR operations. *Phys. Lett. A* **369**(1–2), 23–30 (2007)
13. Roman, R., Najera, P., Lopez, J.: Securing the Internet of Things. *Computer* **9**, 51–58 (2011)
14. Williams, R.S.: How we found the missing memristor. In: *Chaos, CNN, Memristors and Beyond: A Festschrift for Leon Chua With DVD-ROM*, Composed by Eleonora Bilotta, pp. 483–489. World Scientific (2013)





# Recommendations for Effective Security Assurance of Software-Dependent Systems

Jason Jaskolka<sup>(✉)</sup>

Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada  
jason.jaskolka@carleton.ca

**Abstract.** Assuring the security of software-dependent systems in the face of cyber-attacks and failures is now among the top priorities for governments and providers of electric, financial, communication, and other essential services. Practical and foundational solutions for systematic, secure, and trustworthy system development are needed to support developers, regulators, and certification bodies in providing assurance that security threats faced by the software systems used in these environments have been adequately mitigated. Using recent experiences reported in the literature as a basis, we discuss the challenges of providing security assurance for software-dependent systems. We also explore the barriers to adoption of existing approaches and techniques which can play an important role in security assurance efforts. Ultimately, we present a set of recommendations which outline a collection of follow-on research directions that can advance the state-of-the-art and support the development of more effective security assurance solutions for critical software-dependent systems.

**Keywords:** Security assurance · Security evaluation · Security-by-design · Software certification · Security requirements

## 1 Introduction

Nowadays, many of our most critical systems, such as those operating in national security, financial, transportation, communication, and healthcare domains, are increasingly dependent on complex software systems to provide their critical services. Over time, these systems have grown to become very large, complex, and connected, and have begun operating in open environments, which inevitably leaves them susceptible to a wide range of security vulnerabilities and threats. This has led to many of these systems being classified as *high-assurance software-dependent systems*, for which compelling evidence that the system delivers its services in a manner that satisfies certain critical properties such as security, safety, survivability, fault tolerance, and real-time performance, is required [1].

Due to the nature of these systems, attacks or failures can have disastrous consequences and many destabilizing effects on dependent systems, humans, the

environment, and/or the economy [2]. As a result, engineering high-assurance systems tends to place a strong emphasis on rigorous requirements and specifications, verification and validation, risk management, and certification. Guarantees, backed by a verifiable body of evidence that the system will function exactly as intended at all times, are expected [1]. As such, mechanisms capable of identifying and addressing the root causes of attacks or failures, and ways in which we can provide security assurance of such complex systems are growing in importance and are sought after by many research and development programs initiated by governments and providers of critical services (e.g. [3,4]).

Security assurance of critical software-dependent systems demands evidence-driven, built-in, systematic approaches for software development capable of providing early evidence of mitigating security risks, attacks, and vulnerabilities. It is often difficult, costly, and time-consuming to gather sufficient evidence supporting assurance claims about the security of a system after it has already been built and deployed [5]. Thus, it is important to incorporate security evaluation and assurance into the system development lifecycle (SDLC). An argument for the security of a system should be developed alongside the system it supports.

In this paper, we explore the challenges of providing security assurance for software-dependent systems. Our discussion is motivated by recent experiences in developing a security assurance cases reported in the literature. In particular, we highlight technology gaps and describe the role that existing approaches and techniques can play to fill these gaps and address the challenges. We also provide insight into the limited adoption of these existing approaches and techniques in providing security assurance. Our novel contribution lies in the presentation of seven recommendations for advancing the state-of-the-art and reducing the barriers to adoption of existing approaches for supporting security assurance efforts. While some of these recommendations may seem obvious to a security expert, they are relevant to many development teams and practitioners who may not be experts in security, but want or need improved security assurance in their SDLC. Furthermore, these recommendations outline a set of follow-on research directions that can stimulate the development of more effective security assurance solutions.

This paper is organized as follows: Sect. 2 gives the necessary background related to security assurance. Section 3 documents the primary challenges and obstacles faced when trying to assure the security of critical software-dependent systems. Section 4 describes the role that existing approaches and techniques can play in addressing the challenges described in Sect. 3. Section 5 provides insight into the barriers to adoption of existing approaches and techniques for supporting security assurance efforts, and proposes recommendations for advancing the discipline. Lastly, Sect. 6 concludes and outlines a research agenda which can drive the development of more effective security assurance solutions.

## 2 Security Assurance

Engineering high-assurance software-dependent systems goes beyond evaluation and aims to guarantee vital system properties supported by verifiable (often

mathematical) evidence that they will function exactly as intended and designed at all times, even when experiencing an attack or widespread failure. *Security assurance* refers to the process of proving or guaranteeing that a system is designed and developed so that it operates at level of security commensurate with the potential risks and associated losses incurred if the system experiences an attack or failure [6]. This involves examining and evaluating many different aspects of a system and its development processes and activities with respect to security. These aspects include the construction of the system and its software, the quality assurance of artifacts produced throughout development, the management of the deployed system, and the ways in which an organization manages the overall development process [7]. However, due to the high costs and required levels of technical expertise, few organizations have the necessary resources to adequately build secure high-assurance software-dependent systems.

Traditionally, high-assurance system engineering approaches were adopted and used to produce software systems for mission-critical national security purposes, where failure is unacceptable and the high costs and levels of technical expertise can be justified. Only recently, has the importance of security assurance been discussed for systems outside traditional mission-critical domains. Software systems are used by governments, industries, and general users in a widening array of domains. Some of these domains are not typically thought of as “critical” and requiring absolute guarantees. For example, “smart” appliances in the home are now requiring unprecedented levels of assurance as their criticality is only now emerging [8]. The conventional testing and evaluation approaches that covered only a subset of potential issues and problems are no longer sufficient and do not provide the evidence and guarantees required for high-assurance [7].

## 2.1 Security Assurance Models and Frameworks

Often, systems will be assigned a *security assurance level* which determines the specific types of evidence and acceptance criteria required for a system to be considered secure. Different security assurance levels are prescribed in various standards and guidelines for different kinds of systems within different jurisdiction (e.g. [9–11]). Based on the evidence generated from these evaluations, a convincing argument must be made to show that the given system provides the required level of security.

Recently, a number of models and frameworks for describing and guiding the activities related to security engineering, evaluation, and assurance have been proposed. The Open Web Application Security Project’s (OWASP) Software Assurance Maturity Model (SAMM) [12] aims to help organizations to formulate and implement a risk-centric software security strategy. SAMM facilitates the evaluation of existing security practices, and the development of a balanced and iterative security assurance program with defined measurements for outcomes of security-related activities in support of security assurance. The Software Assurance Framework (SAF) [13] defines important security practices for process management, project management, engineering, and support. The SAF aims to provide measurable outcomes indicating that security has been

appropriately addressed in the requirements, design, construction, and testing to establish confidence that security is sufficient. NIST’s System Security Engineering Framework [14] defines, bounds, and focuses both technical and nontechnical security engineering activities and tasks towards the achievement of stakeholder security objectives. It aims to facilitate a coherent, well-formed, evidence-based case that demonstrates that those objectives have been achieved in the system development. In particular, the framework emphasizes an integrated, holistic security perspective across all stages of the SDLC. Similarly, NIST’s Framework for Improving Critical Infrastructure Cybersecurity [15] provides a set of security activities, outcomes, and references meant to be applicable across all critical infrastructure sectors. Overall, the framework aims to explicitly incorporate security activities and considerations of security risks as part of the organization’s risk management processes.

Each of these models and frameworks can be used to assist practitioners in understanding what is required for adequately assuring the security of critical software-dependent systems. However, with the exception of the SAF, they often focus more on the risk management practices of the organization developing the system, rather than the system itself.

## 2.2 Security Assurance Cases

An assurance case is a reasoned and compelling argument, supported by a body of evidence, that a system, service, or organization will operate as intended for a defined application in a defined environment [16]. Assurance cases represent combinations of structured claim decompositions in terms of high-level claims, sub-claims, and supporting evidence related to the design and implementation, and an argument—which is an informal proof—demonstrating that the claim decomposition supported by the evidence will achieve a required system property such as safety, security, or reliability [17].

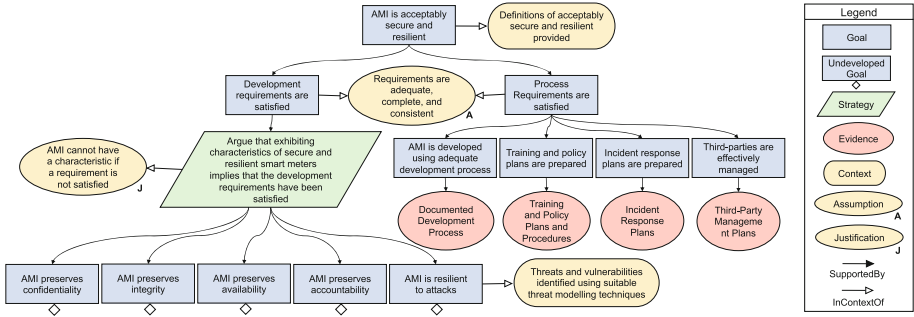
Assurance cases have been adopted for arguing system safety with much success (e.g. [18–20]). However, the adoption and development of assurance cases for arguing system security is still in its infancy. Even so, there has been some important groundwork established in the literature [21,22]. This work provides basic guidance and advice on creating security assurance cases and was incorporated as part of the “Build Security In” initiative in the United States [23].

## 3 Challenges Faced When Providing Security Assurance

In this section, we discuss the primary challenges, obstacles, and technology gaps faced when providing security assurance for software-dependent systems.

### 3.1 An Experience in Providing Security Assurance

In recent work [24], an assurance case template was developed for arguing the security and resilience of advanced metering infrastructure (AMI)—a high-assurance software-dependent system that is part of an energy grid. The size,



**Fig. 1.** Security assurance case template for advanced metering infrastructure using goal structuring notation [24].

complexity, and connectedness of AMI leaves it vulnerable to a wide range of threats that can lead to direct, or indirect, attacks which could disrupt energy supplies and increase the overall risk to the energy sector [25].

A large proportion of the threats to which AMI is vulnerable (e.g., eavesdropping, impersonation, etc.) are related to compromise, forgery, or denial of use of critical services and information. This means that assurance of AMI security must be dependent on adequate evidence that the system maintains the confidentiality, integrity, availability, accountability, and resilience of its critical services and information [26]. These characteristics form the basis of the high-level goals and claims of the developed assurance case.

Recognizing that security assurance is concerned with both the product that is developed and the process by which it is developed, the high-level goals and claims of the assurance case are decomposed based on the primary security and resilience requirements for AMI and are divided into two primary categories: development requirements and process requirements. This is shown in the developed assurance case template (using the goal structuring notation [16]) provided in Fig. 1. *Development requirements* prescribe specific protection mechanisms and functional requirements related to the development of the AMI system (i.e., the specific product). *Process requirements* are concerned with the robust and resilient design of core functionalities and infrastructures, conformance to implementation standards, adequate testing, verification and validation, and suitable specification and documentation for all system aspects, including the physical systems, information systems, and networks [24]. Process requirements are not necessarily tied to the specific functionality of the systems and components, but rather to the preparedness to deal with security and resilience challenges throughout the system’s lifespan.

To support any claim that an AMI system is secure, evidence of the satisfaction of each requirement is needed. Figure 1 gives an idea of what an assurance case for supporting the top-level claim *AMI is acceptably secure and resilient* looks like. The developed assurance case clearly illustrates the relationship between the security and resilience requirements of AMI, and the assurance

argument represented in the assurance case. The argument is centered on the premise that if there is sufficient evidence that the security and resilience requirements of AMI are satisfied, then the AMI can be considered acceptably secure and resilient.

We will use the experience in developing the assurance case template in [24] as a representative concrete example to focus our discussion. Similar experiences and examples can also be found in the literature (e.g. [5, 19, 27–29]) and we will also draw from these experiences to support our arguments. In the following subsections, we discuss the primary challenges, obstacles, and technology gaps faced when providing security assurance for software-dependent systems.

### 3.2 Tradeoffs with Security Requirements

Due to its nature, security is a system quality that often cannot be considered in isolation from other system qualities such as safety, reliability, performance, and usability. Because of this, tradeoffs among competing, and sometimes contradictory, system qualities need to be made during system development [5]. To adequately support security assurance efforts, the justification and rationale for each of these tradeoffs and design decisions needs to be sufficiently documented so that it can be effectively evaluated and submitted as evidence for a security assurance case, and to support the maintenance of system security as the threat landscape changes during system evolution.

As evidenced in Sect. 3.1, the development of a security assurance case is deeply rooted in understanding the security requirements for the system being developed. However, when designing software-dependent systems, the focus is often targeted at the functional requirements, ensuring that the system being developed will do what it is supposed to do. Because of this, tradeoffs with respect to non-functional security requirements that will help to prevent “bad things” from happening are made without the full consideration or understanding of the short- and long-term effects of those decisions. Furthermore, these tradeoffs often lack sufficient documentation, which severely weakens the assurance case that can be developed for the system. Methods to support developers in making and documenting informed decisions about important tradeoffs with security requirements through rigorous analysis are required to support security assurance efforts.

### 3.3 Coping with Size and Complexity

Many critical systems are already extremely complex, and are rapidly becoming more so as the number of devices and connections constituting these systems continues to increase. This problem is exacerbated by the growth of the Internet of Things (IoT), where the number of devices that are part of critical systems is growing substantially. For example, financial, communication, and healthcare systems, as well as the common household are all comprised of vastly more devices today than only a few years ago. Due to this size and complexity, a complete and holistic understanding of the system once all of its components

are all integrated is often out of reach. However, for effective security assurance, such a holistic view of the system is required to account for the complex landscape of the interactions among the system components, as well as with the ever-changing environment in which the system operates. This is a significant challenge for security assurance as it means that taken together, many high-assurance software-dependent systems are quite literally too complex to be completely understood—by a person, a team of people, or by a computer model.

Furthermore, the large size and complexity of a system means that there is often an enormous amount of evidence that needs to be managed when developing a security assurance case. Even for a relatively small system with a limited scope such as that described in Sect. 3.1, an assurance case quickly grows to be quite complex [24]. This issue has also been noted by other experiences reported in the literature (e.g. [5, 19, 27–29]). This is a result of the hierarchical structuring of the assurance case argument. Managing the complexity and scope of the problem to ensure that all relevant aspects of the argument are covered with respect to the vast number of requirements and goals that must be considered is difficult, especially when those goals and requirements become intertwined with one another. Furthermore, the development of assurance cases is still mostly manual, and they are difficult to scale up for complex systems, or down for details in the required level of formality, which adds to the challenge.

To cope with size and complexity, systems need to be built with simplicity in mind and evidence of security needs to be generated and gathered from the outset [30]. The amount of evidence should be proportional to the level of assurance required by the given system [5]. Clear and concise standards and guidelines specifying these levels of assurance, and improved and automated tools and methods to support security analysis, as well as for handling the massive amount of generated and gathered evidence is needed.

### 3.4 Lack of Security Evaluation Approaches

Accurately assessing and evaluating system security is essential for providing security assurance. However, the undecidability of security has long plagued security experts and engineers that are tasked with building and securing critical systems and networks. The intractability of performing a complete coverage of tests to guarantee that a system is secure leads to the problem of trying to show that a system is “secure enough.” Deciding what “secure enough” means and how it is affected by the system application context is not always straightforward. Analysis that incorporate the exploitability and the potential impact of specific attacks on systems is required to make accurate determinations. Current approaches for providing these kinds of risk assessments and classifications (such as those described in [31]) are not standardized, and are often far too subjective, relying on (sometimes false) assumptions about the system and the attackers.

Generally speaking, the strength of a security assurance case depends on the evidentiary foundation supporting the security argument. The ability to measure and evaluate system security properties is needed to produce sufficient evidence of security early in the development of a system. Sufficient and verifiable evidence

is needed to support the claims in assurance case arguments. However, despite an evolving research effort, there is still a lack of widely accepted approaches for evaluating and measuring security. As long as this remains to be the case, providing strong security assurances will remain difficult.

### 3.5 Complications with Legacy Systems

Many critical systems involve a complex mix of insecure legacy systems and new technologies whose security properties are not proven. A large number of these systems are now being used in ways in which they were never intended which contributes to the vast array of security issues plaguing these systems. The development costs to bring these legacy systems up-to-date, or to perform a system overhaul presents a significant barrier to providing security assurance. Instead of swallowing these costs, owners and operators of these systems scramble to patch together security solutions that can help to prevent some kinds of attack or compromise. This presents a significant challenge for evaluation and certification bodies as it becomes very difficult to determine the evidence that is required for such patchwork systems to support any kind of security assurance argument. To make matters worse, even finding evaluators or certification bodies capable of understanding the legacy systems and components can be a major barrier to the security assurance activity.

Beyond the issues related to system size and complexity previously discussed in Sect. 3.3, legacy systems and components are often not well-documented in terms of their behaviours and the processes by which they were developed. This results in a very weakly (or non-existent) documented development process, which as seen in Fig. 1, is a critical piece of evidence for supporting claims that process requirements are satisfied. This emphasizes the need to incorporate security *early and often* in the development of any system to simplify the generation, gathering, management, and evaluation of the required evidence. Without the presence of well-documented and accepted standards, guidelines, and/or requirements, the development of effective security assurance solutions for systems depending on legacy systems and components will remain challenging.

### 3.6 Dealing with Third Party Suppliers

There is now an increasing dependence on the trustworthiness of the components that are being used to build systems. These components are often designed and manufactured by third-party suppliers. There is usually an assumption that the system, while it may be complex, poorly understood, or antiquated, is at least built with components that can be trusted to operate as advertised. However, this is proving less and less to be the actual case.

System development now relies on a vast and growing array of suppliers of software, hardware, and component systems. Despite that testing is routinely done before integrating individual system components, much of that testing is targeted at uncovering accidental defects, as opposed to malicious flaws that



may have been intentionally inserted to enable future attacks against system security. As a result, and as shown in Fig. 1, evidence supporting claims pertaining to the management of third-party suppliers is a significant part of the security assurance case argument. To adequately support these claims, the tacit assumption that all components received through the supply chain are trusted and secure needs to be forgotten. The main challenge here is that more often than not, when a component is received from a third-party supplier, there is no documented assurance that the received component is acceptably secure for integration into new systems. Consequently, this significantly weakens the security assurance argument for the new system. Therefore, an effort towards developing and delivering security assurance cases as part of the final product is required.

### 3.7 Compositionality of Security

Systems that are secure in isolation may not be secure when composed together. Emergent behaviours and feature interaction [32] make it difficult to provide security assurance as we are required to reassess the system each time we integrate a new component. It becomes very challenging to reuse assurance or certification efforts from previous versions or parts of systems due to this compositionality issue. For example, we do not currently have an effective way in which we can reuse the security assurance case for AMI described in Sect. 3.1 in a larger system such as an entire smart grid. This issue is exacerbated when thinking about all of the previously mentioned issues related to coping with the size and complexity of systems, dealing with legacy systems, and managing the security of components acquired from third-party suppliers. Beyond the possibility of ad hoc approaches, we would have to reassure the entire system which can become prohibitively costly and time-consuming.

The inability to quickly and easily reassess assurance claims is a major challenge for the assurance community. There is a need for more modular, incremental, reusable, and compositional approaches for providing security assurance for software-dependent systems.

### 3.8 It's About More Than Just a Product

Providing security assurance is not just about assuring the product, but also the process by which the product has been developed, and the people involved. For example, in Sect. 3.1, the developed assurance case template clearly indicates the importance and role of process requirements in the claim decomposition and argument structure. The importance of incorporating the people and process in assurance arguments has also been noted by other experiences (e.g. [5, 13, 14, 19, 27, 29]). However, generating and measuring suitable evidence of the people and process in the development of a high-assurance software-dependent system is not also easy to come by. Moreover, such requirements are not always explicitly or clearly stated (or even thought about) which often makes this portion of an assurance case quite weak.

A better awareness of the importance of documenting the process used in the development of a product for security assurance is needed. This extends to the explicit inclusion and consideration of process requirements from early stages of development so that sufficient evidence can be generated and gathered. It also encompasses the need to adequately train developers, as well as evaluators, regulators, and certification bodies, with the nature of the evidence that is required to support a security assurance case, and how to facilitate the gathering, generation, and preservation of such evidence [5]. The ability to demystify the idea that security assurance is just another set of processes to follow that puts a strain on project budgets and timelines, when “*all the customer cares about is the product*” is urgently required.

## 4 Ways to Address the Challenges

Failure to incorporate system-level security measures and considerations from early stages in system development can have a significant impact on the ability to provide assurance of such system properties. In this section, we argue that a significant number of the challenges discussed in Sect. 3 can be addressed by adopting security-by-design (and related) approaches and techniques. We reiterate that while these solutions may seem obvious to security experts, there are many development teams without security expertise that regularly face and must address the presented challenges, for which this section is especially relevant.

### 4.1 The Role of Security-By-Design

Security is deeply rooted within the complexity of a system and it is intractable to think that solutions can address security issues after-the-fact. Security should therefore be considered at all stages of development. The current approach of having security retrofitted or “bolted-on” to the software systems that we build is not sufficient. Instead, we must take into account the critical security requirements for these systems and design them so that security is “baked-in”. This is the fundamental idea behind the *security-by-design* paradigm which promotes the systematic construction of systems with security considerations at each and every stage of the development process. The goal is to enable developers to include appropriate security analyses and controls early in the development of a system to avoid relying on retrospective security audits, and to have a more secure system at release [33].

As discussed in Sect. 3, the ability to generate, gather, and preserve evidence in support of security assurance efforts is a widespread challenge faced by the security assurance community. Security-by-design enables developers to think about security (and its requirements) from early stages of system development. In general, it helps to generate artifacts, in the form of requirements specifications, design documentation, test plans and results, etc. that can be used to support verification and validation efforts which provide the evidentiary basis for security assurance. This has been demonstrated with the development and use

of *Multiple Independent Levels of Security/Safety* (MILS) architectures [34] and the notion of *assurance-based development* [6] which support the implementation of documented and verifiable security solutions for both products and processes from early stages in the development process. It has also been the focus of assurance models and frameworks such as OWASP's SAMM [12], SAF [13], NIST's System Security Engineering Framework [14], and NIST's Framework for Improving Critical Infrastructure Cybersecurity [15]. Additionally, the emergence of DevOps [35] practices which place an emphasis on continuous testing, evolution, and maintenance has shown promise for supporting security-by-design. Such approaches have the potential to address the many challenges faced when assuring the security of software-dependent systems.

## 4.2 The Role of Adequate Threat Modeling

Providing strong security assurance requires a thorough understanding of the security requirements, threats, risks, and countermeasures so that they can be incorporated at all stages of development. If you do not think about the correct security requirements at all stages development process then it becomes difficult, or even impossible, to provide assurance that the resulting product is adequately secure. For example, not having adequate requirements showing that system components are isolated to ensure adequate security controls early can make it difficult to gather the required evidence to support any assurance claims at later stages, resulting in the need to rework or completely overhaul the system. This requires an adequate threat modeling approach.

A *threat model* is a structured representation of all of the security-relevant aspects of a system, and often consists of a list of assets that need protection, a list of vulnerabilities that threaten those assets, a description of the attacks that exploit those vulnerabilities, and a prioritized list of security improvements and countermeasures to reduce the risk and potential impact of exploits of the identified vulnerabilities [36]. Threat modeling is an integral part of eliciting system security requirements and understanding possible vulnerabilities and attack scenarios for the purpose of clearly identifying risk and impact levels [37]. It can be done at any stage of development, though it is very often recommended that it be done early so that the findings can inform later stages of development and guide the generation and gathering of supporting evidence for security assurance. The purpose of a threat model is to enable system defenders to systematically analyze a probable attacker's profile, the most likely attack vectors, and the assets most desired by an attacker, which thereby enables informed decision-making about security risk. This is very important for providing security assurance as it can often be viewed as a roadmap for security assurance efforts and is a first step in generating supporting evidence. This is certainly needed to address the challenges associated with managing tradeoffs with security requirements, and for dealing with legacy systems and complex supply chains.

### 4.3 The Role of Formal Methods for Security

Formal methods are highly desirable when developing systems with high standards of safety, security, and reliability [32]. Integrating the use of formal methods in the SDLC can further help to address many assurance-related challenges, especially those pertaining to generating sufficient and verifiable evidence to support security assurance claims. The ability to formally specify and verify critical systems and their components can help to address the lack of security evaluation methods, and can provide a strong evidentiary basis for security assurance.

Without the broad use of formal methods, security and resilience will always remain fragile, and therefore, their adoption and development for security has been encouraged whenever they can be demonstrably effective [38]. By taking a formal methods-based approach, mathematical proofs of assurance of system security properties can be provided [39]. Therefore, methods and techniques such as those in [39,40] can help to practically support a systematic and rigorous security assurance process founded on detailed, precise, and verifiable evidence.

### 4.4 The Role of a Systematic Process

A rigorous, systematic, and well-documented process is critical to providing the required assurance for a developed system. Development processes that do not generate the desired forms of evidence may be considered inadequate for the production and certification of software-dependent systems with respect to security [5]. A streamlined development process that emphasizes the explicit consideration and justification of decisions related to critical security requirements and properties is desired. This can help to facilitate security evaluation and assist in managing the size and complexity of software-dependent systems.

Furthermore, a systematic process can help to generate, gather, and manage the supporting evidence needed for security assurance. For example, with consideration to the V-model process for engineering high-assurance secure systems shown in Fig. 2, supporting evidence can be obtained from a number of sources including design methodologies, testing, verification, simulation, and analysis tools. This can then support assurance justification at each stage of the process which can help to form a convincing security assurance case that can be easily evaluated by third-party evaluators, regulators, and certification bodies. Moreover, an initial security assurance case can guide development teams in generating artifacts throughout the SDLC that can serve as evidence to support the eventual assurance argument. This emphasizes the need for, and importance of, integrating security evaluation and assurance activities within the SDLC.

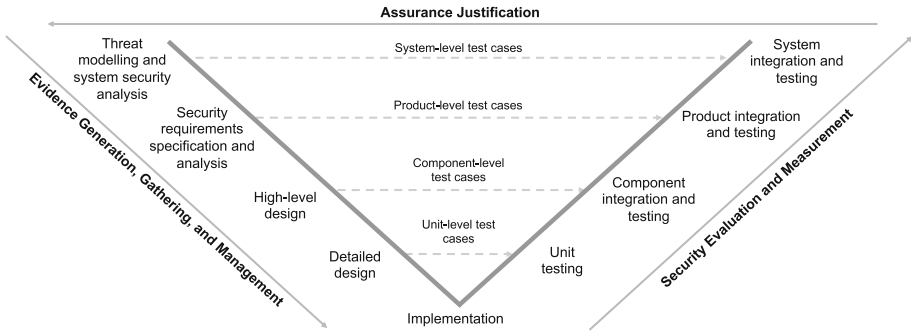


Fig. 2. V-Model for engineering high-assurance secure systems

### 4.5 The Role of Standards and Guidelines

An important aspect of security assurance, certification, and evaluation comes in the form of standards and guidelines related to the secure development of systems and their constituent components. Such standards and guidelines vary by domain and jurisdiction. For instance, in the domain of smart electricity grids, standards and guidelines outlining the minimum security requirements for smart grid components have been published by the IEC [41], NIST [42], and in the SafSec standard [43]. Several other major sources of security evaluation criteria, and assurance and accreditation processes include the DoD Instruction 8510.01, Risk Management Framework [44], TCSEC Orange Book [31], ITSEC [45], CTCPEC [46], and the Common Criteria [9].

Standardized approaches are desired as they provide clear and detailed processes and procedures that guide what needs to be shown in an assurance case and, often, the specific evidence that needs to be produced to evaluate a system for assurance purposes. Standards and guidelines can help to encourage trends and actions to reduce barriers to adoption of more integrated, security-by-design approaches that enable the costs of security assurance to be justified and weighed in a fair and just manner. As existing standards and guidelines have limitations, new or updated standards and guidelines need to be clearly defined and maintained to allow for changes to be made as new technologies are developed and adopted. The main risk here is that as standards change, older systems (i.e., legacy systems) will need to be re-evaluated and re-certified which can be very costly. We strongly believe that such limitations and concerns can be addressed with the development of more modular, compositional, and incremental security assurance solutions that are integrated into rigorous and systematic development processes. Such solutions can help to localize change and reduce the costs and effort required to re-evaluate and assure system security.

### 4.6 The Role of Automated Security Analysis

A major component of NIST’s System Security Engineering Framework [14] is the incorporation of system security analysis to support the security engineering activities leading towards the development of a security assurance case. The

role of such system security analysis is to produce data to support engineering and stakeholder decision-making and for generating evidence to support security evaluation and assurance activities. However, as noted in Sect. 2, carrying out such analysis at all stages of the SDLC can be very costly and time-consuming. Beyond that, there is also a challenge associated with the personnel involved in manually performing system security analysis. Human-driven processes are bound to fail one way or another. Therefore, the ability to automate these analysis and seamlessly incorporate them into the existing SDLC can have significant rewards and can assist in ensuring that the people involved in developing, evaluating, and assuring a given system cannot do the wrong thing leading to the deployment of vulnerable systems. Recently proposed approaches and tools, such as those proposed by Feiler [47] for automating attack tree analysis and attack impact analysis, have taken some steps in this direction.

## 5 Recommendations to Reduce Barriers to Adoption and Advance the State-of-the-Art

In Sect. 4, we described the role that a number of existing approaches and techniques can play in addressing the challenges outlined in Sect. 3. If approaches that can aid in addressing the noted challenges already exist, then why do we still face the challenges when evaluating and assuring the security of software-dependent systems? Learning from the experience reported in the literature, we provide some insight into this issue and propose a set of recommendations for how we can advance towards more effective security assurance solutions.

***Recommendation 1: Develop More Modular, Compositional, and Incremental Security Assurance Solutions.*** Despite that numerous approaches have been proposed to facilitate security assurance in the engineering of software-dependent systems, there is still a perception that integrating these approaches into the SDLC and performing the associated activities still requires too much time, effort, and resources [24]. As a result, there has been a lack of adoption of such approaches. This is worsened when we consider the speed at which technology is developed. Developers often succumb to business pressures to put a product on the market. Thus, they primarily focus on building a product that does what it is supposed to as quickly as possible. This often means that considerations to engineer a resilient system in the face of security threats, and further to demonstrate that it meets its security requirements are often sacrificed.

We need a stronger push towards the development of more modular, compositional, and incremental solutions for securing systems from the outset and for generating sufficient evidence of their built-in resilience to a range of cyberattacks and failures. Existing modular, compositional, and incremental system security solutions are far from satisfactory [38]. Because systems are constantly evolving, there is a need for approaches that ensure that changes in a system, do not trigger another long and costly security evaluation process. Rather, we would like to see a security evaluation process where the evaluation effort is proportional to the degree of change in the system. Such efforts can reduce the costs and time-to-market in a secure-by-design development process.

***Recommendation 2: Harmonize Security Assurance Efforts with Agile and Adaptive Development Processes.*** Modern software-dependent systems operate under high volatility and in high scale. These factors call for more flexible and adaptable architectures and designs. This is why more agile approaches have been adopted, and these are the same reasons that have caused paradigms like DevOps to rise to prominence. For a number of highly critical systems, adaptability and flexibility may not be possible to the highest degree, but for systems like e-banking applications and IoT-enabled systems, adaptability is a “must-have” property. However, the rigidity and heaviness of many existing approaches supporting security assurance do not easily accommodate agile development processes in practice, despite boasting the ability to do so in principle. Establishing a process by which security considerations are taken into account at each and every stage of the SDLC is a daunting task. This is especially true when the threat landscape for a given system changes rapidly. Developers of high-assurance software-dependent systems can very easily be overwhelmed by the size and complexity of the system that they are developing, evaluating, or assuring. There is a perception that there is too much to do, and carrying out security evaluation and assurance activities are overly burdensome.

To overcome these barriers, we must strive towards developing more evolvable and adaptable, secure architectures. Recent research efforts have started exploring this direction [34, 40, 48]. Moreover, the emergence of DevOps has spun off the sub-area of DevSecOps which endeavours to embed security engineering activities into agile development operating practices [49].

***Recommendation 3: Provide Documented Demonstrations of the Effectiveness of Existing or Newly Developed Approaches.*** Despite the existence of numerous approaches, methods, techniques, and tools that can address a number of the challenges identified in Sect. 3, many practitioners simply do not know where to start. Determining which approach, method, technique, or tool is applicable or suitable for addressing the security challenges in the development of high-assurance software-dependent systems remains difficult for practitioners.

Documented demonstrations of the effectiveness of existing approaches, as well as newly developed methods, techniques, and tools can enable practitioners to clearly and easily see their value, or when and how to apply them in the SDLC. The research community has been poor in providing such support to demonstrate that proposed solutions can be widely applicable, or in clearly stating the limitations of such approaches to enable practitioners to make effective decisions about when and how to apply a potential approach, method, technique, and/or tool. More effort in this area can greatly advance the state-of-the-art and reduce the barriers of adoption of more integrated security assurance efforts.

***Recommendation 4: Develop More Accessible Tool Support for Integrating Automated Security Analyses into the SDLC.*** Effectively performing the system security analyses required at all stages of the SDLC to provide sufficient evidence in support of security assurance claims heavily relies on adequate tool support for automation. Automated analyses built into the SDLC

can be run (and re-run) throughout the development process to support the generation of evidence in support of security assurance arguments. This is especially true when using formal methods and assurance case approaches as these methods are burdensome and unscalable when done manually. However, we do not currently have the full fleet of tools required to support many activities in the security engineering process. Despite the existence of formal tools such as model checkers and theorem provers, the integration of such tools in the SDLC to address security concerns is not well-recognized. As a research community, we need to develop tools that are applicable in a variety of application domains, that are more accessible to practitioners, and that can be seamlessly integrated into the SDLC workflow. This must be combined with Recommendation 3 to aid practitioners in understanding the usage, benefits, and limitations of such tools which will help to reduce the barrier to adoption of more formal methods and approaches.

***Recommendation 5: Develop More Rigorous, Specific, and Focused Security Standards, Guidelines, and Best Practices.*** There is an organizational and operational context which needs to be applied when considering security assurance. Different contexts demand different tolerances of risk for the system assets that need protection. For example, different considerations are needed if we are dealing with smart energy infrastructure than if we are dealing with a healthcare or transportation system. However, classification systems for expressing risk tolerances in different domains are not standardized, and when they are it is a convoluted mess as was the case with the TCSEC Orange Book [31]. Furthermore, current evaluation methods and criteria such as the Common Criteria [9] have been criticized for focusing on assuring mainly the functional security requirements of a system or product [6,7]. In addition, the existence of well-defined or documented sets of standards, guidelines, or requirements for developing secure high-assurance software-dependent systems are limited, or lack focus and specificity making compliance either too difficult or too easy. As a result, many practitioners are not sure what needs to be done to demonstrate that they have taken appropriate measures to adequately secure their systems. There is not enough guidance to allow them to see the value in adopting existing approaches or to even understand when and how to apply existing approaches.

Without readily available guidance documents, assuring the security of software-dependent systems will remain challenging. Further research efforts in developing more rigorous standards and best practices with support for guiding practitioners to incorporate suitable security measures into the development of software-dependent systems at all stages of system development, as well as the evidence to be produced to support assurance claims are needed. These standards, guidelines, and best practices need to be outcome-oriented and based on sound technological principles. To achieve this we can build upon existing frameworks (e.g., [12–15]) which aim to provide such outcome-oriented guidance for engineering, evaluating, and assuring secure software-dependent systems.



***Recommendation 6: Improve Collaboration Among Stakeholders in the SDLC.*** The development of large and complex software-dependent systems historically been carried out by a number of teams and stakeholder groups that functioned in relative silos. However, failing to include and collaborate with system stakeholders regularly throughout the SDLC significantly contributes to the challenges of building-in security and providing security assurance [24]. Therefore, we need to include and improve the collaboration between developers and other critical stakeholders, such as domain experts and regulators, in the process of engineering high-assurance software-dependent systems. For example, improving collaboration between developers, domain experts, regulators, evaluators, and certification bodies can help to determine the completeness of security and resilience requirements and to determine appropriate decompositions of security assurance claims into sub-claims based on the particular evidence expected to support a particular claim, as well as the acceptance criteria for that evidence. Recent advances and the emergence of software development practices such as DevOps [35] have targeted this kind of collaboration. These practices have shown promise for improving accountability for addressing security concerns throughout the SDLC, but also for creating more efficiency in the process.

***Recommendation 7: Improve Training, Education, and Awareness of All Personnel Involved in the Development of High-Assurance Software-Dependent Systems.*** Training, education, and awareness about the emerging discipline of security engineering and the development and usage of security assurance cases are essential not only for system and software engineering practitioners and their managers, but also for policy makers (including regulators) and for all organizations and individuals who increasingly rely on the critical services provided by modern software-dependent systems [5]. However, there is often a lack of expertise to carry out the activities required in engineering high-assurance systems. We mentioned above that there has been an emergence of DevOps in security engineering. Organizations are even going so far as to advertise positions for *Security DevOps Engineers*. In practice, most personnel do not have the full complement of expertise to perform all of the activities required by these roles to the standards needed for high-assurance software-dependent systems. This is not to say that such positions are not needed. In fact, the existence of such positions underscores the importance of incorporating security at all stages of the SDLC. The point to be made here is that more effort needs to be put forth in training more engineers to be capable of considering the complex nature of security assurance in the development of software-dependent systems. We propose the development of awareness campaigns to train practitioners to understand and manage such complexities and considerations. This can help to cultivate a culture of security-consciousness in organizations so that security is no longer considered as an afterthought.

## 6 Concluding Remarks

The challenges of providing effective security assurance for software-dependent systems are well-founded and a number of these challenges have been experienced in the recent past (e.g. [5, 13, 14, 19, 24, 27–29]). In this paper, we discussed the challenges and obstacles that make security assurance so difficult. We highlighted technology gaps related to managing tradeoffs with security requirements, coping with the size and complexity of systems and associated assurance cases, the lack of security evaluation approaches, dealing with legacy systems and third-party suppliers, and handling the evaluation and incorporation of people and process in security assurance efforts. We explored the role of security-by-design and related approaches and techniques in leading to a more comprehensive solution capable of overcoming many of the challenges and obstacles faced when providing security assurance. We also provided some insight, based on experience, into the barriers that currently limit the adoption of existing approaches and techniques capable of addressing the noted challenges. We proposed seven recommendations for reducing these barriers and advancing the state-of-the-art.

Although our main focus has been on security assurance, these open issues and challenges apply to assurance for any software or system quality attribute such as safety, reliability, dependability, etc. Our hope is that our recommendations can be used as a roadmap for a variety of future research efforts to fully realize the vision presented in this paper. For example, there is plenty of work to be done in developing and adopting streamlined development processes aimed at producing evidentiary artifacts, and methodologies, standards and guidelines supporting awareness and enforcement of assurance concerns in the development of software-dependent systems. There is also room for developing tools to assist in managing the generation, gathering, evaluation, and preservation of enormous amounts of security assurance evidence, and for preparing security assurance cases that can be incorporated and delivered as part of the final product. We are actively engaged in ongoing work in this area. We believe that such efforts, founded on the principles of security-by-design, can dramatically improve assurance efforts so that we can be confident in the operation of our ever-growing collection of critical software-dependent systems.

**Acknowledgment.** This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant RGPIN-2019-06306.

## References

1. McLean, J., Heitmeyer, C.L.: High assurance computer systems: a research agenda. In: America in the Age of Information, National Science and Technology Council Committee on Information and Communications Forum (1995)
2. Mead, N.R.: SEHAS 2003: the future of high-assurance systems. *IEEE Secur. Priv.* **1**, 68–72 (2003)
3. Government of Canada: National electric grid security and resilience action plan, December 2016. <https://www.publicsafety.gc.ca/cnt/rsrscs/pblctns/pln-crtcl-nfrstrctr-2014-17/index-en.aspx>

4. U.S.A. Department of Homeland Security: National critical infrastructure security and resilience research and development plan, November 2015
5. Weinstock, C.B., Lipson, H.F.: Evidence of assurance: laying the foundation for a credible security case. Technical report, Software Engineering Institute, August 2013
6. Agudo, I., Vivas, J.L., López, J.: Security assurance during the software development cycle. In: International Conference on Computer Systems and Technologies, CompSysTech 2009, pp. 20:1–20:6 (2009)
7. Winograd, T., McKinley, H.L., Oh, L., Colon, M., McGibbon, T., Fedchak, E., Vienneau, R.: Software Security Assurance: A State-of-the Art Report (SOAR). Information Assurance Technology Analysis Center (IATAC), July 2007
8. Federal Trade Commission: Internet of things: privacy and security in a connected world. FTC Staff Report, Federal Trade Commission, January 2015
9. Common Criteria Recognition Arrangement: Common Criteria for Information Technology Security Evaluation (CC). No. CCMB-2009-07, Common Criteria Recognition Arrangement, July 2009
10. Communications Security Establishment Canada: Annex 2 - Information System Security Risk Management Activities: IT Security Risk Management: A Lifecycle Approach. Communications Security Establishment Canada (2012)
11. Gilsinn, J.D., Schierholz, R.: Security assurance levels: a vector approach to describing security requirements. NIST, October 2010
12. Chandra, P.: Software assurance maturity model, a guide to building security into software development, version 1.0 (2009). <http://www.opensamm.org/downloads/SAMM-1.0.pdf>
13. Woody, C.C., Ellison, R.J.: Software assurance measurement - establishing a confidence that security is sufficient-establishing a confidence that security is sufficient. *J. Cyber Secur. Inf. Syst.* **5**(3), 28–36 (2017)
14. Ross, R.S., McEvilly, M., Oren, J.C.: Systems security engineering: considerations for a multidisciplinary approach in the engineering of trustworthy secure systems. Special Publication (NIST SP) 800-160, NIST, November 2016
15. National Institute of Standards and Technology: Framework for improving critical infrastructure cybersecurity, version 1.1, April 2018. <https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf>
16. GSN Working Group: GSN community standard version 2, January 2018
17. Rushby, J., Xu, X., Rangarajan, M., Weaver, T.L.: Understanding and evaluating assurance cases. NASA Contractor Report NASA/CR–2015-218802, NASA Langley Research Center, September 2015
18. Rinehart, D.J., Knight, J.C., Rowanhill, J.: Current practices in constructing and evaluating assurance cases with applications to aviation. NASA Contractor Report NASA/CR–2015-218678, NASA Langley Research Center, January 2015
19. Rushby, J.: The interpretation and evaluation of assurance cases. Technical report, SRI-CSL-15-01, SRI International, July 2015
20. Wassыng, A., Maibaum, T., Lawford, M., Bherer, H.: Software certification: is there a case against safety cases? In: Calinescu, R., Jackson, E. (eds.) *Monterey Workshop 2010: Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems*. LNCS, vol. 6662, pp. 206–227. Springer, Heidelberg (2011)
21. Alexander, R., Hawkins, R., Kelly, T.: Security assurance cases: motivation and the state of the art. Technical report CESC/TR/2011/1, University of York, April 2011

22. Weinstock, C.B., Lipson, H.F., Goodenough, J.B.: Arguing security - creating security assurance cases. Technical report, Software Engineering Institute, January 2007
23. U.S.A. Computer Emergency Readiness Team: Build security in: setting a standard for software assurance (2015). <https://www.us-cert.gov/bsi>
24. Jaskolka, J.: Challenges in assuring security and resilience of advanced metering infrastructure. In: 18th Annual IEEE Canada Electrical Power and Energy Conference, EPEC 2018, pp. 1–6 (2018)
25. U.S.A. Department of Homeland Security: Sector risk snapshots, March 2014
26. Asghar, M.R., Dán, G., Miorandi, D., Chlamtac, I.: Smart meter data privacy: a survey. *IEEE Commun. Surv. Tutor.* **19**(4), 2820–2835 (2017)
27. Ibarra, I., Ward, D.: Assurance cases to argue system resilience properties for road vehicles. In: 2013 Workshop on Human Factors in the Safety and Security of Critical Systems, March 2013
28. Pantazopoulos, P., Haddad, S., Lambrinouidakis, C., Kalloniatis, C., Maliatsos, K., Kanatas, A., Varádi, A., Gay, M., Amditis, A.: Towards a security assurance framework for connected vehicles. In: 19th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, pp. 1–6 (2018)
29. Wassying, A., Singh, N.K., Geven, M., Proscia, N., Wang, H., Lawford, M., Maibaum, T.: Can product-specific assurance case templates be used as medical device standards? *IEEE Des. Test* **32**(5), 45–55 (2015)
30. Jackson, D., Thomas, M., Millett, L.I. (eds.): *Software for Dependable Systems: Sufficient Evidence?* National Academies Press, Washington, DC (2007)
31. U.S.A. Department of Defense: *Trusted Computer System Evaluation Criteria (TCSEC)*. No. DoD 5200.28-STD in Defense Department Rainbow Series (Orange Book), Department of Defense/National Computer Security Center, December 1985
32. Nhlabatsi, A., Laney, R., Nuseibeh, B.: Feature interaction: the security threat from within software systems. *Prog. Inform.* **5**, 75–89 (2008)
33. Deogun, D., Sawano, D., Bergh Johnsson, D.: *Secure by Design*. Manning Publications Company, Shelter Island (2018)
34. Tverdyshev, S.: Security by design: introduction to MILS. In: *International Workshop on MILS: Architecture and Assurance for Secure Systems* (2017)
35. Bass, L., Weber, I., Zhu, L.: *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, New York (2015)
36. Shostack, A.: *Threat Modeling: Designing for Security*. Wiley, Hoboken (2014)
37. UcedaVélez, T., Morana, M.M.: *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*, 1st edn. Wiley, Hoboken (2015)
38. Chong, S., Guttman, J., Datta, A., Myers, A., Pierce, B., Schaumont, P., Sherwood, T., Zeldovich, N.: Report on the NSF workshop on formal methods for security. Technical report (2016). <http://arxiv.org/abs/1608.00678>
39. Mandrioli, D.: The role of formal methods in developing high assurance systems: some old and some less old thoughts. In: *Workshop on Software Engineering for High Assurance Systems, SEHAS 2003*, pp. 29–32 (2003)
40. Rouland, Q., Hamid, B., Jaskolka, J.: Formalizing reusable communication models for distributed systems architecture. In: *8th International Conference on Model and Data Engineering, MEDI 2018*, pp. 198–216 (2018)
41. International Electrotechnical Commission: IEC Standard: 62351, May 2007. <http://www.iec.ch/smartgrid/standards/>

42. The Smart Grid Interoperability Panel–Smart Grid Cybersecurity Committee: Guidelines for smart grid cybersecurity: Volume 1 – smart grid cybersecurity strategy, architecture, and high-level requirements. Interagency Report NISTIR 7628 Revision 1, NIST, September 2014
43. Dobbing, B., Lautieri, S.: SafSec methodology: Standard 3.1. SafSec: Integration of Safety & Security Certification S.P1199.50.2, Altran Praxis, November 2006
44. U.S.A. Department of Defense: DoD Instruction 8510.01, Risk Management Framework (RMF) for DoD Information Technology (IT), March 2014. <http://www.nist.gov/cyberframework/upload/cybersecurity-framework-021214.pdf>
45. U.K. Department of Trade & Industry: Information Technology Security Evaluation Criteria (ITSEC), COM(90) 314. Department of Trade & Industry, June 1991
46. Communications Security Establishment Canada: Canadian Trusted Computer Product Evaluation Criteria (CTCPEC). Communications Security Establishment Canada (1993)
47. Feiler, P.: Automated assurance of security-policy enforcement in critical systems. SEI Blog, February 2018. [https://insights.sei.cmu.edu/sei\\_blog/2018/02/automated-assurance-of-security-policy-enforcement-in-critical-systems.html](https://insights.sei.cmu.edu/sei_blog/2018/02/automated-assurance-of-security-policy-enforcement-in-critical-systems.html)
48. Sljivo, I., Gallina, B.: Building multiple-viewpoint assurance cases using assumption/guarantee contracts. In: 10th European Conference on Software Architecture Workshops, ECSAW 2016, pp. 39:1–39:7. ACM (2016)
49. Hsu, T.H.C.: Hands-On Security in DevOps: Ensure Continuous Security, Deployment, and Delivery with DevSecOps. Packt Publishing Ltd., Birmingham (2018)



# On Generating Cancelable Biometric Templates Using Visual Secret Sharing

Manisha and Nitin Kumar<sup>(✉)</sup>

National Institute of Technology, Uttarakhand, Srinagar, India  
{manisharawatphd,nitin}@nituk.ac.in

**Abstract.** Cancelable Biometrics have been gaining popularity due to security and privacy concerns of an individual's Biometric image(s). The main objective in Cancelable Biometrics is to generate templates which are non-invertible in nature and result from repetitive distortion of the Biometric image. In this paper, we propose a novel framework for generating Cancelable Biometric templates using *Visual Secret Sharing*. In the proposed scheme,  $n$  different shares are generated corresponding to one Biometric image with the help of  $n - 1$  other images called *Cover images*. The generated *Secret Shares* are stored in a distributed manner instead of the original Biometric image. For generating  $n$  shares, we propose three different methods (M1) One Biometric image and  $n - 1$  randomly chosen natural gray images (M2) One Biometric image with  $n - 1$  randomly permuted version of Biometric image (M3) Both Secret image and Cover images are randomly permuted version of the Biometric image. To show the efficacy of the proposed approach, we have used the publicly available IIT Delhi Iris database (version 1.0). The performance of these three approaches have been compared in terms of average Correlation Coefficient, False Accept Rate (FAR), False Reject Rate (FRR) and Genuine Accept Rate (GAR), True Error Rate (TER) and True Success Rate (TSR). The experimental results show that M3 performs best among the proposed methods in terms of all the performance measures and in qualitative terms.

**Keywords:** XOR · Cancelable · Random · Secret · Shares · Database

## 1 Introduction

Biometrics play an important role in everyday life due to its applications in critical areas such as surveillance, ATM, access control, corpse identification [1], etc. In today's digital world scenario, security becomes a major issue in every domain such as access control, border immigration control, website access control, forensic sector etc. The major drawback of traditional Biometric authentication systems is that the original Biometric features or patterns of a person are stored in a database. If some intruder succeeds in accessing the database, then person's original Biometric data may be compromised. Consequently, the

security and privacy of a person may get breached. Due to burgeoning applications of Biometric systems, a person must use his or her Biometric in one or other applications. In Cancelable Biometric technique a repetitive distortion is applied on original features by various transformation functions. Finally, a distorted pattern is obtained as shown in Fig. 1. Now, this distorted or meaningless pattern is stored in the database instead of the original one. If intruder gets success in accessing this database, then he or she will receive only transformed or distorted template instead of the original one. The backbone of this technique is its transformation function. This function should be non-invertible in nature such that the intruder will not be able to get original features by reverse engineering process from the distorted Cancelable Biometric templates.

## 1.1 Motivation

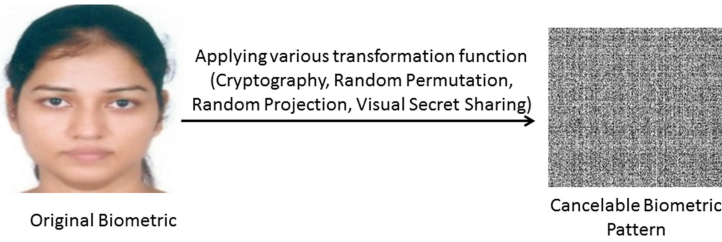
Motivated by Visual Secret Sharing technique suggested by Deshmukh et al. [6], we propose a novel approach for generating Cancelable Biometric template in which one Biometric image and  $n - 1$  other images are employed to generate  $n$  Secret Shares. These  $n - 1$  other images are called Cover images. Based on how we can choose these Cover images, three schemes have been proposed as discussed later. After generation of  $n$  Secret Shares, one is assigned to the user and others are stored in the distributed database. The crux of the proposed technique is that, when a user presents his/her Secret Share along with some key such as PIN during the authentication phase, the corresponding shares will be combined together to authenticate his/her identity. If the recovered image results in a match, the probe is allowed access otherwise he/she is denied. The matching is performed with similarity score and threshold.

The rest of the paper is organized as follows: Sect. 2 discusses the available techniques in literature which focus on generation of Cancelable Biometric templates using Visual Secret Sharing. Section 3 describes three different proposed methods for generation of Cancelable Biometric templates. In Sect. 4, experimental set up and results have been presented and discussed. Some concluding remarks and future directions are given in Sect. 5.

## 2 Related Work

In literature, there are various methods proposed by different authors for generating Cancelable Biometric templates. A Cancelable Biometric template must possess four important characteristics, viz. (i) Diversity, (ii) Reusability or Revocability, (iii) Non-invertibility, and (iv) Performance. Recently, a comprehensive survey on various Cancelable Biometric is carried out by Manisha and Kumar [4].

In recent scenario of Cancelable Biometric field, these template generation methods are broadly classified into six major categories: (i) Cryptography based, (ii) Transformation based, (iii) Filter based, (iv) Hybrid based, (v) Multimodal based, and (vi) Other methods. For detail working of these methods, please



**Fig. 1.** Generation of Cancelable Biometric Pattern from the original biometric after applying various transformation functions

refer [4]. The proposed work comes under Cryptography based Cancelable Biometric template generation method. This method is generally used for sending or storing very sensitive or important information. In this method, with the help of some transformation functions original message/information is totally changed into another meaning less form. Various Cancelable template generation methods are available in literature e.g. Index of Max hashing [2], Hill Cipher based encryption [8], Bio-Hashing [3], etc. Visual Secret Sharing Scheme (VSS) is a variant of Cryptography based methods or Visual Cryptography [7]. In VSS, we have a Secret image and other Cover images. Secret image is generally the message which we need to hide and Cover images will help to hide this Secret image. We can encrypt this Secret image with help of Cover images to various distorted patterns, which are known as *Secret Shares*. The motivation behind this technique is to provide more security in original Secret image. The key of this technique is its Secret Shares. The original Secret image can only be recovered if all of its corresponding Secret Shares are available. In this technique, among all the generated Secret Shares, one share is given to the genuine user (corresponding to which original Biometric image belongs) and others are usually stored in decentralize database. At the time of authentication, the query user needs to provide his/her share and corresponding to this share, other shares are fetched from all decentralize databases using a key or PIN to authorize him/her. In VSS  $(k, n)$  technique, original Secret will only be recovered if  $k$  out of  $n$  Secret Shares are combined together. We cannot obtain the Secret by combining less than  $k$  Secret Shares. In VSS  $(n, n)$  technique, the Secret will only recovered if all the  $n$  Secret Shares are combined together, less than  $n$  Secret Shares will not reveal any information about the Secret image.

In literature, various researchers have suggested Visual Cryptography based methods for generation of Cancelable Biometrics templates in the domains of iris [9–12], face [13], fingerprint [15] etc. Revenker et al. [12] have suggested a method for Cancelable Biometric template in which firstly an image is segmented using Circular Hough transform [11], secondly normalization is performed using Daughman's rubber sheet model [10] and lastly feature extraction is carried out using Log-Gabor wavelets. Thereafter, pixel expansion using XOR function is employed to encrypt the iris template thereby generating two Secret Shares



typically called  $S_1$  and  $S_2$ . The original iris template features can be formed by stacking of these two Secret Shares. The main drawback of this method is only color coding is used for encryption which is easy to break. Ross and Othman [13] have proposed another method for Cancelable face template generation in which two images similar to the Biometric image are selected from publicly available datasets based on face geometry and appearance. Experiments were performed on IMM and XM2VTS face datasets. Gray level Extended Visual Cryptography Scheme (GEVCS) is employed for hiding the private image or Biometric image in the selected host images or Cover images thereby generating two host sheets or Secret Shares. The bottleneck of this method is to find the publicly available host images or Cover images based on geometry and appearance which requires huge computational resources. The authors have further extended the research work [13] for other Biometrics modalities such as Iris and fingerprint [14]. Further, Thomas and Babu [15] have generated Cancelable Biometric templates using (2, 2) VSS technique on fingerprint images. Two Secret Shares are generated using 4 subpixel layout using random basis column pixel expansion technique. The original Secret image or Biometric image can be retrieved after combining of these two Secret Shares using XNOR operation. The limitation of this method is that this coding is only applicable on binary images. In all the methods discussed above, pixel expansion is used for generation of Secret Shares [16].

To address all the above limitations, in this paper we have proposed a novel framework for generating Cancelable Biometric templates using Visual Secret Sharing. The proposed framework possesses several advantages as given below:

- i) No pre-processing required
- ii) No external Biometric dataset is required by the proposed framework
- iii) The proposed method can work for gray scale images.

### 3 Proposed Method

In the proposed method, we generate  $n$  Secret Shares by employing one Secret Biometric image and  $n - 1$  Cover images. Based on how we use these Cover images, we have proposed three different schemes:

*M1:* Original Biometric image is used as Secret image and  $n - 1$  different natural Gray images are chosen as Cover images.

*M2:* Original Biometric image is used as Secret image and  $n - 1$  Cover images are generated from the Secret image by 2-D random permutation [5] of the original image.

*M3:* Secret image as well as  $n - 1$  Cover images, all are randomly permuted version of the original Biometric image.

**Table 1.** Acronyms used

S	Secret image
C	Cover images
SS	Secret Shares
RM	Reverse Matrix
IS	Intermediate Shares
NI	Number of Imposters
FA	Number of False Acceptance
FR	Number of False Rejection
NG	Number of Genuine Users
FAR	False Acceptance Rate
FRR	False Rejection Rate
TER	True Error Rate
TSR	True Success Rate
GAR	Genuine Acceptance Rate
ARM	Authentication phase Reverse Matrix
AIS	Authentication phase Intermediate Shares
MXOR	XOR Matrix of Secret image and Cover Images
AXOR	Authentication phase XOR Matrix of Secret Shares

Each of these methods M1, M2 and M3 are explained next. We have also proposed two different Algorithms i.e. (i) Enrollment for Secret Shares generation, and (ii) Authentication for Decryption to the original Biometric image. A list of acronyms used in algorithms and rest of the paper is given in Table 1. Next, the algorithms for Enrollment and Authentication are discussed.

### Enrollment Algorithm

The step-wise details of Enrollment are given in Algorithm 1. The inputs to the Algorithm are (i) original Biometric image (i.e. Secret image  $\mathbf{S}_1$ ) and  $n-1$  Cover images ( $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \dots, \mathbf{C}_{n-1}$ ), while the outputs are  $n$  different Secret Shares. In step 1, XOR ( $\oplus$ ) operation is performed among all the  $n$  input images and output is stored in a matrix called  $\mathbf{M}$ . Afterwards, reverse bit operation is performed in left bit-wise reversal order of  $\mathbf{M}$  and the result is stored as another matrix  $\mathbf{R}_M$ . Next, Intermediate shares ( $\mathbf{I}_i, i = 1, 2, \dots, n$ ) are generated by taking  $\oplus$  of input images (Secret and Cover images) with Reverse Matrix ( $\mathbf{R}_M$ ). Lastly, Secret Shares ( $\mathbf{SS}_i, i = 1, 2, \dots, n$ ) are generated using Intermediate Shares ( $\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n$ ) as given in step 4 of the algorithm.

### Authentication Algorithm

The step-wise details of Authentication are given in Algorithm 2. The inputs to the Algorithm are  $n$  Secret Shares ( $\mathbf{SS}_1, \mathbf{SS}_2, \dots, \mathbf{SS}_n$ ) while the outputs are (i) recovered Secret image, and (ii)  $n-1$  Cover images. In step 1, XOR ( $\oplus$ )

operation is performed among all the  $n$  input images and output is stored in a matrix called  $\mathbf{A}_M$ . Afterwards, reverse bit operation is performed in left bit-wise reversal order of  $\mathbf{A}_M$  and the result is stored as another matrix  $\mathbf{A}_{RM}$ . Next, Authentication Intermediate shares ( $\mathbf{A}_i, i = 1, 2, \dots, n$ ) are generated by taking  $\oplus$  of Secret Shares and Authentication Reverse Matrix ( $\mathbf{A}_{RM}$ ). Lastly, Secret Image ( $\mathbf{S}_1$ ) and  $n - 1$  Cover images ( $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \dots, \mathbf{C}_{n-1}$ ) are recovered using Authentication Intermediate Shares ( $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n$ ) as given in step 4 of the algorithm.

---

**Algorithm 1:** Algorithm for Secret Shares generation in Enrollment phase

---

**Input:** Secret image  $\mathbf{S}_1$  and  $n - 1$  Cover images  $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \dots, \mathbf{C}_{n-1}$

**Output:**  $n$  secret shares  $\mathbf{SS}_1, \mathbf{SS}_2, \mathbf{SS}_3, \dots, \mathbf{SS}_n$

- 1 Calculate  $\mathbf{M} = \mathbf{S}_1 \oplus \mathbf{C}_1 \oplus \mathbf{C}_2 \oplus \mathbf{C}_3 \oplus \dots \oplus \mathbf{C}_{n-1}$
  - 2  $\mathbf{R}_M = \text{ReverseBit}(\mathbf{M})$
  - 3 Generate Intermediate Shares  $\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3, \dots, \mathbf{I}_n$  as follows:
    - (i)  $\mathbf{I}_1 = \mathbf{S}_1 \oplus \mathbf{R}_M$
    - (ii)  $\mathbf{I}_i = \mathbf{C}_{i-1} \oplus \mathbf{R}_M$  for  $i = 2$  to  $n$
  - 4 Generate Secret Shares  $\mathbf{SS}_1, \mathbf{SS}_2, \mathbf{SS}_3, \dots, \mathbf{SS}_n$  as follows:
    - (i)  $\mathbf{SS}_1 = \mathbf{I}_1$
    - (ii)  $\mathbf{SS}_2 = \mathbf{I}_2$
    - (iii)  $\mathbf{SS}_i = \mathbf{I}_i \oplus \mathbf{I}_{i-1} \oplus \mathbf{I}_{i-2}$  for  $i = 3$  to  $n$
- 

---

**Algorithm 2:** Algorithm for retrieval of Secret and Cover images in Authentication phase

---

**Input:**  $n$  Secret Shares  $\mathbf{SS}_1, \mathbf{SS}_2, \mathbf{SS}_3, \mathbf{SS}_4, \dots, \mathbf{SS}_n$

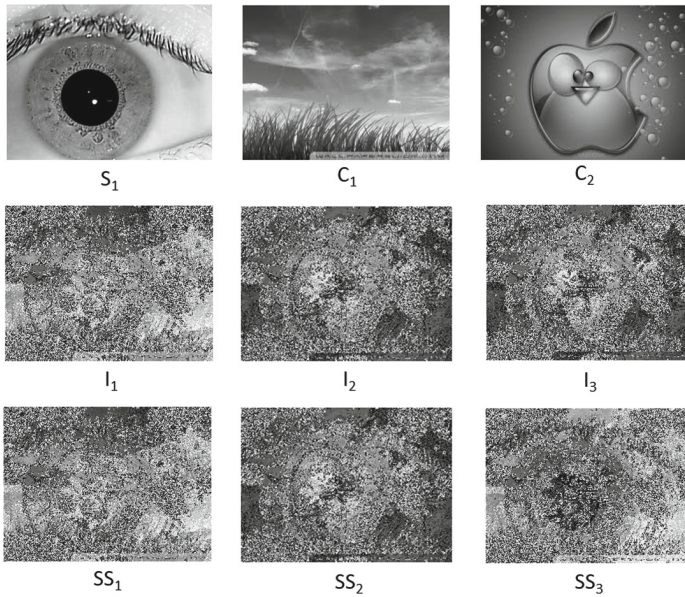
**Output:** one Secret image  $\mathbf{S}_1$  and  $n - 1$  Cover images  $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \dots, \mathbf{C}_{n-1}$

- 1 Calculate  $\mathbf{A}_M = \mathbf{SS}_1 \oplus \mathbf{SS}_2 \oplus \mathbf{SS}_3 \oplus \dots \oplus \mathbf{SS}_n$
  - 2  $\mathbf{A}_{RM} = \text{ReverseBit}(\mathbf{A}_M)$
  - 3 Generate Authentication phase Intermediate Shares  $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \dots, \mathbf{A}_n$   
 $\mathbf{A}_i = \mathbf{SS}_i \oplus \mathbf{A}_{RM}$  for  $i = 1$  to  $n$
  - 4 Generate Secret image  $\mathbf{S}_1$  and Cover images  $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \dots, \mathbf{C}_{n-1}$ 
    - (i)  $\mathbf{S}_1 = \mathbf{A}_1$
    - (ii)  $\mathbf{C}_1 = \mathbf{A}_2$
    - (iii)  $\mathbf{C}_{i-1} = \mathbf{A}_i \oplus \mathbf{A}_{i-1} \oplus \mathbf{A}_{i-2}$  for  $i = 3$  to  $n$
- 

**3.1 M1: Original Biometric Image (Secret) and  $n - 1$  Natural Gray Scale Images (Cover)**

In this method, one Biometric image ( $\mathbf{S}_1$ ) with  $n - 1$  different gray Cover images ( $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{n-1}$ ) are combined using XOR operation, which results in the formation of  $\mathbf{M}$  matrix. A reverse matrix  $\mathbf{R}_M$  is formed after taking the left bit-wise reversal of the individual pixel value in  $\mathbf{M}$  matrix, which results in generation

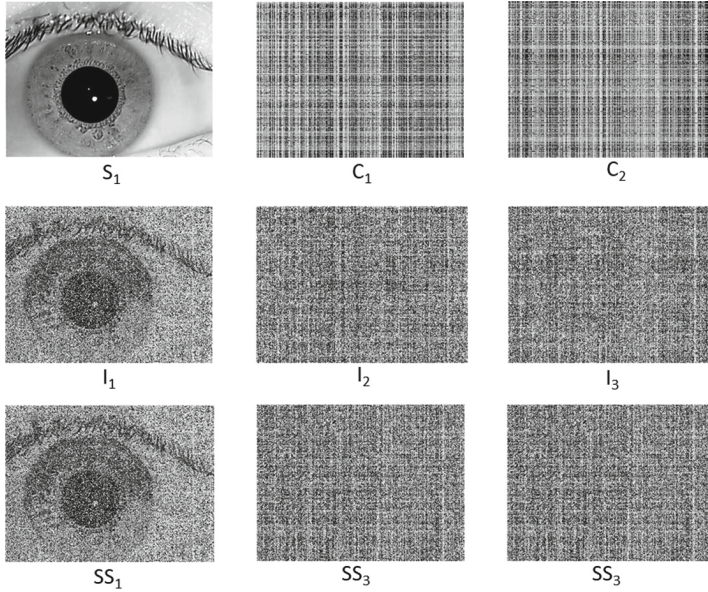
of Intermediate Shares ( $I_i, i = 1, 2, \dots, n$ ). Further, XORing these Intermediate Shares results in generation of  $n$  different Secret Shares ( $SS_1, SS_2, \dots, SS_n$ ) as shown in Fig. 2. For experiment, one original Biometric image ( $S_1$ ) with two Cover images ( $C_1$  and  $C_2$ ) are chosen, which results in generation of three Secret Shares ( $SS_1, SS_2$  and  $SS_3$ ). The main drawback of this method is that, it reveals some Biometric information as shown in Fig. 2.



**Fig. 2.** Generation of Secret Shares using M1 (top row) Secret image ( $S_1$ ) and Cover images ( $C_1$  &  $C_2$ ), (middle row) Intermediate Shares ( $I_1$ – $I_3$ ), (bottom row) Secret Shares ( $SS_1$ – $SS_3$ )

### 3.2 M2: Original Biometric Image (Secret) and $n - 1$ Randomly Permuted Versions (Cover) of Original Biometric Image

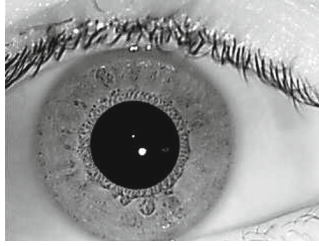
In this method, original Biometric image ( $S_1$ ) and  $n - 1$  Cover images ( $C_1, C_2, \dots, C_{n-1}$ ) are combined. In contrast to M1, Cover images are Randomly Permuted versions of Secret image ( $S_1$ ). For experimental work, we have chosen one Biometric image ( $S_1$ ) with two Cover images ( $C_1$  and  $C_2$ ), which generates three Secret Shares ( $SS_1, SS_2$  and  $SS_3$ ). The main objective of any Cancelable Biometric based technique is that, it should not leak any partial or full information regarding its original features or Biometric image. The main disadvantage with this method is, its Secret Share ( $SS_1$ ) reveals original Biometric information as shown in Fig. 3. Hence, the user may be assigned any of the Secret Shares except  $SS_1$ .



**Fig. 3.** Generation of Secret Shares using M2 (top row) Secret image ( $S_1$ ) and Cover images ( $C_1$  &  $C_2$ ), (middle row) Intermediate Shares ( $I_1$ – $I_3$ ), (bottom row) Secret Shares ( $SS_1$ – $SS_3$ )

### 3.3 M3: Secret and Cover Images Are Randomly Permuted Version of Original Biometric Image

In this method, both Secret image ( $S_1$ ) as well as  $n - 1$  Cover images ( $C_1, C_2, \dots, C_{n-1}$ ) are Randomly Permuted versions of the original Biometric image as shown in Fig. 4. For experiment work, one Secret image ( $S_1$ ) and two Cover images ( $C_1$  and  $C_2$ ) are chosen, which results in generation of three Secret Shares ( $SS_1, SS_2$  and  $SS_3$ ). Among these three Secret Shares, one Secret Share is given to the genuine user and other two Secret Shares are stored in the decentralized database. The difference between this method with other two methods is that, whole the Algorithm is applied on Randomly Permuted versions of the original Biometric or Secret image. The main advantage of this technique over the other methods i.e. M1 and M2 is that, Secret Shares generated using M3 do not reveal any visual information about the original Biometric image, which is the major requirement for any Cancelable Biometric based system. Another advantage with this technique is, if anyhow an intruder accessed one or all Secret Shares, and tries to retrieve original Biometric image and Cover images by reverse engineering process. At the end, he/she will receive only Random Permuted version of Secret image, which is again distorted one. This supports the non-invertible property of this method for Cancelable Biometrics which makes it more secured than others.



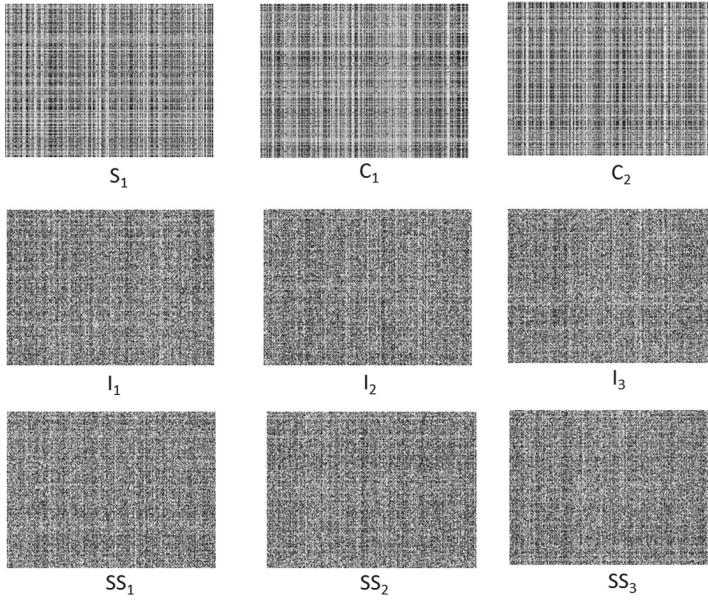
**Fig. 4.** Original iris image

## 4 Experimental Set Up and Results

For experiment work, one Biometric/Secret image and two Cover images are selected for proposed three methods M1, M2 and M3, which results in generation of three Secret Shares as shown in Fig. 2, 3, and 5. Experiments are performed on IIT Delhi Iris Database (version 1.0). This database consists of 1120 iris images collected from 176 males and 48 females between age group of 14–55 years. All the images are in bitmap (\*.bmp) format with  $320 \times 240$  pixel resolution. We have selected 10 users from database with 3 images corresponding to each user (total of 30 images). We have performed two types of experiment: (i) Intra Co-relation Coefficient (ii) Cross Co-relation Coefficient. In first one, we have generated Secret Shares corresponding to all 3 images of individual users separately. After this, we find the similarity between the original Shares ( $S_1, C_1$  and  $C_2$ ) and generated Secret Shares ( $SS_1, SS_2$  and  $SS_3$ ) of individual user respectively. This means, Co-relation between ( $S_1$  &  $SS_1, C_1$  &  $SS_2, C_2$  &  $SS_3$ ) of single user. The motive behind doing this is to show, the dissimilarity between the original Shares and generated Secret Shares of single user. In Cross Co-relation Coefficient, we have calculated the Co-relation between Secret Shares generated for individual user with Secret Shares of rest of users i.e Co-relation between ( $SS_1, SS_2$  and  $SS_3$ ) of user 1 with ( $SS_1, SS_2$  and  $SS_3$ ) of other users (2, 3, ..., 10) respectively. The rationale behind Cross Co-relation Coefficient is to show, how individual user's Secret Shares are not related to each other. This supports the diversity characteristics of Cancelable Biometric based system i.e. each generated template for different users should be different with others. The Co-relation between the original Shares and generated Secret Shares should be minimum. Lower value of Co-relation results in higher security of Cancelable Biometric system. Low Co-relation value is the major requirement in most Cancelable Biometric based applications. It means that even if an intruder accessed Secret Shares corresponding to genuine user by some forgery attacks, he/she will be unable to access the original Biometric. This is due to no information leakages by any Secret Shares about the original Shares. Hence, users original Biometric will remain safe and secure by our proposed method *M3*.

The performance of the proposed schemes is also evaluated in terms of False Accept Rate (FAR), False Reject Rate (FRR), Genuine Accept Rate (GAR),





**Fig. 5.** Generation of Secret Shares using M3 (top row) Secret image ( $S_1$ ) and Cover images ( $C_1$  &  $C_2$ ), (middle row) Intermediate Shares ( $I_1$ – $I_3$ ), (bottom row) Secret Shares ( $SS_1$ – $SS_3$ )

**Table 2.** Performance of the proposed methods on IIT Delhi iris dataset by Intra Co-relation Coefficient

Method	Average Intra Co-relation Coefficient
Method 1	0.3400
Method 2	0.0079
Method 3	<b>0.0077</b>

**Table 3.** Performance of the proposed methods on IIT Delhi iris dataset by Cross Co-relation Coefficient

Method	Average Cross Co-relation Coefficient
Method 1	0.3800
Method 2	0.0158
Method 3	<b>0.0157</b>

True Error Rate (TER) and True Success Rate (TSR). Here NI and NG are total number of imposters and total number of genuine users respectively. The performance measures are computed as follows:

$$FAR = FA/NI \quad (1)$$

$$FRR = FR/NG \quad (2)$$

$$GAR = 1 - FRR \quad (3)$$

$$TER = (FA + FR)/(NG + NI) \quad (4)$$

$$TSR = 1 - TER \quad (5)$$

In above performance measures, *FAR* is defined as how many imposters are accepted as genuine users by the system. In other words, it can be defined as number of impostors scores exceeding the threshold. *FRR* is defined as how many genuine users are rejected by the system. It is defined as number of genuine user(s) scoring less than threshold. *GAR* is defined as the number of actual genuine users accepted by the system. TER stands for true error rate of the system as we can see by Eq. 4 while TSR is true success rate of the system which denotes the accuracy of the system.

#### 4.1 Discussion

As shown in Fig. 2, it can be observed that in method M1, Secret Shares that have been generated reveal some amount of information about the original Biometric and the Biometric trait can be easily guessed by the intruder. Another important requirement in method M1 is that two natural images are required for generation of Secret Shares. For method M2, there is no requirement of external natural images but the first Secret Share still reveals information about the Biometric modality. Rest of the Secret Shares do not reveal any visual cues about the Biometric. Lastly, in method M3, Secret image as well as Cover images, all are randomly permuted versions of the original Biometric image and we are able to achieve satisfactory performance in terms of generated Secret Shares.

Average Intra Co-relation and Average Cross Co-relation values are also depicted in Tables 2 and 3. The average Intra Co-relation value of M3 is 97.74% is better than M1 and 2.53% better than M2. The average Cross Co-relation value of M3 is 95.87% better than M1 and 0.63% better than M2. Lower value of Co-relation signifies the lower Co-relation among the Shares. In our result, Tables 2 and 3, we can see M3 has best value of Intra Co-relation as well as Cross Co-relation than M1 and M2.

Ideally, the performance measures FAR, FRR and TER should be close to 0 while GAR and TSR should be close to 1. The experimental results for all the three methods in terms of above five performance measures are shown in Table 4. It is easy to observe that the performance of M3 is best among all the proposed methods across all the performance measures. In respect of FAR, method M3 is 50.06% and 44.52% better than M1 and M2 respectively. Similarly in terms



of FRR, method M3 is 87.22% more effective than M1 and 59.99% than M2 respectively. The performance of M3 in terms of GAR is more than 93% which is better than M1 and M2 by 95.33% and 12% respectively. True Error Rate (TER) for M3 is 85.05% and 41.71% better than M1 and M2 respectively while True Success Rate (TSR) of M3 is 60.58% and 49.80% better than M1 and M2 respectively. Hence, we can say that M3 performs best in terms of all the performance measures. This is also shown by marking the numerical values in bold for the best method in Table 4.

**Table 4.** Performance measures of proposed methods

Method	FAR	FRR	GAR	TER	TSR
M1	0.0741	0.5222	0.4778	0.4160	0.5840
M2	0.0667	0.1667	0.8333	0.1067	0.8933
M3	<b>0.0370</b>	<b>0.0667</b>	<b>0.9333</b>	<b>0.0622</b>	<b>0.9378</b>

## 5 Conclusion and Future Work

In this paper, a solution for Cancelable Biometric template generation using Visual Secret Sharing is proposed. For this purpose,  $n$  Secret Shares are generated corresponding to one Biometric image and  $n - 1$  Cover images. Three different methods for generating Cancelable Biometric Templates using Visual Secret Sharing technique have been proposed i.e. M1, M2 and M3 in which original Biometric image, natural gray scale images and randomly permuted versions of original Biometric have been used in various combinations. The performance of above methods are measured in terms of Co-relation Coefficient, FAR, FRR, GAR, TER and TSR. Among the three proposed schemes, M3 performs best in terms of all the performance measures. The drawback of method M1 is that the Secret Shares reveal some information about the original Biometric. Moreover, the average Intra and Cross Co-relation Coefficient is also very high as compared to methods M2 and M3. Similarly, M2 also reveals some information about the original Biometric but only in the first Secret Share and significantly improves the Intra and Cross Co-relation Coefficient than M1. In method M3, none of the Secret Shares leaks any information about the original Secret image and Cover images. However, the storage space requirement for Secret Shares puts extra burden for computational resources and is dependent upon the number of Secret Shares generated. In future, we shall explore this domain by generating the Secret Shares in some other manner and which can give better performance than the one proposed currently.

**Acknowledgment.** We acknowledge Ministry of Human Resource Development, Govt. of India for supporting this research by providing fellowship to one of the authors, Ms. Manisha. One of the authors, Dr. Nitin Kumar is thankful to Uttarakhand State Council for Science and Technology, Dehradun, Uttarakhand, India for

providing financial support towards this research work (Sanction No. UCS & T/R & D-05/18-19/15202/1 dated 28-09-2018).

## References

1. Prabhakar, S., Pankati, S., Jain, A.K.: Biometric recognition: security and privacy concerns. *IEEE Secur. Priv.* **1**, 33–42 (2003)
2. Jin, Z., Hwang, J.Y., Lai, Y.L., Kim, S., Teoh, A.B.J.: Ranking-based locality sensitive hashing-enabled cancelable biometrics: index-of-max hashing. *IEEE Trans. Inf. Forensics Secur.* **13**(2), 393–407 (2018)
3. Lumini, A., Nanni, L.: An improved biohashing for human authentication. *Pattern Recogn.* **40**(3), 1057–1065 (2007)
4. Manisha, Kumar, N.: Cancelable biometrics: a comprehensive survey. *Artif. Intell. Rev.* (2019). <https://doi.org/10.1007/s10462-019-09767-8>
5. Kumar, N., Singh, S., Kumar, A.: Random permutation principal component analysis for cancelable biometric recognition. *Appl. Intell.* **48**(9), 2824–2836 (2018)
6. Deshmukh, M., Nain, N., Ahmed, M.: Efficient and secure multi secret sharing schemes based on boolean XOR and arithmetic modulo. *Multimed. Tools Appl.* **77**(1), 89–107 (2016)
7. Kaur, H., Khanna, P.: Biometric template protection using cancelable biometrics and visual cryptography techniques. *Multimed. Tools Appl.* **75**(23), 16333–16361 (2016)
8. Kaur, H., Khanna, P.: Non-invertible biometric encryption to generate cancelable biometric templates. In: *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1 (2017)
9. Sinduja, R., Sathiya, R.D., Vaithyanathan, V.: Sheltered iris attestation by means of visual cryptography (SIA-VC). In: *IEEE International Conference on Advances in Engineering, Science and Management*, pp. 650–655 (2012)
10. Daugman, J.G.: Biometric personal identification system based on iris analysis. U.S. Patent 5: 291-560 (1994)
11. Cherabit, N., Chelali, F.Z., Djeradi, A.: Circular hough transform for iris localization. *Sci. Technol.* **2**(5), 114–121 (2012)
12. Revenkar, P.S., Anjum, A., Gandhare, W.Z.: Secure iris authentication using visual cryptography. arXiv preprint [arXiv:1004.1748](https://arxiv.org/abs/1004.1748) (2010)
13. Ross, A., Othman, A.: Visual cryptography for face privacy. *Biometric Technol. Hum. Identif. VII* **7667**, 766–70 (2010)
14. Ross, A., Othman, A.: Visual cryptography for biometric privacy. *IEEE Trans. Inf. Forensics Secur.* **6**(1), 70–81 (2010)
15. Monoth, T.: Tamperproof transmission of fingerprints using visual cryptography schemes. *Procedia Comput. Sci.* **2**, 143–148 (2010)
16. Naor, M., Shamir, N.: Visual cryptography. In: *Proceedings of the Advances in Cryptology–Eurocrypt*, pp. 1–12 (1995)



# An Integrated Safe and Secure Approach for Authentication and Secret Key Establishment in Automotive Cyber-Physical Systems

Naresh Kumar Giri<sup>1</sup>, Arslan Munir<sup>2(✉)</sup>, and Joonho Kong<sup>3</sup>

<sup>1</sup> Intel Corporation, Santa Clara, CA 95054, USA  
ngiri@ksu.edu

<sup>2</sup> Kansas State University, Manhattan, KS 66506, USA  
amunir@ksu.edu

<sup>3</sup> Kyungpook National University, Daegu 41566, South Korea  
joonho.kong@knu.ac.kr

**Abstract.** In this paper, we propose an integrated safe and secure approach for operation in automotive cyber-physical systems (CPS). The proposed approach incorporates a novel protocol for authentication and secret key establishment for electronic control units (ECUs) in automotive CPS. The approach leverages certificates and elliptic curve cryptography (ECC) for authentication and secret key establishment, and symmetric encryption and hash-based message authentication codes for providing confidentiality and integrity, respectively, for messages on in-vehicle bus. To incorporate safety primitives, the approach leverages multi-core ECUs and provide fault tolerance by redundant multi-threading (FT-RMT), FT-RMT enhanced by quick error detection (FT-RMT-QED), and FT-RMT with lightweight check-pointing (CP). The proposed approach ensures that the simultaneous integration of security and safety primitives in intra-vehicle ECU communication does not violate real-time constraints of automotive CPS applications. We demonstrate the proposed approach through a steer-by-wire case study. Results verify that our proposed approach integrates confidentiality, integrity, authentication, and secret key establishment in intra-vehicle networks without violating real-time constraints even in the presence of errors in computation and transmission.

**Keywords:** Automotive · Cyber-physical systems · Fault tolerance · Security · Authentication · Key establishment

## 1 Introduction and Motivation

Modern vehicles are equipped with a multitude of sensors, radio interfaces, and digital processors, also known as electronic control units (ECUs), that are connected with each other via in-vehicle networks, such as controller area network

(CAN), CAN with flexible data-rate (CAN FD), local interconnect network (LIN), and media oriented systems transport (MOST) [12]. However, most of the contemporary automotive ECUs and in-vehicle networks do not have built-in security and/or safety primitives thus making automotive systems susceptible to security and safety vulnerabilities. Attackers can infiltrate into in-vehicle networks through a compromised ECU and can read/alter messages, which enables the attackers to control many safety critical systems such as disabling brakes, stopping the engine, opening doors, changing heating and cooling, and turning on/off lights [8, 11]. Cyber-physical attributes of modern automotive systems directly relates security vulnerabilities to automobile's physical safety and dependability.

In addition to security vulnerabilities, modern automobiles are also susceptible to electronic failures. Harsh operating environments, external noise, and radiations make automotive cyber-physical systems (CPS) susceptible to *permanent*, *transient* and *intermittent faults*. Automotive CPS have stringent safety requirements as stipulated by ISO 26262 [6]. ISO 26262 requires that at least one critical fault must be tolerated by automobiles without loss of functionality. Thus, both security and safety primitives need to be incorporated in automotive CPS. However, simultaneous integration of security and safety in automotive CPS is challenging. The biggest challenge in the simultaneous integration of security and safety is to avoid violation of the automotive CPS application's hard real-time constraints.

Previous works [12, 14] have incorporated symmetric cryptography primitives, such as advanced encryption standard (AES) for confidentiality along with hash-based message authentication code (HMAC) for message integrity. The symmetric cryptography, however, requires pre-shared symmetric keys between communicating parties. Most of the existing works assume that these keys are pre-programmed by the original equipment manufacturers (OEMs) during vehicle manufacturing. Nevertheless, OEMs tend to use identical keys across series of ECUs and even vehicles, which makes an entire series of ECUs and vehicles vulnerable to security attacks when a single key is compromised. Furthermore, stored symmetric keys can be easily extracted using side-channel analysis (SCA) attacks, which render storage of permanent symmetric keys in ECUs susceptible to vulnerabilities.

In this paper, we propose a safe and secure approach for *symmetric (session) key establishment* as well as regular operation in automotive CPS. The proposed key establishment protocol eliminates the need for storing symmetric keys permanently in ECUs thus preventing an attacker from gaining access to symmetric keys through SCAs. Our proposed scheme ensures that only authenticated ECUs can participate in communication over the intra-vehicle network. The ECU authentication and secret key establishment in our approach leverages certificates and elliptic curve cryptography (ECC). ECC is chosen over other asymmetric cryptography approaches (e.g., RSA, Elgamal) because ECC provides higher security with comparatively shorter key lengths. ECC implemen-

tations, therefore, have lower computation complexity and are more suitable for applications having real-time deadlines such as automotive CPS.

To address the safety requirements stipulated by ISO 26262 [6], we have incorporated various fault tolerance (FT) approaches such as FT by redundant multi-threading (FT-RMT), FT-RMT enhanced with quick error detection (FT-RMT-QED), and FT-RMT with checkpointing (FT-RMT-CP). Our main technical contributions are as follows:

- Proposal of an integrated safety and security approach that simultaneously incorporates security (key establishment, confidentiality, integrity, and authentication) and FT primitives while adhering to real-time constraints of automotive CPS. We demonstrate this approach through a steer-by-wire (SBW) case study.
- Proposal of a certificate-based authentication scheme for ECUs leveraging ECC to ensure that only authenticated ECUs can participate in in-vehicle communication.
- Proposal of a novel symmetric (session) key establishment protocol to enable the ECUs to communicate securely over the in-vehicle networks.
- Safety integration through various FT approaches such as FT-RMT, FT-RMT-QED, and FT-RMT-CP.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents the proposed integrated safety and security approach. The proposed certificate-based ECU authentication and symmetric key establishment mechanisms are elaborated in Sect. 4. Section 5 discusses the SBW system and its timing model, which is used as a case study to verify our proposed approach. Section 6 discusses the results. Finally, Sect. 7 presents concluding remarks.

## 2 Related Work

Many previous works have studied security of automotive systems. Koscher et al. [8] analyzed internal and external attack surfaces through which an attacker could control automotive subsystems. The authors practically demonstrated an attack on a car through onboard diagnostics (OBD) port by using a self-developed software. The authors successfully controlled radio, instrument panel cluster, body controller, engine, brakes, and heating, ventilation, and air conditioning (HVAC) subsystems. Rouf et al. [5] studied the security and privacy of a tire pressure monitoring system (TPMS). Huang et al. [4] classified the in-vehicle network in three different layers: control layer, middle layer and external interface layer, and studied the security vulnerabilities at each layer. All these works have motivated the necessity of security and authentication mechanisms within in-vehicle networks for realizing secure and dependable automotive CPS.

Lin et al. [9] proposed integration of message authentication codes (MACs) in CAN data frames to prevent masquerade and replay attacks. Wolf et al. [17] proposed a vehicular hardware security module (HSM) that provided hardware support for symmetric cryptography, asymmetric cryptography, hash function,

and pseudorandom number generator. However, the HSM did not support any FT features which are crucial for safe operation of modern automobiles. Fassak et al. [2] proposed a protocol for authenticating ECUs and establishing session keys between them on the CAN bus using ECC. However, if a new ECU was added to the CAN bus, the protocol required the storage of all other ECUs to be updated with the public key of the new ECU. Furthermore, the protocol utilized truncated MACs which could increase the probability of collision between hashes.

The safety for automotive embedded system has been studied in some previous works. Beckschulaze et al. [1] have studied different FT approaches on dual-core micro-controllers. Munir et al. [12] and Poudel et al. [14] have proposed multicore ECU based design for secure and dependable cybercars. However, these works did not discuss symmetric key establishment and distribution for automotive CPS.

### 3 Integrated Safety and Security Approach

Fig. 1 provides an overview of our proposed integrated safe and secure approach for cybercar design. In this work, we focus on CAN FD as the vehicular network, however, our approach is equally applicable for CAN and FlexRay. The figure shows the operations involved at both the sending and receiving CAN FD nodes to integrate safety and security primitives.

#### 3.1 Safety

To address the safety requirements stipulated by ISO 26262 [6], we incorporate various FT approaches such as FT-RMT, FT-RMT-QED, and FT-RMT-CP. The FT-RMT uses two different threads and a dual-core architecture to compute the same safety-critical computation. The results of the two threads are matched at the end of the computation to detect any error. If an error occurs during computation, recomputation is carried out in both the threads. This recomputation fixes the errors that are caused by transient faults which constitutes majority of errors in automotive CPS. The FT-RMT-QED enhances FT-RMT with quick error detection (QED) mechanism [12]. In FT-RMT-QED, the main thread executes original instructions and the check instructions, which are inserted at different points in the program/computation, whereas another thread executes duplicated instructions. The FT-RMT-QED permits earlier detection of errors in the program via inserted checks as compared to error detection at the end of the program in FT-RMT. The FT-RMT-CP introduces checkpoints at various portions of the safety-critical code. When an error/fault occurs in the program, the execution resumes from previous (last) checkpoint, which removes the need of re-executing the whole program during errors. The checkpoint is lightweight because the checkpointing in FT-RMT-CP stores minimum state information just enough to resume the computation.

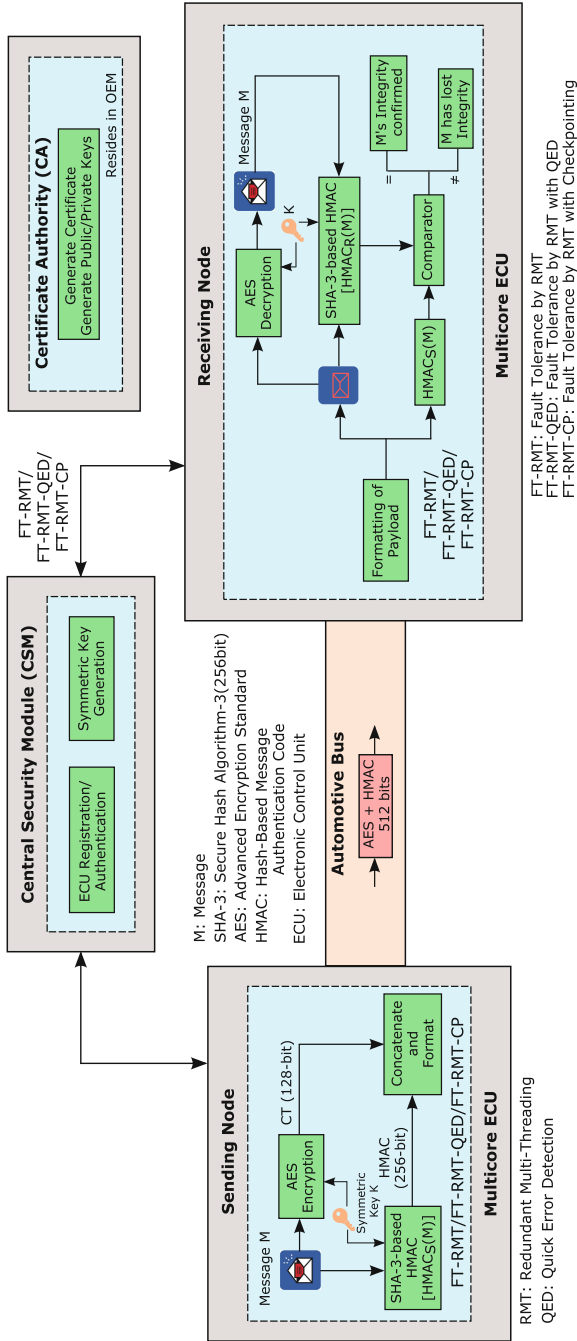


Fig. 1. An integrated safe and secure approach for in-vehicle networks.

### 3.2 Security Threat Model

Assuming an adversary has gained access to in-vehicular network, this section briefly discusses the associated security threat model against which our proposed approach provides resilience [12, 14].

**Threat 1—Passive Eavesdropping & Traffic Analysis:** An attacker may perform passive eavesdropping which means he/she can sniff, steal, and analyze all the traffic information from intra-vehicular network to obtain critical information about driver, vehicle, and navigation routes, which can put the driver and passenger at great risk.

**Threat 2—Active Eavesdropping & Message Injection:** An attacker may conduct spoofing attacks by actively injecting/modifying messages in the in-vehicle network. Moreover, by injecting well targeted messages, an adversary might be able to gain additional information from the system reaction via active eavesdropping.

*Threats 1 and 2* are possible in the absence (or breaking) of data confidentiality, integrity, and authentication in in-vehicle networks. To address the threat of active and passive attacks, security primitives can be incorporated, such as encryption for providing confidentiality to discourage passive attacks and message authentication codes to discourage active attacks.

**Threat 3—Key Extraction from Storage:** The approaches that can be used to counter *Threats 1 and 2* typically use symmetric key cryptography because of less computation overhead as compared to public key cryptography. The symmetric key cryptography, however, requires a symmetric key which the previous works [12, 14] assume is stored in a secure memory. However, an adversary can conduct SCAs on memory to extract the stored symmetric key, which can compromise not only the single ECU or single vehicle but may also compromise a whole series of vehicle, because same series of ECUs tend to use the same symmetric key. To tackle *Threat 3*, a symmetric key needs to be generated during vehicle startup to prevent key extraction attacks from storage. Furthermore, the key needs to be refreshed periodically to prevent replay attacks. The proposed approach develops a solution for symmetric key generation and distribution for automotive systems.

### 3.3 Security

To provide resilience against the considered threat model (Sect. 3.2), we use AES-128 for providing message confidentiality and HMAC based on SHA-3 (Secure Hash Algorithm-3) for ensuring message integrity. The proposed approach uses “encrypt-and-MAC”. The receiving node decrypts the message and compare the HMAC calculated on the receiving side with the one received from the sender. If the two HMACs are equal, the message is authentic otherwise the message has lost its integrity. Another key aspect of the proposed approach is ECU authentication and key establishment. Our proposed approach incorporates a central



security module (CSM), which is responsible for ECU registration, authentication, and symmetric key establishment. In Fig. 1, CA is certificate authority that generates the necessary keys (i.e., public and private keys) for all ECUs and the CSM. The CA has its own public and private keys. The CA's public key is shared with all ECU nodes, whereas the private key is stored secretly. The CA can be automotive OEM.

## 4 Authentication and Secret Key Establishment

This section discusses the proposed certificate-based ECU authentication and symmetric key establishment mechanisms.

### 4.1 Certificate Generation

Each ECU  $i \forall i \in \{1, 2, \dots, n_E\}$ , where  $n_E$  denotes the total number of ECUs on the in-vehicle bus, will have a public key  $k_{pub,E_i}$  and a private key  $k_{pr,E_i}$ , that is,  $k_{E_i} = (k_{pub,E_i}, k_{pr,E_i})$ . The CA is responsible generating these public and private keys for each ECU. The CA also generates the certificate for each  $ECU_i$  by combining the ECU's public key  $k_{pub,E_i}$  and its identity  $ID_{E_i}$  with the signature  $S_{E_i}$  generated over  $k_{pub,E_i}$  and  $ID_{E_i}$  using the private key of the CA  $k_{pr,CA}$ . The ID of each ECU is assigned by the OEM (e.g., an ECU's serial number can serve as the ECU's ID).

### 4.2 ECU Authentication

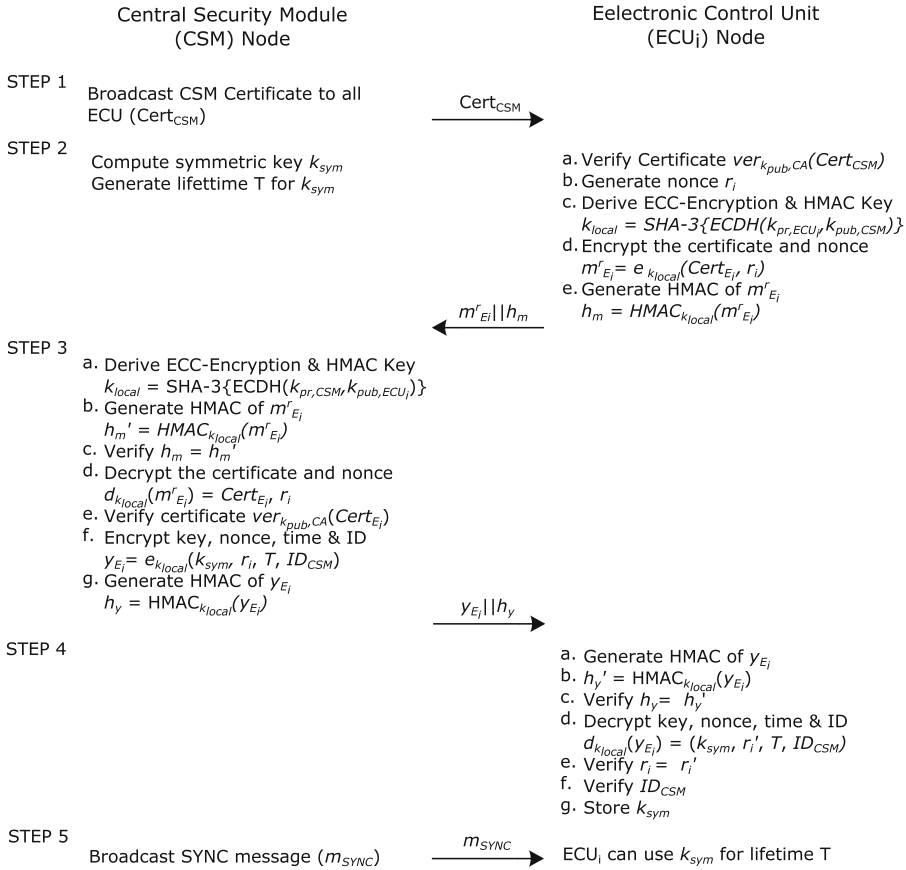
In our approach, authentication of ECUs is done by certificate verification. By using  $k_{pub,CA}$ , we can verify if the signature is legitimate or not. The approach uses elliptic curve digital signature algorithm (ECDSA) [10] for the signature generation and verification.

### 4.3 Symmetric Key Establishment Protocol

After verifying all the ECUs, the CSM generates new symmetric key and distributes it to all ECUs, the process known as *key establishment*. This process is repeated after certain time period  $T$  to refresh the keys periodically. Figure 2 presents the proposed certificate-based protocol for symmetric key establishment. The proposed protocol comprises of five key steps as described below.

Step 1: At the vehicle start up, the CSM advertises/broadcasts its public key, ID, and certificate to all other ECUs in the vehicle. All ECUs verify the authenticity of CSM.

Step 2 (ECU Side): After CSM authentication, the ECU  $i$  which requires registration or authentication generates a random nonce  $r_i$ . For authentication of ECU  $i$ , its certificate and nonce is sent to the CSM in encrypted form because



**Fig. 2.** Proposed symmetric key establishment protocol.

only the CSM (and no other malicious ECU) should be able to retrieve the certificate and nonce of the ECU  $i$ . Hence, a common secret is generated using the private key of ECU  $i$  and the public key of CSM, which is then transformed to obtain local key  $k_{local}$ . The local key is used to encrypt the message (ECU  $i$  certificate and  $r_i$ ) and generate HMAC of the message  $h_m$ . The encrypted message  $m_{E_i}^r$  and  $h_m$  are concatenated ( $m_{E_i}^r || h_m$ ) and sent to the CSM. Here, HMAC is appended to the message to maintain the message integrity.

**Step 2 (CSM Side):** While ECUs are authenticating the CSM, the CSM generates a new symmetric keys  $k_{sym}$  and the lifetime  $T$  for  $k_{sym}$ .

**Step 3:** After receiving the request message  $m_{E_i}^r$  along with the hash  $h_m$ , the CSM verifies the HMAC to find out the integrity of the message. To calculate the HMAC at CSM, the CSM needs to have local key  $k_{local}$  which was used at ECU side (step 2). The local key is again generated using the common secret of private key of CSM and public key of ECU  $i$ . After verification of HMAC, the

CSM decrypts the message, and verifies the certificate  $Cert_{E_i}$  using the public key of CA. After this verification, the CSM encrypts the generated symmetric keys  $k_{sym}$  (in step 2 at CSM), nonce  $r_i$ , lifetime  $T$ , and ID of the CSM  $ID_{CSM}$ , denoted as message  $y_{E_i}$  with  $k_{local}$ . The CSM also calculates the hash  $h_y$  of  $y_{E_i}$  and then send this response message  $y_{E_i}$  concatenated with  $h_y$  back to the ECU  $i$ .

Step 4: The received message packet  $y_{E_i}$  is tested for message integrity by verifying the HMAC, and then decrypted by the ECU  $i$  using its local key (generated in step 2). The ECU  $i$  also verifies the random nonce  $r_i$  received from the CSM and its original random nonce. Furthermore, the ECU  $i$  verifies  $ID_{CSM}$  with the one obtained from the CSM certificate in Step 1. After successful verification, the ECU accepts the symmetric key  $k_{sym}$ .

Step 5: Finally, the CSM broadcast SYNC (synchronization) message instructing all ECUs to use the newly generated symmetric key for time  $T$ .

Each ECU starts communicating with other nodes using this newly established key for symmetric encryption and HMAC for regular operation. After time period  $T$ , each ECU requests for a new key for key refreshment by sending a message request as in step 2 ECU side (Fig. 2). The CSM then distributes new symmetric keys to ECUs as shown in Fig. 2. The proposed protocol uses ECC for the computation of steps 2, 3 and 4 explained above. The algorithm used in step 2(c, d, and e) and step 3(a, b, c, d) are based on the steps of elliptic curve integrated encryption scheme (ECIES) [10].

## 5 Case Study: Steer-by-Wire Subsystem

A SBW system replaces the heavy mechanical steering column with an electronic system. The SBW subsystem provides two functions: front axle (FA) control (FAC) and hand-wheel (HW) force feedback (HWF). The FAC controls the wheel direction according to hand-wheel, whereas HWF provides the mechanical like feedback to hand-wheel. The rotation on hand-wheel is sensed by hand-wheel sensors and sensed values are fed as the input to HW sensor (HWS) ECU1. HWS ECU1 processes the information to determine the commands for the front axel actuator (FAA) ECU1, which are then sent through the in-vehicle network (e.g., CAN FD bus) to the FAA ECU1. The FAA ECU1 processes the received CAN FD packet to extract the commands sent from HWS ECU1 and then turns the actuators accordingly to rotate the wheels. In this work, we only focus on FAC part to compute the response time and error resilience of our proposed approach.

The delay between the driver's request at HWS and the corresponding response at FAA has significant impact on the reliability of SBW subsystem. The end-to-end delay/response time ( $\tau_r$ ) is regarded as a quality of service (QoS) metric, however, it becomes a reliability metric, which can be defined in terms of *behavioral reliability*, if this delay exceeds a critical threshold value  $\tau_r^{max}$  as the driver can lose control of the vehicle beyond this threshold. The behavioral reliability is the probability that the worst-case response time is less than the

critical threshold. This threshold value is defined by automotive OEMs. The response delay  $\tau_r$  time is given by following equation

$$\tau_r = \tau_p + \tau_m + \tau_s, \quad (1)$$

where,  $\tau_p$  is pure delay,  $\tau_m$  is mechatronic delay, and  $\tau_s$  sensing delay. The mechatronic delay is introduced by the actuators (electric motor in SBW system case). The sensing delay is the delay introduced due to sensing and sampling of measurements. Since  $\tau_m$  and  $\tau_s$  can be bounded by a constant and can vary for different kind of sensors and actuators, we focus on  $\tau_p$  for our analysis [16]. Here  $\tau_p$  includes ECUs computational delay for processing the control algorithm, computational delay for processing the incorporated security and safety primitives, and transmission delay including bus arbitration. To ensure safe operation of the vehicle as governed by behavioral reliability,  $\tau_p$  should be less than or equal to the maximum tolerable pure delay  $\tau_p^{max}$ , that is,  $\tau_p \leq \tau_p^{max}$ . Mathematically,  $\tau_p$  for the FAC function can be written as,

$$\tau_p = rcc1 \cdot \tau_{hws}^{ecu1} + rtc \cdot \tau_{bus} + rcc2 \cdot \tau_{faa}^{ecu1} \leq \tau_p^{max}, \quad (2)$$

where  $\tau_{hws}^{ecu1}$  and  $\tau_{faa}^{ecu1}$  denote the computation time at HWS-ECU1 and FAA-ECU1, respectively;  $\tau_{bus}$  represents the transmission time for a message on in-vehicle bus from HWS-ECU1 to FAA-ECU1;  $rcc1$  and  $rcc2$  represent the number of recomputations that are needed to be done at HWS-ECU1 and FAA-ECU1, respectively; and  $rtc$  represents the number of retransmissions required for an error-free transmission of a secure message over in-vehicle bus.

The pure delay  $\tau_p$  for FAC can be considered for two cases: (i) delay during regular operation  $\tau_p^R$ , and (ii) delay during key refreshment operation  $\tau_p^K$ .

**1. Delay During Regular Operation:** The delay during regular operation comprises of encryption/decryption and HMAC computation delay at the sending and receiving nodes plus the message transmission delay on in-vehicle network, that is,

$$\tau_p^R = rcc1 \cdot \tau_{hws}^{ecu1,R} + rtc \cdot \tau_{bus} + rcc2 \cdot \tau_{faa}^{ecu1,R} \leq \tau_p^{max}, \quad (3)$$

where  $\tau_{hws}^{ecu1,R}$  and  $\tau_{faa}^{ecu1,R}$  denote the computation time at HWS ECU1 and FAA ECU1, respectively, during regular operation.

**2. Delay During Key Refreshment Operation:** The delay during key refreshment operation comprises of the delay during regular operation and the delay due to key refreshment operation. The  $\tau_p^K$  for the CSM refreshing the key for HWS ECU1 (Fig. 2) can be written as

$$\tau_p^K = rcc1 \cdot \tau_{hws}^{ecu1,R} + rtc \cdot \tau_{bus} + rcc2 \cdot \tau_{faa}^{ecu1,R} + rcc3 \cdot \tau_{hws}^{ecu1,K} \leq \tau_p^{max}, \quad (4)$$

where  $\tau_{hws}^{ecu1,K}$  denotes the delay for key refreshment at HWS ECU1 and  $rcc3$  represents recomputations required to yield an error-free result at HWS ECU1.

## 6 Result and Discussion

### 6.1 Experimental Setup

For ECC implementation in key establishment protocol, we have used a prime field curve P-192 from NIST [3]. We have used AES-128 for providing confidentiality and SHA-3 256 for providing message integrity. We have implemented the proposed approach that includes ECU authentication, and key establishment protocol in NVIDIA's Jetson TX2 platform, which has four 64-bit ARM Cortex-A57 cores running *Ubuntu 14.04.4 LTS* at 2.0 GHz. The future generations of automotive ECUs are expected to possess comparable compute capabilities [13]. The code for providing confidentiality, integrity, authentication, and key establishment is written in *C* language. *OpenMp* is used for FT-RMT implementation. For the SBW system, we assume the steering wheel sensor sampling rate of 400 Hz, which corresponds to the sampling/sensing delay  $\tau_s$  of 2.5 ms [7]. We simulate our SBW system in Vector CANoe [15].

### 6.2 Timing Analysis

We have measured the timing response of the key-establishment process in different FT settings. For timing analysis, we inject soft errors at different points in the program. Our approach emulates bit flipping in the program/memory due to noise and/or radiation from the environment.

**Performance Analysis of Key Establishment:** Table 1 shows execution times of different steps of the key establishment protocol (Fig. 2) in various FT operational modes assuming no error in the program/computations. Table 1 does not depict computation time for Step 1 and Step 5 because in these steps, CSM broadcasts a precomputed message packet, which does not require computation. Results indicate that the overhead incurred by various FT approaches is insignificant (less than 1%) because computation time of the protocol steps is large as compared to the overhead.

**Effect of Errors on Performance:** Table 2 shows the execution time of Step 2 on ECU side of the key establishment protocol (Fig. 2) in the presence of errors. In this experiment, a single soft error is injected at different points in the program. If a single error occurs in FT-RMT mode, the entire function is recomputed to rectify the error as the error is detected at the end of the computation in FT-RMT. This results in the computation time with error to be  $2\times$  of the computation time without the error. In FT-RMT-QED, if the error occurs during the start of program/computation, the computation overhead is much less as compared to FT-RMT. However, if error occurs near the end of the computation, the computation overhead is almost the same as that of FT-RMT. The FT-RMT-CP only repeats the execution of those parts where the error has occurred and thus the recomputation overhead depends on the size of code between the two checkpoints. Results indicate that FT-RMT-CP performs best as compared to other FT approaches and considerably reduces the computation

time when error occurs near the end of computation. For example, FT-RMT-CP provides 44.21% and 43.94% reduction in computation time as compared to NFT and FT-RMT-QED, respectively, when the error occurs in ECC encryption of Step 2 on ECU side.

**Table 1.** Performance analysis of key establishment

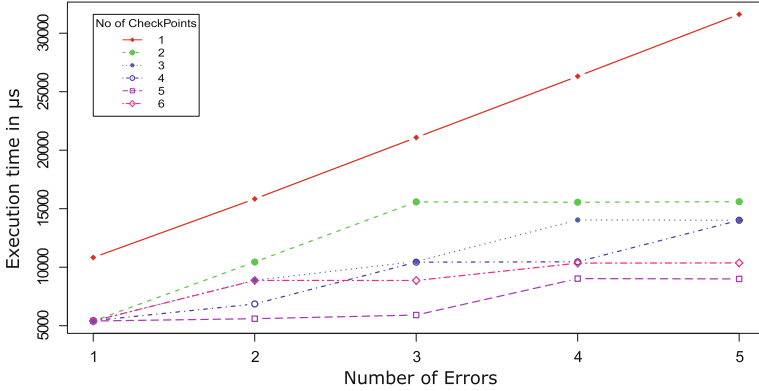
Operational modes	Algorithm steps (time in $\mu sec$ )			
	Step 2		Step 3	Step 4
	ECU side	CSM side	CSM side	ECU side
NFT	5214.63	216.51	5224.71	1563.63
FT-RMT	5259.53	219.80	5226.57	1565.86
FT-RMT-QED	5306.83	220.42	5300.08	1571.29
FT-RMT-CP	5352.41	219.71	5300.93	1585.75

**Table 2.** Effect of errors on performance (for Step 2 ECU side)

Error location	Operational modes (time in $\mu sec$ )		
	NFT	FT-RMT-QED	FT-RMT-CP
@ Verification of certificate	10223.79	8483.39	8760.96
@ Key generation for ECC	10191.91	9951.42	6837.83
@ ECC-encryption	10202.22	10153.84	5692.02
@ ECC-HMAC	10323.72	10224.69	5747.74

**Performance Analysis of Checkpointing:** Fig. 3 shows the execution time of Step 3 of the key establishment protocol (Fig. 2) with varying number of checkpoints and multiple errors injected at different points in the computation. The number of checkpoints varies from one to six whereas the number of errors introduced varies from one to five. The errors have been uniformly distributed over the program in order to provide a fair evaluation. Results indicate that as the number of errors increases, the computational time increases linearly. Furthermore, as the number of inserted checkpoints increases, the time required to rectify the error decreases and thus the overall computation time decreases.

**Performance Analysis of Regular Operation:** Table 3 shows the temporal performance of regular operation, which comprises of encryption/decryption of 2 blocks of 128-bit AES and a 256-bit HMAC, at sender and receiver nodes. The results show that encryption and HMAC at the sender node takes longer time as compared to decryption and HMAC at the receiver node. This is because at the receiver node, the decryption computation is accelerated by using precomputed



**Fig. 3.** Effect of checkpoints and errors on performance.

tables. We observe that FT-RMT at the sender node has 16.2% overhead and the receiver node has 25.2% overhead.

**Table 3.** Timing of regular operation at sender and receiver nodes

Operational mode	Sender node ( $\mu sec$ )	Receiver node ( $\mu sec$ )
NFT	130	87
FT-RMT	151	109

### 6.3 Feasibility Analysis

We have conducted the feasibility analysis of the proposed approach (including the key establishment protocol) to verify that the proposed mechanisms do not violate the real-time constraints of automotive CPS. From extrapolation of QoS score  $\mathcal{S}$  versus  $\tau_p$  [16], we determine that for  $\mathcal{S}$  of 11.08, the critical limit for  $\tau_p$  is 16 ms.

**Regular Operation:** As shown in Table 3, the computational time at the sender node for encryption (2-blocks of 128-bit) and HMAC (256-bit digest) is 0.151 ms, and the computational time at the receiving node for decryption and HMAC is 0.109 ms. The message transmission time using CAN FD obtained through Vector CANoe simulations is 0.12 ms (for a packet of 512 bits) [14]. The total computational and transmission delay without error is 0.38 ms (i.e., 0.151 ms + 0.109 ms + 0.12 ms) for one 512-bit packet of CAN FD (payload of CAN FD is 64 bytes). Furthermore, in a period of 16 ms at least 6 readings are taken by the HW sensor as  $\lfloor \tau_p^{max} / \tau_s \rfloor = \lfloor 16 \text{ ms} / 2.5 \text{ ms} \rfloor = \lfloor 6.4 \rfloor = 6$ . Considering one block of AES can store the reading of one sample, 3 CAN FD frames (as

1 frame contains two AES blocks and thus can hold 2 sensor readings) need to be transmitted within a period of 15 ms without losing any sample value. We note that for CAN protocol, the encrypted and HMAC-ed message transmission would require multiple CAN frames [12].

To resolve the errors in computation, FT-RMT performs recomputation and in case of errors in transmission, erroneous packets are retransmitted. For this study, we assume that in one end-to-end communication, at most two errors can occur in each component/node (i.e., maximum two errors at the sender node, two errors at the receiver node, and two errors in transmission) in the worst case. Experimental results reveal that the total time taken to resolve (i.e., recomputation in case of computation errors and retransmission in case of transmission errors) two errors occurring in each of the components (sender, receiver, transmission) is 0.76 ms. Hence, within  $\tau_p^{max}$  of 15 ms and for 3 packets, time taken to compute and resolve at most 2 errors is 2.28 ms. These results verify the feasibility of regular operation (Sect. 5) of SBW system using the proposed approach.

**Key Establishment and Refreshment:** We also measure the timing of key establishment and refreshment protocol. Considering that each of the communication messages between the ECU and the CSM in the key establishment protocol can be accomplished with one CAN FD packet, then using results from Table 1, the time taken for the key establishment protocol using FT-RMT-CP can be calculated as 0.12 ms (Step 1 communication) + 0.22 ms (Step 2 CSM side) + 5.35 ms (Step 2 ECU side) + 0.12 ms (Step 2 communication) + 5.3 ms (Step 3) + 0.12 ms (Step 3 communication) + 1.58 ms (Step 4) + 0.12 ms (Step 5 communication) = 12.93 ms. The time taken for key refreshment is 12.81 ms (12.93 ms – 0.12 ms = 12.81 ms) since key refreshment takes Steps 2 to 5 in Fig. 2. These results verify that the key refreshment can take place along with the regular operation within the time constraints specified by desired QoS, that is, 12.81 ms + 2.28 ms = 15.09 ms  $\leq$  16 ms, even in the presence of faults. Hence, these results verify the feasibility of the proposed key establishment and refreshment protocol for automotive CPS.

## 7 Conclusion

In this paper, we have proposed a safe and secure approach for secret key establishment and operation in automotive cyber-physical systems (CPS). The proposed approach realizes authentication and symmetric key establishment using certificates and ECC. To incorporate safety, the proposed approach leverages multicore ECUs and various FT approaches such as FT-RMT, FT-RMT-QED, and FT-RMT-CP. Results reveal that FT-RMT-CP performs best as compared to other FT approaches. Furthermore, results verify that our proposed approach provides confidentiality, integrity, authentication, and secret key establishment in intra-vehicle networks without violating real-time constraints of the vehicle even in the presence of errors in computation and transmission.



## References

1. Beckschulze, E., Salewski, F., Siegbert, T., Kowalewski, S.: Fault handling approaches on dual-core microcontrollers in safety-critical automotive applications. In: International Symposium On Leveraging Applications of Formal Methods, Verification and Validation, pp. 82–92. Springer (2008)
2. Fassak, S., El Idrissi, Y.E.H., Zahid, N., Jedra, M.: A secure protocol for session keys establishment between ECUs in the CAN bus. In: Proceedings of IEEE International Conference on Wireless Networks and Mobile Communications (WINCOM), Rabat, Morocco (November 2017)
3. Federal Information Processing Standards Publication: 186-4. Digital signature standard (DSS) (2013)
4. Huang, T., Zhou, J., Wang, Y., Cheng, A.: On the security of in-vehicle hybrid network: status and challenges. In: International Conference on Information Security Practice and Experience, pp. 621–637. Springer (2017)
5. Ishtiaq Roufa, R.M., Mustafaa, H., Travis Taylor, S.O., Xua, W., Gruteserb, M., Trappeb, W., Seskarb, I.: Security and privacy vulnerabilities of in-car wireless networks: a tire pressure monitoring system case study. In: 19th USENIX Security Symposium, Washington DC, pp. 11–13 (2010)
6. ISO: ISO 26262-1:2018: Road vehicles – Functional safety (December 2018). <https://www.iso.org/standard/68383.html>. Accessed 7 June 2019
7. Klobedanz, K., Kuznik, C., Thuy, A., Mueller, W.: Timing modeling and analysis for autosar-based software development: a case study. In: Proceedings of the Conference on Design, Automation and Test in Europe, pp. 642–645. European Design and Automation Association (2010)
8. Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., et al.: Experimental security analysis of a modern automobile. In: 2010 IEEE Symposium on Security and Privacy (SP), pp. 447–462. IEEE (2010)
9. Lin, C.W., Sangiovanni-Vincentelli, A.: Cyber-security for the controller area network (CAN) communication protocol. In: 2012 International Conference on Cyber Security (CyberSecurity), pp. 1–7. IEEE (2012)
10. Menezes, A., Hankerson, D., Vanstone, S.A.: Guide to Elliptic Curve Cryptography. Springer, Berlin (2004)
11. Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. Black Hat USA **2015**, 91 (2015)
12. Munir, A., Koushanfar, F.: Design and analysis of secure and dependable automotive CPS: a steer-by-wire case study. IEEE Trans. Dependable Secur. Comput. (TDSC) (2018). <https://doi.org/10.1109/TDSC.2018.2846741>
13. NVIDIA: NVIDIA Self-Driving Cars. <https://www.nvidia.com/en-us/self-driving-cars/>. Accessed 5 Sep 2019
14. Poudel, B., Munir, A.: Design and evaluation of a reconfigurable ECU architecture for secure and dependable automotive CPS. IEEE Trans. Dependable Secur. Comput. (TDSC) (2018). <https://doi.org/10.1109/TDSC.2018.2883057>
15. Vector: ECU Development and Test with CANoe. <https://www.vector.com/us/en-us/products/products-a-z/software/canoe/>. Accessed 3 June 2019
16. Wilwert, C., Navet, N., Song, Y.Q., Simonot-Lion, F.: Design of Automotive X-by-Wire Systems. The Industrial Communication Technology Handbook. CRC Press, Boca Raton (2005)
17. Wolf, M., Gendrullis, T.: Design, implementation, and evaluation of a vehicular hardware security module. In: International Conference on Information Security and Cryptology, pp. 302–318. Springer (2011)



# How Many Clusters? An Entropic Approach to Hierarchical Cluster Analysis

Sergei Koltcov, Vera Ignatenko<sup>(✉)</sup>, and Sergei Pashakhin

National Research University Higher School of Economics, 55/2 Sedova Street,  
St. Petersburg 192148, Russia  
{skoltsov,vignatenko,spashahin}@hse.ru

**Abstract.** Clustering large and heterogeneous data of user-profiles from social media is problematic as the problem of finding the optimal number of clusters becomes more critical than for clustering smaller and homogeneous data. We propose a new approach based on the deformed Rényi entropy for determining the optimal number of clusters in hierarchical clustering of user-profile data. Our results show that this approach allows us to estimate Rényi entropy for each level of a hierarchical model and find the entropy minimum (information maximum). Our approach also shows that solutions with the lowest and the highest number of clusters correspond to the entropy maxima (minima of information).

**Keywords:** Hierarchical clustering · Rényi entropy · Number of clusters · User profiles · Online social networks

## 1 Introduction

The importance of information as a resource in modern society is growing significantly due to the high speed of dissemination and importance for decision-making. At the same time, online social networks (OSN) increasingly become more critical infrastructure in the process of disseminating information. On the one hand, networks represent the environment for the distribution of information; on the other hand, networks themselves generate information capable of affecting significantly economic and political preferences of people. The political turmoils of recent years in various countries (the Arab Spring, the Occupy Wall Street movement, the Ukrainian crisis), the apparent imbalance in news coverage on various online platforms (i.e. the US presidential elections), generation of numerous fake informational events and their explosive distribution through social networks demonstrate the need for a clear understanding of the information transmission and transformation processes.

In the study of news dissemination through OSN, networks should be considered as complex social systems (complex systems), requiring the use of various methodologies. There are many models of news spread which account for network topology [7], the role of ‘influential users’ [19] and the topical component of

the distributed messages [2]. However, one of the critical factors in news spread through OSN is a set of social attributes of users, such as gender, age, political preferences or religious affiliation [14]. Thus, when analyzing the distribution of information through OSN, it is necessary to solve the problem of estimating the influence of users' social attributes on the depth and speed of dissemination. This problem can be solved either by constructing regression models [14,31], or by including user features in a unified probabilistic framework [4]. However, despite the importance of adding user attributes to a model for transmitting information over OSN [12], the inclusion of a large set of features in probabilistic models is a big problem due to their extreme heterogeneity.

Another solution is to cluster users on their features and reduce them to one variable of 'user similarity'. Accordingly, 'user similarity' can replace the many user features in probabilistic models of information dissemination. However, clustering of OSN users by their socio-demographic characteristics with classic models such as K-means, C-means or hierarchical model, despite the developed techniques [23,26], causes problems, as it is necessary to determine the right number of clusters. Moreover, our experience shows that such techniques as the gap statistic [29], the jump method [27] or the elbow method [18] are unable to find the optimal number of clusters on large user data from OSN. These methods are developed on relatively small datasets and involve expensive in terms of time and memory computations, which is a critical issue with large data. Moreover, these approaches still require human judgment as to where is the optimal number given a set of measures. Therefore, it is necessary to develop other techniques for determining the optimal number of clusters.

In the framework of this work in progress, we consider the direction of 'network thermodynamics' [8], which allows one to organize data clustering, or rather, determine the number of clusters, based on the thermodynamic formalism [25,32]. In this paper, an entropy approach is proposed for determining the optimal number of clusters for profile data of OSN users with the classical hierarchical clustering method. In other words, rather than developing a new algorithm of hierarchical clustering, we use classic algorithms and aim at determining the optimal number of clusters (i.e., the optimal cut off) of a hierarchical solution.

The distinctiveness of the hierarchical method is in the construction of a hierarchical structure (dendrogram) of folded clusters. Here, at the highest level of a hierarchy, all nodes are assigned to one cluster, and at the lowest level, each element is a separate cluster. Hence, one can determine the entropy of two borderline situations and organize a search for the number of clusters inside these boundaries.

## 2 Background

The study of complex systems with methods of statistical physics is a leading stream in the network analysis research. Here one can distinguish several areas, each with specific goals and tasks. One is the area of network modeling such as Erdős-Rényi, Bollobás-Riordan, Watts-Strogatz models and other [6,11,22]. However, the other two areas are more relevant to our problem.

The second area studies clustering models of network structures, where researchers develop metrics for graph partitioning [13]. For instance, when dealing with large in terms of nodes and edges networks, researchers describe a network with methods of statistical physics such as annealing models for modularity optimization [8, 15] or with thermodynamic formalism [9, 32]. Additionally, the concept of entropy, as in classic Gibbs-Shannon or Rényi-Tsallis definition based on deformed logarithm, could be found in the literature of network analysis [24, 28]. This area could be referenced to as the ‘network thermodynamics’ [8].

Another area is closely related to the two already mentioned and involves models of hierarchical cluster analysis. Such clustering procedures attempt to restore the structure as a dendrogram, or one may say that such procedure is the sequential merging of smaller clusters into increasingly larger. One feature of the hierarchical approach to data clustering is the formation of parent-child relations, where parents are merged child clusters. In such a structure, the top level has all nodes in one cluster, and the bottom level has each node in a separate cluster.

When hierarchical clustering is applied to small data, where dendrogram is no larger than ten levels, the analysis is not so problematic. However, when data consists of several thousand or more units, the problem of choosing a dendrogram cut (the number of clusters) becomes complicated. For hierarchical clustering, the standard approach is to manually examine the dendrogram and try to put a cut-off line so that the distance distributions below the line are more heterogeneous than the distributions above the line. However, this approach is often ambiguous as it relies on human judgment. A solution to this problem could be found with the thermodynamic formalism from non-extensive statistical physics.

We ground our approach in the following works. The first [25] is proposing to search for the free energy minimum in data clustering. However, this criterion is developed only for the K-means type of algorithms. The second work [28] shows that the Tsallis entropy obtained with  $q$ -deformed Stirling formula may be used to describe hierarchical statistical systems where each level has its value of the Tsallis entropy. Such a description allows for exploring the hierarchical structure using the Tsallis entropy. As for hierarchical cluster analysis, Gibbs-Shannon entropy was used for evaluating solutions in [1, 10].

Thirdly, we build on the work of Olemskoi, who proposed using the concept of internal energy to describe a hierarchical system, which allows us to determine the free energy of the entire hierarchical system, as well as at each level [24]. However, unlike Olemskoi, who considers the transition from level to level in a hierarchical tree in terms of a diffusion process on branching trees, we propose to view the transition process as a process of hierarchical clustering which is characterized by the measure of the Rényi entropy. The Tsallis entropy could be obtained from the Rényi entropy with simple transformations [3].

A similar use of the deformed Rényi entropy is considered in [20, 21] for clustering of large document collections. The tests showed that the minimum of the Rényi entropy corresponds to the human choice of the number of clusters. At the same time, the maximum of entropy corresponds the lowest and the largest

numbers of clusters (from one-two to hundreds and more). In such cases, the Rényi entropy becomes larger as the distribution of features becomes uniform. However, these approaches have not been adapted for hierarchical models.

### 3 An Entropic Approach

Based on the discussed works, we formulate an entropic approach for determining the optimal level (the number of clusters) in hierarchical clustering.

We start from the proposition by Beck that information is related to entropy in the following way:  $S = -I$  [5]. Thus, information maximum corresponds to entropy minimum. Next, we consider a set of objects (nodes) as a statistical system. At the starting point, such a system is characterized by entropy maximum (information minimum) because at the initial state each object belongs to a separate cluster. Next, we consider a number of clusters as a temperature of such system which is a function of a level in hierarchical clustering. Given that, the hierarchical clustering procedure transforms a system from the state of maximum entropy to the state of the entropy minimum by changing the number of clusters (temperature). Therefore, the optimal clustering for large and heterogeneous data would be at the state of the entropy minimum for a system.

In the framework of hierarchical clustering, one can find two borderline situations: (1) All objects belong to one cluster. Such clustering has minimal information value, and, correspondingly, such solution has large entropy. (2) Each object is a unique cluster where the probability that a particular object belongs to a cluster is constant. In this case, as it is a uniform distribution, entropy is also large.

A hierarchical clustering procedure constructs a hierarchical tree, where each level has a certain number of clusters. Each cluster may contain a different number of objects  $N_{ik}$ , where  $k$  is a cluster on level  $i$ . However, the total number of elements on each level always equals the total number of system elements  $N$ . We define the probability of elements in cluster  $k$  on level  $i$  as follows:

$$p_{ik} = \frac{N_{ik}}{N}.$$

If each cluster contains the same number of elements, we obtain a uniform distribution. Notice that we also obtain a uniform distribution on the lowest level, when each element is a cluster. Therefore, we introduce a threshold  $1/N$  and investigate obtained distributions with respect to this initial uniform distribution.

Correspondingly, one can describe each level  $i$  of a dendrogram with following variables: (1) The total number of clusters  $K_i$  on level  $i$ . (2) The total number of elements with probability over the threshold  $p_{ik} > \frac{1}{N}$  of level  $i$ , namely,  $M_i = \sum_k N_{ik} \cdot \mathbb{1}\left(\frac{N_{ik}}{N} - \frac{1}{N}\right)$ , where the step function  $\mathbb{1}(\cdot)$  is defined by  $\mathbb{1}(x-y) = 1$  if  $x \geq y$  and  $\mathbb{1}(x-y) = 0$  if  $x < y$ . (3) The sum of high probabilities  $\tilde{P}_i$ , i.e., probabilities larger than  $1/N$ , namely,  $\tilde{P}_i = \sum_k p_{ik} \cdot \mathbb{1}(p_{ik} - \frac{1}{N})$ .

We can measure all these variables in the process of data clustering. With these values, one can determine internal energy and Gibbs-Shannon entropy at a given level in the following way:

$$E_i = -\ln\left(\frac{\tilde{P}}{K_i}\right),$$

$$S_i = \ln\left(\frac{M_i}{N}\right).$$

With Gibbs-Shannon entropy and internal energy, one can define free energy and Rényi entropy for each level of a hierarchy. Free energy of a hierarchical level  $i$  is expressed as

$$F_i = E_i - K_i S_i.$$

And Rényi entropy of level  $i$  can be expressed as follows [5]:

$$S_i^R = \frac{F_i}{1 - q},$$

where  $q = \frac{1}{K_i}$  is a deformation parameter. Thus, our approach allows us to estimate the process of hierarchical clustering from a perspective of behaviour of Rényi entropy under transition between levels, i.e., to estimate the dependence of entropy on the number of clusters.

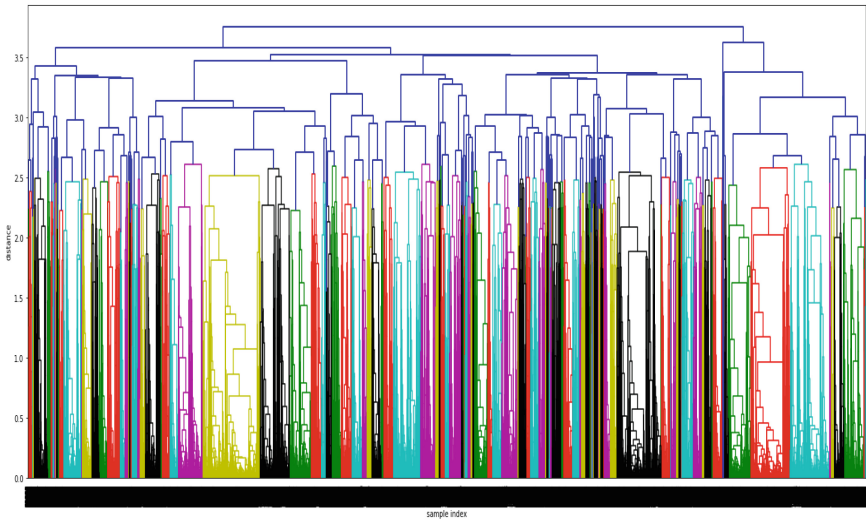
The process of clustering begins with minimum information (maximum Rényi entropy) and also ends with maximum Rényi entropy. Hence, minimum Rényi entropy (information maximum) is located somewhere in between these maxima. Particular data features will define the location of the global minimum and a set of local minima.

## 4 Experiment

We test our approach on data of user-profiles from the leading Russian OSN *Vkontakte* (VK). We collected the data through VK API [30]. Then, we anonymized user data, i.e., names, surnames and IDs were deleted to avoid the possibility of revealing real users. The dataset includes digital traces of user activity such as numbers of likes, posts, reposts, comments; indicators of subscribing to one or more pages from 12 national news channels publishing news on VK; as well as user stated political beliefs (one of eight). In total, the dataset has 47 user attributes of a total 50,000 users. Our attempts to cluster this dataset with K-means and C-means while searching for the optimal number of clusters with gap statistics, jump and silhouette methods were unsuccessful.

On a machine with 64 GB RAM and i7-6700 CPU @ 3.40 GHz (four cores), we were unable to run hierarchical clustering and to test our approach on more massive datasets since the algorithm has time complexity  $O(n^2)$  and uses  $O(n^2)$  memory, where  $n$  is the number of samples. However, the hierarchical clustering of our data on the mentioned machine takes about 8 h (28,798 s).

We test our approach in two stages. First, we conduct hierarchical clustering using `scipy.cluster.hierarchy` Python package [16] with the ‘complete’ method of calculating the distance between newly formed clusters [17], namely, the distance between clusters  $u$  and  $v$  is expressed as  $d(u, v) = \max_{i,j}(\text{dist}(u[i], v[j]))$ , where ‘dist’ refers to Euclidean distance,  $u[i]$  and  $v[j]$  are objects contained in cluster  $u$  and cluster  $v$ , correspondingly. In each iteration, we select and merge two or more clusters with the smallest distance. This stage produces a hierarchy of clusters which can be visualized in the form of the dendrogram (Fig. 1). One can see how manual analysis of such dendrogram could be problematic.



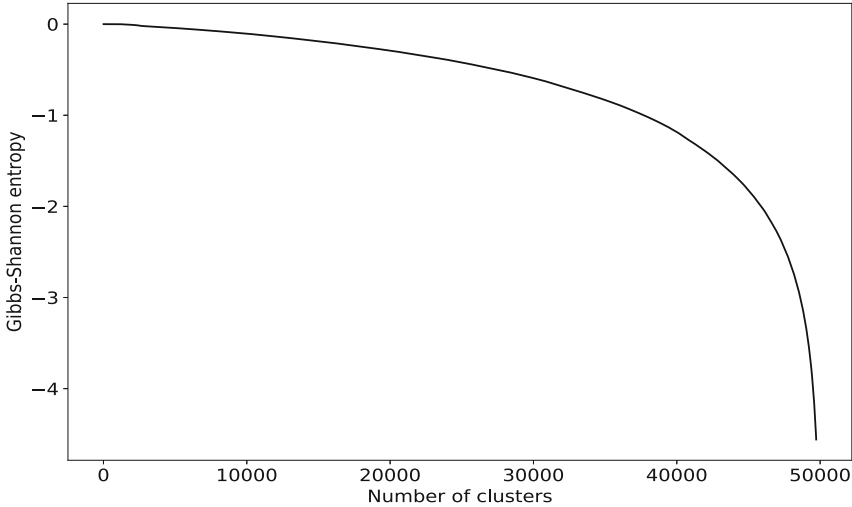
**Fig. 1.** The dendrogram of clustering 50,000 OSN user profiles.

Then, we calculate the number of obtained clusters on each level of the hierarchy and the number of users in each cluster. Here, all users belong to the same cluster on the upper level of the hierarchy, and each user belongs to a separate cluster on the bottom level of the hierarchy, i.e., the lowest level contains 50,000 clusters. Then, we compute Gibbs-Shannon entropy, internal energy, free energy and Rényi entropy.<sup>1</sup> Finally, we will consider our approach valid if (1) it will show a clear entropy minimum (a maximum of information) and (2) the entropy maxima (the minima of information) will correspond to the borderline states.

<sup>1</sup> An example of calculations in Python is available here: <https://github.com/hse-scila/entropic-approach-hierarchical-clustering>.

## 5 Results and Conclusion

Figure 2 and 3 show two mutually opposite processes present during hierarchical clustering of social media users. The first process is the decrease of Gibbs-Shannon entropy with a rising number of clusters, which means that the equilibrium state corresponds to the minimum of a given entropy. The equilibrium corresponds to the state when each user is a separate cluster.

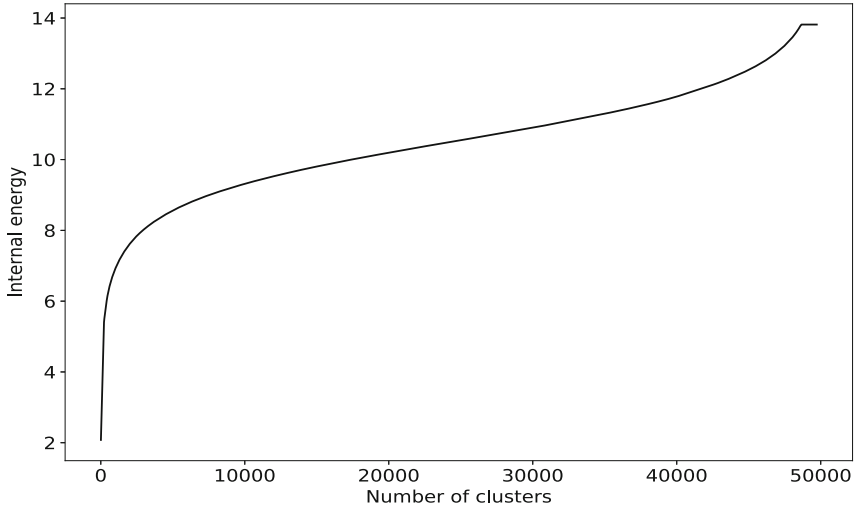


**Fig. 2.** Distribution of Gibbs-Shannon entropy over the number of clusters.

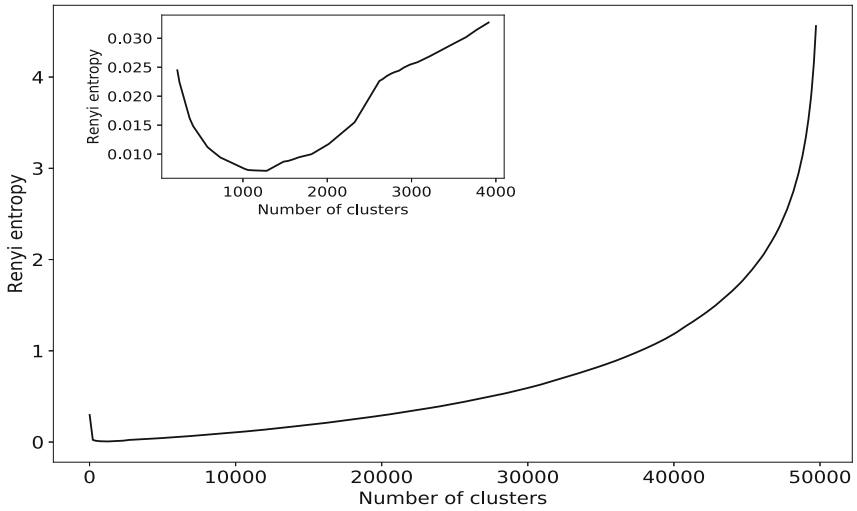
The second process is the increase of internal energy with a rising number of clusters (Fig. 3). The difference between these two processes has an area where they balance each other (Fig. 4). In this area, the Rényi entropy has its minimum value. Hence, the minimum of the Rényi entropy corresponds the maximum of information of a hierarchical model. For this dataset, the minimum of the Rényi entropy lies at the 1,281 clusters or 50,000 users could be grouped in 1,281 clusters (Fig. 4). In the machine learning terms, the left branch of the Rényi entropy indicates underfitting while the right branch to overfitting. Thus, the minimum of the Rényi entropy indicates the optimal parameters of hierarchical clustering.

In this work, we propose a criterion of finding the optimal number of clusters for hierarchical clustering, using entropic formalism with deformed Rényi entropy where the parameter of deformation is the number of clusters. This approach could be used for such algorithms as Infinite Mixture Models with Nonparametric Bayes and the Dirichlet Process with various implementations (Chinese restaurant process, stick-breaking algorithm).





**Fig. 3.** Distribution of internal energy over the number of clusters.



**Fig. 4.** Distribution of Rényi entropy over the number of clusters.

In further, we plan to test our approach with large synthetic data with a pre-defined number of clusters. One potential area of further testing is to consider if various combinations of user features affect the global Rényi minimum location. Another direction is to consider other than Euclidean distances to assess their fitness for hierarchical clustering of common types of data from OSN.

**Acknowledgments.** The reported study was funded by RFBR according to the research project No 18-011-00997 A.

## References

1. Aldana-Bobadilla, E., Kuri-Morales, A.: A clustering method based on the maximum entropy principle. *Entropy* **17**(1), 151–180 (2015)
2. AlSumait, L., Barbará, D., Domeniconi, C.: On-line LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. In: *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM 2008*, pp. 3–12, Washington, DC, USA. IEEE Computer Society (2008)
3. José, A., Balogh, S., Hernández, S.: A brief review of generalized entropies. *Entropy* **20**(11), 813 (2018)
4. Bao, Q., Cheung, W.K., Liu, J.: Inferring motif-based diffusion models for social networks. In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016*, pp. 3677–3683. AAAI Press (2016)
5. Beck, C.: Generalised information and entropy measures in physics. *Contemporary Phys.* **50**(4), 495–510 (2009)
6. Bollobás, B., Riordan, O.M.: Mathematical results on scale-free random graphs. In: Bornholdt, S., Schuster, H.G. (eds.) *Handbook of Graphs and Networks: From the Genome to the Internet*, 1st edn, pp. 1–34. Wiley, Weinheim (2003)
7. De Choudhury, M., Lin, Y.-R., Sundaram, H., Candan, S.K., Xie, L., Kelliher, A.: How does the data sampling strategy impact the discovery of information diffusion in social media? In: *ICWSM (2010)*
8. Dehmer, M., Emmert-Streib, F.: *Analysis of Complex Networks: From Biology to Linguistics*. Wiley, Hoboken (2009)
9. Dehmer, M., Emmert-Streib, F., Chen, Z., Li, X., Shi, Y. (eds.): *Mathematical Foundations and Applications of Graph Entropy*. Wiley, Weinheim (2016)
10. Elayat, H., Murphy, B., Prabhakar, N.: Entropy in the hierarchical cluster analysis of hospitals. *Health Serv. Res.* **13**(4), 395–403 (1978)
11. Erdős, P., Rényi, A.: On the evolution of random graphs. In: *The Structure and Dynamics of Networks*, pp. 38–82. Princeton University Press, Princeton (2011)
12. Fogués, R.L., Such, J.M., Minguet, A.E., García-Fornes, A.: Open challenges in relationship-based privacy mechanisms for social network services. *Int. J. Hum. Comput. Interaction* **31**, 350–370 (2015)
13. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**(3–5), 75–174 (2010)
14. Guille, A., Hacid, H.: A predictive model for the temporal dynamics of information diffusion in online social networks. In: *Proceedings of the 21st International Conference on World Wide Web, WWW 2012 Companion*, pp. 1145–1152. ACM, New York (2012)
15. Guimerà, R., Nunes Amaral, L.A.: Functional cartography of complex metabolic networks. *Nature* **433**(7028), 895–900 (2005)
16. Hierarchical clustering (scipy.cluster.hierarchy)—SciPy v1.3.1 Reference Guide
17. Hierarchical clustering (scipy.cluster.hierarchy.linkage)—SciPy v1.3.1 Reference Guide
18. Ketchen, D., Shook, C.: The application of cluster analysis in strategic management research: an analysis and critique. *Strategic Manage. J.* **17**, 441–458 (1996)
19. Kitsak, M., Gallos, L., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H., Makse, H.: Identification of influential spreaders in complex networks. *Nat. Phys.* **6**(11), 888–893 (2010)

20. Koltcov, S.: Application of rényi and tsallis entropies to topic modeling optimization. *Phys. A: Stat. Mech. Appl.* **512**, 1192–1204 (2018)
21. Koltcov, S., Ignatenko, V., Koltsova, O.: Estimating topic modeling performance with sharma-mittal entropy. *Entropy* **21**(7), 660 (2019)
22. Newman, M.E.J.: Models of the small world. *J. Stat. Phys.* **101**(3), 819–841 (2000)
23. O'Donovan, F.T., Fournelle, C., Gaffigan, S., Brdiczka, O., Shen, J., Liu, J., Moore, K.E.: Characterizing user behavior and information propagation on a social multimedia network. In: 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)
24. Olemskoi, A.: Synergetics of Complex Systems: Phenomenology and Statistical Theory [Sinergetika slozhnyh sistem. Fenome-nologiya i statisticheskaya teoriya]. KRASAND, Moscow (2009)
25. Rose, K., Gurewitz, E., Fox, G.C.: Statistical mechanics and phase transitions in clustering. *Phys. Rev. Lett.* **65**(8), 945–948 (1990)
26. Rytsarev, I.A., Kupriyanov, A.V., Kirsh, D.V., Liseckiy, K.S.: Clustering of social media content with the use of BigData technology. *J. Phys. Conf. Ser.* **1096**, 012085 (2018)
27. Sugar, C.A., James, G.M.: Finding the number of clusters in a dataset: an information-theoretic approach. *J. Am. Stat. Assoc.* **98**(463), 750–763 (2003)
28. Suyari, H., Wada, T.: Scaling property and the generalized entropy uniquely determined by a fundamental nonlinear differential equation. arXiv (2006)
29. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. *J. Royal Stat. Soc. Ser. B (Statistical Methodology)* **63**(2), 411–423 (2001)
30. VK API guide. <https://vk.com/dev/manuals>
31. Wang, Y., Zhang, Z.-M., Peng, Z.-H., Duan, Y.-Y., Gao, Z.-Q.: A cascading diffusion prediction model in micro-blog based on multi-dimensional features. In: Barolli, L., Zhang, M., Wang, X.-A. (eds.) *Advances in Internetworking, Data & Web Technologies*, pp. 734–746, Springer, Cham (2018)
32. Zhang, Q., Li, M., Deng, Y.: A new structure entropy of complex networks based on nonextensive statistical mechanics. *Int. J. Modern Phys. C* **27**(10), 1650118 (2016)



# Analysis of Structural Liveness and Boundedness in Weighted Free-Choice Net Based on Circuit Flow Values

Yojiro Harie<sup>1</sup>(✉) and Katsumi Wasaki<sup>2</sup>

<sup>1</sup> Interdisciplinary Graduate School of Science and Technology,  
Shinshu University, 4-17-1 Wakasato, Nagano 380-8553, Japan  
16st207c@shinshu-u.ac.jp

<sup>2</sup> Division of Electrical and Computer Engineering, Faculty of Engineering,  
Shinshu University, 4-17-1 Wakasato, Nagano 380-8553, Japan

**Abstract.** A Petri net is a mathematical method that can be used to represent and analyze discrete event systems. Although research on structural liveness and safety in ordinary free-choice (FC) nets has been reported, analysis methods for weighted Petri nets have not yet been developed. In this study, we propose a method for determining the structural liveness and safety of strongly connected FC nets. The flow rate of tokens for strongly connected marked graphs is defined as the *circuit flow value*. In addition, the circuit flow value of a strongly connected FC net is obtained by calculating the superposition of the circuit flow value.

**Keywords:** Petri nets · Structural liveness and boundedness · Circuit flow value

## 1 Introduction

A Petri net is a mathematical method that can be used to represent and analyze discrete event systems [6–8]. The properties of Petri nets can be classified into two categories as follows: dynamic properties, which characterize changes in the dynamic behavior of systems, and structural properties, which depend on the structure of the Petri nets. Structural properties include liveness, boundedness, and other properties that can be determined by the calculating an incidence matrix of Petri nets. These conditions can be further examined using linear algebra techniques [1, 5, 6]. Structural analysis cannot be used to analyze the distribution of tokens or the increase or decrease in the number of tokens because the state dynamically changes when tokens are moved between various places by firing. However, examining structural liveness of weighted Petri nets can be performed by behavioral verification. Behavioral verification such as generating state spaces is expensive, however, because it generates the entire state space from the initial marking. Therefore, a flow net has been proposed to view both

the structural properties as a graph and the behavior of the tokens [4]. Flow nets are graphs that are converted into weighted directed graphs, with the exception of Petri net transitions.

Workflow modeling and analysis methods for managing tasks and data flow have been proposed as applications of Petri nets [11]. Soundness is a criterion of logical correctness in workflow Petri nets, and marked graph (MG) workflow nets that do not include directed circuits are mathematically guaranteed to be sound [2]. A short-circuit net has been proposed that constitutes a circuit by connecting the start point and end point in a workflow net with a single transition [10]. The soundness decision problem in short-circuit nets can result in liveness and boundedness decision problems [10]. Although workflow nets allow weighted arcs, several conventional studies have targeted nets with input/output conditions or ordinary Petri nets due to the complexity of the analysis [11].

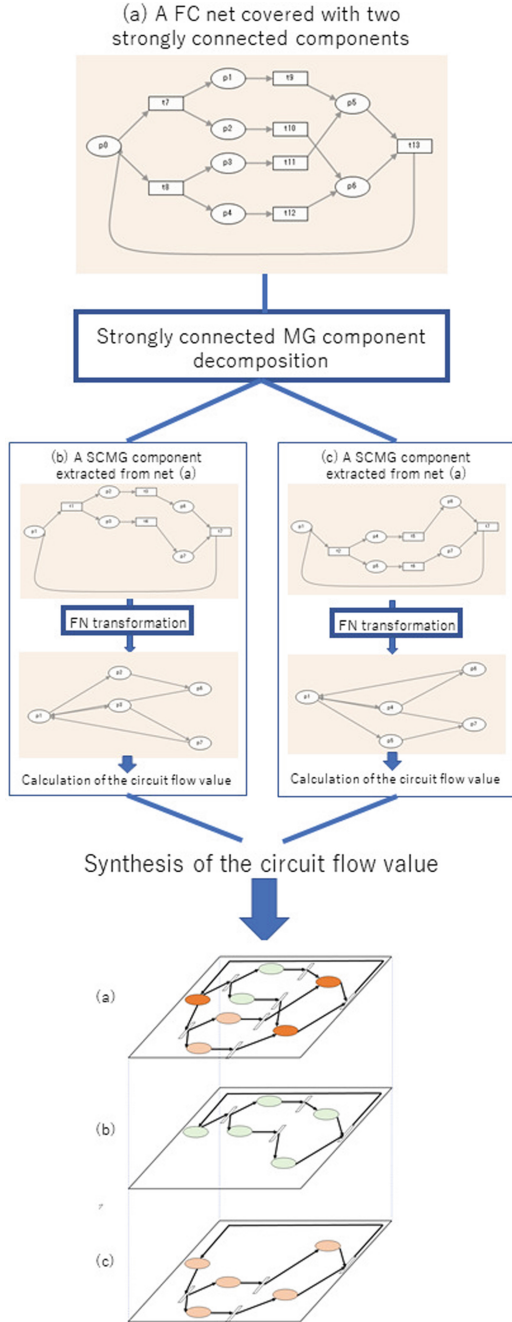
In this study, we propose a weighted net liveness/safety decision method that involves a token flow calculation algorithm using flow net (FN) transformation. The number of input/output locations, which is the configuration requirement of the workflow net, is not one. The token flow of free-choice (FC) net circuits is obtained by calculating the token flow from the start point to the end point of MG circuits and superimposing the token flow for each structure of the MG circuits. This analysis method can be applied to both workflow nets and short-circuit nets.

Figure 1 presents an outline of the proposed method. The net in the Fig. 1 represents a well-formed flow FC net that is covered by two strongly connected MG components, (a) and (b). The net in Fig. 1(a) can be decomposed into two components of the MG circuit to obtain strongly connected MG components in Fig. 1(b) and (c). By applying a flow net transformation to each of the strongly connected MG components in Fig. 1(b) and (c), the circuit flow net value can be calculated by the calculation algorithm. The entire net does not satisfy liveness/boundedness if each circuit does not satisfy liveness/boundedness. The circuit superimposition method does not involve compositing the union as a subgraph structure, but rather, synthesizing the flow values of each circuit. The flow value of the entire net is calculated by superimposing the circuit flow values of the subgraph layer.

This paper is organized as follows. Section 2 introduces a Petri net, while Sect. 3 introduces a flow net. Section 4 presents the calculation of the circuit flow value in an MG circuit, while Sect. 5 presents an analysis of liveness and boundedness by superimposing circuit flow values. Finally, Sect. 6 presents the conclusions of this study.

## 2 Place/Transition (P/T) Net

The original definitions of Petri nets can be found in previous studies [6–8]. A Petri net is denoted  $N = (P, T, F, W, M_0)$ , where  $P$  is a finite set of places,  $T$  is a finite set of transitions with  $P \cap T = \phi$ ,  $F \subseteq (P \times T) \cup (T \times P)$  is a finite set of arcs,  $W$  is the function  $W : F \rightarrow \mathbb{N}$  specifying the arc weights, and  $M_0$  is the initial marking (i.e., a mapping  $M : P \rightarrow \mathbb{N}$ , indicating the number of tokens in each place).



**Fig. 1.** Extraction of strongly connected marked graph (MG) components and superimposition circuit flow values.

Let  $a \in P \cup T$ ,  $\bullet a = \{b \in P \cup T \mid (b, a) \in F\}$  be the pre-set and  $a \bullet = \{b \in P \cup T \mid (a, b) \in F\}$  be the post-set. Let  $t \in T$ ,  $t$  be the source transition if  $\bullet t = \phi$ , and  $t$  be the sink transition if  $t \bullet = \phi$ . A transition  $t \in T$  is said to be fireable when it satisfies  $\forall p \in \bullet t : M(p) \geq W(p, t)$ . Here,  $t$  is fireable from marking  $M$  to  $M'$ , denoted  $M[t > M']$ . A nite sequence  $\sigma = t_1 \dots t_n$  is fireable at a marking  $M$  if there are markings  $M_1, M_2, \dots, M_n$  such that  $M[t_1 > M_1, M_1[t_2 > M_2, \dots, M_{n-1}[t_n > M_n$ . Here,  $\sigma$  is called a firing sequence.  $M_n$  is reachable from  $M$  when there is a firing sequence that has a marking  $M$  to a marking  $M_n$ . We represent the set of all firing sequences from  $M_0$  as  $L(N, M_0)$  or  $L(M_0)$ .  $A = \{a_{ij}\}$  is an  $m \times n$  matrix, and  $a_{ij} = a_{ij}^+ - a_{ij}^-$ , where  $a_{ij}^+ = W(t_i, p_j)$ ,  $a_{ij}^- = W(p_j, t_i)$ .

Let  $N = (P, T, F, W)$  be a net. A transition  $t \in T$  of  $N$  is live if for every marking  $M$  that is reachable from  $M_0$ , there exists a marking  $M'$  that be enabled to fire  $t$  and is reachable from  $M$ . Here,  $M_0$  is called the live initial marking in  $N$ . We use structural liveness to indicate that there exists a live initial marking in  $N$ . For some  $k \in \mathbb{N}$ ,  $\forall M \in L(M_0), p \in P : M(p) \leq k$ ,  $N$  is  $k$ -bounded or bounded. A net structure  $N$  is considered structurally bounded if it is bounded for any finite initial marking  $M_0$ . A Petri net is called *conservative* if  $Ay = 0$  for some  $y > 0$ , and is called *consistent* if  $A^T x = 0$  for some  $x > 0$ .

A classification based on conditions regarding the composition of Petri nets is called a subclass. A Petri net  $N = (P, T, F, W)$  is an MG if for all  $p \in P$ ,  $|\bullet p| = |p \bullet| = 1$ . A Petri net  $N = (P, T, F, W)$  is a state machine (SM) if for all  $t \in T$ ,  $|\bullet t| = |t \bullet| = 1$ . A Petri net  $N = (P, T, F, W)$  is a Free choice net (FC) if for all  $p \in P$ ,  $|p \bullet| = 1$  or  $\bullet\{p \bullet\} = \{p\}$ .

$t, t'$  are said to be in structural conflict, where  $\bullet t \wedge \bullet t' \neq \phi$ . A coupled conflict relation is defined as the transitive closure of the structural conflict relation. The equivalence class of transition  $t$  is denoted  $CSS(t)$ , and the quotient set is  $SCCS$ . The following theorem pertains to the structural properties [3, 8].

The target net in this study is a weighted FC net covered with strongly connected MG components. We define a well formed flow Petri net (WFFP) to allow flow superposition.

**Definition 1.** Let  $N = (P, T, F, W)$  be a net, and  $X \subseteq P \cup T$  be any strongly connected MG component in  $N$ . If an arbitrary path  $\alpha = \{x, y_1, y_2, \dots, x'\}$  connected to  $X$  is a TP- or PP- handle of  $X$ ,  $N$  is called a WFFP.

Let  $A$  be the incidence matrix of  $N$ . If  $N$  is structurally bounded and live, then  $N$  is consistent and conservative, and  $rank(A) = |SCCS| - 1$ . In an MG, there is no structural conflict; thus  $n = |T| = |SCCS|$  always holds. With respect to Petri nets, it is known that the rank of the incidence matrix of connected weighted MG nets is  $n - 1$  or  $n$  [9]. It is also known that the rank of a neutral weighted MG is  $n - 1$ . Calculating the circuit flow value to apply our proposed algorithm is equivalent to analyzing the net as neutral.

### 3 Flow Net Definition and Flow Capacity

#### 3.1 Flow Net

The structural properties of Petri nets are analyzed by solving algebraic equations and inequalities for incidence matrices. Transition firing generates and consumes tokens, and the total number of tokens in the (sub)net changes. We refer to the ratio of token increase and decrease due to transition firing as flow, and propose a flow net that defines the connection weight between nodes by flow [4]. The definition of a flow net  $FT$  is as follows.

**Definition 2.** A flow net is a 3-tuple,  $FT = (V, E, W_f)$ , where  $V = \{v_1, v_2, \dots, v_n\}$  is a finite set of nodes,  $n = |V| = |T|$ ,  $E \subseteq V \times V$  is a finite set of edges, and  $W_f$  is the function  $W_f : E \rightarrow \mathbb{Q}$ .  $F_T = \{f_{r_{ij}}\}$  is a rational matrix of  $n \times n$ , where each component is given by

$$f_{r_{ij}} = \frac{W(t_i, p)}{W(p, t_j)} (t_i \in \bullet p, t_j \in p \bullet)$$

Here, for arbitrary nodes  $v_i, v_j \in V$ ,  $W_f(v_i, v_j) = f_{r_{ij}}$ .

If the FT transformation is applied to Fig. 2, the directed graph  $FT$  presented in Fig. 3 is obtained as well as the following adjacency matrix  $F_p$ .

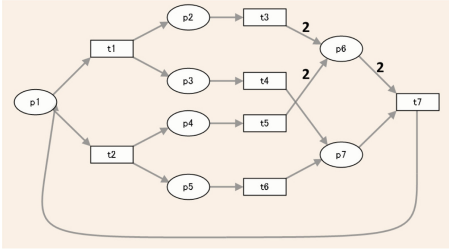


Fig. 2. Example of non-ordinary P/T net

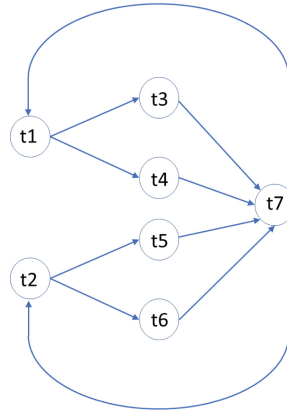


Fig. 3. Example of flow net converted from Fig. 2

$$F_T = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



When the PN structure  $N$  satisfies conditions (i) and (ii), defined below,  $N$  is said to be convertible to graph  $FT$ .

**Definition 3.** Let  $N = (P, T, F, W)$  be a PN structure.  $N$  is convertible to graph  $FT$ , which signifies that  $N$  satisfies conditions (i) and (ii) as follows:

- (i) For any  $p \in P$ ,  $\{\bullet p\} \neq \emptyset$  and  $\{p\bullet\} \neq \emptyset$
- (ii) For different  $p_1, p_2 \in P$   $\{\bullet p_1\} \cap \{\bullet p_2\} \neq \emptyset \rightarrow \{p_1\bullet\} \cap \{p_2\bullet\} = \emptyset$

We define a bijective function  $h : T \rightarrow V; h(t_i) = v_i$  ( $t_i \in T$ ) that associates the index numbers of transitions with the index numbers of nodes of the flow net. In this paper, flow net nodes are associated with transitions even if  $v_i = h(t_i)$  and  $V_S = h(S) = \{v_i \mid v_i = h(t_i), \forall t_i \in S, S \subseteq T\}$  are not written.

### 3.2 Flow Capacity

In a live circuit Petri net, it is important how the token of starting point  $p$  moves as input in the loop, after which it is output to  $p$ . Here, we consider the capacity of an MG circuit in terms of the increase and decrease of tokens in the circuit.

**Definition 4.** Let  $N = (P, T, F, W)$  be a strongly connected MG net, and  $FT = (V, E, W_f)$  be a flow net obtained by converting  $N$ , where  $t_{join} \in T$  is a merging transition. For any natural number  $k = 1, 2, \dots$ , we denote the following function sequence  $f_k$  and node set  $S_k$  as the flow product for  $k$  in  $FT$  and the flow calculation set for  $k$ , respectively.

$$S_1 = v_{join}\bullet$$

$$f_1(v) = \begin{cases} 1 & (v \in S_1) \\ 0 & (\text{otherwise}) \end{cases}$$

$$\begin{aligned} S_k &= S_k' \cup S_k'', \\ S_k' &= \{v' \mid v \in S_{k-1}, v' \in v\bullet, \bullet v' \subseteq S_{k-1}\}, \\ S_k'' &= \{v \mid v \in S_{k-1}, \bullet\{v\bullet\} \not\subseteq S_{k-1}\} \end{aligned}$$

$$f_k(v) = \begin{cases} cal(v) & (v \in S_k') \\ f_{k-1}(v) & (\text{otherwise}) \end{cases}$$

$$cal(v) = \begin{cases} c & (\forall v' \in \bullet v, W_f(v', v)f_{k-1}(v') = c) \\ \omega & (\text{otherwise}) \end{cases}$$

When there exists an upper bound  $\varphi \in \mathbb{R}$ ,  $\varphi$  is referred to as the flow capacity in flow graph  $FT$ . If  $k_1$  is the smallest  $k$  where  $S_k = S_1$ , then the value of  $f_{k_1}(v)$  ( $v \in S_{k_1}$ ) is called the circuit flow value.

Here, the calculation rule of  $\omega \notin \mathbb{Q}$  is defined as  $\forall a \in \mathbb{Q}, \omega \times a = \omega, a \times \omega = \omega, \omega \times \omega = \omega$ . The circuit flow value  $f_{k_i}$  contains information regarding whether the token output from the start node set is input to the start node via the a closed loop.

## 4 Circuit-Flow

### 4.1 Primitive Circuit Net

In simple flow nets, edges between nodes are considered to be the ratio of the number of tokens moved between the places of Petri nets. As an example, Fig. 4 consists of  $P = \{p_1, p_2, \dots, p_n\}$  and  $T = \{t_1, t_2, \dots, t_{n-1}\}$ . Letting  $s_1, s_2, \dots, s_{n-1} \in \mathbb{N}$ , there are  $T_1 = s_1w(p_1, t_1)s_2w(p_2, t_2) \dots s_{n-1}w(p_{n-1}, t_{n-1})$  tokens in place  $p_1$  as the initial marking. When transition  $t_1$  fires  $s_1s_2w(p_2, t_2) \dots s_{n-1}w(p_{n-1}, t_{n-1})$ ,  $s_1w(t_1, p_2)s_2w(p_2, t_2) \dots s_{n-1}w(p_{n-1}, t_{n-1})$  tokens move to place  $p_2$ . Similarly, transitions  $t_2 \dots t_{n-1}$  are fired sequentially to cause the movement of tokens. Finally, there are  $T_n = s_1w(t_1, p_2) \dots s_{n-1}w(t_{n-1}, p_n)$  tokens in place  $p_n$ . Comparing  $T_1$  and  $T_n$ , the following can be written:

$$T_n = \frac{w(t_1, p_2)}{w(p_1, t_1)} \dots \frac{w(t_{n-1}, p_n)}{w(p_{n-1}, t_{n-1})} \times T_1$$

If  $p_1 = p_n$ , then  $N$  is a simple circuit net. If

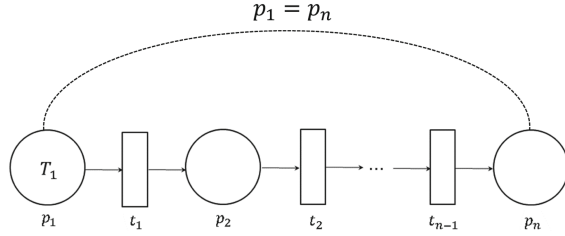
$$\frac{w(t_1, p_2)}{w(p_1, t_1)} \dots \frac{w(t_{n-1}, p_n)}{w(p_{n-1}, t_{n-1})} = 1 \tag{1}$$

is satisfied,  $N$  is consistent because the total number of tokens does not change.  $T_1$  takes the minimum marking when  $s_1 = s_2 = \dots s_{n-1} = 1$ . The component of (5) corresponds to the edge weight  $W_f(v_i, v_{i+1})$  from node  $v_i$  to  $v_{i+1}$  in the flow net. If  $t_{n-1}$  is  $t_{join}$ , then  $S_0 = p_1$ , and the circuit flow value in  $FT$  obtained by converting  $N$  satisfies 1.

**Theorem 1.** *Let  $N = (P, T, F, W)$  be an MG simple circuit, and  $FT$  be a flow net obtained by converting  $N$ . Here,  $|P| = |T| = m$ . If  $v_1 = v_{m+1}$ , then the following are equivalent:*

- (i)  $\prod_{v_i, v_{i+1} \in V} w_f(v_i, v_{i+1}) = 1$  ( $i = 1, 2, \dots, m$ ).
- (ii) The circuit flow value of  $FT$  is 1.
- (iii)  $N$  is conservative and repetitive.

*Proof.* It is clear that (i) and (ii) are equivalent, because  $N$  is a simple circuit. Statements (iii) and (ii) can be proven by solving the determinant of the incidence matrix because  $N$  is conservative. To prove (i)–(iii), if  $\prod_{v_i, v_{i+1} \in V} w_f(v_i, v_{i+1}) = 1$  ( $i = 1, 2, \dots, m$ ) is satisfied, then  $N$  is consistent. This has already been demonstrated. Here, we can demonstrate that the simple circuit  $N$  is conservative. Let  $A = A^+ - A^-$  be an incidence matrix of  $N$ . From



**Fig. 4.** P/T net,  $|P| = n, |T| = n - 1$

the definition of conservative, we can demonstrate that there exists  $y > 0$  such that  $Ay = 0$ . If  $f_k = W_f(v_k, v_{k+1}) = \frac{w(t_k \cdot p_{k+1})}{w(p_k, t_k)}$  is satisfied, then

$$\begin{aligned}
 y &= (y_1, y_2, \dots, y_{m-1}, y_m) \\
 &= (f_1 f_2 \dots f_{n-1} c, f_2 f_3 \dots f_{n-1} c, \dots, f_{n-1} c, c)
 \end{aligned}
 \tag{2}$$

holds for all  $c \in \mathbb{N}$ . If  $c = w(p_1, t_1)w(p_2, t_2) \dots w(p_{n-1}, t_{n-1})$  is satisfied, then  $y$  consists of all integer solutions. Equation (2) can be proven inductively; however, it is omitted due to space limitations.

### 4.2 MG Circuits Excluding a Simple Circuit

Before introducing a general MG circuit, we discuss the relationship between a conditional net and flow product. First, we consider a net in which the number of merging transitions included in the MG circuit path is at most one. Let such a net be  $N_1 = (P_1, T_1, F_1, W_1)$ . The number of degrees of freedom for the incidence matrix  $A_1$  of  $N_1$  matches the number of input places for the merging transition  $t_{last} \in T_1$ . Let  $t \in T_1$ , where  $t$  is not a merging transition. There is only one input place for  $t$  because the subclass of  $N_1$  is an MG and does not overlap with the input places for other transitions. Therefore,  $|P_1| = |T_1| + |\bullet t_{last}| - 1$  holds. If  $|P_1| = m$ , then the number of degrees of freedom of  $A_1$  is  $m - rank(A) = m - (n - 1) = m - ((|P_1| - |\bullet t_{last}| + 1) - 1) = |\bullet t_{last}|$ . This fact can be used to demonstrate the following.

**Theorem 2.** *Let  $N$  be an MG circuit net  $N$ , where it is assumed that there is only one merging transition. The circuit flow value of the converted flow net  $N$  is 1 if and only if  $N$  is structurally live and bounded.*

**Theorem 3.** *Let  $N$  be an MG circuit net, and  $FT = (V, E, W_f)$  be a flow net converted from  $N$ , where the number of merging transitions is greater than one, and there is one merging transition that appears in all circuits in  $N$ . Assuming that the circuit flow is  $f = f_k(v)$  ( $v \in S_k$ ) of  $FT$ , the following conditions are required for  $N$  to be conservative and consistent.*

- (i) *The following equation holds for any  $v_x, v_y \in \bullet v: j < k, v \in S_j', |\bullet v| > 1$ . Then, the following equation holds:  $f_{j-1}(v_x)w_f(v_x, v) = f_{j-1}(v_y)w_f(v_y, v)$*

(ii)  $f_k(v) = 1, v \in S_k$

When multiple merging transitions appear in all circuits in  $N$ , the total product of the flow products for each subgraph can be expressed as follows.

**Theorem 4.** *Let  $N = (P, T, F, W)$  be an MG circuit net, and  $FT = (V, E, W_f)$  be a net obtained by flow transformation of  $N$ , where  $T_{join} \subseteq T$  is the set of merging transitions that appear in all circuits, consisting of  $\{t_{join_1}, \dots, t_{join_i}, \text{ and } \dots, t_{join_l}\}$  ( $i = 1, 2, \dots, l$ ). Given  $t_{join_i} \bullet = P_{join_i}$ ,  $N$  can be expressed as a union of the following subnets:  $P = \bigcup_{i=1}^l P_i$ ,  $T = \bigcup_{i=1}^l T_i$ ,  $\bullet T_i = \{p \mid p \in \bullet t, \text{ and } t \in T_i\} \subset P_i$ . Here, for  $j$  and  $k$ , if  $k = j + 1$ , then  $P_k \cap P_j = P_{join_j}$ , where  $P_{l+1} = P_1$ . Otherwise,  $P_j \cap P_k = \phi$ . If  $f$  is a circuit flow value of  $FT$ , then  $f = \prod f_{i k_i}(v)$  holds. If  $N_i = (P_i, T_i, F_i, W_i)$  is a partial net of  $N$  consisting of arc set  $F_i$  and weighted function  $W_i$  for  $P_i$  and  $T_i$ , the function sequence  $f_i$  is a flow product of flow net  $FP_i$ , where  $S_{i0} = P_{join_{i-1}}, S_{i k_i} = P_{join_i}$ . Subscript  $k_i$  is the smallest integer for which the value of  $f_{i k_i}(v)$  ( $v \in S_{i k_i}$ ) is updated from 0.*

## 5 Flow Calculation

### 5.1 Synthesis of Circuit-Flow Value

In this section, we discuss nets that are strongly connected structures in FC subclasses. The token flow of an FC circuit is calculated by superimposing the token flow of an MG circuit. When the union of two MG circuits is constructed by the structure of the handle for each circuit, the calculation of the circuit flow value is defined for each type of handle. The TT-handle is not discussed because it is already included in the calculation of the circuit flow value.

### 5.2 PP Handle

Let two distinct strongly connected MG circuits be denoted  $c_1$  and  $c_2$ ,  $c_1 \cup c_2 = c_1 \cup h_1$  or  $c_2 \cup h_2$ , where  $h_1, h_2$  is the PP handle. Let  $f_1, f_2$  represent  $c_1, c_2$  of the circuit flow value, respectively. Using natural numbers  $a$  and  $b$ , we can represent the circuit flow value of  $c_1 \cup c_2$  as  $f_1^a + f_2^b$ . This signifies that  $c_1$  and  $c_2$  token flows synchronously through the transition, which is the start point of the TP-handle. In addition, it signifies that the output transition of the place that is the starting point of the PP-handle is selective and that the tokens are selectively transferred  $a$  times on the  $c_1$  circuit or  $b$  times on the  $c_2$  circuit.

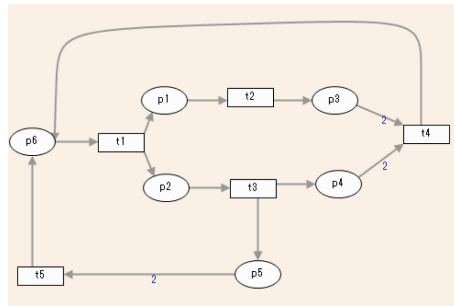
### 5.3 TP Handle

Let two distinct strongly connected MG circuits be denoted  $c_1, c_2$ ,  $c_1 \cup c_2 = c_1 \cup h_1$  or  $c_2 \cup h_2$ , where  $h_1, h_2$  is a TP-handle. Let  $f_1$  and  $f_2$  represents  $c_1$  and,  $c_2$  of the circuit flow value, respectively. The circuit flow value of  $c_1 \cup c_2$  is represented by  $f_1 + f_2$ ; this signifies that  $c_1$  and  $c_2$  token flows synchronously through the transition that is the start point of the TP-handle.

### 5.4 Examples of Strongly Connected MG Closed Token Flow Synthesis

Figure 2 displays a strongly connected weighted FC WFFP net, that is covered by MG circuits  $c_1 = \{p_1, p_2, p_3, p_6, p_7, t_1, t_3, t_7\}$ ,  $c_2 = \{p_1, p_4, p_5, p_6, p_7, t_5, t_6, t_7\}$ . The circuit flow values of  $c_1$  and  $c_2$  are  $f_1 = 1$  and  $f_2 = 1$ , respectively, and  $c_1 \circ c_2 = f_1^a f_2^b = 1^a \times 1^b = 1$ . Therefore, this net satisfies structural boundedness and liveness by the circuit calculation of PP-handles.

Figure 5 also displays a strongly connected weighted FC WFFP net, that is structurally bounded and live, however, it becomes unbounded when the weights are ordinary. This net is covered by MG circuits  $c_1 = \{p_1, p_2, p_3, p_4, p_6, t_1, t_2, t_3, t_4\}$ ,  $c_2 = \{p_2, p_5, p_6, t_1, t_3, t_5\}$ . The circuit flow values of  $c_1$  and  $c_2$  are  $f_1 = \frac{1}{2}$  and  $f_2 = \frac{1}{2}$ , respectively, and  $c_1 \circ c_2 = f_1 + f_2 = 1$ . Therefore, this net satisfies structural boundedness and liveness by the circuit calculation of TP-handles.



**Fig. 5.** Weighted free-choice net diagram covered by strongly connected marked graph components

## 6 Conclusion

Up to now, analysis of the structural liveness and boundedness of strongly connected weighted FC nets has not been performed. It has been demonstrated that analysis of the structural liveness and safety of Petri nets can be performed by net circuit flow value calculation using flow net transformation for a strongly connected MG. In this paper, we assume that there are merging transitions included in all circuits, and we propose that this precondition can be eliminated by systematizing the circuit flow value calculation. In future work, we plan to define the composition of circuits related to PT-handles.

## References

1. Bouyekhf, R., El Moudni, A.: On the analysis of some structural properties of petri nets. *IEEE Trans. Syst. Man Cybern. Part A Syst. Humans* **35**(6), 784–794 (2005)

2. Desel, J., Esparza, J.: *Free Choice Petri Nets*. Cambridge University Press, New York (1995)
3. Girault, C., Valk, R.: *Petri Nets for System Engineering: A Guide to Modeling, Verification, and Applications*. Springer, Heidelberg (2001)
4. Harie, Y., Wasaki, K.: Stability subnet detection of petri net by circuit flow-matrix transformation. In: *IEICE Technical Report*, number 296 in CAS2018-64, MSS2018-40, pp. 37–42, Shizuoka, November 2018
5. Ji, G., Wang, M.: Analysis of structural properties of petri nets based on product incidence matrix. *Kybernetika* **49**(4), 601–618 (2013)
6. Murata, T.: Petri nets: properties, analysis and applications. *Proc. IEEE* **77**(4), 541–580 (1989)
7. James Lyle Peterson: *Petri Net Theory and the Modeling of Systems*. Prentice Hall PTR, Upper Saddle River (1981)
8. Reisig, W.: *Petri nets: an introduction*. EATCS Monographs on Theoretical Computer Science, vol. 4 Springer (1985)
9. Teruel, E., Chrzastowski-Wachtel, P., Colom, J.M., Silva, M.: On weighted t-systems. In: Jensen, K. (ed.) *Application and Theory of Petri Nets 1992*, pp. 348–367. Springer, Heidelberg (1992)
10. van der Aalst, W.M.P.: Verification of workflow nets. In: Azéma, P., Balbo, G. (eds.) *Application and Theory of Petri Nets 1997*, pp. 407–426. Springer, Heidelberg (1997)
11. van der Aalst, W.M.P., Basten, T.: Inheritance of workflows: an approach to tackling problems related to change. *Theor. Comput. Sci.* **270**(1), 125–203 (2002)



# Classification of a Pedestrian's Behaviour Using Dual Deep Neural Networks

James Spooner<sup>1,2</sup>(✉), Madeline Cheah<sup>2</sup>, Vasile Palade<sup>1</sup>, Stratis Kanarachos<sup>1</sup>,  
and Alireza Daneshkhah<sup>1</sup>

<sup>1</sup> Centre for Connected and Autonomous Automotive Research, Coventry University,  
Coventry, UK

spoonerj@coventry.ac.uk

<sup>2</sup> HORIBA MIRA Limited, Nuneaton, UK

**Abstract.** Vulnerable road user safety is of paramount importance as transport moves towards fully autonomous driving. The research question posed by this research is of how can we train a computer to be able to see and perceive a pedestrian's movement. This work presents a dual network architecture, trained in tandem, which is capable of classifying the behaviour of a pedestrian from a single image with no prior context. The results show that the most successful network was able to achieve a correct classification accuracy of 94.3% when classifying images based on their behaviour. This shows the use of a novel data fusion method for pedestrian images and human poses. Having a network with these capabilities is important for the future of transport, as it will allow vehicles to correctly perceive the intention of pedestrians crossing the street, and will ultimately lead to fewer pedestrian casualties on our roads.

**Keywords:** Pedestrian prediction · Deep learning · Classification · Neural networks

## 1 Introduction

This paper outlines a method in which a pedestrian's behaviour can be accurately classified. In the evolving world of Advanced Driver Assistance Systems (ADAS), and Autonomous Driving (AD), pedestrian safety is one of the most difficult but necessary benefits of such technology. In particular, the computer vision systems that are devised to identify pedestrians need to be as accurate as possible as to limit the number of missed pedestrians; as even one missed pedestrian could result in a fatal accident, as was seen in the case of Uber in Arizona in 2017 [13].

For a number of years, pedestrian safety has been a concern of many global regions with an emphasis to reduce the number of pedestrian deaths on the roads. Until recently, this was the case. In the UK during 2016, pedestrian fatalities made up 25% of all road fatalities, while the total number of pedestrian deaths saw a rise of 10% when compared to the previous year [16].

Therefore, this research introduces a method which is able to not only identify the pedestrian themselves, but to identify their behaviour as well. This identification of behaviour represents a step forward in being able to better understand a pedestrian and their movements, as it is well documented the extent to which human behaviour can be unpredictable, especially while distracted [20].

Section 2 will discuss some of the relevant research in the area of pedestrian behaviour classification, as well as different methods for estimating a human's pose. Section 3 will introduce the dataset and the datatypes used, as well as how the data was manipulated.

Section 4 will introduce the methods employed in creating a network capable of classifying a pedestrian's behaviour, as well as statistics of the dataset we used. Following this, Sect. 5 showcases the results from our approach, and highlights some examples, while Sect. 6 will discuss what the results mean, and areas in which the research can be improved. Finally, Sect. 7 discussed the applications of this research, and the goals of the wider research topic moving forward.

## 2 Related Work

### 2.1 Pedestrian Detection

Pedestrian detection has been at the forefront of computer vision innovations since a major breakthrough from Dalal and Triggs' paper [2]. They introduced a method named Histograms of Oriented Gradients (HOG), which formed one of the first semi-successful edge detection methods. HOG methods are still used in conjunction with many modern methods. Another work produced in around the same time introduced a rapid object detector built on boosted cascades of simple features [21]. Both of these methods paved the way for pedestrian detectors to improve to a point of 'real time' detection, with pedestrian detections being made at  $\sim 6$  fps in 2010 [5].

After the publishing of AlexNet [15] in 2012, researchers began to investigate the use of convolutional neural networks (CNN) for the pedestrian detection task. The first CNN development in this application was made by Sermanet *et al.* [17] scoring a miss rate of 77.20%, while the state of the art at the time using traditional machine learning techniques scored a miss rate of 45.39% [22].

However, the use of deep learning for pedestrian detection gained traction, and the current state of the art pedestrian detector registered a miss rate of just 7.67% on the Caltech pedestrian dataset [8].

As with any neural network, the quality of the output is dependent on the size and quality of the data provided at training. In pedestrian detection, the most popular dataset for training detectors is the Caltech pedestrian dataset [6]. This dataset comprises of approximately 350,000 annotated bounding boxes collected from 10 h of driving footage. This dataset is certainly good enough to train a network to be able to identify a pedestrian, however it lacks the contextual understanding of the pedestrian to be able to learn and train a network to understand the behaviour of the pedestrian. The dataset used in this research tackles this problem. The Joint Attention for Autonomous Driving (JAAD) [14]



contains all the bounding box information that the Caltech dataset contains, however, it also contains the behaviours of each pedestrian, whether that be ‘crossing’, ‘walking’, ‘looking’, etc.

The methods used, outlined in Sect. 4, highlight the classification approach taken in this research, attempting to classify all the given classes from the dataset. The following results from the dataset will identify areas of improvement in both the methods used and the use of the dataset, finally moving onto methods of how the pose information can be included in the classification.

## 2.2 Pose Estimation and Behaviour Classification

The image classification is the first approach tackled in this research, which is then complemented with the addition of human pose structures extracted using an off the shelf pose estimator. Human pose estimation has become somewhat of a hot topic in machine learning recently, with pose estimators such as OpenPose [1] and AlphaPose [9].

Research has been published using the JAAD dataset with pose estimation, however their research significantly simplifies the problem [10]. Instead of using the pose information for all the classes, outlined in Sect. 3, they simplify the problem to crossing or not crossing, where in reality there can be a lot learnt in the behaviours and poses that lead up to the crossing situation, such as ‘looking’ for example.

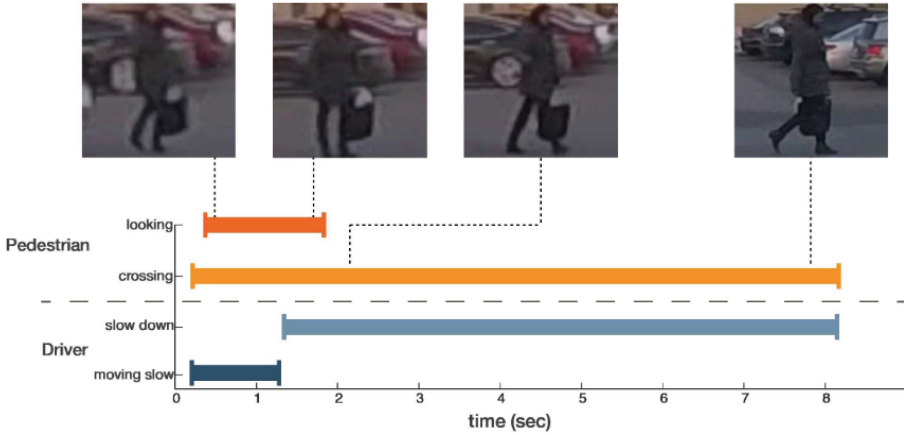
Our research offers a novel solution to use both images and pose estimation to classify the behaviour exhibited by a pedestrian at any given time step.

## 3 Dataset

### 3.1 Image Dataset

The JAAD dataset is introduced with the view that it can be used to help Autonomous Driving (AD) vehicles learn how to interact with pedestrians and the environment that they are in [14]. For the most part in normal driving scenarios between a driver and a pedestrian, there are social cues which allow the driver and pedestrian to communicate, such as eye contact, hand waves, and nodding [19].

There are many pedestrian datasets available, including Caltech Pedestrian [7], Berkeley Pedestrian Dataset, and KITTI [3]. All of these datasets are well known and well used, however JAAD is different, allowing it be used for a more fine tuned classification task. JAAD has the addition of pedestrian behaviour added to the annotations, so that in any given frame, there is bounding box information for the location of the pedestrian, as well information regarding their current behaviour. On top of that, the JAAD provides information about the ego vehicle from which the dataset is collected. This information is again about the behaviour, but this time, the behaviour of the driver. This information can prompt some very interesting research questions, however the driver behaviour



**Fig. 1.** Example of the observed behaviour of a pedestrian from the JAAD dataset. Example taken from [14].

is not considered in this research. Bounding box information is provided for all pedestrians in frame, as well as other vehicles. However, the only bounding boxes that were of interest were those that also contained information regarding behaviour.

The JAAD dataset consists of 346 high-resolution driving clips, of length between 5 and 15 s, collected at 30 fps. These clips were extracted from over 240 h of driving footage in several global locations, namely in Ukraine, Canada, Germany, and USA.

For the dataset to be fed into a Convolutional Neural Network (CNN), the data needed to be pre-processed in such a way that would suit the end requirement of the classifier. The first step was to extract all of the frames in each video sequence, and this had to be done in Matlab to iterate through the video file provided and save each frame as a new image. The next task was to extract all of the bounding box information into a format which was readable in Python. Following this, the bounding box information was correlated to the whole frame for each person in each frame, and the pedestrian(s) of interest were cropped out of each frame and saved as a new file. This decision was made so that the computational requirements when training would be smaller due to smaller image sizes. Also, when the images were resized to be fed into the network, the pedestrians of interest were not made too small to be legible for the network.

The next step was to allocate each cropped pedestrian into their relevant class(es) for each frame. As can be seen in Fig. 1, the behaviour is measured in a time based window. This being that the start frame and end frame of each behaviour is known. It can be noted that there are common classes for the behaviours observed, in these instances, the decision was made to replicate the image in multiple classes rather than try to decide what class each should belong. This is discussed further in the results section.

When all the images were in their class folders, images with a file size smaller than 3 KB were deleted as these were far too small and ambiguous for the network to learn anything from. The final dataset size is over 100,000 images across 9 classes. The classes and number of images in each class can be seen in Table 1. All images were then re-sized to the ImageNet input size of  $224 \times 224$ . This was done by zero padding the images first to a square so that the aspect ratio was not changed, then re-sized to  $224 \times 224$ . This meant that some images would have had a different resolution to their original form, but all images were the same size for the neural network input.

**Table 1.** Observed behaviour classes

Class	Num images
Crossing	53,330
Walking	21,684
Looking	15,436
Standing	11,615
Speed up	2,962
Clear path	2,217
Slow down	1,524
Handwave	309
Nod	95
Total	109,172

### 3.2 Pose Estimation Subset

To create the pose estimation subset, an off the shelf pose estimator was used. The pose estimator chosen was AlphaPose [9], as it showed more accurate and favourable results compared to other pose estimators such as OpenPose [1].

To create the poses, all of the JAAD cropped, padded and resized images were passed through the AlphaPose estimator. In order to limit the number of false positives, a confidence limit of 50% was applied to the estimator. This means that AlphaPose had to be at least 50% confident of its pose prediction in order to generate an output. Due to the type of images contained in the JAAD dataset, namely, some very dark images, AlphaPose was not able to provide a prediction on all images; only the images which had a matching pose were used. Table 1 reflects the number of matched samples in our training.

The pose structure for each image consisted of 17 points on the body, in an  $x, y$  pixel position format. For the use in the neural network, this was reshaped from a (17, 2) array to a flattened (34, 1) array.

An example of the AlphaPose output can be seen in Fig. 2 where the first two images represent individuals in the ‘crossing’ class, while the third image

represents someone in the ‘standing’ class. The red dots are the AlphaPose outputs for the body points, which have been remapped onto the original image for representation purposes. For training the network, only the coordinates of the red dots were used to train the pose branch of the network.



**Fig. 2.** Example of AlphaPose estimation on 3 images from JAAD

## 4 Methodology

### 4.1 Image Classification

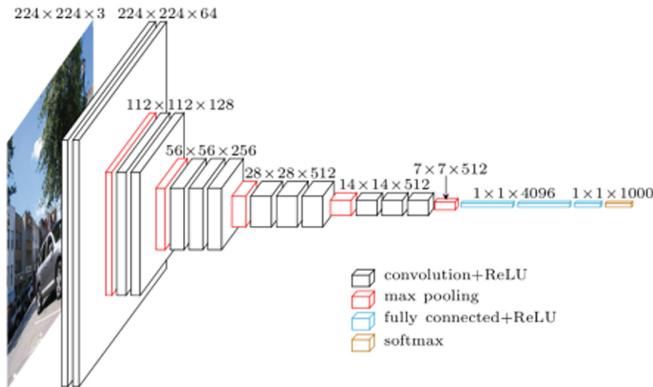
The methodology was based around the use of Convolutional Neural Networks (CNN). The types of networks used are discussed, as well as the training types and data splits for validation. The training process took on a combination of training a network end-to-end from scratch, and transfer learning from a network trained on the ImageNet dataset [4]. The ImageNet challenge is a classification challenge where entrants compete to score the best classification score when classifying images into 1,000 classes. The ImageNet models are trained on over 1 million images.

The first network architecture used was a very simple 6 layer neural network. This consisted of two convolutional layers, with a max pooling layer in between, followed by 3 fully connected layers to give a classification result.

The second network architecture used was AlexNet [15]. This is an architecture which builds on the simple network from before, it consists of several more convolutional, and pooling layers, as well as introducing dropout layers. As this was taken from the ImageNet challenge, the number of final classes has been changed from 1,000 to 9.

Dropout layers are very useful for building a more versatile network. They work by ‘turning off’ a certain percentage of neurons at a certain layer for that epoch, meaning that a network doesn’t over fit an output signal to a common set of neurons in each round of training.

The third network architecture selected was VGG-16 [18]. This is a deep neural network developed by the University of Oxford. The network uses very small convolutional filters of  $3 \times 3$ , which allows for significant improvement on accuracy by creating 16 trainable weight layers. The architecture of which can be seen in Fig. 3<sup>1</sup>. When tested on the ImageNet challenge, VGG16 was able to score a top-5 error of 9.62%; which is quite impressive given that the network is still relatively shallow.



**Fig. 3.** Architecture of VGG16<sup>3</sup>

The next architecture tested was the ResNet architecture created by Microsoft [11]. ResNet architectures are published with networks as shallow as 18 learnable layers, or as deep as 152 layers.

As the name suggests, ResNet is a deep residual learning network which harnesses residual functions for learning with reference, instead of learning unreferenced functions. The addition of this residual element means that the networks are far deeper than many networks, and are also able to optimise, and gain accuracy from the increased depth.

An example of ResNet block learning is seen in Fig. 4. The network is made up of these blocks, connected to one another to create a network of the desired length. The blocks work by taking an input  $x$  through two convolutions and filters, in this case  $F(x)$ . The addition of the original input is then combined with the output by using a shortcut connection; a type of connection that allows information to skip layers. This addition of the input is an identity map which allows the output to create a residual map of the input, meaning that the output

<sup>1</sup> <https://www.cs.toronto.edu/~frossard/post/vgg16/>.

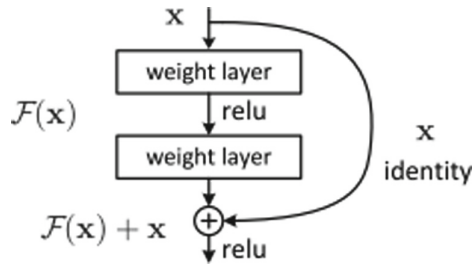


Fig. 4. Example of residual learning in a ResNet block [11]

now has a reference of ‘where it came from’. By completing this process on the inputs and outputs for each block, it enables the network to be able to better optimise deeper into the network. The output residual map,  $F(x) + x$ , is now the input for the next residual block after passing through an activation function.

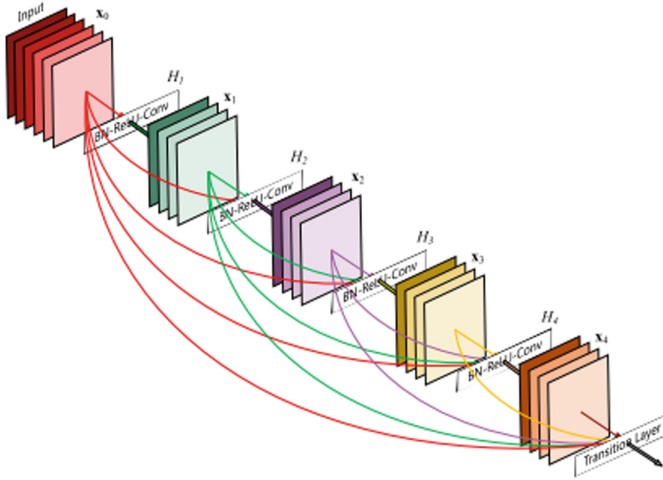
Finally, the last network architecture explored was Densenet [12]. Densenet follows on from the philosophy of ResNet, by building deeper networks with residual features. The distinct difference between ResNet and Densenet is that instead of feeding the input of the residual block to the output of that processed input, Densenet uses the output of every preceding block as the input of every dense block. Therefore, the first dense block only has the input fed through the block, and the second dense block will have both the input from the first block, and the output from the first block (the same as ResNet). However, on every dense block after this, the outputs from previous dense blocks concatenate, so that the third dense block will receive input from the output of the first dense block, and the output from the second dense block. An example of this can be seen in Fig. 5.

### 4.2 Dual Network with Pose Classification

In conjunction with the CNN classifier developed on the images, a classifier was also designed to be able to classify just the poses. Following this, a novel architecture is proposed to dual train both network types at the same time, concatenate their outputs, and then classify the linear layers.

First, the pose classifier was built around the idea that the pose data is of low dimensionality, and therefore does not require complex convolutional layers or a particularly deep network. It was therefore decided to build a simple multi-layer Artificial Neural Network (ANN), using linear layers and widely used activation functions.

Therefore, the network architecture of the pose classifier was obtained through trial and error to achieve the best accuracy. The results of the classifier are discussed in Sect. 5. The best fully connected ANN for the pose classifier was constructed as follows:

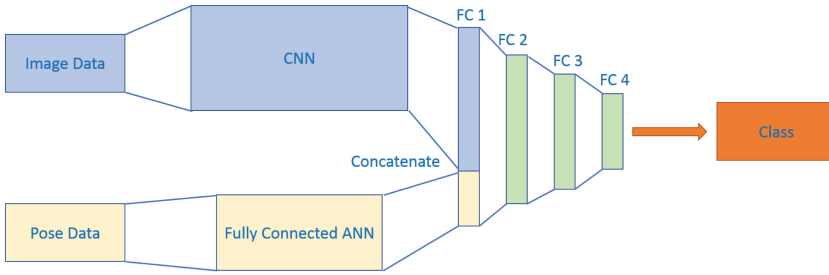


**Fig. 5.** Example of the Densenet architecture [12]

- Input Layer (34,1)
- Hidden Layer (200)
- ReLU activation
- Hidden Layer (300)
- ReLU activation
- Hidden Layer (250)
- Dropout (0.2)
- Hidden Layer (150)
- ReLU activation
- Hidden Layer (50)
- Output Layer (9 classes)

Following this, the construction of the dual trained network was formed. Figure 6 shows a schematic of how the network came together. It can be seen that the CNN and pose classifier form different branches of the network, where the CNN is trained on images, and the pose classifier is trained on the pose estimation. For the CNN classifier, the original model remains the same up until the fully connected layers. Here is where the output of the final hidden layer of the pose classifier is concatenated to the first fully connected layer of the CNN.

This is where the data fusion takes place. The dual network is trained with the matched samples taking the features learnt from the CNN, along with the features learnt from the pose classifier, and classifying the behaviour of the pedestrian based on both sources of data. This architecture and associated results present the novel contribution from this research.



**Fig. 6.** Network architecture for CNN and pose classifier

### 4.3 Training Set-up

For training the network, the data must be separated into three independent groups. One for training, one for validation, and one for testing. For all networks, the data had a 70/10/20 split, respectively. Before the data was split, it was randomised so that all the classes would be mixed up together to give a good representation when training.

All networks were trained with a batch size of 16 images and poses, and a maximum number of epochs of 20, due to trying to limit the training time. The network would identify the best performing epoch out of 20, and save the respective model. Categorical cross entropy loss was used as the function to calculate the losses in the network. This type of error was selected as the most appropriate, as it suits the classification task best. Other loss functions include Mean Square Error (MSE) and Root Mean Square Error (RMSE), however as it is only classification being measured and not regression, MSE would struggle to penalise errors in a compatible way as there is a known number of possible outputs (the classes themselves). Stochastic Gradient Decent was selected as the optimiser with an initial learn rate of 0.001, and momentum of 0.9. The learning rate was decreased at a rate of 0.1 for every 7 epochs until training completed.

The networks were built using Python programming language with the following environments:

- Python 3.6.7
- Anaconda distribution 2018.12
- iPython 7.2.0
- Spyder 3.3.2

In order to load and train the chosen network architectures, the following packages were used in Spyder:

- PyTorch 1.0.1
- CudaToolKit 9.0
- Torchvision 0.2.2
- Numpy 1.16.2
- Pandas 0.24.2



The networks were trained on a Dell Precision 7520 with 4 GB Quadro M2200 GPU. The training, dependent on the network, took between 11 min and 467 min for the full dataset.

## 5 Results

### 5.1 Pose Classification Results

In order to select the optimal pose classifier, trial and error was performed on different fully connected networks, trialling different numbers of parameters and different numbers of hidden layers.

For the purpose of choosing the best network set up, the pose classifier was trained with a final layer outputting the number of classes. For the dual trained network, this layer was omitted.

The results of the fully connected network were compared to other traditional machine learning methods. The chosen benchmarks included a Single Vector Machine (SVM), Radial Basis Function (RBF) network, and a Random Forrest classifier. The classification accuracy for each machine learning method is seen in Table 2.

**Table 2.** Pose classification scores

Method	Accuracy score (%)
SVM	78.4
RBF	81.6
Random Forrest	82.1
<b>Fully Connected ANN</b>	<b>91.7</b>

The results on the pose classification show that the pose classifier was able to correctly classify the pose of a pedestrian to their behaviour with an accuracy of 91.7%.

### 5.2 Dual Image-Pose Network

For the next stage of combining learnt pose features with the CNN features, the same fully connected ANN for the pose was used, while the CNN was changed to discover the best overall accuracy.

The selected network architectures were each trained from scratch with randomised weights and biases, and by using transfer learning. The transfer learning came from the previously defined networks trained on the ImageNet challenge. The idea is that the network has already learnt a lot of the weights and biases that it needs to from the 1 million+ images on ImageNet, therefore when slimming down from 1,000 classes to 9 classes, it is more of a fine tuning task for the network. The results from the transfer learning will be compared to the results from the networks which trained from scratch, with randomly assigned weights and biases.

**Table 3.** Training results for the dataset of images

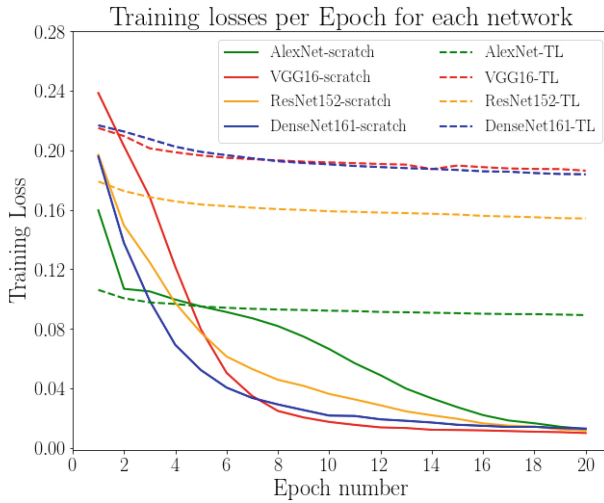
Network	Train type	Time (m)	Accuracy (%)
Simple Net	Scratch	18	69.7
AlexNet	Scratch	22	89.3
	Transfer	11	57.8
VGG16	Scratch	189	92.8
	Transfer	48	48.9
ResNet152	Scratch	384	91.2
	Transfer	223	61.8
<b>Densenet161</b>	<b>Scratch</b>	<b>467</b>	<b>94.3</b>
	Transfer	274	69.1

The results in Table 3 clearly show that Densenet161 trained from scratch is the most accurate classifier, registering a score of 94.3% on the testing partition of the dataset.

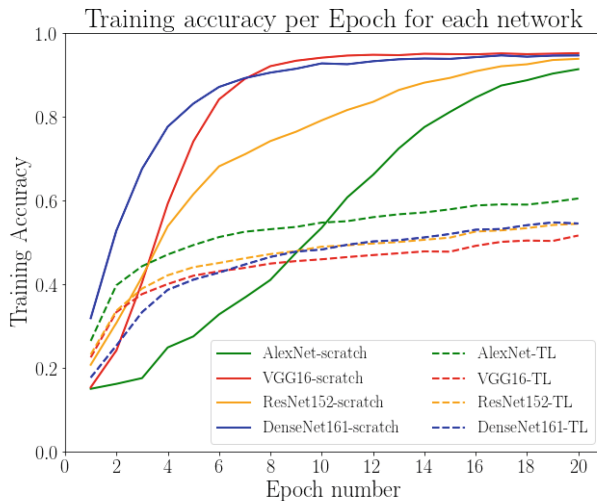
The results are testament to the theory that a greater number of deep convolutional layers can greatly improve the accuracy score in classifications [12]. The two best performing networks in this experiment, Densenet161 and ResNet152, both have in excess of 150 trainable, weighted layers. However, these additional layers come at a cost; the cost of training time.

As is seen in the time column in Table 3, the time taken to train each network varies considerably. It is clear that there is a correlation where the deeper a network becomes, the greater time it takes to train. This makes logical sense as each batch of images has ‘further’ to travel before it registers a classification score. Another trend is in the training time between training from scratch and using transfer learning. Transfer learning is far quicker than training from scratch as the vast majority of its weights and biases have already been learnt, and set in place. In essence, the only layers the networks are learning in transfer learning are the final four layers which were added. The variation in time for transfer learning is due to the fact that all the images still need to pass through the pretrained layers before reaching the trainable layers.

Figure 7 and Fig. 8 show the plotted losses and accuracy of the models over the training period. It is clear to see that over the 20 epochs of training, that the losses reduce and the accuracy improves, as expected. As is confirmed with the accuracy results, it is clear that the transfer learning methods are massively outperformed by the scratch training method, for all network architectures. It is also interesting to see the difference in the rate of convergence seen in the networks; where the loss and accuracy converge far quicker for ResNet, VGG, and Densenet when compared to Alexnet, however they all end with a very similar accuracy. Each network was ran 3 times, with the plots showing the average losses and accuracy from all 3 runs. Each run would have seen a different random split of images, therefore one run could have theoretically been easier or more difficult than the previous.



**Fig. 7.** Training losses for the average of 3 runs on all network architectures



**Fig. 8.** Training accuracy for the average of 3 runs on all network architectures

## 6 Observations and Analysis

As can be seen from the results in Table 3, it is clear that transfer learning is not suitable for this task. This could be due to many reasons, one of which is the size of the dataset it is being trained on. As it is not the smallest of datasets, it means that there is enough training information for the networks to effectively learn the weights and biases to an accurate enough degree. By having the weights

and biases pre-set from ImageNet, it is clear that by not retraining them, that it hinders the classification result.

Another reason for the lack of accuracy in the transfer learning scores is likely due to the types of images and classes that the network is seeing. The core task of this classifier is to be able to classify the behaviour of a pedestrian from a single image. By locking down the weights and biases in the main convolutional layers, it means that the network isn't able to learn the slight intricacies of pedestrians between behaviours. Instead, the network then relies upon the weights and biases it previously learnt when it was trying to classify in image into 1,000 classes; essentially putting a massive constraint on the network's ability to learn.

It is clear when referring to Fig. 7 and 8 that the transfer learning struggles to grasp any learning from the second epoch onwards. The main reason is down to the type of transfer learning employed, where all of the feature layers are locked and the only additional layers are linear, fully connected layers. The problem stems back to the original problem which is trying to be solved; the intricacies and differences between the classes are very slight. Whereas the transfer learning networks have been trained on a huge variety of images. This means that for every image passing through the pre-trained model is likely to fire and activate the same neurons, regardless of class, due to the fact that they would activate the neurons for the network classifying a human. The absence of any feature layers in the new layers further compounds this issue, as the network as a whole never has a chance to try to learn the differentiating features between classes.

When considering the networks which were trained from scratch, it is clear that these networks performed significantly better than the transfer learning networks. Highlighted in Table 3 is the network and score of Densenet161, which was the best performing classifier on this dataset, scoring an accuracy of 94.3%.

Considering the size and complexity of the networks tested, it is clear that they have all performed quite well. Alexnet performed excellently considering it only contains 18 layers, when compared to the result from Densenet161, which has nearly 10 times the number of layers and had an improved accuracy of only 5%.

When looking at this comparison, it is important to consider the training time. Although Densenet161 had the best accuracy of all the networks, it took considerably longer, with its longest run taking 467 min. Depending on the application and need to achieve ultimate accuracy, there may be a balance point between accuracy achieved and the time it takes to achieve that accuracy. An example being that Alexnet achieved a score only 5% less accurate than Densenet, but trained 21 times quicker.

It is also worth noting that the training losses and accuracy of Alexnet have not yet converged at 20 epochs, so therefore, there is still room to increase the accuracy of the classification with a slightly higher number of epochs.

The epochs chosen were based on trying to limit the training time on the machine used for training, however a greater number of epochs would have likely been able to increase the accuracy score on all observed networks. Although Fig. 7 shows that the plots for VGG and Densenet have begun to converge at around

12 epochs, the raw scores of the losses show a continued reduction in loss for every epoch continuing to epoch 20. Therefore, it can be inferred that there is still learning to be made from each network, and more significantly on Alexnet, as it is clear that the accuracy has not begun to converge in Fig. 8.

An important error in the results is from how the images were segmented in their respective classes. As can be seen in Fig. 1, any one frame can be classified into more than one class. This means that, taking the example from Fig. 1, for one or more frames, the pedestrian can be classified into the classes of ‘crossing’ and ‘looking’.

Therefore, the decision was made to include the image of pedestrian once per class for the entire dataset, rather than manually looking and deciding which class it should belong to. This meant that during the training of the network, the networks would have seen many of the same images more than once per epoch, albeit belonging to different classes each time. This would have led to a certain amount of ambiguity when the network came to classify the results, due to the fact that it could classify someone as ‘looking’, when the labelled data for that specific image would have said that they are ‘crossing’, although the classification of ‘looking’ would not have been technically incorrect.

## 7 Conclusions and Future Work

This paper has introduced a novel classifier for the purposes of classifying the behaviour of a pedestrian in a single frame. By training a variety of state of the art classifiers, the results show that the dual trained network, comprising of Densenet161 CNN for the images, and a fully connected ANN for the poses, was able to estimate the behaviour of a pedestrian with a confidence of 94.3%. In comparison, Alexnet registered less favourable, but acceptable results of 89.3% while training was 21 times quicker than Densenet161.

The use of this research will aid the development of a time based recurrent network, where the current and future behaviour and position of a pedestrian will be predicted. This will be done with the addition of sequential data in the form of image sequences and pose sequences. This will require a dynamic label for each time step in the in sequence.

This prediction will have many uses within the scope of autonomous driving, especially from the angle of simulation. The prediction will be able to inform intelligent simulation models, making it more difficult for autonomous systems under test, as they will see stochastic pedestrian movements, which will have been learnt from real life scenarios, such as the ones used in this research.

The main aim of this research topic is to further the safety of vehicles on the road, and in turn, to reduce the number of pedestrian fatalities seen on our roads. This research is a small step along the way, and will be used in the pursuit of the end goal of reducing pedestrian fatalities.

## References

1. Cao, Z., Hidalgo, G., Simon, T., Wei, S., Sheikh, Y.: OpenPose: realtime multi-person 2D pose estimation using part affinity fields, pp. 1–14, December 2018. <http://arxiv.org/abs/1812.08008>
2. Dalal, N., Triggs, W.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, no. 3, pp. 886–893 (2004)
3. [DATASET], Karlsruhe Institute of Technology, and Toyota Technological Institute at Chigaco. KITTI Vision Benchmark Suite (2011)
4. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Li, F.-F.: ImageNet: a large-scale hierarchical image database. In: CVPR, pp. 248–255 (2009)
5. Dollar, P., Belongie, S., Perona, P.: The fastest pedestrian detector in the west. In: Proceedings of the British Machine Vision Conference 2010, pp. 68.1–68.11 (2010)
6. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: a benchmark. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009, pp. 304–311 (2009)
7. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(4), 743–761 (2012)
8. Du, X., El-Khamy, M., Morariu, V.I., Lee, J., Davis, L.: Fused deep neural networks for efficient pedestrian detection, pp. 1–11, May 2018
9. Fang, H., Xie, S., Tai, Y., Lu, C.: RMPE: regional multi-person pose estimation. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2353–2362. IEEE, October 2017
10. Fang, Z., López, A.M.: Is the pedestrian going to cross? Answering by 2D pose estimation. In: Proceedings of IEEE Intelligent Vehicles Symposium, June–July 2018, pp. 1271–1276 (2018)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR 2016 (2016)
12. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks, August 2016. <http://arxiv.org/abs/1608.06993>
13. Kohli, P., Chadha, A.: Enabling pedestrian safety using computer vision techniques: a case study of the 2018 Uber Inc. self-driving car crash (2018). <http://arxiv.org/abs/1805.11815>
14. Kotseruba, I., Rasouli, A., Tsotsos, J.K.: Joint Attention in Autonomous Driving (JAAD). 1–10 (2016). <http://arxiv.org/abs/1609.04741>
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances In Neural Information Processing Systems, pp. 1–9 (2012)
16. Reynolds, S., Tranter, M., Baden, P., Mais, D., Dhani, A., Wolch, E., Bhagat, A.: Reported road casualties Great Britain: 2016. Technical report, September 2017
17. Sermanet, P., Kavukcuoglu, K., Chintala, S., Lecun, Y.: Pedestrian detection with unsupervised multi-stage feature learning. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3626–3633 (2013)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. 1–14 (2014). <http://arxiv.org/abs/1409.1556>
19. Sucha, M., Dostal, D., Risser, R.: Pedestrian-driver communication and decision strategies at marked crossings. *Accid. Anal. Prev.* **102**, 41–50 (2017)

20. Thompson, L.L., Rivara, F.P., Ayyagari, R.C., Ebel, B.E.: Impact of social and technological distraction on pedestrian crossing behaviour: an observational study. *Inj. Prev.* **19**(4), 232–237 (2013)
21. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001*, vol. 1, pp. I–511–I–518 (2004)
22. Yan, J., Zhang, X., Lei, Z., Liao, S., Li, S.Z.: Robust multi-resolution pedestrian detection in traffic scenes. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3033–3040 (2013)



# Towards Porting Astrophysics Visual Analytics Services in the European Open Science Cloud

Eva Sciacca<sup>(✉)</sup>, Fabio Vitello, Ugo Becciani, Cristobal Bordiu, Filomena Bufano, Antonio Calanducci, Alessandro Costa, Mario Raciti, and Simone Riggi

INAF, Catania Astrophysical Observatory, Catania, Italy  
eva.sciacca@inaf.it

[https://www.researchgate.net/profile/Eva\\_Sciacca](https://www.researchgate.net/profile/Eva_Sciacca)

**Abstract.** The European Open Science Cloud (EOSC) aims to create a federated environment for hosting and processing research data to support science in all disciplines without geographical boundaries such that data, software, methods and publications can be shared as part of an Open Science community of practice. This work presents the ongoing activities related to the implementation of visual analytics services, integrated in EOSC, towards addressing the diverse astrophysics user communities needs for data management, mapping and structure detection. These services relies on visualisation to manage the data life cycle process under FAIR principles, integrating data processing for imaging and multidimensional map creation and mosaicing, and, injecting machine learning techniques, for detection of structures in large scale multidimensional maps.

**Keywords:** Visual analytics · Cloud computing · Astrophysics

## 1 Introduction

The European Open Science Cloud<sup>1</sup> (EOSC) initiative has been proposed by the European Commission in 2016 to build a competitive data and knowledge economy in Europe with the vision of enabling a new paradigm of transparent, data-driven science as well as accelerating innovation driven by Open Science [1].

In Astrophysics, data (and metadata) management, mapping and structure detection are fundamental tasks involving several scientific and technological challenges. A typical astrophysical data infrastructure includes several components: very large observatory archives and surveys, rich databases containing several types of metadata (e.g. describing a multitude of observations) frequently

---

<sup>1</sup> EOSC web page: <https://ec.europa.eu/research/openscience/index.cfm?pg=open-science-cloud>.



produced through long and complex pipelines, linking to outcomes within scientific publications as well as journals and bibliographic databases. In this context, visualization plays a fundamental role throughout the data life-cycle in astronomy and astrophysics starting from research planning, and moving to observing processes or simulation runs, quality control, qualitative knowledge discovery and quantitative analysis. The main challenges came to integrate visualisation services within common scientific workflows in order to provide appropriate supporting mechanisms for data findability, accessibility, interoperability and reusability (FAIR principles [14]).

Large-scale sky surveys are usually composed of large numbers of individual tiles, for 2D, 3D or data cubes, each one mapping a limited portion of the sky. This tessellation derives from the observing process itself, when a telescope with a defined field of view is used to map a wide region of the sky by performing several pointings. Although it is simpler for an astronomer to handle single-pointing datasets for analysis purposes, it strongly limits the results for objects extending over multiple contiguous tiles/cubes and hampers the possibility to have a large-scale view on a particular phenomenon (e.g. the Galactic diffuse emission). Tailored services are required to map and mosaic such data for scientific exploitation in a way that their native characteristics (both in 2D and 3D) are preserved.

Additionally, the astrophysics community produces data at very high rates, and the quantity of collected and stored data is increasing at a much faster rate than the ability to analyse them in order to find specific structures to study. Due to the increase of data volume and complexity that will need to be analysed, a suite of an as fully automatic as possible structure detection services, integrating machine learning techniques, is required - consider as an example the ability to recover and classify diffuse emission and to extract and estimate the parameters for compact sources and extended structures.

This work presents the ongoing activities related to the implementation of services, integrated in EOSC, towards addressing the diverse astrophysics user communities needs for: i) putting visualisation at the center of the data life cycle process while underpinning this by FAIR principles, ii) integrating data processing for imaging and multidimensional map creation and mosaicing, and, iii) injecting machine learning techniques, for an as fully automatic as possible detection of structures in large scale multidimensional maps.

## 2 Background and Related Works

Innovative developments in data processing, archiving, analysis and visualization are nowadays unavoidable to deal with the data deluge expected in next-generation facilities for astronomy, such as the Square Kilometer Array<sup>2</sup> (SKA).

The increased size and complexity of the archived image products will rise significant challenges in the source extraction and cataloguing stage, requiring

---

<sup>2</sup> SKA web page: <https://www.skatelescope.org/>.

more advanced algorithms to extract scientific information in a mostly automated way. Traditional data visualization performed on local or remote desktop viewers will be also severely challenged in presence of very large data, requiring more efficient rendering strategies, possibly decoupling visualization and computation, for example moving the latter to a distributed computing infrastructure.

The analysis capabilities offered by existing image viewers are currently limited to the computation of image/region statistical estimators or histogram display and to data retrieval (images or source catalogues) from survey archives. Advanced source analysis, from extraction to catalog cross-matching and object classification, are unfortunately not supported as the graphical applications are not interfaced with source finder batch applications. On the other hand, source finding often requires visual inspections of the extracted catalog, for example to select particular sources, reject false detections or identify the object identity. Integration of source analysis into data visualization tools could therefore significantly improve and speed-up the cataloguing process of large surveys, boosting astronomer productivity and shortening publication times.

As we approach to the SKA era, two main challenges are to be faced in the data visualization domain: scalability and data knowledge extraction and presentation to users. The present capability of visualization softwares to interactively manipulate input datasets will not be sufficient to handle the image data cubes expected in SKA (200–300 TB at full spectral resolution). This expected volume of data will require innovative visualization techniques and a change in the underlying software architecture models to decouple the computation part from the visualization. This is for example the approach followed by new-generation viewers such as CARTA [4]. CARTA uses a “tiled rendering” method and a client-server model, in which computation and data storage is performed on remote clusters with high performance storage, while visualization of processed products is performed on clients with modern web features, such as GPU-accelerated rendering. The volume and complexity of future SKA data will however require not only to import and visualize input data but also, mostly, to maximize user perception efficiency, e.g. allow extraction of meaningful knowledge or discover new unexpected information from data. Indeed the ability to extract science from data represents for SKA the ultimate challenge.

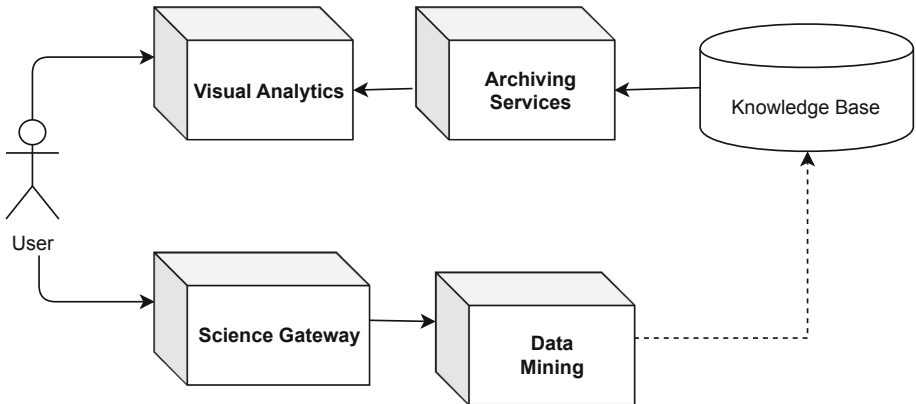
To address these needs under a unified framework, visual analytics (VA) has recently emerged as the “science of analytical reasoning facilitated by interactive visual interfaces” [15]. VA aims to develop techniques and tools to support people in synthesizing information and deriving insight from massive, dynamic, unclear, and often conflicting data [6, 7]. To achieve this goal, VA integrates methodologies from information, geospatial and scientific analytics but also take advantages from techniques developed in the fields of data management, knowledge representation and discovery, and statistical analytics. In this context new developments have been recently done for astronomy. As an example, the encube framework [13] was developed to enable astronomers to interactively visualise, compare and query a subset of spectral cubes from survey data. It provides a large scale comparative visual analytics framework tailored for use with large

tilled displays and advanced immersive environments like the CAVE2 [5] (a modern hybrid 2D and 3D virtual reality environment).

### 3 VisIVO Visual Analytics

VisIVO Visual Analytics [12] is an integrated suite of tools focused on handling massive and heterogeneous volumes of data coming from cutting-edge Milky Way surveys that span the electromagnetic spectrum and provide a homogeneous coverage of the entire Galactic Plane. The tool access the data previously processed by data mining algorithms and advanced analysis techniques with highly interactive visual interfaces offering scientists the opportunity for in-depth understanding of massive, noisy, and high-dimensional data.

Alongside the data collections the tool exposes also the knowledge derived from the data including information related to e.g. filamentary structures, bubbles and compact sources.



**Fig. 1.** Architecture of the VisIVO visual analytics.

Figure 1 shows the VisIVO Visual Analytics integrated framework where the Visual Analytics desktop client, the Science Gateway embedding the Data Mining pipelines and the Knowledge Base can be employed both as independent actors or as interacting components.

### 4 EOSCPilot Science Demonstrator

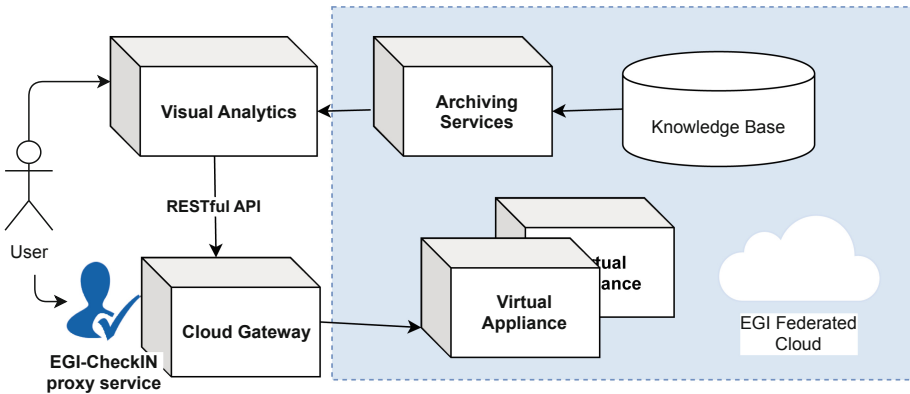
The EOSCpilot project<sup>3</sup> supported the first phase in the development of the European Open Science Cloud, bringing together stakeholders from research

<sup>3</sup> EOSCpilot web page: <https://eoscpilot.eu/>.

infrastructures and e-Infrastructure providers and engaging with funders and policy makers to propose and trial EOSC's governance framework.

The VisIVO project has been selected as science demonstrator [2] functioning as high-profile pilot that integrate astrophysical data and visual analytics services and infrastructures to show interoperability within other scientific domains such as earth sciences and life sciences. Thus, the connection with the European Open Science Cloud has been investigated exploiting the services developed within the European Grid Initiative (EGI) such as the ones to allow federated authentication and authorization and the federated cloud for analysis and archiving services.

The visual analytics application has been extended by exploiting the use of the EOSC technologies for the archive services and intensive analysis employing the connection with the ViaLactea Science Gateway<sup>4</sup> [11].



**Fig. 2.** Architecture of the VisIVO EOSC science demonstrator implementation and employed services.

Figure 2 shows the overall architecture of the VisIVO EOSC Science Demonstrator implementation and employed services. The *Archiving Services* (including the knowledge base) have been deployed within the EGI Federated Cloud toward the assurance of a FAIR access to the surveys data and related metadata. The *Cloud Gateway* has been integrated with the EGI Check-in<sup>5</sup> proxy service to enable the connection from the federated Identity Providers and with the EGI Federated Cloud<sup>6</sup> to expand the computing capabilities making use of a dedicated *Virtual Appliance* stored into the EGI Applications Database<sup>7</sup>. Actually the virtual appliance is exploited for massive calculation of spectral energy distributions but may be expanded for other kind of analysis.

<sup>4</sup> ViaLactea Science Gateway: <https://vialactea-sg.oact.inaf.it/>.

<sup>5</sup> EGI Check-in service: <https://www.egi.eu/services/check-in/>.

<sup>6</sup> EGI Federated Cloud: <https://www.egi.eu/services/cloud-compute/>.

<sup>7</sup> EGI Applications Database: <https://appdb.egi.eu/store/vappliance/visivo.sd.va>.

Furthermore, we have implemented also a lightweight version of science gateway framework developing an ad-hoc *RESTful API* to expose a simple set of functionalities to define pipelines and executing scientific workflows on any Cloud resources, hiding all the details of the underlying infrastructures.

## 5 Future Works: Further EOSC Exploitation

The H2020 NEANIAS project<sup>8</sup> has been recently approved by the European Commission to address the ‘Prototyping New Innovative Services’ challenge set out in the recent ‘Roadmap for EOSC’ foreseen actions. NEANIAS will drive the co-design, delivery, and integration into EOSC of innovative thematic services, derived from state-of-the-art research assets and practices in three major sectors: underwater research, atmospheric research and space research. Each thematic service will not only address its community-specific needs but will also enable the transition of the respective community to the EOSC concept and Open Science principles. From a technological perspective, NEANIAS will deliver a rich set of services that are designed to be flexible and extensible; they will be able to accommodate the needs of communities beyond their original definition and to adapt to neighboring cases, fostering reproducibility and re-usability.

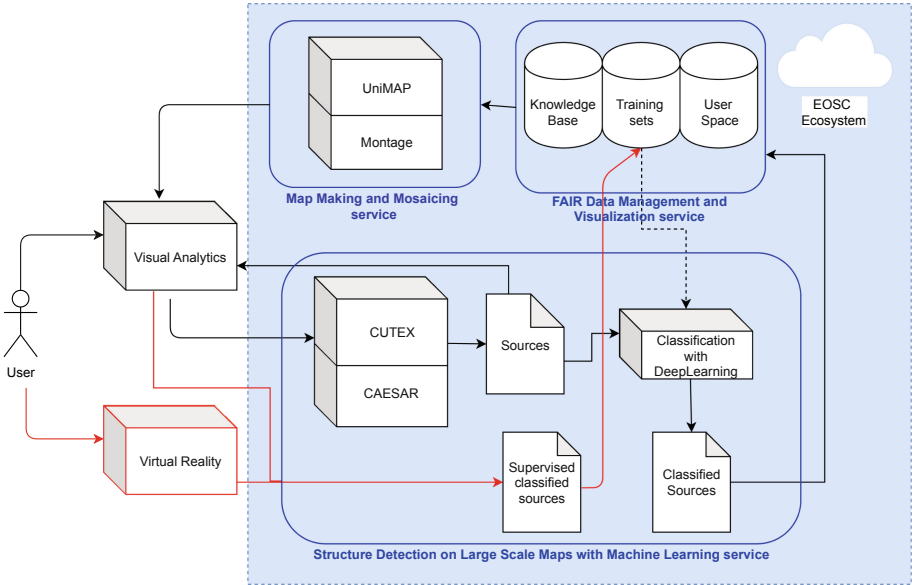
The foreseen services related to the astrophysics visual analytics are:

- The *FAIR Data Management and Visualization service* will provide an advanced operational solution for data management and visualization service for space FAIR data. It will provide tools that enable the efficient and scalable visual discovery, exposed through advanced interaction paradigms exploiting virtual reality.
- The *Map Making and Mosaicing of Multidimensional Space Images service* will deliver a user-friendly cloud-based version of the already existing workflow for map making and mosaicing of multidimensional map images based on open source software such as Unimap [9] and Montage [3]. It will deliver multi-dimensional space maps through novel mosaicing techniques to a variety of prospective users/customers (e.g., mining and robotic engineers, mobile telecommunications, space scientists).
- The *Structure Detection on Large Scale Maps with Machine Learning service* will deliver a user-friendly cloud-based solution for innovative structure detection (e.g. compact/extended sources, filaments), extending the CAESAR [10] and CuTEx [8] tools with machine learning frameworks. The delivered structure detection capabilities will leverage the targeted-users’ opportunities for efficiently identifying and classifying specific structures of interest.

Figure 3 shows the main workflow foreseen to exploit the EOSC ecosystem for visualization, source finding and classification of Big Data images and 3D spectral datacubes coming from Galactic Plane surveys.

The user will employ the Visual Analytics tool to load data from the data management services opportunely mapped and mosaiced. The tool will exploit

<sup>8</sup> NEANIAS web page: <https://www.neanias.eu/>.



**Fig. 3.** Foreseen NEANIAS architecture of the visual analytics services in EOSC.

the source finding applications to extract the sources from the data. Optionally expert users may employ the Visual Analytics tool and/or Virtual Reality application to interactively classify the sources thanks to the aided visual inspection. The results of the supervised classification can be stored to the data services enriching the training set for the Deep Learning networks. The extracted sources are classified automatically with the Deep Learning algorithms and the results will be stored within the data user space and can be optionally published for re-use from other users and/or to enrich the training set.

## 6 Conclusion

We presented the ongoing activities related to the implementation of services, integrated in EOSC, towards addressing the diverse astrophysics user communities needs for visual analytics services. The preliminary demonstration implementation developed within the H2020 EOSCPilot project has been summarized and forthcoming activities, to be developed within the H2020 NEANIAS project, are foreseen to integrated services in EOSC for FAIR data management and visualization integrating data processing for imaging and multidimensional map creation and mosaicing, and, injecting machine learning techniques, for detection of structures in large scale multidimensional maps.

**Acknowledgments.** The research leading to these results has received funding from the European Commissions Horizon 2020 research and innovation programme under the grant agreement No. 863448 (NEANIAS).

## References

1. Ayris, P., Berthou, J.-Y., Bruce, R., Lindstaedt, S., Monreale, A., Mons, B., Murayama, Y., Södergård, C., Tochtermann, K., Wilkinson, R.: Realising the European Open Science Cloud. European Union, Luxembourg (2016)
2. Becciani, U., Vitello, F., Sciacca, E., Costa, A., Calanducci, A., Riggi, S., Molinari, S.: VisIVO visual analytics tool: an EOSC science demonstrator for data discovery. In: *Astronomical Data Analysis Software and Systems XXVIII*, vol. 523, pp. 29–32. ASP Conference Series, 2018VisIVO Visual Analytics Tool: An EOSC Science Demonstrator for Data Discovery (2019)
3. Berriman, G.B., Deelman, E., Good, J.C., Jacob, J.C., Katz, D.S., Kesselman, C., Laity, A.C., Prince, T.A., Singh, G., Su, M.-H.: Montage: a grid-enabled engine for delivering custom science-grade mosaics on demand. In: *Optimizing Scientific Return for Astronomy through Information Technologies*, vol. 5493, pp. 221–232. International Society for Optics and Photonics (2004)
4. Comrie, A., Wang, K.-S., Ford, P., Moraghan, A., Hsu, S.-C., Pińska, A., Chiang, C.-C., Jan, H., Rob, S.: The cube analysis and rendering tool for astronomy. CARTA (2019)
5. Febretti, A., Nishimoto, A., Thigpen, T., Talandis, J., Long, L., Pirtle, J.D., Peterka, T., Verlo, A., Brown, M., Plepys, D., et al.: CAVE2: a hybrid reality environment for immersive simulation and information analysis. In: *The Engineering Reality of Virtual Reality 2013*, vol. 8649, p. 864903. International Society for Optics and Photonics (2013)
6. Ham, D.-H.: The state of the art of visual analytics. In: *EKC 2009 Proceedings of the EU-Korea Conference on Science and Technology*, pp. 213–222. Springer (2010)
7. Keim, D.A., Mansmann, F., Schneidewind, J., Thomas, J., Ziegler, H.: Visual analytics: scope and challenges. In: *Visual Data Mining*, pp. 76–90. Springer (2008)
8. Molinari, S., Schisano, E., Faustini, F., Pestalozzi, M., Di Giorgio, A.M., Liu, S.: Source extraction and photometry for the far-infrared and sub-millimeter continuum in the presence of complex backgrounds. *Astron. Astrophys.* **530**, A133 (2011)
9. Piazza, L., Calzoletti, L., Faustini, F., Pestalozzi, M., Pezzuto, S., Elia, D., di Giorgio, A., Molinari, S.: Unimap: a generalized least-squares map maker for herschel data. *Mon. Not. R. Astron. Soc.* **447**(2), 1471–1483 (2014)
10. Riggi, S., Vitello, F., Becciani, U., Buemi, C., Bufano, F., Calanducci, A., Cavallaro, F., Costa, A., Ingallinera, A., Leto, P., et al.: Caesar source finder: recent developments and testing. *Publ. Astron. Soc. Aust.* **36**, e037 (2019)
11. Sciacca, E., Vitello, F., Becciani, U., Costa, A., Hajnal, A., Kacsuk, P., Farkas, Z., Marton, I., Molinari, S., Di Giorgio, A.M., et al.: ViaLactea science gateway for milky way analysis. *Future Gener. Comput. Syst.* **94**, 947–956 (2017)
12. Vitello, F., Sciacca, E., Becciani, U., Costa, A., Bandieramonte, M., Benedettini, M., Di Giorgio, A.M., Elia, D., Liu, S.J., Molinari, S., et al.: Vialactea visual analytics tool for star formation studies of the galactic plane. *Publ. Astron. Soc. Pac.* **130**(990), 084503 (2018)
13. Vohl, D., Fluke, C.J., Hassan, A.H., Barnes, D.G., Kilborn, V.A.: Collaborative visual analytics of radio surveys in the big data era. *Proc. Int. Astron. Union* **12**(S325), 311–315 (2016)

14. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L.B., Bourne, P.E., et al.: The fair guiding principles for scientific data management and stewardship. *Sci. Data* **3**, 160018 (2016)
15. Yi, J.S., Kang, Y.A., Stasko, J.: Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. Vis. Comput. Graph.* **13**(6), 1224–1231 (2007)





# Computer Graphics-Based Analysis of Anterior Cruciate Ligament in a Partially Replaced Knee

Ahmed Imran<sup>(✉)</sup>

Ajman University, Ajman, UAE

a.imran@ajman.ac.ae, ai\_imran@yahoo.com

**Abstract.** Artificial human knee with partial prosthetic replacement was modelled in the sagittal plane in order to analyze the role of anterior cruciate ligament in an unconstrained artificial knee. The cruciate and collateral ligaments were modelled as non-linear elastic fibers that stretched and resisted relative movements of the bone. Role of fibers in the anterior and posterior fibers of the anterior cruciate ligament was analyzed during simulated tests similar to those used in clinical practice. Anterior half of the ligament was found to resist forces for all simulated flexion positions of the joint. The posterior half resisted forces in low and in high flexion positions and remained unstretched during for nearly 30–90° flexion. The model calculations agreed with experimental observations on cadaver knees reported in the literature. A graphical interface facilitated visual analysis of the joint while the ligament fibers stretched sequentially developing forces and unstretched becoming slack as the joint flexed or the femoral and tibial bones with prosthetic parts moved relative to each other. The cruciate ligaments controlled the joint kinematics after replacement. The model analysis helps in visual analysis and in gaining insight into the joint behavior with clinical relevance.

**Keywords:** Artificial knee · ACL in partial knee replacement · Unconstrained knee prosthesis · Anterior cruciate ligament · Cruciate ligaments in unconstrained arthroplasty

## 1 Introduction

Artificial human knees involving partial resurfacing of the joint with unconstrained prosthetic parts require retention of the cruciate ligaments. Clinical studies suggest that about a third of the patients for knee replacement are suitable for unconstrained type of prosthesis and that correct patient selection and surgical techniques are crucial to successful long-term outcome of such replacements [1]. Studies show that not only the retained cruciate ligaments must be fully intact and functional, but also their proper lengths must be restored during surgery [1]. Such replacements have shown complications like dislocations of prosthetic parts and incorrect patterns of joint laxity after surgery [1]. Cruciate ligaments connect the femoral and tibial bones at the knee and play crucial role in controlling the joint kinematics and provide stability during activity. Unconstrained artificial knee designs assume this role of the ligaments and depend on

their proper function, particularly during flexion motion that takes place in the sagittal plane. The anterior cruciate ligament (ACL) and posterior cruciate ligament (PCL), therefore, play important role in long-term success of such replaced knees. Overtight or slack ligaments result in altered patterns of movement between the bones. Further, the ACL is commonly shown to experience injuries during strenuous activities particular in young active subjects. While ACL constraints posteriors movement of the upper bone or femur on the lower bone or tibia, the PCL constraints anterior movement of the upper bone. Also, anatomical studies show that the ligaments comprise distinct fiber bundles that change their geometry during motion and contribute variably during activity [2–5]. All such geometric patterns and their contributions during the joint activity cannot be observed in the living person. Therefore, in order to understand the joint behavior after replacement, computer graphics could be utilized as a useful tool to visually analyze various patterns during motion or activity.

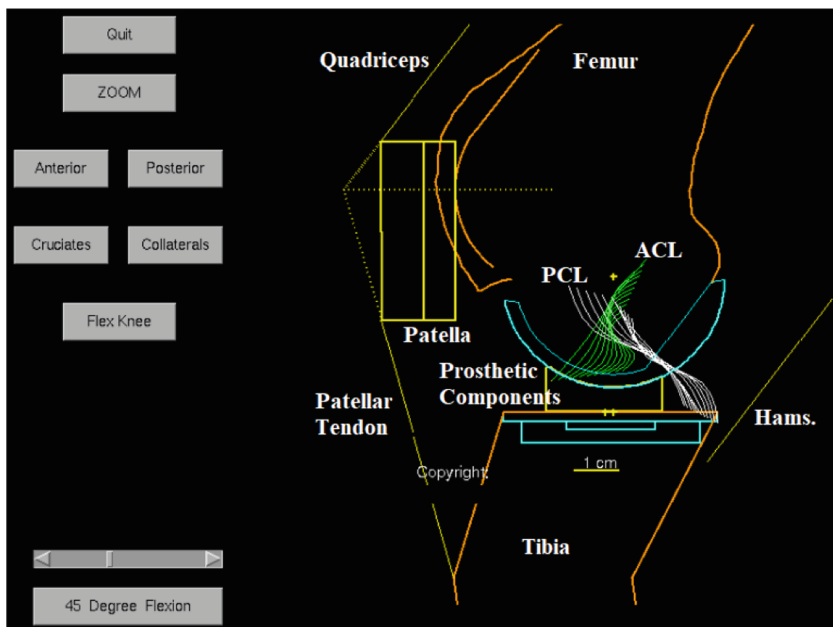
The aim of this study is to present a computer graphics based analysis of the ACL after replacement with unconstrained knee replacement in terms of changing geometries of different ligament fibers as the bones move relative to each other. Input for the visual graphics in the study are derived from mathematical modelling of the replaced joint that is based on anatomical and clinical studies in the literature.

## 2 Methods

A graphics based interactive model of a prosthetic knee was developed with partially replaced joint surfaces for visualization of bones, prosthetic components, ligaments and lines of actions of muscle forces during 0–120° flexion motion. The two cruciate and two collateral (medial and lateral) ligaments of the knee were modelled in the sagittal plane. Parameters for the model for attachments of muscle tendons and ligament fibers on respective bones, material properties of different bundles of ligament fibers and shapes of femoral and tibial bones with implanted joints were obtained from anatomical studies in literature [6–8]. Geometries of the prosthetic components used were similar to those employed for unconstrained partial knee replacement [9, 10].

Figure 1 shows the model bones with attached prosthetic components, ligaments represented as bundles of elastic fibers and lines of actions of the major muscle forces. The elastic fibers of the ligaments offered resistance when stretched and buckled when slack [8–10]. Passive motion of the knee was defined during 0–120° flexion in the absence of external loads or muscle forces. It is shown that during such motion with the prosthetic components placed appropriately, knee kinematics similar to the normal knee can be reproduced [11]. In the model knee with passive motion, the femur rotates and slides relative to the tibia, the muscle forces re-orient and re-locate, selected fibers in the ligaments maintain isometricity while all other fibers either straighten without stretch or slacken [6, 7, 9, 12–14]. The slackness was depicted in the model with buckled fibers [11].

The graphic interface allowed selection of one or more ligaments with their corresponding fibers as required for visual analysis. Further, the interface allows selection of a fixed flexion position and relative translation of the bones anterior or posterior to their position of passive motion. As a consequence of such translation, fibers in the model



**Fig. 1.** Graphical interface of the prosthetic knee showing various options and model elements.

ligaments either stretch sequentially or slacken further, thus providing a clear view of how the ligament fibers respond. The stretched ligaments were shown with thick straight lines. Computational feature of the mathematical model incorporates external loads and muscle forces as experienced during activity. In a simulated clinical test, 90 N anterior force applied externally on the tibia (acting towards front of the knee) resulted in anterior translation of tibia (ATT) and stretched fibers of the ACL, thus, provided the balancing effect. ATT corresponding to the equilibrium position was calculated at selected flexion positions. Model results were compared with *in vitro* measurements of Kondo *et al.* [15] who experimented on 14 cadaver knees to measure ATT corresponding to 90 N anterior force on the tibia as also simulated in the present study. In addition, contributions of anterior and posterior halves of the ACL in terms of developed forces were calculated for known magnitudes of ATT over the flexion range and the results were compared with similar experimental measurements available in the literature [16].

### 3 Results and Analysis

Table 1 compares model calculations with experimental measurements of Kondo *et al.* [15]. Values are given for ATT corresponding to 90 N anterior force applied on the tibia at selected flexion angles.

The model calculations at each flexion angle were similar to those measured experimentally. ATT first increased with flexion, remained at around its maximum during 30 and 60° positions and then decreased in high flexion.

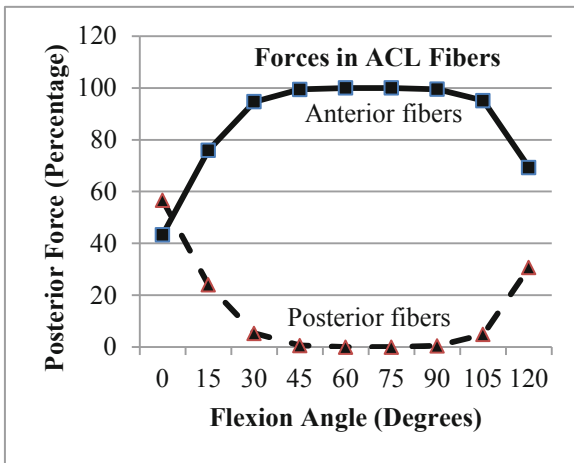
**Table 1.** ATT for 90 N simulated test – as calculated by the model and as measured experimentally by Kondo *et al.* [15] given in mm.

Flexion angle	Model calculations (mm)	Experiment measurements [15] mean (std. dev.) (mm)
0°	2.6	1.8 (1.3)
30°	5	4.0 (1.3)
60°	5	3.8 (2.2)
90°	4	3.0 (1.4)
110°	3.7	2.9 (1.7)

Figure 2 gives forces in the anterior and posterior halves of the ACL plotted over the flexion range. The forces were calculated as a percentage of an anterior force on tibia required to cause 6 mm ATT at each joint position. Near full extension, or 0° degree flexion, both halves of the ACL contributed nearly similarly. While the anterior half remained stretched and developed resistance for all flexion positions, the posterior half remained unstretched and did not develop resistance for nearly 30–90° flexion.

These observations are supported by experimental measurements of Kawagachi *et al.* [16] who used 8 cadaver knees and measured contributions of antero-medial and postero-lateral bundles of the ACL in resisting 6 mm ATT similar to that simulated in the current study. They reported 66% to 84% of the total resistance due to anteo-medial bundle during 0–90° flexion. Corresponding resistance due to the postero-lateral bundle was much less between 16% to 9%.

Visual analysis during each simulation showed changing patterns of ligament fibers and located the stretched fibers with forces.



**Fig. 2.** Forces in anterior and posterior halves of the ACL were calculated as a percentage of total force corresponding to 6 mm ATT plotted during a simulated test over 0–120° flexion.

## 4 Conclusions

The model calculations corroborated the experimental observations and provided insight into the ligament behavior with the facility of visual analysis which is either not amenable to observation or is difficult to observe through real or physically simulated systems. The model is capable of analyzing partial tears by selective sacrifice of the ligament fibers. The role of cruciate ligaments in controlling the joint mechanics after knee replacement with unconstrained prosthetic components can be further analyzed during different activities that can be simulated using the current model.

**Acknowledgments.** The author would like to support the College of Engineering at Ajman University, Ajman, UAE. for support provided for this research project.

## References

1. Hamilton, T.W., Pandit, H.G., Inabathula, A., Ostlere, S.J., Jenkins, C., Mellon, S.J., Dodd, C.A.F., Murray, D.W.: Unsatisfactory outcomes following unicompartmental knee arthroplasty in patients with partial thickness cartilage loss – a medium-term follow-up. *Bone Joint J.* **99-B**(4), 475–482 (2017)
2. Imran, A.: Computer graphics based analysis of loading patterns in the anterior cruciate ligament of the human knee. In: Arai, K., Bhatia, R., Kapoor, S. (eds.) *Advances in Intelligent Systems and Computing*, vol. 998, pp. 1175–1180. Springer, Cham (2019). (2)
3. Lord, B.R., El-Daou, H., Zdanowicz, U., Smigielski, R., Amis, A.: The role of fibers within the tibial attachment of the anterior cruciate ligament in restraining tibial displacement. *Arthrosc.: J. Arthrosc. Related Surg.* **35**(7), 2101–2111 (2019)
4. Imran, A.: Relating knee laxity with strain in the anterior cruciate ligament. In: *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering*, London, pp. 1037–1042 (2017)
5. Imran, A.: Analyzing anterior knee laxity with isolated fiber bundles of anterior cruciate ligament. In: *Proceedings of the World Congress on Engineering*, London, pp 869–872 (2016)
6. Imran, A.: Sagittal plane knee laxity after ligament retaining unconstrained arthroplasty: a mathematical analysis. *J. Mech. Med. Biol.* **12**(2), 1–11 (2012)
7. Imran, A., O'Connor, J.: Control of knee stability after ACL injury or repair: interaction between hamstrings contraction and tibial translation. *Clin. Biomech.* **13**(3), 153–162 (1998)
8. Mommersteeg, M., Blankevoort, L., Huiskes, R., Kooloos, J., Kauer, J.: Characterisation of the mechanical behavior of human knee ligaments: a numerical-experimental approach. *J. Biomech.* **29**(2), 151–160 (1996)
9. O'Connor, J., Shercliff, T., Bide, E., Goodfellow, J.: The geometry of the knee in the sagittal plane. *Proc. Inst. Mech. Eng. (Part H) J. Eng. Med.* **203**, 223–233 (1989)
10. Lu, T.W., O'Connor, J.: Fiber recruitment and shape changes of knee ligaments during motion: as revealed by a computer graphics based model. *Proc. Inst. Mech. Eng. (Part H) J. Eng. Med.* **210**, 71–79 (1996)
11. O'Connor, J., Imran, A.: Bearing movement after Oxford uni-compartmental knee arthroplasty: a mathematical model. *Orthopedics* **30**(5S), 42–45 (2007)
12. Imran, A.: Modelling and simulation in orthopedic biomechanics-applications and limitations. In: Tavares, J.M., Jorge, R.M. (eds.) *Computational and Experimental Biomedical Sciences: Methods and Applications*. Springer, Cham (2015)

13. Imran, A.: Influence of flexing load position on the loading of cruciate ligaments at the knee—a graphics-based analysis. In: Tavares, J., Jorge, R.M. (eds.) *Computational and Experimental Biomedical Sciences: Methods and Applications*. Springer, Cham (2015)
14. Imran, A.: Computer graphics based approach as an aid to analyze mechanics of the replaced knee. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) *Advances in Intelligent Systems and Computing*, vol. 857. Springer, Cham (2019)
15. Kondo, E., Merican, A., Yasuda, K., Amis, A.: Biomechanical analysis of knee laxity with isolated anteromedial or posterolateral bundle deficient anterior cruciate ligament. *J. Arthrosc. Related Surg.* **30**(3), 335–343 (2014)
16. Kawaguchi, Y., Kondo, E., Takeda, R., Akita, K., Yasuda, K., Amis, A.: The Role of fibers in the femoral attachment of the anterior cruciate ligament in resisting tibial displacement. *J. Arthrosc. Related Surg.* **31**(3), 435–444 (2015)



# An Assessment Algorithm for Evaluating Students Satisfaction in e-Learning Environments: A Case Study

M. Caramihai<sup>(✉)</sup>, Irina Severin, and Ana Maria Bogatu

University POLITEHNICA, Bucharest, Romania  
m.caramihai@yahoo.com, irina.severin@upb.ro,  
bogatu\_ana\_maria@yahoo.com

**Abstract.** The aim of this paper is to provide an overview of temporal and spatial properties of eLearning. Firstly, it describes the cognitive factors found both in e-Learning and traditional learning. Also, it gives a comparison between online and traditional learning process emphasizing advantages and disadvantages of both of these two learning processes. Next, it offers a short description of spatial thinking and spatial learning concepts. Also, it details time models and learning process and defines associative processes in temporal-spatial cognition. Finally, this paper gives a comprehensive analysis of temporal and spatial properties of e-Learning by describing the interactions of cognitive factors with traditional learning and online learning, spatial cognition and spatial thinking aspects, learning process and time models, defining associative processes in temporal-spatial cognition and it emphasizes the enhancing of skills acquisition regarding the spatial and temporal properties of eLearning through the student's perspective.

**Keywords:** e-Learning information systems · e-Learning evaluation survey · Student satisfaction · Mathematical approach

## 1 Introduction

Initially, the e-Learning systems were designed to support the traditional method, but nowadays, e-Learning has become one of the most significant developments in the information systems industry [1] because the student satisfaction has become a mandatory requirement, as service stakeholder, in most universities and institutions of higher learning all around the world.

e-Learning which is also known as web-based learning, is defined as the delivery of education in a flexible and easy way through the use of internet to support individual learning or organizational performance goals [2]. However, there are some advantages and disadvantages regarding the e-Learning process.

The advantages consist of the flexibility to select either instructor-Led or self-study courses, the versatility of learning at any place and time, learning through a variety of activities that apply to many different learning styles that students have, the possibility

of reviewing the materials for as long as necessary, self-pacing for slow or quick learners and also helping the students to develop their knowledge of using the latest technologies and the Internet.

The main disadvantages are the lack of a firm framework to encourage students to learn, the low level of contact (due to the lack of interpersonal and direct interaction between students and teachers), the difficulty of learning when problems with the technology within the system arise and the necessity of a high level of self-discipline and motivation from the students.

The purpose of this research is to present a comprehensive e-Learning assessment algorithm incorporating concepts from both information systems and education disciplines.

This study contributes to the e-Learning literature with an instrument providing guidelines for e-Learning systems developers and teachers to better understand the students' perceptions of both social and technical issues associated with e-Learning systems.

## **2 Literature Review**

### **2.1 The Social Framework**

The social setting presents the switching from the real (tangible) environment to the virtual one for students and teachers. Thus, the e-Learning method implies changing the teacher's role into an instructor and the student into a learner. Learners are the external users and interact directly with the system; the instructors are part of the supplier group, they are internal users and interact directly with the e-Learning platforms. Instructors and learners' groups can also interact directly with the system if they promote learning and research activities.

Seeing that interaction is indirect, there's no pressure of time, it's an easy access no matter the location if you have Internet access, the learners express higher confidence interacting through email or social platforms and the instructors can offer information in a more user-friendly way. As a result, the e-Learning system helps increasing the knowledge and the confidence of students.

Therefore, the quality of an instructor (they should find a balance between time and the information given) and the learner's perceived effectiveness (they need motivation, confidence and a dose of excitement) are important determinants for an effective learning management system.

### **2.2 The Technical Entity**

The technical setting involves the software quality (the need of security, ease of use, stability, user-friendliness), the hardware quality (a reliable computer) and the Internet quality. The higher the quality and reliability of used technology, the higher the learning effects will be [3, 4].

The designing and managing of the learning environment are a must for learners for a better understanding of the content and for an easy access. The content quality involves



the useful, flexible and interactive manner which the information is presented, with the purpose of helping learners to easily garner new knowledge.

e-Learning offers a wide range of applications and processes designed to deliver information through the Internet or interactive multimedia. Learning through this medium can engage students' interest because it usually comes together with interactive graphics, texts, sounds and videos. At the same time, it can be accessed anywhere and anytime as long as exists a computer and an Internet connection [5].

### 2.3 Overview Regarding Models for Student's Satisfaction Evaluation in e-Learning Environments

#### The Jaccard Distance [6]

The Jaccard distance is complementary to the Jaccard index, which is a statistic used for gauging the similarity and diversity between finite sample sets. The Jaccard distance measures the dissimilarity between sample sets and is obtained by subtracting the Jaccard index from 1.

#### The Hamming Distance [7]

The Hamming distance is usually used between two strings of equal length to measure the minimum number of substitutions required to change one string into the other. But, in this case, it will be used between two integers to show the distance from zero on the number line.

#### Statistical Analysis

The statistical analysis implies assigning for each criteria and statement a percent and finding the most used grade for each statement.

## 3 Materials and Methods

### 3.1 Presentation of the Algorithms

#### The Jaccard Distance [6]

It uses 2 vectors, one with the grades from each statement from the survey (Check 4.2.2 Presentation of the questionnaire) and one with the ideal grade (5). Then, the algorithm is based on 2 formulas listed below.

$$x = (x_1, x_2, \dots, x_n) = (5, 5, \dots, 5), n = 93 \quad (1)$$

where x is the vector with ideal grades

$$y = (y_1, y_2, \dots, y_n), n = 93 \quad (2)$$

where  $y$  is the vector with real grades from each statement

$$J_W(x, y) = \frac{\sum_i^n \min(x_i, y_i)}{\sum_i^n \max(x_i, y_i)} \text{ (The Jaccard index)} \quad (3)$$

$$d_{JW}(x, y) = 1 - J_W(x, y) \text{ (The Jaccard Distance)} \quad (4)$$

### The Hamming Distance [7]

It uses 2 vectors, one with the grades from each statement from the survey (Check 4.2.2 Presentation of the questionnaire) and one with the ideal grade (5). Then, the algorithm is based on the formula listed below.

$$x = (x_1, x_2, \dots, x_n) = (5, 5, \dots, 5), n = 93 \quad (5)$$

where  $x$  is the vector with ideal grades

$$y = (y_1, y_2, \dots, y_n), n = 93 \quad (6)$$

where  $y$  is the vector with real grades from each statement

$$d_H(x, y) = \frac{\sum_i^n (x_i - y_i)}{\sum_i^n (y_i)} \text{ (The Hamming distance)} \quad (7)$$

## 3.2 Data Collection

### Participants

The survey was carried out by means of an online program which collected data from 93 anonymous students (University POLITEHNICA Bucharest, Automatic Control & Computer Science Faculty, during the academic year 2018–2019) for data analysis in order to obtain the grade of satisfaction for students in the e-Learning environment. Each one of the students answered the questions using a system of grading (from 0 to 5, with 5 the highest).

For a research tool, one of the required conditions is to be considered consistent and reliable [8]. In other words, each item of the questionnaire (questions or statements) must correlate with the additional result of all items (the scale, the global score). The items of the research tool are designed to measure a certain attribute (attitude, knowledge, factor, behaviour). The internal consistency is defined as the property of the items to correlate with the “global score” of the research tool or scale they belong to. Due to the fact that all the items must reflect a certain attribute, they have to correlate with each other and, at the same time individually correlate with the score that reflects that attribute. The correlation between an item and the total score, from which that item is omitted, gives us information regarding the relevance of that item for the global result of the test. When each item is relevant, it can be stated that the research tool has “internal consistency”.

The Cronbach’s-Alfa internal consistency [8] can take values between 0 and 1, where 0 indicates that the research tool only measures random errors, having nothing to do with the real score, and 1 indicates that the research tool measures only the real score, random errors being completely eliminated.

For a scale to be considered consistent, it must overcome the 0.60 value which is accepted as lowest limit by most researchers.

In the present paper, the Cronbach’s-Alfa internal consistency resulted 0.838 (and was determined using the SPSS (Statistical Package for the Social Sciences software), which falls within the accepted limits and reveals that the designed research tool has internal consistency and the items are correlated.

**Presentation of the Questionnaire**

The structured questionnaire and the main themes are presented in Table 1:

**Table 1.** The content of the questionnaire.

Specific domain	Question
Curricular area	It is in line with my expectations
	It allows me to further develop my knowledge
	Online teaching methods are appropriate
	Exercises/homework/quizzes are appropriate
Spatial analysis of the courses	While going through the courses, visual elements are more important than narrative ones
	While going through the courses, you prefer to resume the ideas under the form of written notes or diagrams
	Laboratory tasks should be based on the interaction between the student and the virtual work environment
	The virtual learning environment needs to be more visually inclined rather than narrative
	Because of the richer visual content, the virtual learning environment allows me to better accumulate knowledge
Evaluation	Information related to the structure of exercises/homework/quizzes are appropriate
	The evaluation criteria are clear
	The professor’s evaluation is in line with the results of auto-evaluation
	During solving exercises/homework/quizzes I have used my own notes of the courses
	The evaluation should take into account the student’s capability to solve exercises/homework/quizzes

(continued)

**Table 1.** (continued)

Specific domain	Question
	The evaluation should take into account the student's capability to synthesize gathered knowledge
Temporal analysis of the courses	The online course should always be available to the student
	When I go through an online course, the only limitation I have is own learning schedule
	Online learning should be based on regularly going through the materials
	The online learning activity should be better planned than class learning
	The advantage of online courses is that they can be tackled in any order
	The advantage of online courses is that you can study whenever you feel like it
	Exercises/homework/quizzes should only be solved when the student feels prepared
	Student-teacher interaction should be lower in the eLearning environment
Social opportunities	Online courses offered me more opportunities to interact with other students
	I am part of a group of students that exchange ideas/solving techniques for exercises/homework/quizzes
	The time spent within the online community is higher than the time dedicated to individual study
	It is important that, within the online community, we share any individual notes of the course that we might have

### Input/Output Variables

The input variables are the grades with whom the students rated the questions based on their preferences and opinions. The output variable is the grade of satisfaction for students in the e-Learning environment.

## 4 Discussion and Results

Based on the Table 2 there is a statement (16) with the lowest distance (0.043), which means that the students rated it with a 4.957 out of 5 and a statement (24) with the highest distance (0.531), which means that the students rated it with a 2.345 out of 5.

In addition, the statement with the lowest distance presents the assertion that the students consider almost in unanimity that the online course must be permanently at the student's disposal.

The second one shows that the students agree with the fact that the only limitation they have is their learning curve. And from the third and fourth one it results that the laboratory work is based on the direct interaction of a student with the virtual work environment and that online courses can be browsed without a predefined schedule.

Most of the lowest distances are situated in the time analysis of courses criteria, so based on the survey, the students agree with the helpfulness in terms of time organization

**Table 2.** Distance measurement for each question.

Statements	Jaccard (dJ)	Hamming (dH)
1	0.372	0.372
2	0.329	0.329
3	0.307	0.307
4	0.356	0.356
5	0.221	0.221
6	0.225	0.225
7	0.172	0.172
8	0.208	0.208
9	0.286	0.286
10	0.346	0.346
11	0.286	0.286
12	0.372	0.372
13	0.350	0.350
14	0.264	0.264
15	0.273	0.273
16	0.043	0.043
17	0.169	0.169
19	0.266	0.266
20	0.249	0.249
21	0.172	0.172
22	0.313	0.313
23	0.475	0.475
24	0.531	0.531
25	0.455	0.455
26	0.470	0.470
27	0.318	0.318

in the e-Learning environments. Based on the highest distances' values, the students are against the reduction of the interaction with other students and the fact that online courses offered them more opportunities to interact with colleagues.

Using the Jaccard distance and the Hamming distance on all the ratings given by the students, an average value of 0.3043 is obtained (see Table 3), which means that the grade of satisfaction is 3.478 (between average and high).

**Table 3.** Synthesis of the statistical analysis.

Criteria	Grade for each statement	Percentage result
Curricular area	3 3 4 3	0.650
Analysis of the presentation	4 4 5 4 4	0.680
Evaluation and corrections	3 4 3 4 4 4	0.699
Time analysis of courses	5 5 4 3 5 5 4 3	0.850
Social opportunities	2 4 3 3	0.600

The statistical analysis used implies assigning for each criteria an equal percentage (20%) and every statement a percentage by dividing 20% by the number of statements in the criteria. For each statement it's found the grade with the most appearances. By summing up the values in the column "percentage result" in the table below is obtained the grade of satisfaction (3.479), which is approximately equal (due to the errors made by using approximations) with the value obtained using the Jaccard and Hamming distance algorithm.

This work has continued previous research on students' satisfaction assessment and searching ways for the improvement of education quality [9, 10].

## 5 Conclusions

This paper presented an assessment algorithm for evaluating students' satisfaction in e-Learning environments using three models (the Jaccard distance, the Hamming distance and a statistical analysis) for a better and more precise way of finding the grade of satisfaction.

According to the multi-criteria survey, the grade of satisfaction (3.478) inclines from average to high, which shows that the students appreciate the flexibility of the e-Learning environments regarding the aspects of location and time. Also, the study highlights the lack of interaction student-student and student-teacher.

Therefore, the e-Learning systems are very helpful regarding the process of improving students learning experience and performance, but the students need the traditional way for motivation and social purposes, because the online environment still needs improving in that area. So, for now, a merge between the real and virtual medium is the key for students to succeed in universities and why not, in life.

## References

1. Ozkan, S., Koseler, R.: Multi-dimensional students' evaluation of e-Learning systems in the higher education context: an empirical investigation. *Comput. Educ.* **53**, 1285–1296 (2009)
2. Ragab, A.H.M., Noaman, A.Y., Madbouly, A.I., Khedra, A.M., Fayoumi, A.G.: Essam: an assessment model for evaluating students satisfaction in E-learning environments. *IJAEDU-Int. E J. Adv. Educ.* **4**(11), 175–184 (2018). <https://doi.org/10.18768/ijaedu.455619>
3. Attwell, G.: E-Learning at the Workplace. In: McGrath, S., Mulder, M., Papier, J., Suart, R. (eds.) *Handbook of Vocational Education and Training*, pp. 923–947. Springer, Cham (2019). [https://doi.org/10.1007/978-3-319-94532-3\\_110](https://doi.org/10.1007/978-3-319-94532-3_110)
4. Luaran, J.E., Samsuri, N.N., Nadzri, F.A., Rom, K.B.M.: A study on the students perspective on the effectiveness of using e-learning. *Procedia – Soc. Behav. Sci.* **123**, 139–144 (2014). <https://doi.org/10.1016/j.sbspro.2014.01.1407>
5. Kattoua, T., Al-Lozi, M., Alrowwad, A.: Review of literature on E-Learning systems in higher education. *Int. J. Bus. Manage. Econ. Res.* **7**(5), 754–762 (2016)
6. Hancock, J.M.: Jaccard Distance (Jaccard Index, Jaccard Similarity Coefficient). *Dictionary of Bioinformatics and Computational Biology* (2004). <https://doi.org/10.1002/9780471650126.dob0956>
7. Bravo, J.M.: Calculating Hamming Distance with the IBM Q Experience (2018). <http://doi.org/10.20944/preprints201804.0164.v1>
8. Bonett, D.G., Wright, T.A.: Cronbachs alpha reliability: Interval estimation, hypothesis testing, and sample size planning. *J. Organizational Behav.* **36**(1), 3–15 (2014). <https://doi.org/10.1002/job.1960>
9. Severin, I., Caramihai, M., Khalifa, M.: Assessing Students' Satisfaction: a Case Study, November 16–18, 2010, Manilla, Philipine, “Trends and Prospects of Innovation and Entrepreneurship and its Implications in Engineering and Business Education amidst Global Economic and Environmental Crises”, 3rd Int Conf. on Innovation & Entrepreneurship and the 3rd Int. Conf. on Engineering & Business Education, Editors: Divina M. Edralin, Giovanni R. Barbajera, pp. 374–388, ISSN 2094 – 7607. <http://www.icebe.net/>
10. Bogatu, A.M., Cicic, D.T., Severin, I., Solomon, G.: Value through education. *Faima Bus. Manage. J.* **5**(1), 61–70 (2017). ISSN 2344-408



# The Use of New Technologies in the Organization of the Educational Process

Y. A. Daineko, N. T. Duzbayev, K. B. Kozhaly, M. T. Ipalakova, Zh. M. Bekaulova,  
N. Zh. Nalgozhina<sup>(✉)</sup>, and R. N. Sharshova

International Information Technology University, Almaty, Kazakhstan  
yevgeniyadaineko@gmail.com, nuri.nalgozhina@gmail.com

**Abstract.** This paper discusses the use of new technologies in education. The analysis of the implementation of various innovative developments in education is given. The authors presented the software product using smart technologies, which allows studying physics using virtual reality. Such approach allowed to make interaction with the application more interesting and memorable, and learning more effective. The Unity 3D cross-platform environment was chosen as the development platform. The main functionality was written in C#. Graphic models were created using Substance Painter.

**Keywords:** Smart technologies · Education · Virtual reality · Unity 3D · Leap Motion · Virtual physical laboratory · Physics

## 1 Introduction

Modern society is in a state of global change. “E-government”, “e-learning”, “e-university” – this is not an exhaustive list of concepts and phenomena that did not exist in reality 30 years ago, but now have become an objective reality that changes both social and economic and political mechanisms of society. The education system must respond to the changes of post-industrial society. At the same time, as experience shows, educational services in developed countries are turning into a highly profitable industry, which means that there is a high demand for this “product”. For example, according to some researchers, the export of educational services in the American economy brings an average of \$13 billion per year (5th place among the export sectors of the US economy). Many countries claim that their GDP is based on a knowledge economy of 70–80%. At the same time, 1.5 exabytes of information were created in the world in 2008, which exceeds the volume of 5000 previous years, and the world volume of knowledge has doubled every 72 h (3 days) since 2010 [1]. One of the modern ways of updating the educational process is the use of new teaching methods and new ways of interaction between teachers and students. The great interest is the use of computer-based training systems with the use of new technologies, which should help to master new material, monitor knowledge and help prepare training material. These may be virtual laboratories, which are a computer program or an associated set of programs that performs computer modeling of some processes [2].



Today the virtual reality technologies (VR - Virtual Reality) are increasingly conquering the information technology market and are in great demand when organizing the educational process. Virtual Reality [3] - is the technology that uses software to reproduce a three-dimensional realistic image of the environment in which the physical movement of the user is simulated.

A special role is played by the use of such technologies in the study of natural sciences and technical disciplines, for example, physics [4]. Physics is one of the main subjects for specialists in the natural-technical field, the demand for which is growing every year according to the industrial-innovative policy of the state. Computer modeling of physical processes, implemented in the form of virtual physical experiments are increasingly used in the process of teaching physics. Compared to real laboratory work, virtual laboratory work has several advantages. Firstly, for a detailed study of physical processes, there is no need to buy expensive equipment and hazardous radioactive materials. Secondly, it becomes possible to simulate physical processes in detail, in order to show what is happening inside of the process. Thirdly, virtual laboratory work has more visualization of physical or chemical processes compared to traditional laboratory work. An important factor is security. Compared to using real laboratory equipment, using a virtual laboratory to study processes is undoubtedly a safer way of learning. In addition, software products that simulate physical processes can be written in different programming languages and using various development tools. At the same time, students can be directly involved in the development of that tools.

The current system of higher education in Kazakhstan needs to be transformed. It helps help to maximize the opportunities for training competitive local staff, increase the salaries of teachers, improve computer literacy, actively master modern techniques and technologies, regularly improve the skills of teaching staff through scientific internships and seminars and provide an opportunity for working youth to receive special education without discontinuing work.

The continuity of knowledge, the formation of a national model of multi-level continuing education, integrated into the world educational space and meeting the needs of the individual and society, are the necessary conditions for the development of any civilization. Great plans and accordingly great changes require constant discussion of what and how to teach in a dynamically changing world, how to give young people the same opportunities as their European peers have, how to give education a new impetus, and how to make education – regardless of changes in life – reliable and competitive.

This article presents an example of the use of new technologies for the development of laboratory work in physics. The virtual laboratory works developed by the authors are implemented as an application for studying physics in higher educational institutions.

## 1.1 Related Works

One of the key problems of any education is the problem of retention of students' attention. Virtual Reality allows keeping the attention throughout the lesson with the help of the vivid impressions of things that students have saw. At the same time, attention is not contemplative, but mobilizing. The use of VR expands the possibility of independent work of students, forms the skill of research activities, provides access to various reference systems, electronic libraries, other information resources, and thus contributes to

the quality of education. The feature of the educational process with the use of VR is that the center of activity becomes a student who builds the process of learning based on his individual abilities and interests. The teacher often acts as an assistant, consultant, encouraging original findings, stimulating activity, initiative and independence. To increase the interest of students, a media library, electronic textbooks with VR technology on subjects can be used. These have a number of undoubtedly positive properties that distinguish it from traditional ones. For example, a large number of slides and animations that enhance the students' emotional and personal perception of the material that has to be studied. The use of such a textbook allows to do much more in the lesson than with the help of traditional means, to increase interest in the subject.

The inclusion of both audio and video tools in the learning process allows implementing not only the principle of visualization, but also significantly increases interest in learning. Video, as a medium of information, plays a significant role in the development and education of children. For example, it can be widely used in courses about the knowledge of the world and art, because the bright frames of nature paintings, historical events and places including museum halls, art galleries, provide the scope for children's imagination, arouses desire to know more and explore new knowledge.

The use of VR makes the lesson more dynamic, increases the motivation of students to learn, and allows the teacher to improve the quality of education according with the needs of society. For instance, in [5] has shown modern approaches and examples of systems and applications using augmented and virtual reality (AR/VR) technologies. It means that they contribute to improvement of student learning ability and summarize skills in the real world. Increasing the level of engagement, promoting self-learning, providing multisensory training, increasing spatial abilities, confidence and pleasure, combining virtual and real objects in a real environment and reducing cognitive load are some of the findings presented in this paper. Thus, despite the fact that there are certain problems before introducing virtual reality into educational practices, AR/VR applications provide an effective tool for improving learning and memory, since such technologies provide immersion in an environment enriched with several sensory features. In [6], a meta-analysis of the use of educational games in museums is presented. This analysis is based on a qualitative literature review comparing the educational roles of museums with the earnest training games used to support these roles. This study can help game developers, designers, tour guides in museums and practicing educators make decisions regarding the choice of game type, customization and content design to support informal learning in the specific context of museum educational activities.

Recently, smart mobile devices such as smartphones and tablets have led to the introduction of widespread educational innovations. In [7], the position of teachers to smartphones and tablets was studied, as well as their use in lessons in South Korea. Although the results showed that teachers identified several factors in the inconvenience of use, the main barrier of using digital devices in the classroom is the beliefs of teachers.

In the field of digitalization of education, a special place is given to artificial intelligence. In [8], computational intelligence (CI) methodologies and machine learning techniques were used to develop intelligent learning systems (STS). The integration of artificial intelligence, data science and the Internet of things (IoT) will create a new generation of intelligent systems for all tasks in the field of education and training. The

article identifies and explores the benefits of such intelligent paradigms to improve the effectiveness of intelligent learning systems. They also addressed the problems faced by application developers and engineers in designing and deploying such systems.

Thus, the use of VR technologies opens up unlimited possibilities for improving the quality of students’ knowledge, ensuring the intellectual development of each student, and the effective organization of students’ cognitive activity.

## 2 General

International University of Information Technology has experience in the employment of Smart-technologies in education. The departments “Computer Engineering and Information Security” and “Radio Engineering, Electronics and Telecommunications” has been working on developing their own software using new information technologies. For example, some laboratory works in the course “Physics” are performed virtually using applications developed by the authors. Modeling of physical experiments are carried out in a virtual environment [2]. Students have the opportunity to set the initial parameters and perceive the physical process as many times as desired.

For the development of software, the cross-platform environment Unity 3D game engine from Unity Technologies for computer games has been used [9]. The application works with a wired motion sensor Leap Motion [10], which designed for manual tracking in virtual reality.

The virtual laboratory application using Leap Motion consists of the components shown in Fig. 1. The structure consists of the Main Launch, which opens the application and calls the following two modules: tasks in the sections “Mechanics”, “Electrostatics”, etc. and tasks with support of Leap Motion controller. Only the graphical interface is visible to the user, but there are three more components hidden behind it: a shared folder containing models, scripts and other necessary resources, separate folders for a specific task, and a folder necessary for the correct work of Leap Motion.

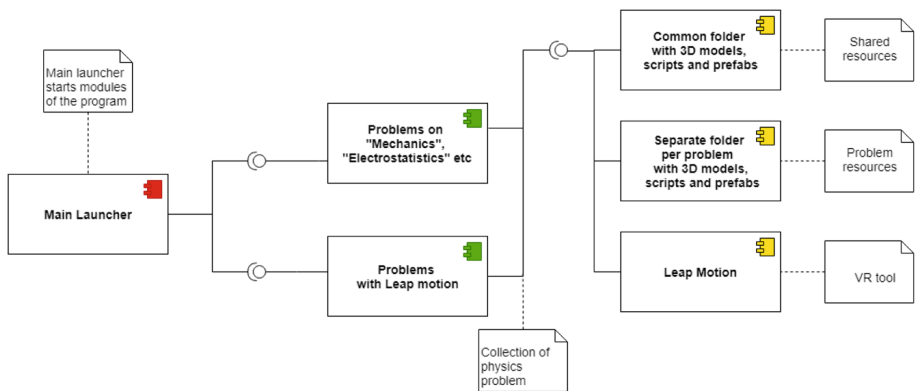
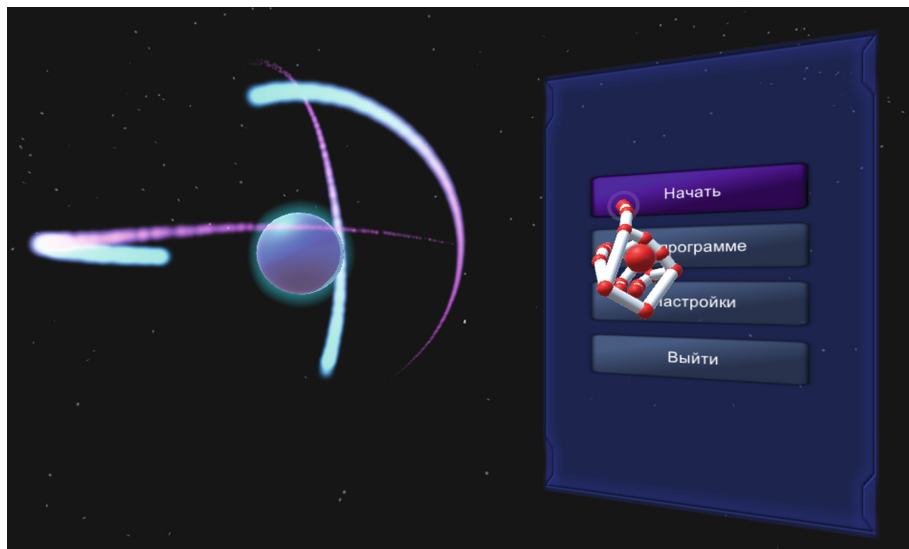


Fig. 1. Application component diagram with Leap Motion

The below Fig. 2 shows the main application menu with the integrated Leap Motion controller.



**Fig. 2.** The main menu of the application with Leap Motion

Interactive control has organized using the Leap Motion controller, keyboard and camera (mouse control), which also allows to rotate 3D scenes in different directions. In addition, the program allows scaling the studied objects for a more detailed overview. Dialog box updates when positions and viewpoints has changed. Interactivity is the main advantage that provides visibility and quick assimilation of the studied materials.

### 3 Conclusion

Thus, the lesson with Virtual Reality not only enlivens the educational process, but also increases the motivation in learning. The use of computer technology in the learning process affects the growth of professional competence of a teacher. This contributes to a significant increase in the quality of education, which leads to the solution of the main task of educational policy. This article is devoted to the development of a software application for studying physics using virtual reality technology. The future work is to expand the functionality of the developed application by integrating into it new practical tasks from other physics sections, animations that shows the physical processes in detail to conduct the new physical experiments.

**Acknowledgment.** The work was done under the funding of the Ministry of Education and Science of the Republic of Kazakhstan (No. AP05135692).

## References

1. Ivanov, A.V.: Adaptivnye sistemy obucheniya. In: Theses of the International Conference “Information Technologies in Education”, Moscow (2019)
2. Daineko, Y., Dmitriyev, V., Ipalakova, M.: Using virtual laboratories in teaching natural sciences: an example of physics. *Comput. Appl. Eng. Educ.* **25**(1), 39–47 (2017)
3. Stanney, K.M.: *Handbook of Virtual Environments: Design, Implementation, and Applications*, p. 23 (2002)
4. Daineko, Y., Ipalakova, M., Tsoy, D.: Development of the multimedia virtual reality-based application for physics study using the Leap Motion controller. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11613, pp. 150–157. LNCS (2019)
5. Papanastasiou, G., Drigas, A., Skianis, C., Lytras, M., Papanastasiou, E.: Virtual and augmented reality effects on K-12, higher and tertiary education students’ twenty-first century skills. *Virtual Real.* **23**(4), 425–436 (2018)
6. Wang, M., Nunes, M.B.: Matching serious games with museum’s educational roles: smart education in practice. *Interact. Technol. Smart Educ.* **16**(4), 319–342 (2019)
7. Leem, J., Sung, E.: Teachers’ beliefs and technology acceptance concerning smart mobile devices for SMART education in South Korea. *Br. J. Educ. Technol.* **50**(2), 601–613 (2019)
8. Salem, A.-B.M., Nikitaeva, A.Y.: Knowledge engineering paradigms for smart education and learning systems. In: *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics*, pp. 1571–1574 (2019)
9. Unity Technologies Homepage. <https://unity3d.com>. Accessed 7 Nov 2019
10. Leap Motion Homepage. <https://www.leapmotion.com/>. Accessed 1 Nov 2019



# Design and Implementation of Cryptocurrency Price Prediction System

Milena Karova, Ivaylo Penev<sup>(✉)</sup>, and Daniel Marinov

Department of Computer Science and Engineering,  
Technical University of Varna, Varna, Bulgaria

{mkarova, ivailo.penev}@tu-varna.bg, daniel\_marinov96@abv.bg

**Abstract.** The paper presents conceptual design of a cryptocurrency price prediction system. Algorithm for data collection and a LSTM neural network for predicting future prices are presented. A brief explanation of the system implementation is shown. The structure of the neural network and the tuning of the hyperparameters are explained. Finally, experimental results with predicted future price of Bitcoin cryptocurrency are presented. The results are compared to the prediction of the Bitcoin price for the same time periods obtained by the Cryptomon system.

**Keywords:** Cryptocurrency · Prediction · Machine learning · Neural network · LSTM · Bitcoin

## 1 Introduction

### 1.1 Motivation

Having knowledge about the past and being able to find dependencies and patterns in it, helps us understand the present and predict the future. Cryptocurrency trading has been an area of growing interest in the past years. In order to profit from selling and buying “virtual money”, cryptocurrency markets must be monitored to make a proper assessment of what the exchange rate for the respective cryptocurrency is expected to be. The process of continuous monitoring cryptocurrency market prices and reading thousands of news, affecting cryptocurrencies’ trading course, is a time consuming task. The high processing power of modern computers gives opportunity for this process to be automated and performed by neural networks, which analyze and process large amounts of information available on the Internet.

### 1.2 Review of Existing Solutions

The majority of the publications concerning cryptocurrency price prediction describe algorithms and methods, which are based on analysis of news and tweets extracted from social media. Most of the published papers use text and sentiment analysis. Typically, user comments and replies are used to predict the price of various cryptocurrencies for different future periods, e.g. [1, 6–8].

As the cryptocurrency prediction problem concerns forecasting unknown data on the basis of past historical data, the proposed solutions usually use machine learning methods. Neural networks are the dominating method for predicting the price of different virtual cryptocurrencies, though some authors use other classifiers, e.g. [4], who use Logistic Regression, Random Forest, K-Nearest Neighbors, Linear SVM, and Gaussian Naïve Bayes. Most of the papers expose prediction of popular cryptocurrencies as Bitcoin, Litecoin, Ethereum [2, 3, 5, 7, 9, 10, 12] and others expose different cryptocurrencies as ZClassic [8]. For example very good results with high accuracy of price prediction of six cryptocurrencies using Long-Short Term Memory (LSTM) neural network are published in [11].

In common, the majority of publications focus on the prediction method. The authors have not found publications concerning the design and implementation of a cryptocurrency price prediction system. Although the prediction methods and algorithms are no doubt an essential component of the system, there are other parts, which are also significant for the effective work of the prediction system. For example the system cannot work without collecting information from remote sources. This function is performed by the collector component of the system. Other important components of cryptocurrency price prediction systems are those providing secure access of users to the functionalities of the system.

Some of the implemented software products that predict cryptocurrency market prices are reviewed in the comparative Table 1. Each of them is a web-based platform and has a machine learning algorithm as part of its back-end logic.

**Table 1.** Review of existing solutions.

	Cryptomon	WalletInvestor	CoinPredictor
Application type	Web-based	Web-based	Web-based
Type of used machine learning algorithm	K-nearest neighbors, Multilayer perceptron network	Neural networks	Neural networks
Data sources	No information is available on the data sources used	Cryptocurrency rates and world news	Historical information on the cryptocurrency exchange rate
Predictions visualization	Plot	Plot	Plot
Interactive visualization	Yes	No	Yes

The authors have also not found in literature comparison of the results from the published methods and algorithms for cryptocurrency price prediction with the predictions obtained by the above-mentioned systems.

### 1.3 Improvement Over Existing Solutions

Designing a predictive algorithm, which reflects the dynamically changing world and improves its prediction accuracy over time, would automate the workflow of a human expert in cryptocurrency market price analysis. The ability of machine learning (and in particular LSTM neural networks) to detect correlations and patterns in time series, goes beyond the capabilities of human memory. Building a platform for accurately predicting the exchange rate of cryptocurrencies increases the profits in crypto trading and provides expert knowledge in the hands of the average user. Implementing a secure communication layer, involving user authentication for gaining access to the predictions, made by the machine learning algorithm, and providing simplified user interface containing interactive diagrams, greatly improves the available software solutions. Incorporating the previously stated concepts with the existence of a web application, as well as a native Android mobile application, leads to high level user experience and makes the platform stand out among other products available on the market.

This paper presents the design and implementation of a cryptocurrency prediction system. The prediction method used is LSTM neural network. The main steps of the design and implementation of the system are described. The structure of the neural network and the tuning of the hyperparameters are explained. The work of the system is tested with prediction of the price of Bitcoin cryptocurrency. The accuracy of the predictions is compared to the results, obtained by the wide-spread system Cryptomon using the same time periods and the same cryptocurrency.

## 2 Cryptocurrency Price Prediction System Concepts

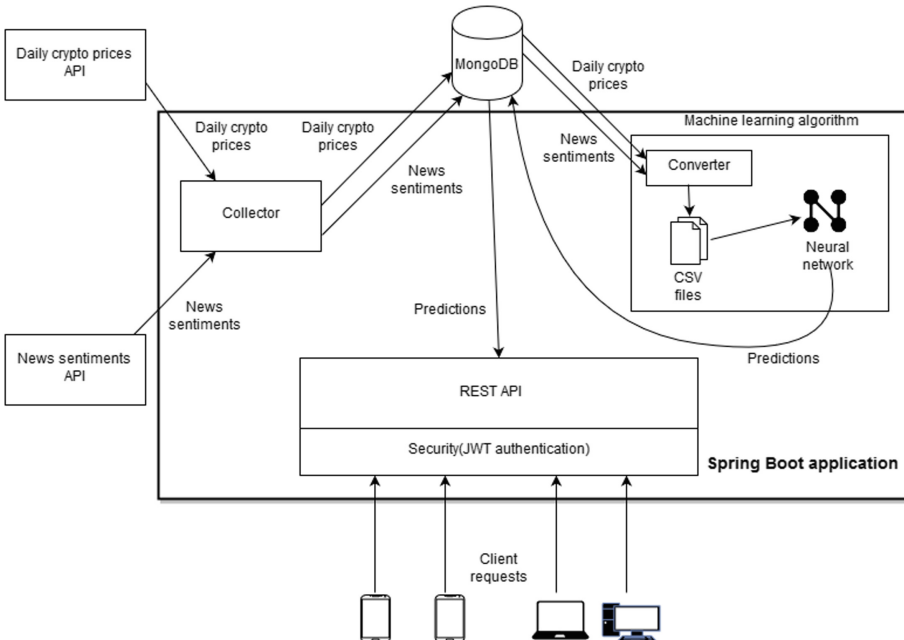


Fig. 1. Conceptual model of the developed system



The system consists of the following logical modules (see Fig. 1):

1. **Collector** – collects data from remote sources (APIs). The data is then stored in a database for further usage by the machine learning algorithm;
2. **Machine learning algorithm** – converts the stored data to a suitable format for machine learning (CSV files). The data is then processed by a LSTM neural network, which outputs one day ahead predictions for the cryptocurrencies’ close prices. The predictions are stored in the database.
3. **REST API** – exposes the predictions made by the machine learning algorithm to the end users. Gives permission only to authorized users to access the secured resources (predictions).
4. **Security** – implemented with JWT authentication.
5. **Front-end applications** – Android and web application for displaying the predictions to the users.

Components numbered from 1 to 4 form the back-end logic of the system and from a development perspective are separate modules in a Spring Boot application.

From a conceptual and scientific point of view, the collector and the machine learning algorithm are of major interest. They form the core functionality of the system.

### 2.1 Collector

The collector is a module in the system, responsible for gathering historical data for cryptocurrencies, which is necessary for the machine learning process. The collected data is of two types:

- The closing market price of the cryptocurrencies for each day.
- Numeric values, describing daily ratings for each cryptocurrency. The ratings are in the scale from 1 to 10 and are calculated as a result of news classification.

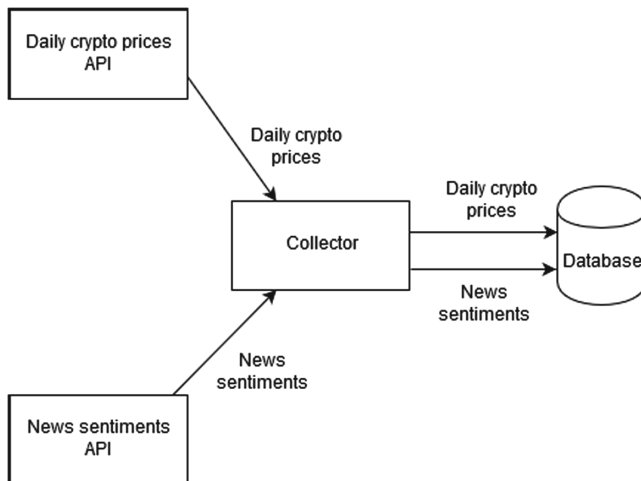


Fig. 2. Conceptual model of collector

The data is collected from two remote APIs – one for the market prices and one for the ratings. A conceptual model of the collector is presented in Fig. 2.

The collector is responsible for handling the following logic (see Fig. 3):

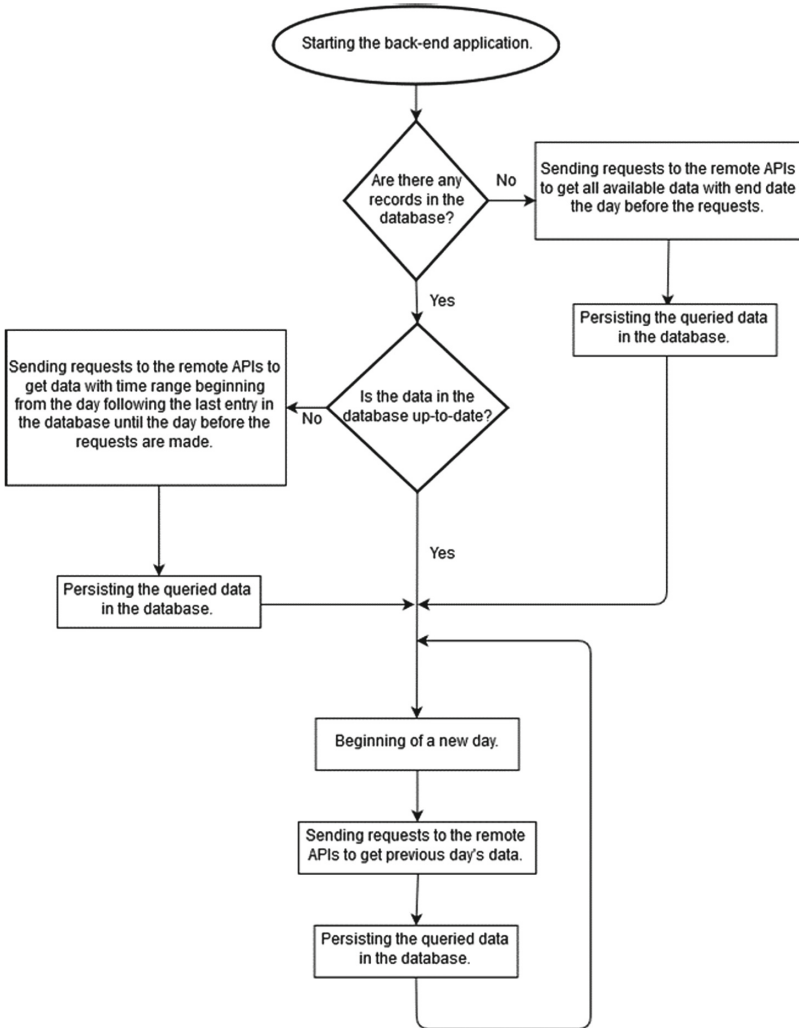


Fig. 3. Workflow of the collector

• Checks for existing records in the database

When the back-end application is started, the collector first checks if any records in the database are available. If the database is empty, the collector sends requests to the remote APIs in order to gather all the missing data. The requested data is with time

range beginning from the first day, for which information is available at the remote APIs, until the day before the request is made. The reason why the end date for the request is not equal to the date, on which the request is made, but the day before, is that the closing price of the monitored cryptocurrency is determined every day at 23:59 by the remote API. Another reason is that the remote API, which exposes ratings, based on news classification, calculates ratings for each hour of the day, but the machine learning algorithm uses the aggregated value for the whole day. After receiving the data, the collector stores it in the database.

- **Checks for up-to-date data**

In case there is data available in the database, the collector checks its current status. If data leaks are detected, the collector queries the remote APIs to collect the missing information. The requested data is with time range beginning from the day, following the last entry in the database, until the day, before the request is made;

- **Collects data for the previous day**

In case of up-to-date data with no missing time intervals in it, the collector queries the remote APIs to provide data from the previous day. After getting a response from the remote services, the collector persists the collected data in the database. The operation is repeated daily.

## 2.2 Machine Learning Algorithm

The machine learning algorithm consists of two parts. The first one is loading the data and transforming it in a suitable format for training the neural network. This process is known as ETL (extract, transform, load). The second part is related to operations with the neural network.

### ETL

#### *Selection of Input Variables According to the Problem Being Solved*

Selection of appropriate input variables is essential in order to achieve accurate predictions. Forecasting market prices, and in particular predicting cryptocurrency prices, is not a trivial problem. Neural networks cannot be successfully trained to predict future cryptocurrency prices only with an input variable, which represents the closing price for the cryptocurrency. The reason for this is that no cyclicity can be observed in the rate of cryptocurrencies. Networks are not capable to predict future jumps or downturns that they have not seen before, without additional parameters to suggest similar upcoming events. On the other hand, news report global events such as wars, financial crises, disasters and more, that strongly affect cryptocurrency trading.

Three additional input variables are chosen in order to overcome the problem – news sentiments. As a result, the input variables used for machine learning are:

- Close price;
- News sentiment;
- Twitter sentiment;
- Reddit sentiment.

### *Transforming the Input Data*

Predicting cryptocurrency prices can be classified as a supervised learning task of learning a function that maps an input to an output, based on example input-output pairs. After transformation the input pairs look as described in Table 2. Each row in the table contains numeric values for one day and the row sequence represents continuous time series. Each row in Table 3 contains a single numeric value, which describes the tomorrow's close price of the cryptocurrency for the day that appears at the same row position in the first table.

**Table 2.** Input data for the neural network

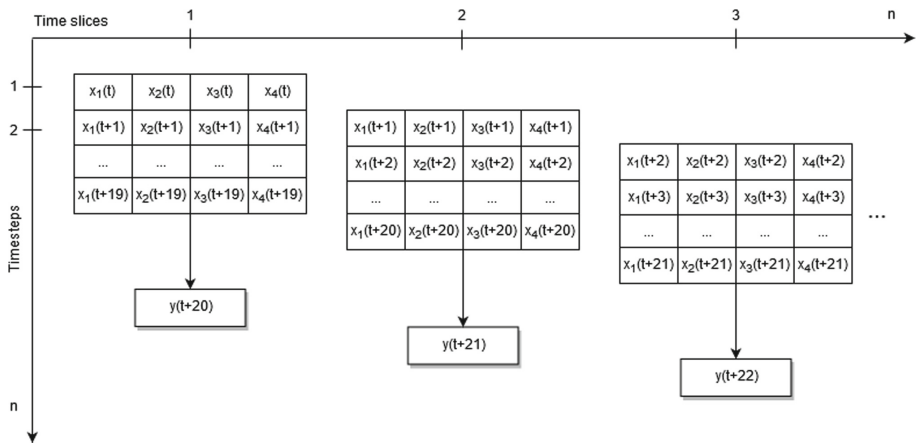
Close	News_sentiment	Twitter_sentiment	Reddit_sentiment
7725.43	7.16717	5.85146	5.00671
7603.99	6.36183	4.24117	4.29825
...	...	...	...

**Table 3.** Output data for the neural network

<b>close</b>
7603.99
7533.92
...

### *Data Modeling*

The next step is to model the data in a way that the LSTM neural network can learn from it and make accurate predictions. The data is transformed into overlapping sequences of 20 days, used as inputs for predicting the 21st day's value, following the respective sequence. During the training process, the 21st day's value acts as a label (after making a prediction for the 21st day's closing price, the real closing price for that day is presented and the network tunes its weights in order to minimize its prediction error). The process of data modeling is presented in Fig. 4.



**Fig. 4.** Organization of overlapping sequences for a period of 20 days

### *Splitting the Data in Training and Test Sets*

The data is split in ratio 8:2 – 80% of the data is used for training the neural network and the rest 20% is used for testing it.

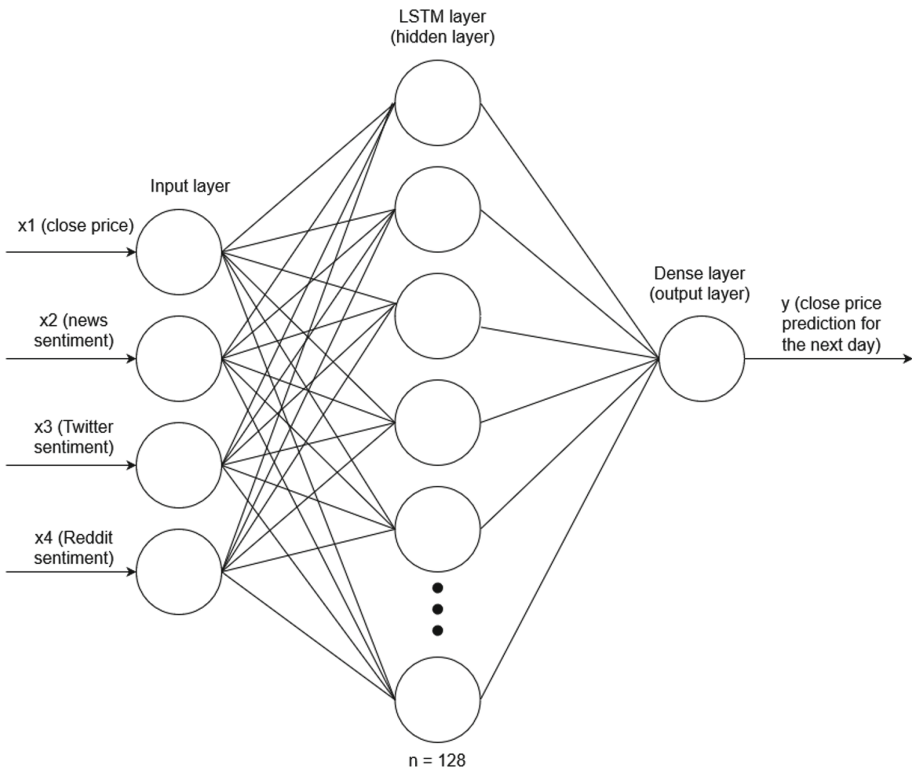
### *Data Normalization*

The purpose of normalization is to transform the values of the input variables in such a way, that they belong to a single numerical range. If the variables belong to different numerical ranges, those whose values exceed many times the values of the others, will have a greater impact on the output. The sentiments have values in the range [1, 10], while the closing price varies in the range [3000, 10000]. Thus, the closing price has a significant advantage over the other variables. The four input variables are of equal importance to the problem being solved, so data normalization is required. Another reason why the data needs to be normalized, is that the neural network is trained by the optimization algorithm of gradient descent, and its activation functions have an active range between  $-1$  and  $1$ .

## **Neural Network**

### *Designing the Neural Network Model*

The neural network consists of a LSTM layer and a dense layer. The LSTM layer is a variation of a recurrent layer with memory. It is able to find long-term dependencies in time series data. The dense layer limits the number of output parameters to one (corresponding to the closing price for the next day) by applying an activation function to the outputs from the previous layer. The activation function for the dense layer is a linear activation function because it allows the neural network to predict higher values than those, with which it was trained. This cannot be achieved by hyperbolic tangent activation function or logical sigmoidal activation function. The input layer of the neural network has 4 neurons - one for each of the input parameters, the hidden LSTM layer - 128, and the output dense layer - 1, which is equal to the number of outputs, due to the regression nature of the problem being solved (see Fig. 5).



**Fig. 5.** Neural network model

### *Hyperparameter Tuning*

Choosing the right hyperparameters' values is essential for successful network training. Poorly selected hyperparameters can lead to slow or failed training. Below are listed the hyperparameters and their respective values.

- **Batch size** – 32
- **Epochs** – 25
- **Iterations per epoch** – 100
- **Learning rate** –  $1.10^{-3}$
- **Activation functions** – The LSTM layer uses a sigmoidal activation function to control the inputs of LSTM cells, because the sigmoidal function has output values in the range from 0 to 1. The activation function for the dense layer is a linear activation function.
- **Loss function** – Mean Square Error (MSE). The error is measured as the arithmetic mean of the sum of the differences between the predictions and the actual observations squared (1).

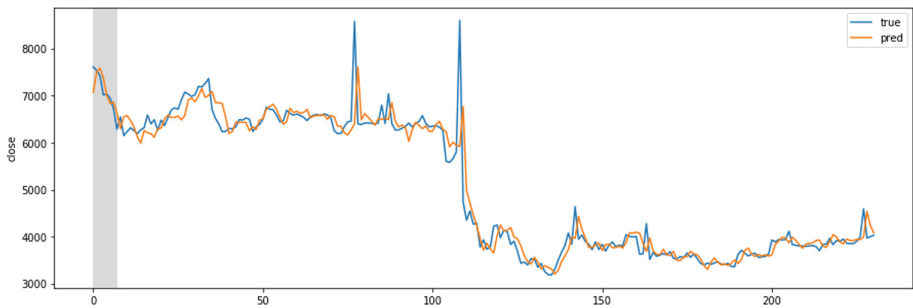
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (1)$$

where  $y_i$  is the prediction, made by the neural network, and  $\hat{y}_i$  is the actual observation.

### *Training the Neural Network*

After the neural network model is configured, the network is trained on the training data.

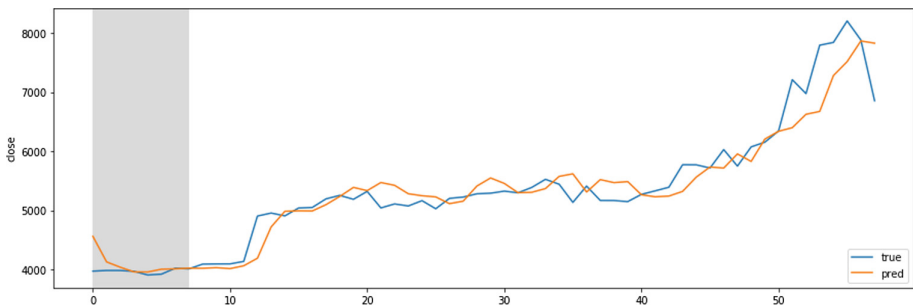
The results from training the neural network with the above hyperparameter values are presented in Fig. 6. The blue line shows the correct outputs and the orange one presents the predictions made by the network during the training process.



**Fig. 6.** Results from training the neural network

### *Testing the Neural Network*

Testing determines how satisfactory the neural network estimates are. The network predicts on data it has not seen before and the predicted values are compared with the correct outputs. The smaller the deviations between the two values are, the better the forecasts are. The results are presented in Fig. 7.



**Fig. 7.** Results from testing the neural network.

### 3 System Implementation

The crypto price prediction system consists of the following software modules:

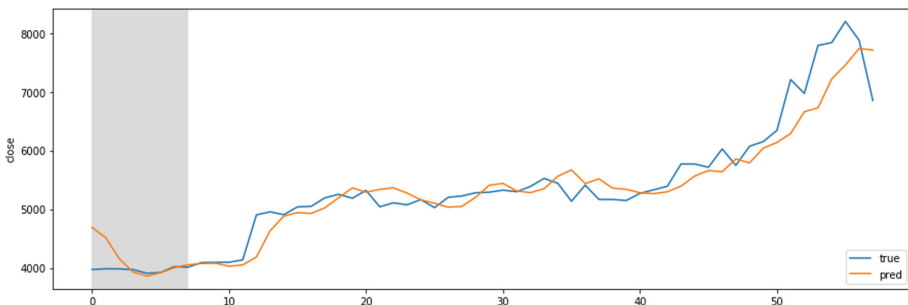
- **Spring Boot application** (back-end)  
Handles data collection (via the collector). For consuming RESTful services is used Spring Boot's HTTP RESTClient, which fetches data from the remote APIs. The collected data is persisted in a MongoDB database and is then converted to CSV files, used in the process of machine learning. The machine learning algorithm is implemented, using the DL4J library for Java. The predictions, made by the LSTM neural network, are persisted in the MongoDB database. A REST API is created for exposing the predictions to the end users of the system. It is authenticated with JWT (JSON Web Token) standard. In order to access secured resources, users must authenticate themselves by providing a valid JWT.
- **MongoDB database** – a non-relational database, which provides dynamic schema design and high throughput and latency.
- **Angular web application** (front-end) – provides high level user experience via interactive diagrams and animations. Requires the users to register themselves and login, in order to get access to the system.
- **Android mobile application** (front-end) – implemented MVVM (Model-View-ViewModel) architecture. Developed using the reactive programming paradigm.

### 4 Experimental Results

#### 4.1 Tuning the Neural Network

The training data is obtained from CoinAPI (<https://rest.coinapi.io/v1/ohlcv/BTC/USD/history>).

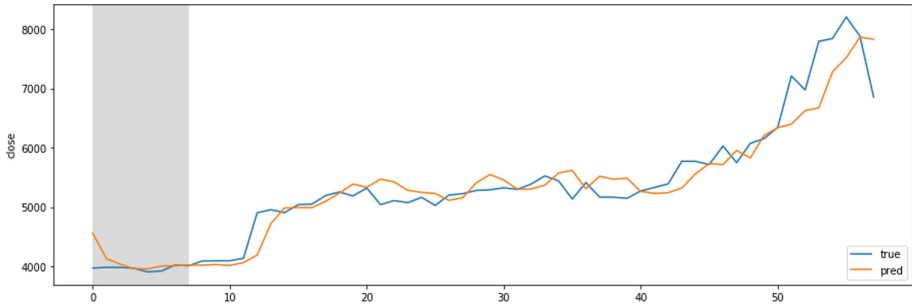
In order to obtain the most accurate predictions for the course of the monitored cryptocurrency, the hyperparameters of the neural network are adjusted until the most satisfactory results are achieved. The following are series of testing the neural network's accuracy with different combinations of hyperparameter values, with which the network was previously trained. The results of the experiments are presented in Fig. 8, 9, 10 and 11.



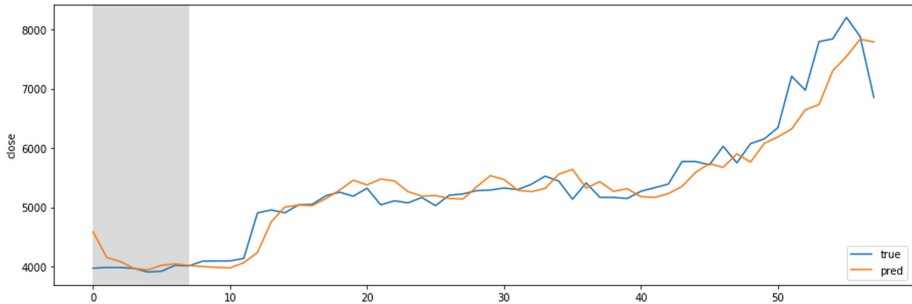
**Fig. 8.** Learning rate:  $10^{-3}$ , batch size: 32, number of neurons in the hidden layer: 64, loss function value: 0.0042



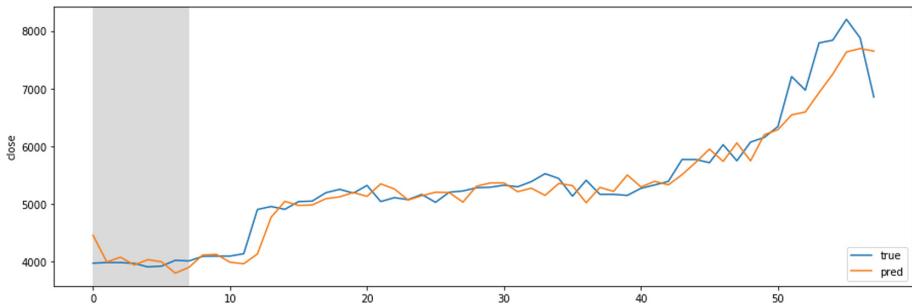
As it can be observed from the experimental results, the neural network’s predictions improve with increasing the batch size and the number of neurons in the hidden layer. The most accurate predictions are shown in Fig. 11 with respective values of the hyperparameters: learning rate:  $10^{-3}$ , batch size: 128, number of neurons in the hidden layer: 512.



**Fig. 9.** Learning rate:  $10^{-3}$ , batch size: 32, number of neurons in the hidden layer: 128, loss function value: 0.0038



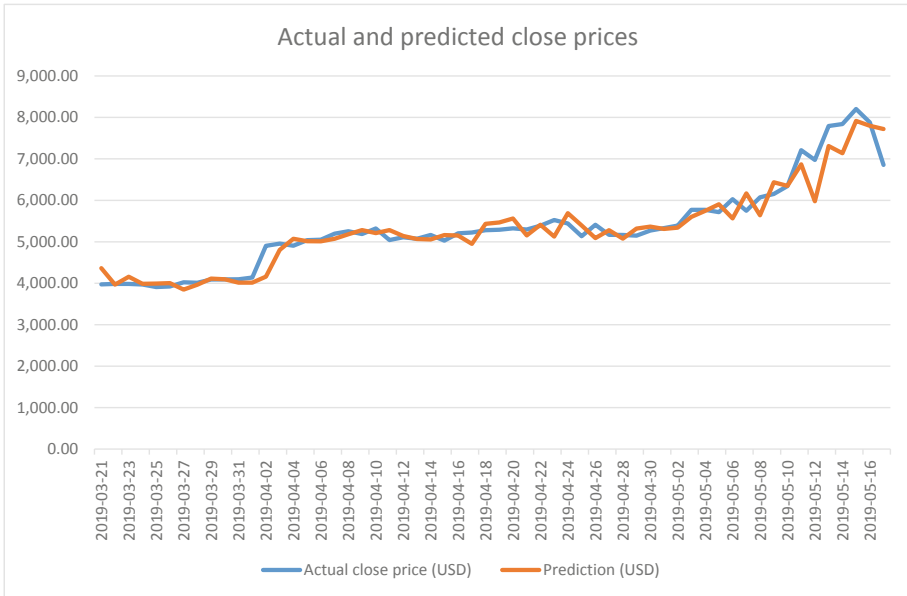
**Fig. 10.** Learning rate:  $10^{-3}$ , batch size: 64, number of neurons in the hidden layer: 512, loss function value: 0.0036



**Fig. 11.** Learning rate:  $10^{-3}$ , batch size: 128, number of neurons in the hidden layer: 512, loss function value: 0.0029

### 4.2 Estimating the Neural Network’s Predictions Accuracy

In Fig. 12 are presented the predictions made by the neural network and the actual corresponding close prices of Bitcoin in the time period 25.03.2019–13.05.2019. After that, a formula for estimating the predictions’ error is presented.



**Fig. 12.** Comparison between actual close prices and predicted close prices

For measuring the predictions’ accuracy of the forecasting algorithm is used the Mean absolute percentage error (MAPE). It expresses the accuracy as a percentage, and is defined by the formula (2):

$$M = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \tag{2}$$

where  $A_t$  is the actual value and  $F_t$  is the forecast value.

After performing the necessary calculations, the error is estimated to be equal to 3,61%, which means that the accuracy of the neural network’s predictions is 96,39%.

### 4.3 Comparison of the Implemented Algorithm’s Accuracy with Other Implemented Crypto Price Prediction System (See Table 4 and Table 5)

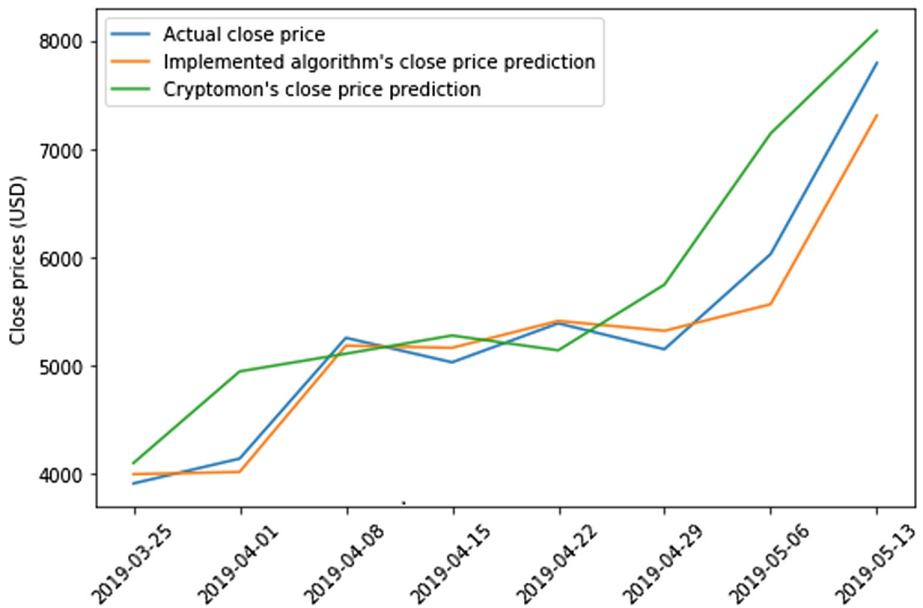
Figure 13 presents a comparison between the actual close prices and the predictions made by the implemented algorithm and the Cryptomon system.

**Table 4.** Comparison between actual close prices and predicted close prices

Date	Actual close price (USD)	Implemented algorithm	Cryptomon
25.3.2019	3907,53	3992,93	4096,08
1.4.2019	4137,00	4013,95	4943,10
8.4.2019	5253,62	5181,36	5106,67
15.4.2019	5027,31	5161,32	5275,06
22.4.2019	5387,60	5409,76	5138,21
29.4.2019	5148,25	5318,54	5745,56
6.5.2019	6027,90	5564,17	7142,76
13.5.2019	7793,47	7309,62	8091,55

**Table 5.** Comparison of estimated errors of both systems

	MAPE (%)
Implemented algorithm	3,353
Cryptomon	8,823

**Fig. 13.** Comparison between actual close prices and predictions made by both systems.

## 5 Conclusion

Cryptocurrency price prediction is always on the top of the list of uses for machine learning and neural network algorithms and it makes a major contribution in crypto trading. This work focuses on the development of project based on the collector (responsible for gathering historical data for cryptocurrencies) and on the machine learning algorithm (responsible for neural network training and for neural network prediction operations). The neural network is designed to study and classify values of hyperparameters. The predictions from the neural network improve with increasing the batch size and the number of neurons in the hidden layer. A comparison between actual close prices and predicted close prices is made for defined test period. The accuracy of the neural network's predictions is 96,39%. The comparison of the accuracy of the implemented algorithm with another implemented crypto price prediction system is presented. The results show that the implemented algorithm produces more accurate predictions of the cryptocurrency price.

It would be interesting for future research to identify and to collect additional data sources. The prediction model would be more effective if the number of monitored cryptocurrencies is increased or more analytic data strategies are developed. In order to improve the prediction process, a parallelization of machine learning algorithms would be effective for training data.

**Acknowledgments.** This paper is partially supported by the National Scientific Program “Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)” (grant agreement DO1-205/23.11.18), financed by the Ministry of Education and Science.

The authors would like to thank the colleagues from the software company DISC (Digital Solutions Consulting GmbH - <http://disc.com.de/>) for the collaborative work on the implementation of the cryptocurrency price prediction system.

## References

1. Abraham, J., Higdon, D., Nelson, J., Ibarra, J.: Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Sci. Rev.* **1**(3), 1 (2018)
2. Alessandretti, L., ElBahrawy, A., Aiello, L.M., Baronchelli, A.: Machine learning the cryptocurrency market. [https://www.researchgate.net/publication/325320374\\_Machine\\_Learning\\_the\\_Cryptocurrency\\_Market](https://www.researchgate.net/publication/325320374_Machine_Learning_the_Cryptocurrency_Market). Accessed 29 Sep 2019
3. Alessandretti, L., ElBahrawy, A., Aiello, L.M., Baronchelli, A.: Anticipating cryptocurrency prices using machine learning. *Complexity* **2018**, 16 (2018). Article ID 8983590
4. Chongyang, B., Tommy, W., Linda, X., Subrahmanian, V.S., Ziheng, Z.: C2P2: a collective cryptocurrency up/down price prediction engine (2019)
5. Indera, N.I., Yassin, I.M., Zabidi, A., Rizman, Z.I.: Non-linear autoregressive with exogeneous input (NARX) bitcoin price prediction model using PSO-optimized parameters and moving average technical indicators. *J. Fundam. Appl. Sci.* **9**(3S), 791–808 (2017)
6. Kim, Y.B., Kim, J.G., Kim, W., Im, J.H., Kim, T.H., Kang, S.J.: Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PLoS ONE* **11**(8), e0161197 (2018)

7. Lamon, C., Nielsen, E., Redondo, E.: Cryptocurrency price prediction using news and social media sentiment. <http://cs229.stanford.edu/proj2017/final-reports/5237280.pdf>. Accessed 29 Sep 2019
8. Li, T.R., Chamrajnagar, A.S., Fong, X.R., Rizik, N.R., Fu, F.: Sentiment-based prediction of alternative cryptocurrency price fluctuations using gradient boosting tree model. *Front. Phys.* **7**, 98 (2019)
9. McNally, S.: Predicting the price of bitcoin using machine learning. Master thesis, Dublin, National College of Ireland (2016)
10. Mittal, R., Arora, S., Bhatia, M.P.S.: Automated cryptocurrencies prices prediction using machine learning (2018). <https://doi.org/10.21917/ijsc.2018.0245>
11. Prashanth, J.R., Vineetha, S.: Cryptocurrency price prediction using long-short term memory model. *Int. J. Res. Sci. Innov. (IJRSI)* **V(VII)** (2018). ISSN 2321–2705
12. Zhengyao, J., Jinjun, L.: Cryptocurrency portfolio management with deep reinforcement learning. *IEEE* (2018)



# Strategic Behavior Discovery of Multi-agent Systems Based on Deep Learning Technique

Boris Morose, Sabina Aledort<sup>(✉)</sup>, and Gal Zaidman

Afeka - Tel-Aviv Academic College of Engineering, Tel Aviv-Yafo, Israel  
borism@afeka.ac.il, sabinaaledort@gmail.com

**Abstract.** Intelligent agents in multi-agent systems may have different strategies to reach the common goal. Discovering these strategies gives an advantage to the opponent systems. Deep Learning algorithms were applied to find parameters of pre-defined types of agent strategies. Different types of models were compared to reach discovery accuracy rate. Numerical experiments show that Deep Learning technique may be successfully applied to discover agent strategies. Simulation shows as well that it may be used effectively to optimize parameters so that strategy discovery of the opponent system will be much more challenging.

**Keywords:** Strategy discovery · Multi-agent system · Deep learning

## 1 Introduction

These days it is common to use complex systems consisting of multiple communicating components that cooperate to reach a common goal. For example, in computer games, each player is an individual component and together they form a multi-agent system of intelligent agents [6]. These agents operate based on chosen strategy to win the game. The strategy is usually hidden from the opponent players, and often are learned and improved throughout the game.

As multi-agent systems become more evolved, their strategy becomes more complex and harder to understand and predict. Classification of the strategic behavior of a multi-agent system in order to know its strategy can have many benefits. It allows users to understand and evaluate a multi-agent system's behavior, determining its strengths and weaknesses [3]. In addition, it allows a competitor facing a multi-agent system to anticipate and counter the system's actions.

Deep Learning (DL) algorithms are commonly recognized as effective tools to find hidden dependencies within a large dataset. In that way, they can learn a given problem, assuming that enough data is provided. DL algorithms are being extensively used by large corporations to learn about their users and improve the performance of their complex systems and are used by governments to predict potential terror attacks.

While research has been concentrated on discovering strategic plays from multi-agent actions [1], and on using DL for predicting human strategic behavior in normal games [2], an extensive search of leading academic resources suggests that research

into whether DL can be effective in discovering and classifying strategic behavior of multi-agent systems still remains to be done.

The main purpose of this study was to examine applications of Deep Learning techniques to strategic behavior discovery and classification.

The paper is organized as follows: first we describe the research environment and the algorithms which were chosen for this paper, following presentation of results and conclusions from multiple numeric experiments.

## 2 Strategic Behavior Discovery

The research environment consists of certain number of agents and a moving target. Each agent has a predefined strategy which it follows during the simulation. Three different parameter-based strategies were defined:

- The agent calculates the target’s estimated path and moves towards the next estimated target’s location.
- The agent evaluates who from all the agents in the environment can reach the target in the shortest time. If this agent is the closest one, it moves towards the target’s estimated next location. If not, the agent keeps moving in its current direction.
- The agent calculates who from all the agents in the environment can get to the target in the shortest time. If this agent is the closest one, it moves towards the target’s estimated next location. If not, the agent remains in its current location.

Each of those strategies depends on the following parameters:

*Acceleration factor* - The agent’s acceleration magnitude is multiplied by this factor to make its velocity increase or decrease.

*Direction factor* - The agent’s direction is multiplied by this factor to change its direction.

*The number of time frames to wait* - The agent can stay in the same location for a few time frames.

The environment consists of few interconnected modules, as shown in Fig. 1.

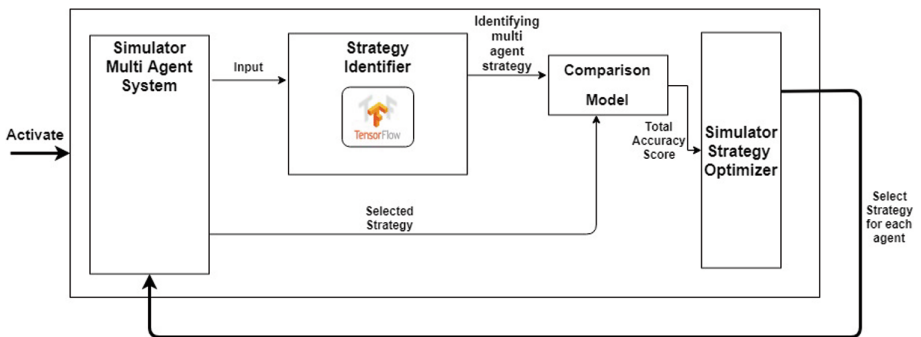


Fig. 1. System architecture block diagram.

## 2.1 Multi-agent System Simulator

This module simulates a group of agents that move in a predefined area with a set of parameter-based strategies. The goal of the agents is to catch a moving target within a predefined time frame. The simulator creates the dataset for the further processing of DL algorithms. A pre-configured number of simulations is performed for a set of strategies. Once all the simulations are created, the dataset is passed to the strategy identifier.

## 2.2 Strategy Identifier

The Identifier consists of a set of Deep Learning models. All models are trained on the given dataset created by the simulator and produce a prediction for each agent about its associated strategy. The predictions and the known values are passed to the Comparison Module which calculates the overall average accuracy score for all the predictions. Once the accuracy score is calculated it is passed to the Simulator Strategy Optimizer which creates a feedback loop between the strategy identifier and the multi-agent system. Table 1 summarizes comparison of three different DL models:

- *GoogleNet model* – Selected because of its ability to work effectively with a large number of parameters. It has only 4 million parameters thanks to usage of the inception models [4].
- *VGGNet model* – Chosen for its simplicity. It is a very deep convolutional neural network that has been proven to produce very good results [5].
- *SimpleVGG model* – 4 convolutional layers and 2 fully connected layers, with a total of 21 million parameters. Uses asymmetric filters with the first layer of  $5 * \{\text{number of axis}\}$  size. This model samples each agent individually and better fit to the dataset. We proposed this model to produce better results for the problem with fewer parameters to provide faster training.

To make a prediction for each agent possible, the final layer for each model is a multi-head node with each head wrapped with a softmax function to generate a prediction for a specific agent.

## 2.3 Simulator Strategy Optimizer

The multi-agent system operates with a given set of parameter-based strategies. The optimizer selects strategy parameters based on results of previous iterations. On each iteration the Optimizer evaluates the time DL needs to discover the strategy and choose new strategy parameters for the next iteration. The last are chosen by analyzing accuracy scores with intent to make the discovery harder for the next iteration.



**Table 1.** Summary of the models used

GoogleNet	VGGNet	SimpleVGG
$7 \times 7$ Convolution layer	$3 \times 3 \times 64$ Convolution layer	$5 \times \text{*{number of axis}} \times 96$ Convolution layer
$3 \times 3$ max pool layer	$3 \times 3 \times 64$ Convolution layer	$5 \times 5 \times 256$ Convolution layer
$3 \times 3$ Convolution layer	$2 \times 2$ max pool – stride of 2	$2 \times 2$ max pool – stride of 2
$3 \times 3$ max pool layer	$3 \times 3 \times 128$ Convolution layer	$2 \times 5 \times 384$ Convolution layer
inception layer	$3 \times 3 \times 128$ Convolution layer	$2 \times 5 \times 384$ Convolution layer
inception layer	$2 \times 2$ max pool – stride of 2	$2 \times 2$ max pool – stride of 2
inception layer	$3 \times 3 \times 256$ Convolution layer	FC-1024
inception layer	$3 \times 3 \times 256$ Convolution layer	FC-100
inception layer	$3 \times 3 \times 256$ Convolution layer	
inception layer	$2 \times 2$ max pool – stride of 2	
inception layer	$3 \times 3 \times 512$ Convolution layer	
inception layer	$3 \times 3 \times 512$ Convolution layer	
inception layer	$3 \times 3 \times 512$ Convolution layer	
	$2 \times 2$ max pool – stride of 2	
	$3 \times 3 \times 512$ Convolution layer	
	$3 \times 3 \times 512$ Convolution layer	
	$3 \times 3 \times 512$ Convolution layer	
	$2 \times 2$ max pool – stride of 2	
	FC-4096	
	FC-4096	
	FC-1000	

### 3 Results

In this research we wanted to understand the efficiency and the boundaries of different DL models to discover the strategies of multi-agent systems. The numerical experiments were provided to measure accuracy of discovery and to optimize strategy parameters to challenge the discovery system. To find specific strategies that challenge the discovery system we performed optimization using different direction and acceleration parameters to find the lowest accuracy rate.

Figure 2 contains the measured performance rates of the VGGNet model. The VGGNet model consists of 138 million parameters. It takes an average of 2 h to train each run on 50k training examples, on a 4k GPU. The results indicate that the VGGNet model recognized the right strategy in 99% of the times, in a setup of 3 agents operating with 3 strategies. The lowest accuracy rate of this model is 48%, in a setup of 10 agents

operating with 10 strategies. The VGGNet model reached 70% accuracy in setups of 3 to 9 agent operations with 3 to 6 strategies.

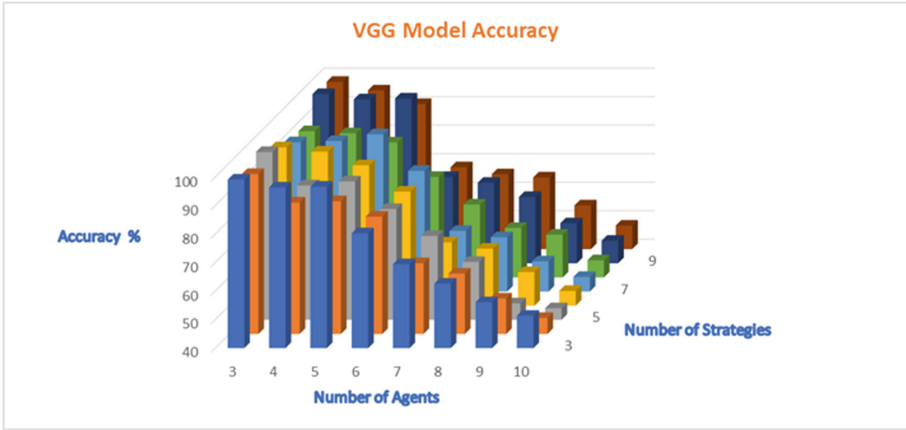


Fig. 2. Performance rates of the VGGNet model

The SimpleVGG model, similarly to the VGGNet model, recognized the right strategy 99% of the time, in a setup of 3 agents operating with three strategies, as described in Fig. 3 below. The lowest accuracy rate of this model is 47%, in a setup of 10 agents operating with 10 strategies. This model consists of 21 million parameters and it takes an average of 1.5 h to train each run on 50k training examples, on a 4k GPU.

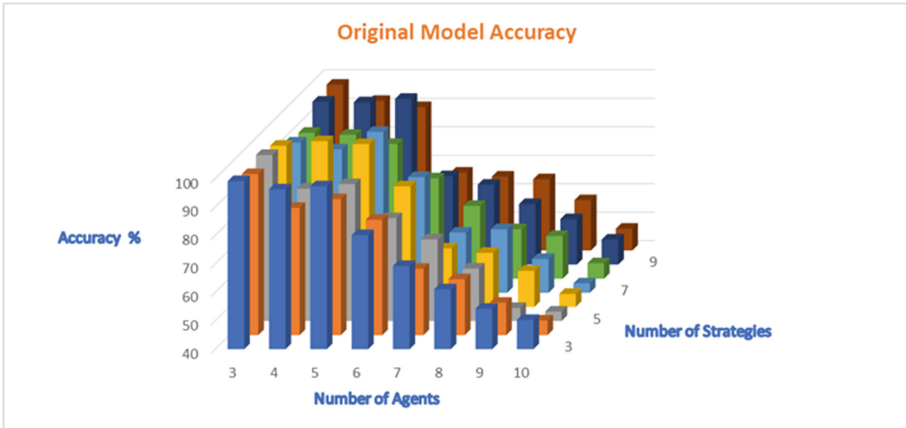


Fig. 3. Performance rates of the SimpleVGG model

Contrary to the VGGNet model, and the SimpleVGG model, the highest accuracy rate the GoogleNet model was able to reach was 85%. The accuracy rates of this model are shown in Fig. 4. In a setup of 10 agents operating with 10 strategies, the GoogleNet

model achieved an accuracy rate of only 17%. The GoogleNet model consists of 21 million parameters and it takes an average of 1.5 h to train each run on 50k training examples, on a 4k GPU.

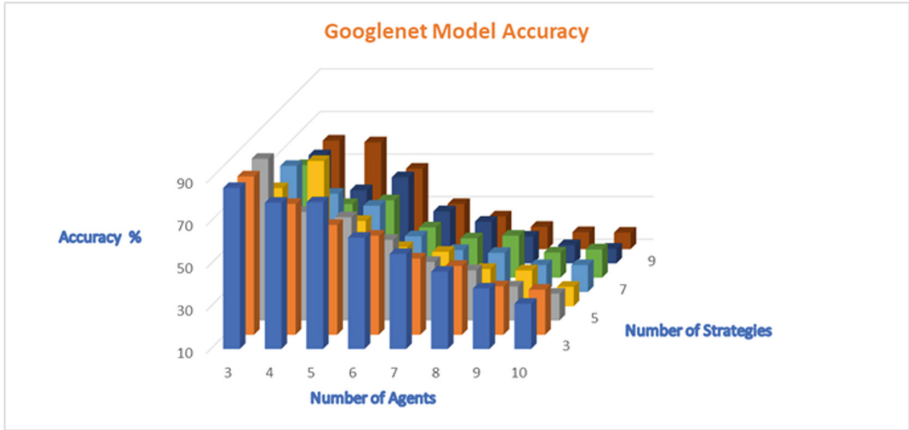


Fig. 4. Performance rates of the GoogleNet model

Figure 5 below shows the optimization process of the direction factor parameter in a configuration of 6 agents using 6 strategies. The lowest algorithms' accuracy was found after 20 iterations.

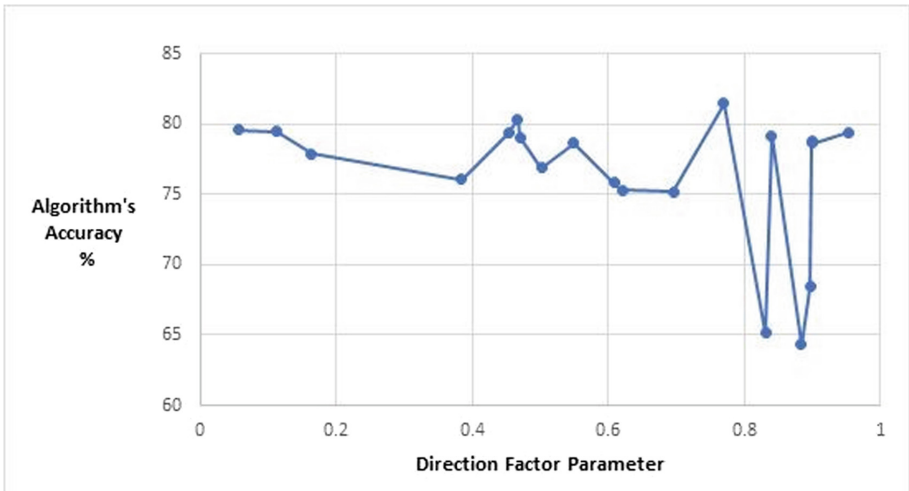


Fig. 5. Discovery accuracy optimized with direction factor parameter

Figure 6 describes the optimization process of the acceleration factor parameter, in the same setup as mentioned above. The value of this parameter was 0.3, when the algorithms' accuracy rate was the lowest (64%).

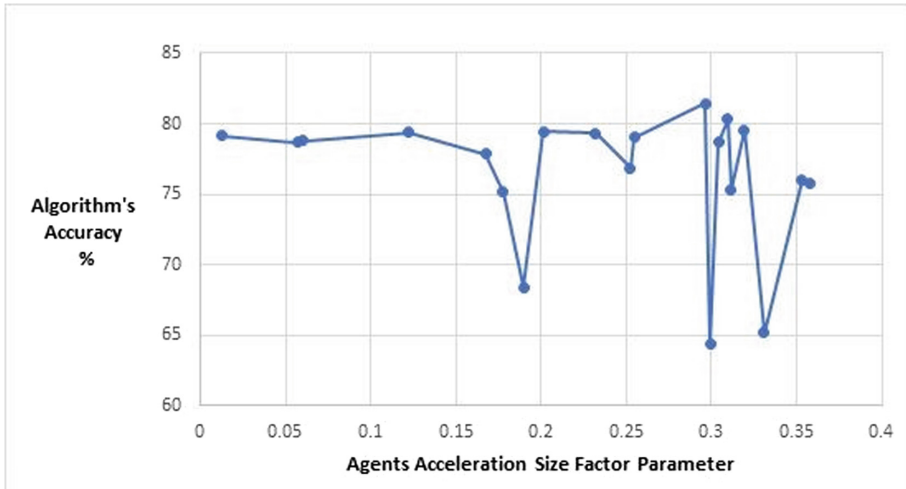


Fig. 6. Discovery accuracy optimized with size factor parameter

## 4 Conclusion

This study investigates the implementation of Deep Learning techniques to strategic behavior discovery of multi-agent systems. Provided numeric experiments were limited to 10 agents and strategies. The results indicate that the algorithm manages to classify the strategies of 3–9 agents operating with 3–6 strategies successfully. We found that the number of strategies has a great influence on the algorithm's performance, while the number of agents has little influence. Parallelization of the testing and training processes was a very efficient way to make massive numerical experiments. The proposed approach successfully found strategies whose discovery rate with DL do not exceed more than 64.33%.


The results show that between the proposed SimpleVGG model and the GoogleNet model there is a big difference in performance, in all the configurations. The unique inception model doesn't perform as well as we assumed it would. However, between VGGNet network and the SimpleVGG model there is almost no difference in terms of performance even during tests with many agents/strategies. This is surprising as VGGNet consists of almost 6 times more parameters than the SimpleVGG model. From a certain point the increase in parameters and convolution layers (the depth of the neural network) does not help the algorithm to achieve better performance. However, the number of agents and strategies has significant influence on the algorithm's results, as the accuracy of the algorithm decreases when the number of agents and strategies increase.

## References

1. Mirchevska, V., Lustrek, M., Bezek, A., Gams, M.: Discovering strategic behavior of multi-agent systems in adversary settings, *Comput. Inform.* **33**, 79–108 (2014)
2. Hartford, J.S., Wright, J.R., Leyton-Brown, K.: Deep learning for predicting human strategic behavior. In: *Advances in Neural Information Processing Systems* (2016)
3. Kaminka, G., Fidanboyly, M., Chang, A., Veloso, M.M.: Learning the sequential coordinated behaviour of teams from observations. *robocup: robot soccer world cup VI. Lecture Notes in Artificial Intelligence* **2003**(2752), 111–125 (2002)
4. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going Deeper with Convolutions, *Arxiv* (2014)
5. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, Visual Geometry Group, Department of Engineering Science, University of Oxford, *arXiv* (2014)
6. Russell, S.J., Norvig, P., Davis, E.: *Artificial Intelligence: A Modern Approach*, 3rd edn. Prentice Hall, Upper Saddle River (2010)



# Development of Prediction Methods for Taxi Order Service on the Basis of Intellectual Data Analysis

N. A. Andriyanov<sup>(✉)</sup> 

Ulyanovsk State Technical University, Severny Venets 32, 432027 Ulyanovsk, Russia  
nikita-and-nov@mail.ru

**Abstract.** The work considers the urgent task of collecting and analyzing information received during the work of the taxi order service. The data obtained by the taxi service can be easily represented by different time series. Particular attention is also paid to the use of neural networks to solve the predicting problem. The relevance of using neural networks in comparison with statistical models is substantiated. The special software used allows one's to collect information on the operation of the service in a variety of SQL tables. Particular attention is paid to existing programming languages that allow to implement data mining processes. The strengths and weaknesses are highlighted for this languages. Based on the accumulated data on the numbers of taxi service orders, the algorithms for predicting the operation of a taxi service were studied using both neural networks and mathematical models of random processes. Comparative predicting characteristics are obtained, variances of predicting errors are found. The results of construction using autoregressive and doubly stochastic models, as well as using fuzzy logic models, are presented. It is shown that the use of neural networks provides smaller errors in predicting the number of taxi service orders.

**Keywords:** Data mining · Fuzzy logic · Mathematical modeling · Prediction · Taxi order service

## 1 Introduction

Currently, there is a rapid introduction of information technology in almost all areas of economic and social activity. One of the most important areas of research is data mining [1–3]. The tasks of the intellectual analysis of multidimensional data are of particular relevance. It is clear that the results of such an analysis greatly facilitate the work of a person both in solving various applied problems and in scientific research. At the same time, the number of tools providing such processing is growing. Indeed, modern computer technology allows us to accumulate huge amounts of data. In addition to high-quality processing of such material, such as, for example, when solving image processing problems [4–6], it is also necessary to be able to analyze the available information. Such an analysis can often be associated with the prevention of critical situations in the operation of a technical object. However, the introduction of information technology in the field

of transport activity began relatively recently. In this regard, a qualitative description of the data on the work of the taxi order service has not yet been received. Nevertheless, attempts were made to analyze the data of taxi order services [7–12]. However, they are all local in nature. And the introduction of existing software into existing systems that would perform real-time analysis and signal the presence of overloads or lack of drivers has not yet been implemented. Thus, it is relevant to study the algorithms of data mining of a taxi order service and implement them in a software package that provides a taxi order service.

## **2 Neural Networks for Prediction Tasks**

In the current economic situation and a sharp increase in the pace of development of science and technology to obtain effective profits in the market, the issues of planning and decision-making based on predicting are becoming increasingly relevant. In this regard, the task of predicting time series is urgent, because in the conditions of a market economy, an enterprise needs to study data on the state of activity in the past in order to assess future conditions and results of work. If an enterprise does not clearly and efficiently analyze and predict the economic indicators of its activities, constantly collect and accumulate information on both environmental factors and its own prospects and opportunities, then it will not be able to achieve stable success. Using various prediction methods, we can draw conclusions about the financial condition of the enterprise, about the current situation in the market of goods and services. Until recently, statistical methods remained the main methods for predicting time series. However, the mathematical models associated with these methods [13, 14] are not always linear, and therefore they cannot predict complex phenomena and processes in which the data model may be nonlinear. In these cases, the apparatus of neural networks comes to the rescue. A neural network is a mathematical tool that allows you to simulate various kinds of dependencies, examples of which are linear models, generalized linear models, nonlinear models. The ability to model nonlinear relationships is the main advantage of neural networks. The ability of a neural network to generalize and highlight hidden relationships between input and output data leads to the ability of a neural network to predict. A trained neural network is able to predict the future significance of some factors that currently exist on the basis of their previous values. To predict future values, it is necessary to prepare data for training and testing the network, select the topology, basic characteristics and training parameters of the neural network.

## **3 Architecture of Software Package for the Taxi Order Service**

For organization of taxi order service it is possible to consider a project based on a contact center. At the same time, telephony is delivered to operators directly via the Internet, which requires only a computer with a headset. For the organization of a taxi dispatch, a powerful hardware and software complex is needed. Its application allows several thousand cars to work in real time.

Obviously, the use of this technology allows to effectively manage resources, increase the speed of processing orders, always have actual customer numbers, and reduce the time for receiving applications.

For the contact center to work, a multi-channel phone number is required, which will allow taxi service to receive many calls at the same time. To do this, it is necessary to use the technology of IP-telephony. One of the most common telephony servers is the Asterisk server, which allows one's to work with SIP telephony. Such a telephone exchange should be configured to distribute calls to taxi service operators. The call is processed using a special program that provides the operator with a taxi order form based on an Internet browser. To store information about calls, a database server is used, for example, MySQL. Tariffs are set using a separate module having the name Tariff, which is programmed for its use on the web.

Thus, it is advisable to use virtualization methods to separate various servers, including a telephony server, a telephony database server and a web server. In addition, an application server is also needed, through which information is transferred from the contact center to the drivers. This is provided by a special taxi program. And here it is recommended to use another database server to store information on orders.

Figure 1 shows the full architecture of the considered service.

The application for the Taxi program can have versions that work just under java (for old devices), or oriented to modern devices running Android and iOS.

With the acceptance of the order by a specific driver, the database is updated. For example, information about the car, time of taking the order, etc. is recorded. This can be used to inform the client about the assigned car.

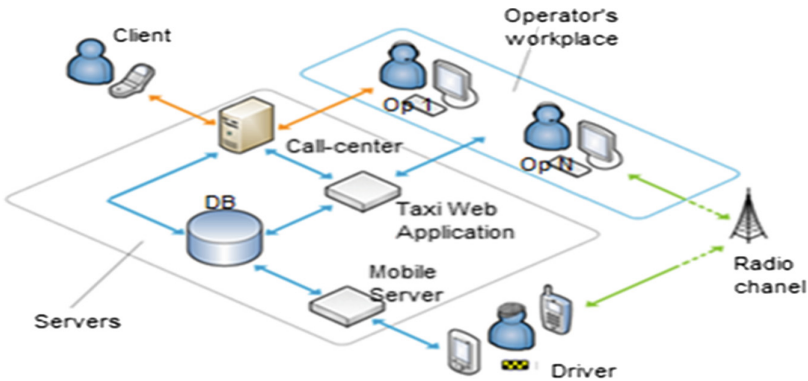
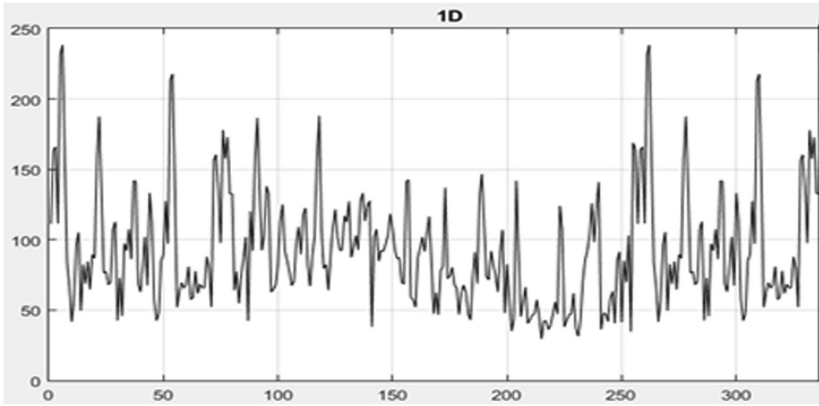


Fig. 1. The structure of the taxi order service

Statistics are collected using database servers, however, the information is presented in a convenient form using the Tariff module, which allows to display statistics either in a text document or in an excel format document. Figure 2 presents the processed information on the distribution of orders, preserving the properties of a real sequence.



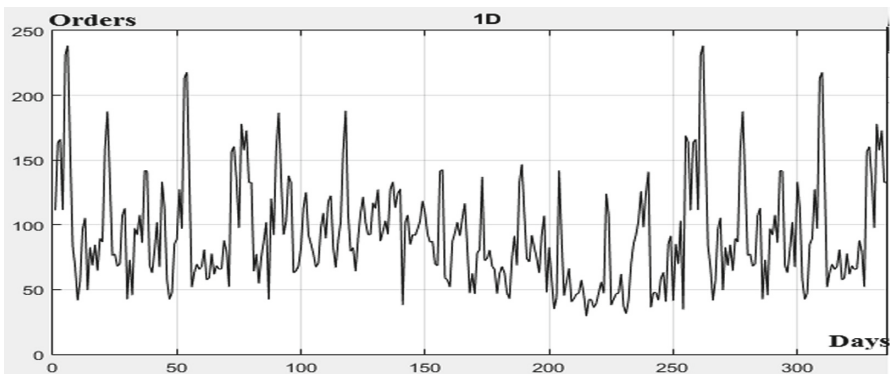


**Fig. 2.** Distribution of orders by days with conversion (the number of orders is postponed on the Y axis, days during the year are postponed on the X axis)

It should be noted that the process in Fig. 2 has a heterogeneous structure, as well as some periodic features. Therefore, it is necessary to select the most adequate model in order to more accurately describe all characteristics of the distribution.

#### 4 Presentation of Taxi Service Data

Given the analysis, it is necessary to predict the sequence shown in Fig. 3 by mathematical models of random processes and models of fuzzy logic. It should be noted that taxi service data is confidential that's why it is necessary to transform this sequences. However the transformation preserve all main properties in covariance and ranges.



**Fig. 3.** Orders distribution for all days in year (from January, 1<sup>st</sup> to December 31<sup>st</sup>)

Table 1 presents possible models for predicting. It is worth noting that in the doubly stochastic model there is a change in correlation parameters. The doubly stochastic model is considered in more detail in [15, 16].

**Table 1.** Data representation models for describing taxi service

Model	Equation
Autoregressive model	$O_i = \rho O_{i-1} + \xi_i, i = 1 \dots N$
Doubly stochastic model	$O_i = \rho_i O_{i-1} + \xi_i, i = 1 \dots N, \rho_i = \tilde{\rho}_i + m_\rho$ $\tilde{\rho}_i = r \tilde{\rho}_{i-1} + \sqrt{\sigma_\rho^2(1 - r^2)} \zeta_i$

To adjust the resulting time series, it is possible to use the mathematical models presented in Table 1. In this case, it is worth to divide the existing time series into 344 and 21 values. In the first group, the models will be trained, and the second group is test in the sense of prediction by the model. In order to estimate the parameters of non-neural network models (Table 1), the Yule-Walker equations or pseudo-gradient search [17] may be used.

However, models based on machine learning have also proved their worth. In particular, the forecasting algorithms from Table 1 can be expanded using fuzzy logic models. In this paper, we choose fairly simple knowledge bases. The first knowledge base refers to a model of the Mamdani type, the second knowledge base refers to a model of the Sugeno type. These models differ in that the fuzzy inference of Mamdani will consist of fuzzy terms at the output and input, and the fuzzy conclusion of Sugeno is constructed in such a way that the output variables (inference) appear to be a functional dependence on the input variables. To describe Mamdani’s knowledge base, the following method should be considered [18]:

$$\begin{aligned} &\text{IF } (x_1 = a_{i1} \ \& \ x_2 = a_{i2} \ \dots \ x_n = a_{in}), \\ &\text{THEN } y = d_i, \text{ having weights } w_i, \quad i = \overline{1, N}, \end{aligned} \tag{1}$$

where  $a_{ij}$  is fuzzy term that is used for a linguistic variable to evaluate factor  $x_j$  in the  $i$ -th decision rule,  $i = \overline{1, N}, j = \overline{1, n}; N$  is number of knowledge base rules;  $d_i$  is the consequent for  $i$ -th rule represented in the form of a fuzzy term;  $w_i \in [0; 1]$  is weight of the  $i$ -th rule, which reflects the confidence of the expert in its reliability.

To build a fuzzy model of the Mamdani knowledge base, one should use the  $c$ -means algorithm adapted for fuzzy sets. Then the task of dividing the available data into classes (learning without a supervisor) is easily formulated on the basis of a characteristic function. It is clear that the values of such a function cannot be less than 0 and more than 1, since it reflects the probability that a particular sample reference belongs to any of  $c$  classes. It is further proposed to describe the fuzzy cluster separation matrix based on the characteristic function in the following convenient form  $F = [\mu_{ki}]$ , where  $\mu_{ki} \in [0, 1], k = \overline{1, M}, i = \overline{1, c}$ . Here the  $k$ -th is the characteristic parameter of the probability of belonging of the element  $X_k = (x_{k1}, x_{k2}, \dots, x_{kn})$  to classes  $A_1, A_2, \dots, A_c$ . Matrix  $F$  should be constructed so that it will be met the following requirements [18]:

$$\begin{aligned} &\sum_{i=\overline{1,c}} \mu_{ki} = 1, \quad k = \overline{1, M}, \\ &0 < \sum_{k=\overline{1,M}} \mu_{ki} < M, \quad i = \overline{1, c}. \end{aligned} \tag{2}$$

The point of using fuzzy clustering based on *c*-means is to calculate matrices *F* and class centers by successive iterations. In this case, classes can be called clusters for convenience, since they denote a region on a data set. The scattering criterion or, conversely, the closest proximity is used as an objective function. The task of building the Mamdani knowledge base is to minimize the objective function, as in many other machine learning algorithms [19]:

$$\sum_{i=1,c} \sum_{k=1,M} (\mu_{ki})^m \|V_i - X_k\|^2 \rightarrow \min, \tag{3}$$

where  $V_i = \frac{\sum_{k=1,M} (\mu_{ki})^m X_k}{\sum_{k=1,M} (\mu_{ki})^m}$  are centers of fuzzy clusters;  $m \in (1, \infty)$  is exponential weight.

Abnormal points for the cluster are calculated based on the Euclidean distance. After that, there are functions responsible for classifying the output variables as defined in the cluster based on the rules of the knowledge base. Input variables are clustered in the same way.

As noted earlier, the Sugeno knowledge base model, unlike the Mamdani knowledge base, makes the value of the output variables dependent on the input. In general, the Sugeno model can be represented as follows [20]:

$$\begin{aligned} \text{IF } (x_1 = a_{i1} \ \& \ x_2 = a_{i2} \ \dots \ x_n = a_{in}), \\ \text{THEN } d_j = b_{j0} + \sum_{i=1,n} b_{ji}x_i, \end{aligned} \tag{4}$$

where  $b_{j0}, b_{j1}, \dots, b_{jm}$  are some real numbers.

Usually the majority of fuzzy knowledge bases are developed on the basis of clustering methods. Examples of such algorithms are subtractive clustering or *c*-means algorithm for fuzzy sets. The synthesis of a fuzzy Sugeno knowledge base, when there is a set or sequence of data, takes place in two main stages. First, you need to choose a knowledge base model architecture that can apply IF-THEN statements using subtractive clustering to all input parameters and their totality. Then it is necessary to evaluate the parameters of the model for generating output variables based on the machine learning algorithm ANFIS. The full name of the ANFIS algorithm is a network-based adaptive fuzzy inference system. During the evaluation of parameters, they are tuned to the membership functions, and the weights of the rules from the fuzzy knowledge base are also adjusted. Obviously, fuzzy clustering can provide identification of characteristic similar data groups (clusters) among a huge set of information. This is important because explicit model training is not provided. Moreover, this approach allows the identification and construction of the structure of a fuzzy knowledge base in an acceptable time. The subtractive clustering algorithm also ensures that the model independently selects the optimal number of clusters. Subtractive clustering is based on the following steps. First, samples of the initial set are considered, which could be used as cluster centers. Then, the probability is calculated that the current element is indeed the center of the cluster. Further, as a result of such division, it is necessary to find the point with the

greatest potential among the given centers of the clusters. In this case, the deduction of the contribution of the newly discovered cluster is taken into account. Finally, an iterative procedure is implemented that provides the calculation of other potentials and the search for cluster centers. Iterations stop when the maximum potential exceeds the threshold selected at the start of clustering [20]. It should be noted that the subtractive clustering algorithm does not actually belong to the class of fuzzy algorithms, but it is often used in systems that allow you to independently (without a supervisor) generate fuzzy rules for any data set.

Given the above, it is possible to identify fuzzy knowledge bases which use subtractive clustering. Moreover, such a method, firstly, creates clusters based on the data space that includes the most similar data samples, and, secondly, provides a knowledge base with a set of fuzzy rules. Applying these fuzzy rules to new arbitrary data, you can determine which cluster they belong to. Otherwise, they can create a new cluster. When the cluster membership probabilities are reflected in the entire input space, some approximation is performed. On its basis, the membership functions of data set elements to knowledge base clusters are calculated. The clustering procedure is based on fuzzy rules. To make the rule inference effective, the least squares method is used.

Thus, the second stage in the Sugeno model involves the use of the ANFIS algorithm with further adjustment of the parameters within the developed knowledge base. Since the conclusion is presented as a formulation of a neuro-fuzzy model, training algorithms for neural networks, for example, with the back propagation of error, are easily applicable to such a base. By its architecture, the ANFIS network is isomorphic to a fuzzy knowledge base. This network consists of five fully connected layers. In this case, the signal in the network spreads only directly. Each layer has its own task. The first layer is required to accept the input. The second layer operates using fuzzy rules. In the third layer, the input and responses of the second layer are normalized. The fourth layer is used to form fuzzy logical conclusions. Finally, on the fifth layer, all the results are complexed and the network output signal is given. The ANFIS network uses a classic combination of back-propagation error algorithm and least squares method for training.

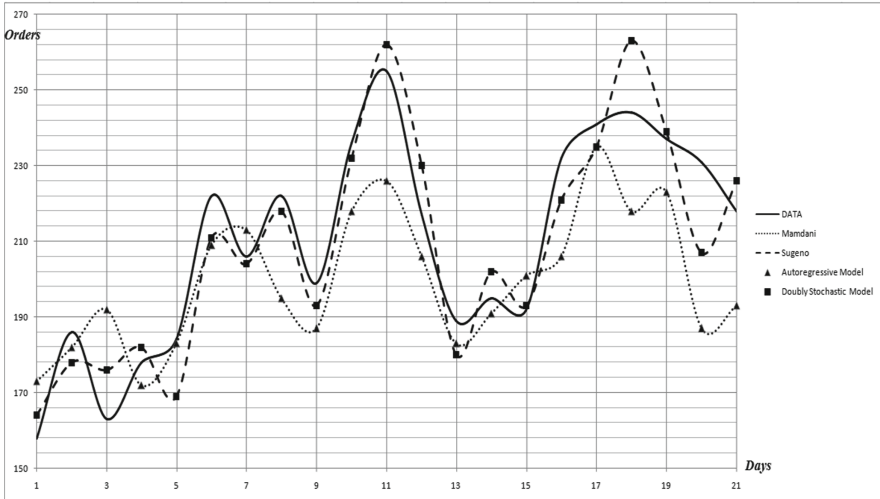
## 5 Prediction Efficiency Analysis

Using the programming languages Matlab and Julia, predictions were simulated based on autoregressive and doubly stochastic random sequences, as well as using fuzzy Mamdani and Sugeno sets. The relative variances of prediction errors of the last twenty-one values, respectively, are following:

- 1) variance of predicting error when using autoregressive model is 0.718;
- 2) variance of predicting error when using doubly stochastic model is 0.381.
- 3) variance of predicting error when using Mamdani fuzzy logic model is 0.304;
- 4) variance of predicting error when using Sugeno fuzzy logic model is 0.206.

Figure 4 shows the predictions themselves and the real data. The solid line characterizes the initial number of orders (converted value), the dashed line shows the prediction based on the Mamdani model, the dashed line shows the prediction based on the Sugeno

model, the predicted values using the autoregressive random process are marked with a triangular marker, and the predicted values using the doubly stochastic random process are marked with a square marker.



**Fig. 4.** Prediction of the number of taxi service orders by mathematical models of random processes and fuzzy logic models

Analysis of Fig. 4 shows that the use of a doubly stochastic model provides a more accurate prediction than conventional autoregression. It should also be noted that fuzzy logic provides more efficient predictions. Moreover, a number of interesting features inherent in fuzzy logic predictions can be highlighted. First, the variances of prediction errors based on fuzzy logic models do not differ as significantly as the variances of errors for predictions based on autoregressive processes. Secondly, both models of fuzzy logic provide a smaller variance of error in prediction.

To study the pricing algorithm based on neural networks, the following factors were selected as significant factors: order time, probability of rainfall, minimum trip cost, number of free cars in the order area, trip distance. Accordingly, a control action was formed at the output that affected only the minimum cost of the trip in these conditions, as well as the price of the trip itself. Decisions were made by taxi order managers. Figure 5 shows the dependence of the order price distribution depending on distance.

Several neural networks with back propagation of errors were trained to predict the control action. Despite the fact that general regression networks provide absolute accuracy in the training set, their application in the test set leads to significant errors. A network based on 5 neurons best approximates data on the formation of a control action. Figure 6 shows the forecast of the control action of the test sample: the solid (red) line is the real value, the dashed (blue) line is the worst network, and the crosses are the best network.

Thus it is possible to predict control action for taxi service order using neural networks.

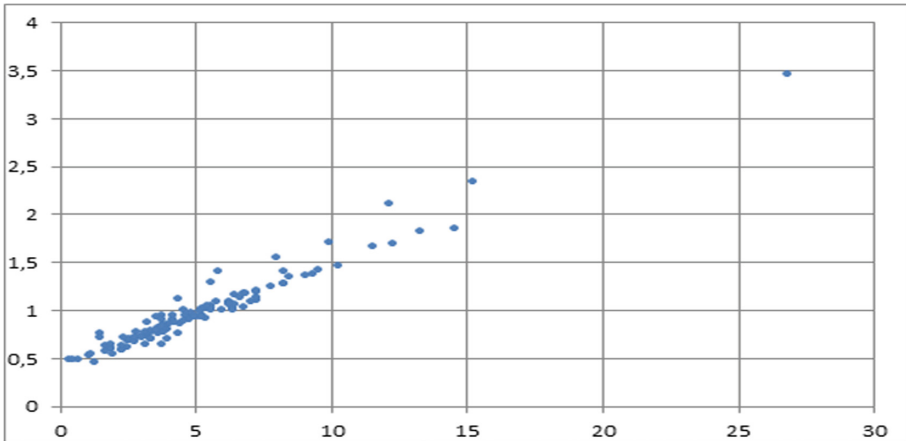


Fig. 5. Price distribution for different distances

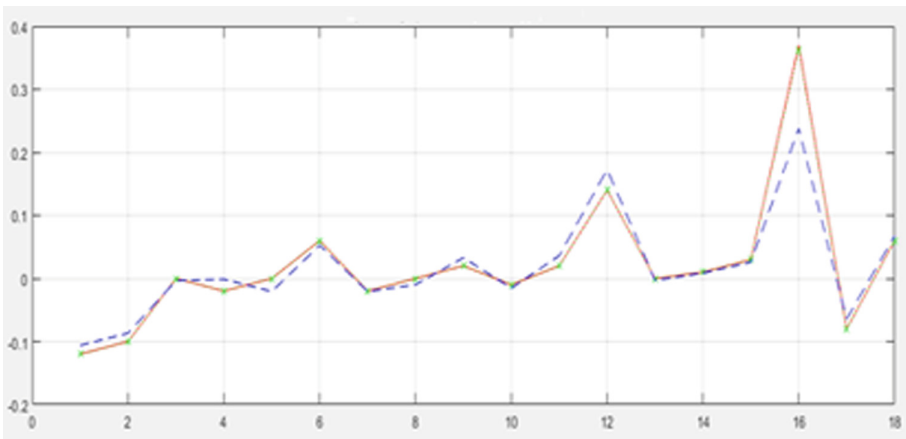


Fig. 6. Control action prediction

## 6 Conclusion

The main results of the study are as follows:

- 1) The programming languages for data mining were analyzed, including data on the operation of the taxi order service. For primary research, the popular Matlab language and the relatively new Julia language were chosen.
- 2) A software architecture has been developed and implemented to organize the collection of taxi order service data, which includes a Web server, a database server, and an application server.
- 3) Stochastic models of random processes and fuzzy logic models that can be used in predicting the number of taxi service orders are presented.

- 4) Predictions of the number of taxi service orders using various models have been made and comparative characteristics of such predictions have been obtained.
- 5) It is shown that models based on fuzzy logic provide a gain in comparison with mathematical models of random sequences when predicting taxi service data about 20-50% of the variance of prediction errors.

Thus, the results obtained allow us to say that the implementation of data mining algorithms in taxi order services is real today using machine learning methods, but additional control by a person is required.

In the future, it is planned to consider deep learning for the analysis of taxi service data. It is also planned to cluster multidimensional data on the operation of a taxi service using Gaussian mixtures models.

## Appendix

Before implementing the data mining algorithms in a specific programming language, it is worth to consider the main trends in this area [21].

A huge number of tasks today are associated with the processing of experimental data, or with mathematical modeling of some real process. These tasks are successfully solved by such hardware as personal computers and, in some cases, even computing clusters and supercomputers. In the software part, there are many programming languages that can be used for numerical calculation. Compared to general-purpose languages, they provide a simple (often intuitive) program syntax, as well as a large library of specialized functions. All of them are interpretable, which speeds up the implementation and debugging of algorithms, but negatively affects the speed of programs. These include Matlab, with its implementations such as Octave and Scilab. These programs operate perfectly with matrix calculations. Python is also gradually gaining popularity in the scientific community, along with the optional NumPy and SciPy modules.

Unfortunately, increasing the speed of programs requires moving the code to one of the traditionally used static languages (C/C++, Fortran). Obviously, the need to rewrite the program creates additional difficulties for the researcher.

Let consider some already proven tools and relatively new languages, such as Julia [22] in more detail. The analysis shows that the following table (see Table 2) can be compiled quite fully characterizing the studied programming languages.

Thus, an analysis was performed on programming languages that can now be successfully applied to data processing. It is important to understand what tasks need to be solved in order to choose the necessary language. After all, some languages have specificity and versatility, and some languages have properties of convenience and efficiency. Nevertheless, for our research on predicting the number of taxi service orders, we will choose the languages Matlab (for implementing mathematical models of random processes) and Julia (for implementing models of fuzzy logic).

**Table 2.** Programming languages for data analysis

Programming language	Advantages	Disadvantages
R	<p>Freeware</p> <p>The language has an open license; it consists of open source packages oriented towards data processing tasks. For example, neural network technologies, nonlinear regression algorithms, libraries for graphic display, etc. This language also works well with matrix algebra data. Good data visualization, for example, through the ggplot2 library</p>	<p>The main disadvantage is low productivity. In addition, this language is specific, and it cannot be used as a general programming language. It should also be noted that R has some features unusual for programming languages, in particular, it starts indexing from 1, not 0</p>
Python	<p>Freeware</p> <p>This language can be used as a general-purpose programming language. It includes special data processing modules. It has broad integration with online services. In addition, it is believed that Python is a fairly easy language to learn. Machine learning is implemented through TensorFlow, pandas, scikit-learn</p>	<p>Low type safety associated with type mismatch errors. The lack of a huge number of application packages for data analysis compared to R. There are analogues with greater speed and security</p>
SQL	<p>In addition to paid versions, there are free ones</p> <p>The language is used to process queries and is used in relational databases. The syntax of the language will be clear even to a beginner. Often integrates with other languages through some modules</p>	<p>The types of implementations are too different in terms of characteristics and functionality</p>
Java	<p>Free use</p> <p>Java is a universal programming language with strong typing of variables. It can be effectively used in machine learning tasks</p>	<p>There is a sufficiently small number of special libraries for data mining</p>
MATLAB	<p>Language provides a lot of built-in specialized functionality. In addition, it provides the user with convenient visualization</p>	<p>It is paid software</p> <p>It is poorly suited for solving general-purpose problems</p>

*(continued)*



**Table 2.** (continued)

Programming language	Advantages	Disadvantages
C++	C++ is extremely popular and high-performance language	It is absolutely not effective in solving data analysis problems
Perl	Freeware Perl is similar to Python, it is a dynamic typing language. Libraries and methods exist for quantitative data analysis	The syntax is difficult for programmers. No new libraries for data science are being released
Julia	Freeware Julia is a compiled JIT language (just-in-time). It provides high performance. Simple enough to learn. In addition, it can be used as a general-purpose programming language. It is intuitive, always readable language	The instability of work associated with the immaturity of the language. Quite a few data science programmers working at Julia. Few specialized libraries compared to R

## References

- Schuh, G., Reinhart, G., Prote, J., Sauermann, F., Horsthofer, J., Oppolzer, F., Knoll, D.: Data mining definitions and applications for the management of production complexity. *Procedia CIRP* **81**, 874–879 (2019)
- Oluwaseun, A., Chaubey, M.: Data mining classification techniques on the analysis of students performance (2019). <https://doi.org/10.11216/gsj.2019.04.19671>
- Bharati, M., Ramageri, M.: Data mining techniques and applications. *Indian J. Comput. Sci. Eng.* **1**, 301–305 (2010)
- Andriyanov, N.A., Gavrilina, Yu.N.: Image models and segmentation algorithms based on discrete doubly stochastic autoregressions with multiple roots of characteristic equations. In: *CEUR Workshop Proceedings*, vol. 2076, pp. 19–29 (2018)
- Hamuda, E., Glavin, M., Jones, E.: A survey of image processing techniques for plant extraction and segmentation in the field. *Comput. Electron. Agric.* **125**, 184–199 (2016)
- Andriyanov, N.A., Vasil'ev, K.K., Dement'ev, V.E.: Investigation of the filtering and objects detection algorithms for a multizone image sequence. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, pp. 7–10 (2019)
- Umang, P.: NYC taxi trip and fare data analytics using BigData (2015). <https://doi.org/10.13140/RG.2.1.3511.0485>
- Andriyanov, N.A., Sonin, V.A.: Using mathematical modeling of time series for forecasting taxi service orders amount. In: *CEUR Workshop Proceedings*, vol. 2258, pp. 272–462 (2018)
- Guo, J.: Analysis and comparison of Uber, Taxi and Uber request via Transit. *JISC*, p. 29 (2018)
- Danilov, A.N., Andriyanov, N.A., Azanov, P.T.: Ensuring the effectiveness of the taxi order service by mathematical modeling and machine learning. In: *Journal of Physics: Conference Series*, vol. 1096 (2018). <https://doi.org/10.1088/1742-6596/1096/1/012188>

11. Deri, J., Moura, J.: Taxi data in New York city: A network perspective, pp. 1829–1833 (2015). <https://doi.org/10.1109/acssc.2015.7421468>
12. Azanov, P.T., Danilov, A.N., Andriyanov, N.A.: Development of software system for analysis and optimization of taxi services efficiency by statistical modeling methods. In: CEUR Workshop Proceedings, vol. 1904, pp. 232–238 (2017). <https://doi.org/10.18287/1613-0073-2017-1904-232-238>
13. Vasiliev, K.K., Dementyiev, V.E., Andriyanov, N.A.: Using probabilistic statistics to determine the parameters of doubly stochastic models based on autoregression with multiple roots. In: Journal of Physics: Conference Series, vol. 1368, pp. 1–8 (2019). <https://doi.org/10.1088/1742-6596/1368/3/032019>
14. Andriyanov, N.A., Dementyiev, V.E.: Determination of borders between objects on satellite images using a two-proof doubly stochastic filtration. In: Journal of Physics: Conference Series, vol. 1353, pp. 1–7 (2019). <https://doi.org/10.1088/1742-6596/1353/1/012006>
15. Vasiliev K.K., Andriyanov N.A.: Synthesis and analysis of doubly stochastic models of images. In: CEUR Workshop Proceedings, vol. 2005, pp. 145–154 (2017)
16. Andriyanov, N.A., Dementiev, V.E., Vasiliev, K.K.: Developing a filtering algorithm for doubly stochastic images based on models with multiple roots of characteristic equations. Pattern Recogn. Image Anal. **29**(1), 10–20 (2019). <https://doi.org/10.1134/S1054661819010048>
17. Andriyanov, N.A., Sluzhiviyi, M.N.: Solution for the problem of the parameters identification for autoregressions with multiple roots of characteristic equations. In: CEUR Workshop Proceedings, vol. 2391, pp. 79–85 (2019)
18. Wood, G., Batt, J., Appelboam, A., Harris, A., Wilson, M.: Exploring the impact of expertise, clinical history, and visual search on electrocardiogram interpretation. Med. Decis. Making **34**(1), 75–83 (2014)
19. Yager, R., Filev, D.: Essentials of Fuzzy Modeling and Control, p. 387. Wiley, New York (1984)
20. Wood, G., Knapp, K.M., Rock, B., Cousens, C., Roobotton, C., Wilson, M.: Visual expertise in detecting and diagnosing skeletal fractures. Skeletal Radiol. **42**(2), 165–172 (2013)
21. <https://bigdata-madesimple.com/top-8-programming-languages-every-data-scientist-should-master-in-2019/>. Accessed 28 Sept 2019
22. <http://julialang.org/teaching/>. Accessed 28 Sept 2019



# Discourse Analysis on Learning Theories and AI

Rosemary Papa<sup>1</sup>(✉), Karen Moran Jackson<sup>1</sup>, Ric Brown<sup>2</sup>, and David Jackson<sup>3</sup>

<sup>1</sup> Soka University of America, Aliso Viejo, CA 92656, USA

rpapa@soka.edu

<sup>2</sup> Laguna Niguel, CA 92677, USA

<sup>3</sup> Stealth Mode Startup, Dana Point, CA 92629, USA

**Abstract.** The intent of the study was to identify the dialogue and discourse on how AI development includes and/or excludes pedagogical educational learning theories focused on the learner. Through identifying areas of intersection between AI development and learning theories, educational leaders can interface with developers and content experts to establish optimal teaching skills and strategies for the ethical ‘good’ of the learner. The review of the discourse in the literature revealed surprisingly limited intersections between AI and learning theories, with a tool-centric literature, coupled with efficiency evaluations and developmental narratives. The following three conceptual questions to engage in dialogue between AI developers and educational leaders surrounding AI and learning theories were proposed: (1) Who ultimately controls the curriculum? (2) Are cognitive theories primarily utilized in constructing AI algorithms? (3) What is encapsulated in AI’s hidden curriculum and how is bias/discrimination accounted for? The opportunity for educational leaders and theorists of learning to engage with AI developers and super-intelligence is necessary for the ‘good’ of what is developed artificially. If teachers are viewed as only content experts without acknowledgement of the multiple strategies they use to inspire and encourage students, then AI development may get teaching very wrong.

**Keywords:** Artificial intelligence · Learning theories · Educational leadership · Discourse analysis

## 1 Introduction

The intent of the study was to identify the dialogue and discourse on how AI development include and/or exclude the intersect of pedagogical educational learning theories to strategies focused on the learner. By identifying these areas within the literature, these researchers are taking a long-term futuristic view to outline where educational leaders of learners can interface with developers and content experts to establish optimal pedagogy/andragogy teaching skills and strategies for the ethical ‘good’ of the learner. Our intention is a dialogue about these issues now, before they become immediate problems. A primary concern is the AI focus on efficiency. Much of the literature review discussed the goal of have “efficient” educational tools and educational systems. Chounta notes

---

R. Brown—Educational Consultant.

© Springer Nature Switzerland AG 2020

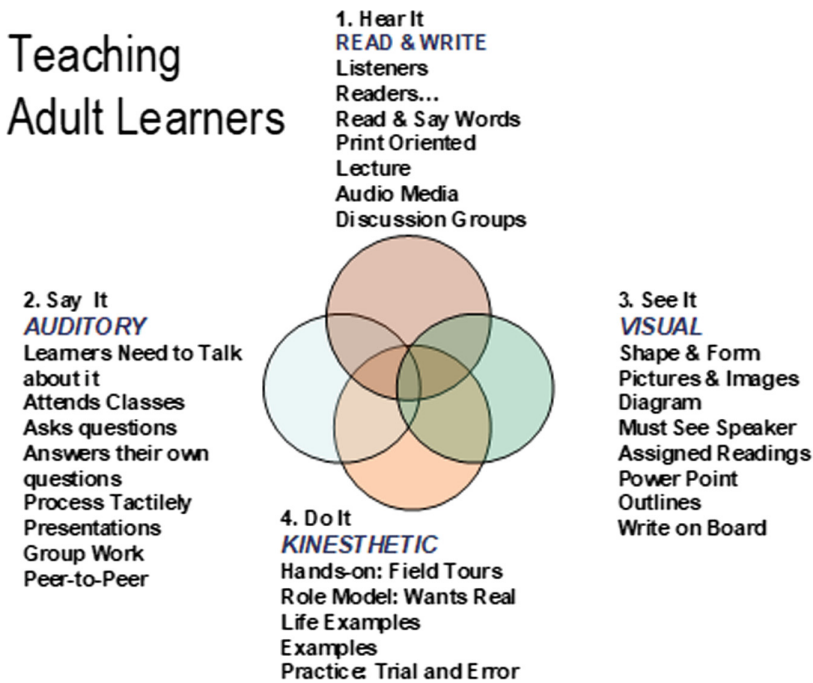
K. Arai et al. (Eds.): SAI 2020, AISC 1230, pp. 665–672, 2020.

[https://doi.org/10.1007/978-3-030-52243-8\\_50](https://doi.org/10.1007/978-3-030-52243-8_50)

that educational systems should develop to assist learners “in using new technologies and digital resources in an efficient and effective way in order to achieve their goals” [1, p. 6] and that AI, along with machine learning, can be used to assist teachers to “orchestrate learning activities more efficiently” [1, p. 12). Morrison and Miller offer a slightly different perspective in that they hypothesize that human intelligence and machine intelligence working together are “likely to be more effective, efficient, and ethical than systems that rely on machine intelligence alone” [2, p. 441].

## 2 Background

Great teaching is defined by the ability to inspire learners [3–5] Motivate the learner and you will grab their attention. Keeping a learner’s attention is more difficult; this is described as the human teacher elements and their strategies to contextually teach all to the one student. Educational pedagogics and andragogics need many strategies at their fingertips to keep others’ attention. In the learning environment, the educational teacher leader acknowledges their role as learner. Figure 1 describes how adults can be taught to reach all learners.



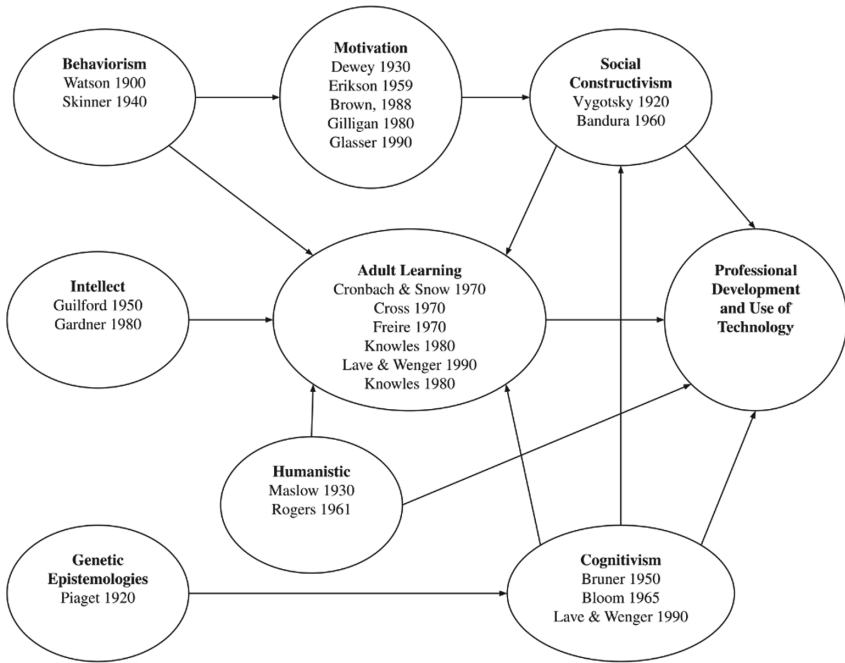
**Fig. 1.** Teaching adult learners requires a combination of different techniques beyond traditional reading and writing, to auditory, visual, and kinesthetic methods (Figure adapted from [3, 4].)

“This chart has the educational leader understand that by changing the strategies for the learner, all adult learners are engaged. Hearing something said, saying something,

doing something, and seeing something acknowledge that adults learn differently” [6]. The goal is to keep the learner’s attention: To optimize engaged learners demands the use of strategies and techniques that support the varied learning styles of both children and adults. How does this interface with AI and the creation of siloed learning units for the learner?

### 2.1 Discourse Analysis

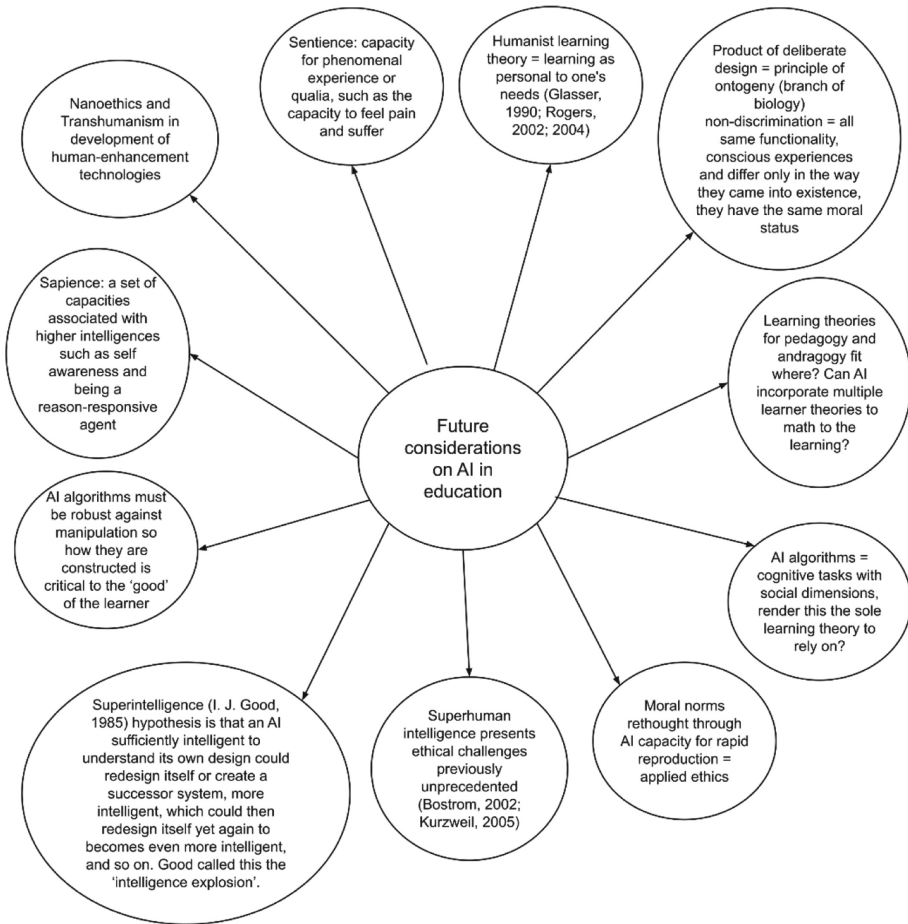
The methods used in this study included a selected literature review and qualitative, theoretically guided content analysis, following the conceptual framework of transhumanism [7], as well as educational learning theories [4–6, 8–23]. See Fig. 2.



**Fig. 2.** A conceptual map of learning theories and the major theorists that continue to impact adult learning, professional development, and the use of technology in classrooms. (Figure adapted from [6].)

Will AI model developers focus on cognition learning theories only? As Fig. 2 depicts, the evolution of learning theories have lead us to where we are today. “Cognitive and social constructivism’s are strong underpinnings to adult learning, as are humanist and motivation-personality theories” [6, p. 98]. The best adult strategies are found foremost in the work of Knowles, Cross, Lave and Wenger, and Cronbach and Snow [6, p. 98]. The evolution of these theories can be found in the cognitivists, social constructivists, motivation theory, intellect theory and humanism.

The content topics that guided the literature were broadly related to artificial intelligence and education. The goal of the literature search was to identify dialogue and discourse about how these topics intersect in terms of ethical concerns and long-term educational outcomes. By identifying these areas of concern within the literature, researchers outlined where educational leaders can interface with developers and technological thought leaders. The interactions between these two groups would then be guided by questions not only about implementation of AI in schools, but also questions about the ethical creation of these AI-enhanced educational tools and systems. See Fig. 3.



**Fig. 3.** A conceptual map depicting future considerations and questions about the implementation and ethical creation of AI-enhanced educational tools and systems. [Author created.]

### 3 Preliminary Results

The review revealed a surprisingly limited literature base. While the field is broad, the literature based is tool-centric, with a preponderance of efficiency evaluations and developmental narratives. For example, a search of the International Journal of Artificial Intelligence in Education (IJAIE), published since 2013, for the terms ‘ethics’ or ‘ethical’ returned 20 articles, none of which have the terms in the title nor in the article keywords, indicating that while ethics may be discussed within the work, it is not the focus. We note that a similar search of education-related journals produced limited work related to AI. For example, *Educational Researcher*, a premier education journal published by the American Educational Research Association, only returns seven articles using the search terms ‘artificial intelligence’ or ‘AI’, although one includes ‘AI’ in the title and two of these include ‘machine learning’ in the keywords. Literature that deals with the intersection of AI with ethical concerns of education appears both sparse and siloed. This finding was echoed in some of the work found. For example, Morrison and Miller note that the association that publishes IJAIE is “insular, enriched but arguably limited by contributions from a fairly small set of disciplines, including cognitive science (and psychology generally), computer science, and computational linguistics” [2, p. 442].

Despite the scarcity of resources, the works that do exist that touch on these issues are rich and offer deep discourse for analysis. The following three opportunities for dialogue between the content areas were discovered and are phrased as questions for discussion: (1) Who ultimately controls the curriculum? (2) Are cognitive theories primarily utilized in constructing AI algorithms? (3) What is encapsulated in AI’s hidden curriculum and how is bias/discrimination accounted for? The speed of reproduction in AI if not ethically constructed on many learning theories may do more harm than good.

#### 3.1 Who Ultimately Controls the Curriculum?

Schools are, for the most part, controlled environments. Student schedules are followed to the minute with times for learning, eating, and play marked by audible bells or buzzers. Materials, such as textbooks and laboratory equipment, are signed in and out as needed, their access guarded by adults. Computers and other devices that can reach into external, out-of-school environments are regulated, with strict filters policing access. Adults in these cases are controlling the lives of children. Yet, the vision of super-intelligence in computers [24–26] presupposes a loss of control by the educator; it is the machine that has the control. What happens when a child deliberately answers all questions on a content assessment wrong? How does a computer respond? How does that failure follow the child and continue to impact them? Walker and Ogan term the programmed reaction to these types of situations as ‘failure management strategies’ pointing out that the level of engagement between students and the technology drives the need for these types of modules [27]. Yet, educators encounter these types of situations everyday with students. Rather than programmed strategies, years of classroom experience and discussions with other educators is often how teachers develop various strategies to deal with the loss of control in the classroom. This renders the human characteristics situational and emotional in deciding what is best within a given context, i.e., classroom as a whole, to the ability to meet student needs. Rather should we consider, as Baker [28] does, the advantages of

simple technology tools that leverage display and reporting system designs that allow teachers to make conclusions and decisions about students' educational pathways?

### 3.2 Are Cognitive Theories Primarily Utilized in Constructing AI Algorithms?

How technology serves student learning is rarely based within humanistic learning paradigms that focus on the overall value of education for the good of society. Individualized learning is not associated with organic students' interests, a core of humanistic education [29, 30], but rather with pacing and exposure through pre-set curriculum [28]. The linear format of much of current AI-enhanced learning tools pre-suppose a competency level that must be met on some content knowledge before exposure to later knowledge. Yet, we know learning is a much more organic process. Sometimes this non-linear nature of learning is acknowledged in the literature. For example, social network analysis and Bayesian models are attempts to deal with this messiness [1].

Students, and efficient retrieval practices of content knowledge in academic situations, such as testing, may be viewed as a product of the reliance on cognitive learning theories that focuses on efficient processing of information. The drive to create efficient systems can also be viewed as a product of the neo-liberal paradigm that insists technology have the goal of producing profit, rather than producing societal good. Within educational practice and common educational goals, however, efficiency is a secondary concern. As educators are trained using a variety of learning theories, other goals besides efficiency are a foremost concern.

### 3.3 What Is Encapsulated in AI's Hidden Curriculum and How Is Bias Accounted for?

AI is dependent on the gathering of new data to inform new models and this dependence on data will drive how AI becomes implemented in educational systems [31]. The kind of data collected on students and how this data is used in systems is often hidden from the end users, students and teachers. Ubiquitous uses of AI-enhanced learning systems, systems that Morrison and Miller note are "inherently amoral" [2, p. 441], hide content choices from teachers and parents. Similar to the 'teacher-proof' curriculum that was used in American schools in the last half of the 20<sup>th</sup> century, when scripted lines were fed to teachers through textbooks, in the case of AI-enhanced tools, it is the textbook that is doing the talking. This loss of social interactions between learners and teachers is a particular red flag for educators who work with children from non-majority groups, as scripted curriculums are often repetitions of dominant narratives and are not inclusive of culturally relevant curriculum and pedagogy [32]. While there is some research on how educational tools have different outcomes in different cultural situations, there is less research on cultural relevant, AI enhanced tools [31]. A simple, and incredibly naïve, solution to the problem would be to create modules that are context dependent or dependent on the inputted demographics of children; these would be particularly vulnerable to charges of stereotyping, essentialism, and discrimination. Even teachers, trained in curriculum presentation and differentiation, struggle with choosing appropriate and intellectually engaging curriculum for students they see on a regular basis and have personal relationships with. Teachers are trained with a focus on multi-dimensional



strategies to reach all learners, and still at times are not able to reach a given student. How then will AI software be developed to get it correctly?

## 4 Conclusion

The opportunity to engage with AI developers and super-intelligence is necessary for the “good” of what is developed artificially. As Bostrom [24] stated, superintelligence may pose as an existential risk to humans as we know them now. If teachers teaching the learners are only content experts with no thought to the multi-dimensional aspects of learners and the strategies that ultimately “inspire” and encourage their human “passions”, then AI development may well get it very wrong.

## References

1. Chounta, I.-A.: A review of the state-of-art of the use of machine-learning and artificial intelligence by educational portals and OER repositories. European Schoolnet LRE Sub-Committee (2019). [http://www.eun.org/documents/411753/3138455/LRE\\_WhitePaper\\_1\\_14022019.pdf](http://www.eun.org/documents/411753/3138455/LRE_WhitePaper_1_14022019.pdf)
2. Morrison, D.M., Miller, K.B.: Teaching and learning in the pleistocene: a biocultural account of human pedagogy and its implications for AIED. *Int. J. Artif. Intell. Educ.* **28**, 439–469 (2018). <https://doi.org/10.1007/s40593-017-0153-0>
3. Papa, R.: *The Art of Mentoring*. Center for Teaching and Learning, Sacramento (2002)
4. Papa, R.: *How We Learn*. Center for Teaching and Learning, Sacramento (2002)
5. Papa, R.: Transitions in teaching and eLearning. In: Papa, R. (ed.) *Media Rich Instruction: Connecting Curriculum To All, Learners*, Chap. 1. Springer, Cham (2015)
6. Papa, R., Papa, J.: Leading adult learners: preparing future leaders and professional development of those they lead. In: Papa, R. (ed.) *Technology for School Improvement*, Chap. 5. Sage, Thousand Oaks (2011)
7. Bostrom, N.: The ethics of artificial intelligence. In: Ramsey, W., Frankish, K. (eds.) *Cambridge Handbook of Artificial Intelligence*. Cambridge University Press, Cambridge (2011)
8. Bloom, B.S.: *Taxonomy of Educational Objectives*. Longman, London (1965)
9. Dewey, J.: *Experience and Education*. Macmillan, New York (1938)
10. Erikson, E.H.: Identity and the life cycle. *Psychol. Issues* **1**, 1 (1959)
11. Erikson, E.H.: *Identity, Youth and Crisis*. W. W. Norton, New York (1968)
12. Gardner, H.: *Frames of Mind: The Theory of Multiple Intelligences*. Basic Books, New York (1983)
13. Gardner, H.: *The Disciplined Mind: Beyond Facts and Standardized Tests: The K–12 Education that Every Child Deserves*. Simon and Schuster, New York (1999)
14. Gardner, H.: *Intelligence Reframed: Multiple Intelligences for the 21st Century*. Basic Books, New York (1999)
15. Gardner, H.: *The Development and Education of the Mind: The Selected Works of Howard Gardner*. Routledge, London (2006)
16. Knowles, M.S.: *The Modern Practice of Adult Education: From Pedagogy to Andragogy*. Association Press, New York (1980)
17. Knowles, M.S.: *The Adult Learner: A Neglected Species*. Gulf, Houston (1990)
18. Kolb, D.: *Experiential Learning: Experience as the Source of Learning and Development*. Prentice-Hall, Englewood Cliffs (1984)

19. Lave, J., Wenger, E.: *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, Cambridge (1991)
20. Maslow, A.H.: A theory of human motivation. *Psychol. Rev.* **50**, 370 (1943)
21. Rogers, A.: *Teaching Adults*, 3rd edn. Open University Press, London (2002)
22. Rogers, J.: *Adult Learning*, 4th edn. Open University Press, London (2004)
23. Skinner, B.F.: *About Behaviorism*. Random House, New York (1974)
24. Bostrom, N.: Existential risks: analyzing human extinction scenarios. *J. Evol. Technol.* **9** (2002). <https://www.nickbostrom.com/existential/risks.html>
25. Bostrom, N.: Transhumanist values. In: Adams, F. (ed.) *Ethical Issues for the 21st Century*. Philosophical Documentation Center Press, Bowling, Green University, Bowling Green (2003)
26. Kurzweil, R.: *The Singularity is Near: When Humans Transcend Biology*. Viking, New York (2005)
27. Walker, E., Ogan, A.: We're in this together: intentional design of social relationships with AIED systems. *Int. J. Artif. Intell. Educ.* **26**, 713–729 (2016). <https://doi.org/10.1007/s40593-016-0100-5>
28. Baker, R.S.: Stupid tutoring systems, intelligent humans. *Int. J. Artif. Intell. Educ.* **26**(2), 600–614 (2016)
29. Maslow, A.H.: A theory of human motivation. *Psychol. Rev.* **50**, 370 (1930)
30. Rogers, C.: *On Becoming a Person: A Therapist's View of Psychotherapy*. Constable, London (1961). ISBN 1-84529-057-7
31. Pinkwart, N.: Another 25 years of AIED? Challenges and opportunities for intelligent educational technologies of the future. *Int. J. Artif. Intell. Educ.* **26**(2), 771–783 (2016)
32. Ladson-Billings, G.: Culturally relevant pedagogy 2.0: aka the remix. *Harv. Educ. Rev.* **84**(1), 74–84 (2014)



# False Asymptotic Instability Behavior at Iterated Functions with Lyapunov Stability in Nonlinear Time Series

Charles Roberto Telles<sup>(✉)</sup>

Director Board, Research Advisory, Secretary of State for Education and Sport, Curitiba,  
PR 80240-900, Brazil  
charlestelles@seed.pr.gov.br

**Abstract.** Empirically defining some constant probabilistic orbits of iterated high-order functions, the stability of these functions in possible entangled interaction dynamics of the environment through its orbit's connectivity (open sets) provides the formation of an exponential dynamic fixed point as a metric space (topological property) between both iterated functions for short time lengths. However, the presence of a dynamic fixed point can identify a convergence at iterations for larger time lengths (asymptotic stability in Lyapunov sense). Qualitative (QDE) results show that the average distance between the discontinuous function to the fixed point of the continuous function (for all possible solutions), might express fluctuations of on time lengths (instability effect). This feature can reveal the false empirical asymptotic instability behavior between the given domains due to time lengths observation and empirical constraints within a well-defined Lyapunov stability.

**Keywords:** Coupling functions · Asymptotic instability analysis · Discretization · Qualitative theory of differential equations · Bifurcation · Topology

## 1 Introduction

The first main problem addressed in this research is to establish a difference between empirical experiments and theoretical mathematical simulations regarding asymptotic stability and bifurcation phenomena [1]. Although asymptotic iterated functions can be often considered empirically invisible in their nature of proportionality and convergence [2–4] due to time dependent outcomes and stability scenarios where high level of complexity is present in a given phenomenon and as noted by Newman *et al.* [4], all the solutions in convergence to a dynamic and time-varying fixed point can be empirically visible if time nonlinear lengths of phenomenon can be accelerated by mathematical tools or physical properties in connectivity with the event [3].

Based on this methodologies views, a given phenomenon and analysis always can point to the correct observation of an event as far as mathematical frameworks are used to fit in the event. However, this framework can be unsuited to empirical analyses where

perturbations over the initial conditions and during phenomenon expression are possible to occur or asymptotic effects have its occurrence not defining a specific probabilistic distribution over time [4–6]. Following this path, the opposite aspect of the mathematical and theoretical framework can be given, when in one empirical analysis, the data is analyzed and a mathematical model is fitted to it, leading to a conclusion where events occur as something independent [3] of the empirical dimension of analysis. In this view, theoretical schemes that don't show any different aspect other than the initial formulation and mathematical prediction of a given phenomenon can't be a good model for nonautonomous equations, where empirical constraints can be the cause of bifurcation phenomena [1, 4, 7].

Differing theoretical qualitative partial differential equations analysis and empirical predictions or experiments towards event's convergence and stability behaviors, the frequency of iterations with which these functions occur partially on time intervals in their theoretical formulations (as time invariants and autonomous) or empirical observations (nonautonomous and time-varying), affecting the physical nature of a phenomenon, are subject to observation and discrimination to the extent that they can be empirically observed, knowing the metric of spaces that constitute all the stages of a given event in terms of its iterated functions be a product of complex interactions [2, 3, 7, 8]. Concerning qualitative analysis of partial differential functions, mathematical theoretic parameters might be defined without the observation of physical and time constraints, whose properties of events in evolve, only in the light of an empirical experimentation, could express unpredicted mathematical expectation, and thus differing from empirical data expressions. However, as the metric space of these iterated functions oscillates in their expressions considering an empirical view of phenomena and can increase distances from each other in terms of possible complex interactions, it is possible, deductively, to describe the event as presenting a dynamic asymptotic instability between functions as a constant defined as  $b$  in terms of its expression at short time lengths for all finite solutions directed to a given fixed point as noted by Lundberg (1963) [9]. But considering the same phenomenon as defined in this research, as a dynamic exponential fixed point, that evolves as far as instability turns it into new patterns of stability, resembling Williamson's concept (1991) [4, 10], the phenomenon in larger time lengths, inductively [3], might present stability evolution (convergence) in the Lyapunov sense of stability [11]. In this sense, for identifying at Lyapunov stability an asymptotic stable behavior, for empirical observations, it is necessary to describe an event considering its escaping and approximation orbits similarly to a KPZ Brownian aspect of convergence [12] or the stochastic Lorentz system [13] in order to make visible, mathematically and empirically, the asymptotic stability formation feature within the apparent unstable dynamics. However, in this research it will be demonstrated that the apparent asymptotic instability observed in nature's function expression in connectivity and time lengths (iterated and created by high level of variable's possible metric spaces within disjoint open sets) is in true a false asymptotic instability effect. This feature occurs as far as the time interval of phenomenon observations confers to a given fixed point, oriented as an exponential function of  $f(x)$  and its complex not *i.i.d.* string of variable's interactions  $g(x)$ , a quantized empirical result that will express oscillatory overall proportionality (not the

asymptotic equipartition property) and the overall convergence for all dynamical fixed point observed at larger time lengths within a nonlinear time series.

The camouflage effect of instability resides in the sense that the asymptotic instability definition in terms of empirical observation does not match with true functions correlation if compared with the theoretical view of mathematical simulations, being the effect of time lengths a tool in which empirical results can present massive quantified results of a function that can be asymptotically stable as the whole system observations or in an exceptional case, even be not asymptotic at all.

It is defined as a dynamic fixed point in this research the formation by iteration and interaction between distinct iterated functions, on a metric space in which one of the function  $g(x)$  that are defined partially by a fixed point  $f(x)$  assume other distances between two points of the two considered functions, alternating the position in the complete existent metric space between one function  $f(x)$  and the other subsequent iteration of  $f(g(x))$  as a nonempty space that represents in empirical terms the connectivity of  $f(x)$  and  $g(x)$  as an evolution and topological expansion of the system defined as the constant  $b$ , thus generating randomly and dynamical exponential fixed points as the  $b$  constant.

This connective metric space in turn generates a kind of instability between the fixed point metric position that can be defined in a specific order of possible orbits of interactions in the system for  $f(g(x))$  (solutions proximity) [14] and at the same time a region of space in which the iteration reaches their maximum degree of expression presenting higher distance between two points of overall iterated functions (escaping solutions) [15–18].

Thus, Banach's fixed-point theorem objectively illustrates this instability distribution of iterated functions over a single fixed point in a complete metric space, considering for it the dynamic fixed point concept and reflecting this definition as one of the iterated functions partial attraction to a fixed point for each interaction or for some time lengths of event, thus creating a connectivity metric space [12, 13], however generating expansively many distinct fixed points.

This research practical implications are that, empirically, many results presented by scientific community show that a given studied system present asymptotic instability, due to the limitation of time acceleration of events. This aspect can, empirically predicts, how physical phenomena will behave in very complex scenario for many fields of science and also for policy making regarding descriptive statistics methodological limitations [19, 20] in the big cities administration. The main result of this research dealt with a simple example using a nonlinear time series data, where an approaches for nonautonomous techniques were done, giving a general overview of time-varying solutions convergence at nonautonomous partial differential equations.

## 2 Methodology

### 2.1 Connectivity and Iterated Functions

**Theorem 1.** Consider within a nonautonomous system of partial differential equations, empirically designed, of one iterated and continuous function  $f$  defined as a mirror (fixed point) of the discontinuous iterated function  $g$  and having as its derivative  $g$  constantly

iterated in connectivity to  $f$  as a constant  $b$ . A complete metric space as  $\delta = (x_{n+1})f(g(x))$  as the constant  $a$ , present all solutions in convergence as  $\varepsilon = y = b$ , but presenting high oscillations as time expands the event, hence expressing asymptotic instable behavior caused by  $a$  derivative and the connectivity topological property. These two iterated functions  $f(x)$  and  $g(x)$  in terms of their physical nature have defined, but high complex probabilistic behavior, and only in the time acceleration of the phenomenon it is possible to observe that the distributive instability asymptote behavior is rather an effect of unobserved overall  $g(x)$  convergence towards  $f(x)$  or in other words as a dynamical fixed point as the constant  $b$  with overall possible convergence of the constant  $a$ .

In this sense, consider for the function  $f(x)$  the sum of all possible metric spaces produced by the complex interactions of variables present at  $g(x)$ , and thus defining itself as the fixed point of convergence, as Picard–Lindelöf theorem, like  $\varphi_0(t) = f(g(x))_0$ , hence defining the  $a$  constant of convergence for  $g(x)$ . As far as  $f(x)$  define itself as a product of  $f(g(x))_{x+1}$ , as  $\varphi_{x+1}(t) = f(g(x))_0 + \int_0^{x+1} g(x)(s, \varphi_x(s))ds$ , being  $s$  a local uniqueness for the iterative behavior, a fixed point is created for each string of iterated events and it has as its derivatives the partial differential equations of  $g(x)$ , where this later function assumes a dynamical time-varying probabilistic variance that can be observed in terms of exponential empirical behavior caused by complex interactions among  $g(x)$  multivariable functions (not embedding). The  $a$  constant of convergence, by  $g(x)$  expansion, leads  $f(g(x))_{x+1}$  to assume an empty space within the disjoint sets, thus generating a connectivity metric space and topological property defined as the  $b$  constant, or in other words  $f(x)_x$  like time-varying dynamical fixed points. In this sense, fixed points can be defined for all  $f(x)$  constantly, defined as a Banach fixed points, and for  $g(x)$  all solutions goes near  $f(g(x))_{x+1}$  dynamical fixed points, but can remains partially tangent to  $f(x)_x$  formed fixed points for all time, constantly as  $b$  undefined asymptotic behavior (theoretically speaking). Note that the notation  $f(x)_x$  is equal  $f(g(x))_{x+1}$ , except for the iterative aspect. However, in this research the notation  $f(x)_x$ , will be replaced by  $f(g(x))$  for the next explanations.

Note that this trivial expression of  $a$ , in the light of an empirical explanation, is the desirable solution of the event, being this expression, the most observed solution of a given phenomenon, however, not observed as an  $b$  constant due to physical and experimental constraints that will lead the observation of the nonautonomous partial differential equations properties.

The time of occurrence  $x(t_{n+1}) \geq 0$  for each iterated event representing each function in connectivity ( $b$ ) is constant at an expression frequency of time as  $t_{n+1} + t_{n+1}$  and probabilistic distribution as a solution  $\varphi = P_n$ , with trajectories  $X(t_n)$  as input data, the solutions  $\varphi(t_n)$  assumes asymptotic stability of  $X(t_n) - \varphi(t_n) < \delta$  for each single iterated event, thus reflecting the formation of a constant already identified as  $a$ . The interaction of these functions in a given region of the metric space  $x_{n+1} = f(g(x))$  generates, assuming that the quantitative properties of the event remain with partial and asymmetric numerical transformations to their original form ( $\partial$ ), for each interaction such as,

$$\frac{\partial x_{n+1}}{\partial g}(f, g) = or \leq 1 \therefore x_{n+1}(f, g) = f(g(x)) \tag{1}$$

a solution to a fixed point at  $y$  axis that defines itself as a constant and random variable  $\partial x_{n+1} = b\{S_1, \dots, S_n\}$  (dynamic fixed point) for the function  $f(g(x))$  for each one of the iterations. The distance between two points in each iterated function at  $y$  as  $f(g(S_n))$  occurs in a general and the maximum empirical expansion as,

$$d(f(x), g(x)) = d((g(x) = fx + gx) \leq f(x) \text{ or } \leq \sum_{S_{n+1}}^n S_n \tag{2}$$

for short time intervals (embedding properties). Thus confirming the Lyapunov sense of asymptotic stability [11] defined as expressing a high convergence rate to the defined fixed point. Visualizing that scenario as time passes, the function’s behavior does not change under small input perturbations the asymptotic stability of phenomena at each of the iterations, where  $x_{n+1}$  can be represented by a connectivity  $n$  which the global distance between two points of function  $f(x)$  and  $g(x)$  domains assume distances equal to,

$$d(f(g(S_n)), g(x), f(x)) \leq d\left(\sum_{S_{n+1}}^n S_n, f(g(S_n))\right), \tag{3}$$

being the global distance presenting high vector instability and therefore as the overall distance of the constant  $a$  start to increase, the dynamical fixed point for  $f(g(x))$  start its formation, thus generating the already defined constant  $b$ . The overall convergence of the function  $g(x)$  when attracted to a fixed point of  $f(x)$  can be defined, roughly, as  $\lim_{n \rightarrow \infty} f(x) = \sum_{S_{n+1}}^n S_n - g(x) = 0$  such that  $x(S_n + 1) < \delta$ , for each iteration  $S_n + 1$ , when considering only one fixed point condition and  $S_n + 1$  is a metric space in which  $g(x)$  is not defined at the fixed point mutually with another iteration as indicated in number notation (4).

$$\begin{aligned} f &: f(g(S_n)) \rightarrow S_{n+1} \\ g &: g(f(S_n)) \rightarrow S_{n-1}. \end{aligned} \tag{4}$$

In the opposite direction of the  $f(g(x))$  pattern formation, the same function as now the constant  $a$ , the dynamic fixed point, can be defined where compared to the previous given equation, would necessarily assume the monotone definition as  $\lim_{n \rightarrow \infty} f(x) = \sum_{S_{n+1}}^n S_n - g(x) > 0$ , hence defining itself mutually among large time intervals as  $x(S_n + 1) < \delta$ , being  $\delta$  redefined constantly as far as  $f(g(x))$  assumes new states of oscillation and convergence, not assuming a behavior as described in (4).

And also  $g(x)$  by its turn has its metric space among complex variables interactions defined by nonlinear dynamics of event, not being possible to measure how the system input and output evolves for larger time lengths as a nonautonomous functions. Despite of a constant iterated convergence ( $a$ ),  $g(x)$  assumes  $\mathbb{R}$  solutions that empirically expand into nonlinear time series, thus not presenting any visible pattern of oscillation at larger time intervals. This feature resembles the nonlinear time series as pointed by Kantz and Schreiber [21] of the partial differential equation’s system.

This complex and random property of variable’s interaction of  $g(x)$  means that to the extent that  $f(x)$  is dynamically defined by fixed points for each iteration, there is

the formation of fixed positions in time as  $f(g(x))_{x+1}$ , with which the nonlinearity of  $a$  distance between two points of both sets for each iteration assumes a Lyapunov sense of asymptotic instability [11, 22] with a bounded and embedded mapping condition where for all vectors, the solutions will always be less than or equal to the sum of all dynamic fixed points ( $v \leq \sum_{S_n+1}^n S_n$ ), therefore presenting pattern formation for shorten time lengths. When observing the connectivity  $b$  of the iterated event, this characteristic reveals that the maximum dynamical fixed points are formed as an exponential function of  $b$  raised by  $a$  exponent of the iterated functions  $f(x)$  at  $y$  axis and  $g(x)$  vectors functional expression at  $x$  axis, generating a defined phase space, locally stable for every  $y$  point and unstable considering all  $y$  nonlinear time series.

According to the fixed point position of the phenomenon generated by the string sequences (orbits) of interaction between complex variables of  $g(x)$  and the average distances generated between the fixed points convergence for each iteration, this behavior of the event also allows us to observe, as will be described in the results section, that the distance for all solutions in all iterated events of the nonlinear time series in the dynamics of maximum and minimum metric spaces, assume an asymptotic instability that accompanies certain empirical expressions that is the not observable stability convergence effect [22], in which strings of iterated functions in connectivity have with the previous and posterior string events. This feature may not be visible (discretized) in terms of overall convergence or non-convergence due to an asymptotic camouflage effect.

This phenomenon can best be described as an orderly sequence of iterations through time as  $x_{n+1} + t_{n+1} = 1$  with parameters determined in  $a$  as  $S_n + 1$ ,  $P_n$  and  $d$  in the relationship that is defined only from  $b = S_1, \dots, S_n$ , generating a composition like  $(f \circ g)(S_n)$  as  $f(S_n + 1) = \{b(S_n) | S_n \in S_n + 1\}$ .

A linear prediction could be obtained if for certain strings of  $a$  and  $b$ , defined as Markovians [17], the functions  $S_n + 1$  be infinitely discretized by presenting growth or decreasing non exponential value caused by  $a$  in the available proportion of  $S_n + 1$ , revealing the flow of process iterations as  $f_{g(x)}^{f(x)}$  ( $f^m \circ g^n$ ) =  $(S_n^{m+n})$  as possibly ordinary differential equations in its various position for every single fixed point and for the formation of other fixed points. This circumstance in the opposite objective of this research, can be defined as a linear time series where statistically it is possible to determine the direction of all vectors in the field [19, 23].

To visualize the interaction of complex variables in iterated functions and the formation of asymptotic instability, the following sets are defined as a stylized example in Fig. 1.

**Lemma 1.** It is possible to observe that the fixed points  $S_n$  are generated from the  $S_n + 1$  interactions  $g(x)$  and iterations  $f(x)$  event. Thus, event discretization  $S_n + 1$  occurs as a string sequence (trajectory) created from the availability of  $S_n$  of being constant, dependent of  $g(x)$  evolution, but randomly formed from  $g(x)$ , which these fixed points in turn assume a function defined as  $(f \circ g)(x)$ . Therefore, for both functions  $f$  and  $g$ , the iterations of each function remain as an image of interactions occurring at a given moment from  $n$  fixed points  $S_1, \dots, S_n$  randomly or not, generated as  $f : (S_n + 1) + (S_n + 1) \rightarrow S_n \therefore f \times g$ , where, considering the larger time lengths, the metric space of connectivity of the constant  $b$  between two points of each iterated function is defined as



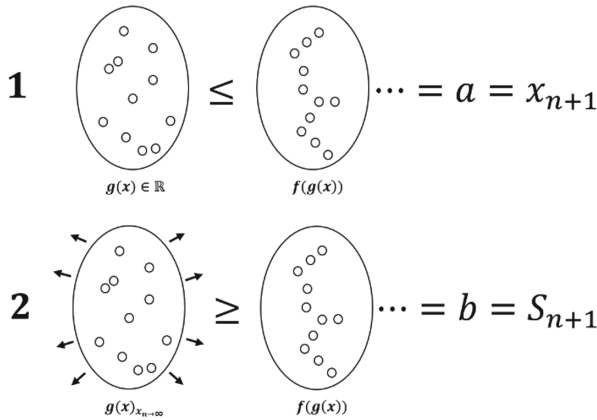


Fig. 1. Formation of dynamic fixed points by connectivity  $(f \circ g)(x)$ .

$d(f(g(S_n)), g(x), f(x)) \leq or \geq d(\sum_{S_{n+1}}^n S_n, f(g(S_n)))$ , expansively compared to the definition of (3), the global distance, where it was considered short time lengths for the iterations as observed analogously by Ignat’ev *et al.* [6]. If this definition be considered in the view of a common linear time series, the whole distance would be defined as (3), even for larger time lengths, due to the existence of only one fixed point as well as average distance and convergence rate among each iteration.

**Proof of Theorem 1 and Lemma 1.** The equation that describes the behavior of Fig. 1 for each iterated function  $f(x)$  in terms of time can be described as:

$$b = f(x) = S_n + 1 = (x_{n+1} + t_{n+1})P_n \tag{5}$$

Since all variables of each function have defined values where  $x \in \mathbb{R}$ , then the interaction orbits of  $g(x)$  have heterogeneous variables (discrete and continuous) interactions, thus presenting as a limit to  $f(g(x))$  as a maximum metric space of heterogeneity. These characteristics of  $f(g(x))$  as a dynamical fixed point formed as far as  $g(x)$  converges to  $f(x)$  for each iteration and as a solution of  $g(x)$  specific pattern random formation (discrete and continuous variables), the string sequence of iterations can be defined as found in Lyapunov’s stability, constantly changing for a new  $g(x)$  towards  $f(x)$ , like,

$$(S_n + 1)n = (x_{n+1} + t_{n+1})P_n, \text{ as the constant } b \tag{6}$$

for each  $f(g(x))$  originated, where, for each  $g(x)$ , the solutions at each iteration presents,

$$g(x)(t_n) - \varphi(t_n) \leq \delta, \tag{7}$$

as a nonautonomous PDE (partial differential equations), or in other words, an adaptive and self-organizing system.

At this point, it is mandatory to note that as briefly described in the introduction section, the mathematical theoretical observation of phenomena can’t induce by its nature of analysis, that the Lyapunov stability is rather fully unstable in the view of empirical results. Also, this feature, in the view of time acceleration of event, it project

an awkward empirical observation of an unstable convergence of  $g(x)$  towards  $f(x)$  (not confirmed at the mathematical point of view) and totally no pattern formation as observation collect data sets of phenomena. Exemplifying it, this feature can be empirically viewed when specimen evolution is theoretically observed through the Darwinian framework of analysis [24].

The empirical observation of asymptotic instability, now better can be defined as the empirical observation of non-convergence of a given phenomenon, can be filtered in a nonlinear time series investigation by time acceleration of the event in to occur. This mathematical or empirical possibility could lead to a new iterated event’s projection defined as the sum of all dynamical fixed point, that are being observed as unstable in most of the nonlinear time series data. The time acceleration of event would lead to the function defined as a constant  $a$  where exponentially defines by its turn the constant  $b$  like the expressions,

$$\begin{aligned}
 a = f(x) &= \sum_{S_{n+1}}^n S_n - g(x) \text{ or } \sum_{S_{n+1}}^n S_n = f(x) + g(x) \text{ or } g(x) = \\
 &\quad -f(x) + \sum_{S_{n+1}}^n S_n \tag{8} \\
 b &= \sum_{t_{n+1}}^n S_n = S_1 + S_2 + S_3 + \dots + S_n \geq 0.
 \end{aligned}$$

Where for all  $g(x)$  at each  $S_n$ , there will be necessarily a value for  $g(x)$  that is not equal 0 hence predicting the system complexity for each iteration as in convergence to  $f(x)$  (asymptotic stability) for each iteration and at the same time the connectivity of iterations reveals in the light of empirical observation a growing or decreasing effect towards  $f(x)$ , but overall asymptote stability towards  $(x_{n+1} + t_{n+1})P_n \rightarrow \infty \therefore f(x) \sim g(x)$ .

For the third Eq. (8) form  $g(x) = -f(x) + \sum_{S_{n+1}}^n S_n$ , the empirical nonlinear time series express internal movements of which unpredictable trajectories are observed constantly as far as the system is perturbed by strong initial input of new variables within the system (system expansion).

Considering the constant  $b$ , roughly, as  $g(x) = -f(x) + \sum_{S_{n+1}}^n S_n$  as unpredictable, but in convergence to the  $f(g(x))$  fluctuations, for the constant  $b$  for only one iteration at a time, the empirical observation of phenomenon for the asymptotic stability effect can only be observed if considering  $g(x)$  as derivative of  $f(x)$  where this later function is assumed to be non-exponential in the sense of Lyapunov convergence, hence without nonlinearity aspects.

In another sense, for all derivatives  $g(x)$  that are produced for  $f(x)$  as a result of strong asymptotic instability and not only for a single observation or short time lengths observations, the iteration process might be better explained in its oscillations through an exponential function defined for  $f(g(x))$ , where the convergence rate of distant metric spaces between  $f(x)$  and  $g(x)$ , created in nonlinear time series observation, can be obtained as the overall growth or decrease of each iteration and nonlinear solutions produced by  $g(x)$  as,

$$(f(g(x))_1, f(x)_1), (f(g(x))_2, f(x)_2), \dots, (f(g(x))_n, f(x)_n), \tag{9}$$

in the form of,

$$- \sum_{S_{n+1}}^n f(g(x))_{x+1} e^{bS_n} + a \sum_{S_{n+1}}^n e^{bS_n} \tag{10}$$

as the  $b$  constant and exponential form, and,

$$\sum_{S_{n+1}}^n f(g(x))_{x+1} S_{n+1} e^{bS_n} - a \sum_{S_{n+1}}^n S_n e^{bS_n} \tag{11}$$

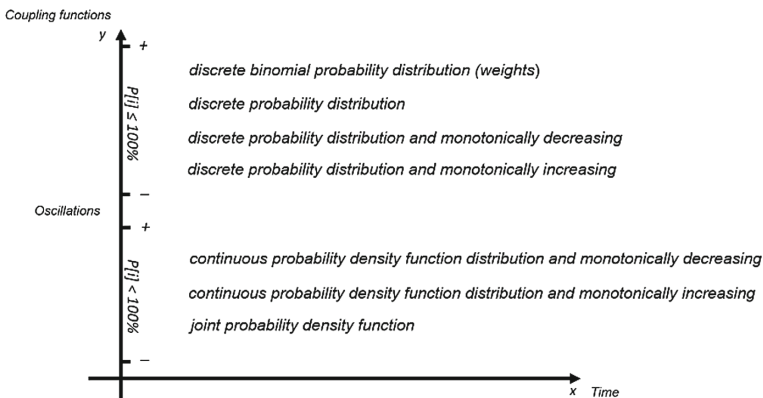
as the constant  $a$  with undefined exponential rate for  $b$ , due to complex variable's nature (discrete and continuous).

Q.E.D. □

### 3 Results

#### 3.1 Dynamical Fixed Points and Connectivity Metrics

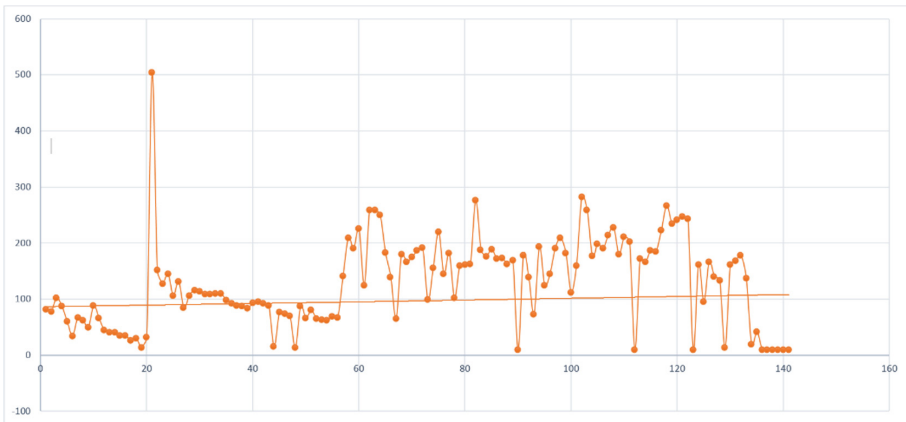
Following this proof, all given possible solutions directly reflect the maximum number of locally stable and unstable maps [22], being it the trajectories of the constants  $a$  and  $b$  generated by the functions  $f(x)$  and  $g(x)$  on time. However, it should be noted that in the expression  $S_n + 1 = f(x) + g(x)$  or  $f(x) - g(x)$ , it can be converted into  $\int_{g(x)}^{f(x)} (f^m \circ g^n)$ , where the sum of existing function  $g(x)$  can reach as many iterations as possible in the unbounded event, and many of the iterations can be identical or not. It is not possible to observe these event's variations in an empirical sense as far as the time length considered for analysis is not enough to express asymptotic exponential stability. Thus, although the asymptotic exponential stability is not visible at short time lengths, both functions  $f(x)$  and  $g(x)$  clearly differ in the initial condition of the event in terms of empirical properties of the iteration and its expressions within a context, for example, probabilistic distribution (Fig. 2) for linear/ nonlinear events which directly affects the ability of the observer to discretize the event and classify it accordingly. So when in  $f(x) \sim g(x)$  in terms of solutions, it is obtained a greater metric space among iterations and consequently  $g(x)$  assumes a greater distance from the fixed point of  $f(x)$



**Fig. 2.** Probabilistic distributions occurring in the complex variables due to coupling effects, where  $P[i]$  represents the deterministic to stochastic strings of distribution for a given event  $i$  to happen with 100% or less probabilities.

when considering the initial conditions of the phenomena. And in the opposite hand, if  $g(x) \sim f(x)$ , the uniqueness of the system can be observed in short time lengths, being this feature observable empirically at the beginning of experimentations, but not necessarily mean a deterministic conclusion.

These statements are very important for determining empirical phenomenon behavior and also many researches can possibly present failures in their findings since no time acceleration is possible to obtain and disproof asymptotic instability observations as it can be seen as one example in Fig. 3. Figure 3 represents the consumption of water by public schools during the period of 2007 until Oct., 2019, month by month [25]. During these nonlinear time lengths, it is possible to observe the oscillations of  $a$  constant that redefines  $b$  during the years. The exponential function (horizontal line) of  $b$  constant presents a gradual water consumption growth that is not observable and not correlated if considering previous year's data or short time periods data as an average or most frequentist data. In this phenomena characterization, the asymptotic instability observed during the years, thus express a stability feature that follows all dynamic fixed point solutions at each pattern formation of convergence and stability. The relative convergence of overall variables towards  $b$  constant reflects also the system adaptation to internal and external variables that influences the system modifying the  $a$  constant with deep instability.

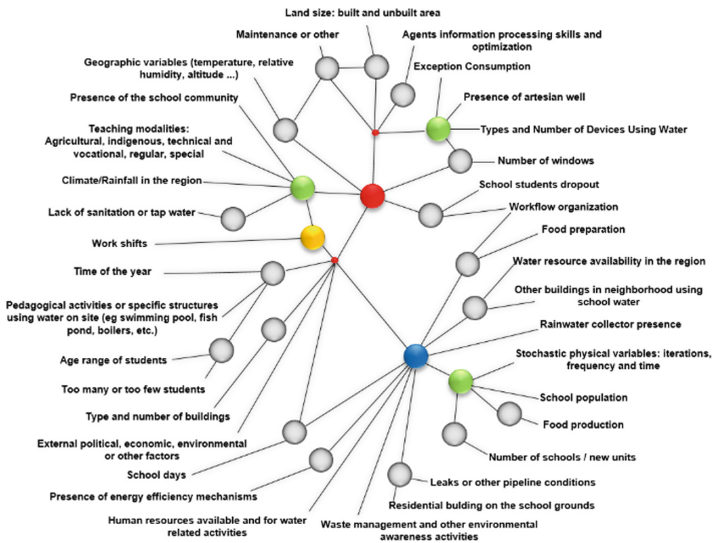


**Fig. 3.** Expression frequency of interactions and iterations on time between iterated functions from a dynamic fixed point in connectivity.

If the frequency with which specific fixed point attracts an iterated function is set to be equals for all  $S_n$  for a long period (considering time here as relative to the phenomena observed), hence exhibiting the same behavior as described in the methodology section as a Cartesian interaction product (linearity of functions behavior), the false asymptotic instability effect ceases as the distances between dynamic fixed points for each of the interactions in connectivity are no matter larger or short. Some resource usage, depending on the context and variables that generate the event, have this function performance, not equal to Fig. 3 properties.

One of the most difficult aspects of determining a fixed point convergence within nonlinear dynamics time series, would be if both functions and iteration frequencies towards instability do not match in either position  $S_n$ ,  $S_{n+1}$  and  $S_{n-1}$  for any kind of connectivity aspect in  $f(g(S_n))$  (attraction) or  $g(f(S_n))$  (pattern formation).

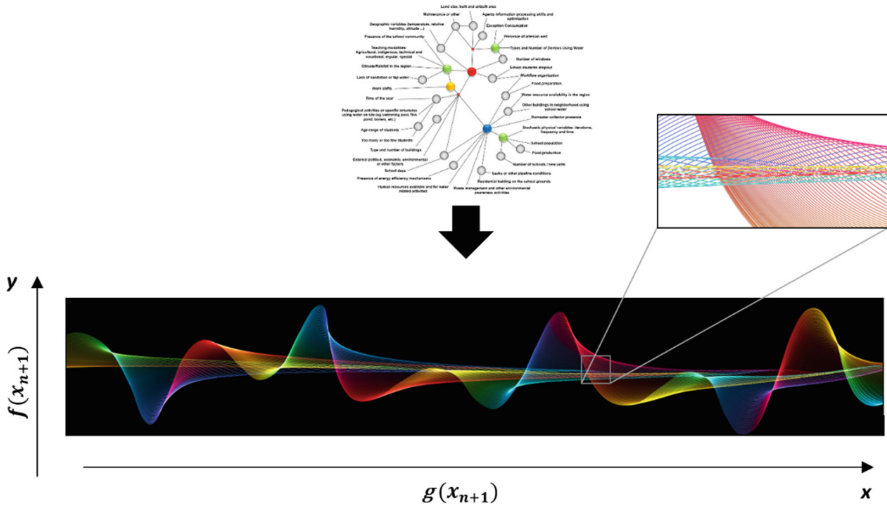
The main point here is to make the difference between theoretical and empirical observations. For the first, a mathematical framework can lead to the correct observation of a given phenomenon as far as defined initial conditions are existent. But consider now an empirical sense of observation. The frequency that it occurs on time for input and output within a control system, where for large periods of time the event does not change as well as its initial conditions leading it to the notion of Lyapunov stability, but as for the input variables, if the event present high frequency and nonlinear behavior of oscillations, it can assume an exponential behavior, thus differing from the perfect mathematical framework with only theoretical analysis. Figure 4 displays all variables involved in water usage in public schools within a sample of 2143 distributed in a geographic region of 199,315 km<sup>2</sup>. Just as an example, this figure illustrates the questions of how to define factors of iteration, interaction, frequency, time, space and other physical property aspects of each one of the 35 variables and each possible composition of variables. Considering that each unit has a specific time-invariant dynamic and there is no formation of determined patterns for the variables as an initial condition, therefore, thinking in public administration where big cities will be the challenge to support massive complex interactions, it is possible to scope the issues of how to define a policy making for all administrative units based in a pattern formation that is ranged for all samples with undefined pattern formation or at least if it is possible to make a classification of groups



**Fig. 4.** Several configurations might appear with distinct strengths at each variable connectivity and coupling effects. What assumes a given probability distribution to one sample, in another one can assume complete different type of distribution, hence modifying phenomena functioning and composition.

of units that share same pattern behavior; what is the ideal sequence, how to control, which dynamics assumes, what variance, which variable has the greatest influence on each other and how often does it occur.

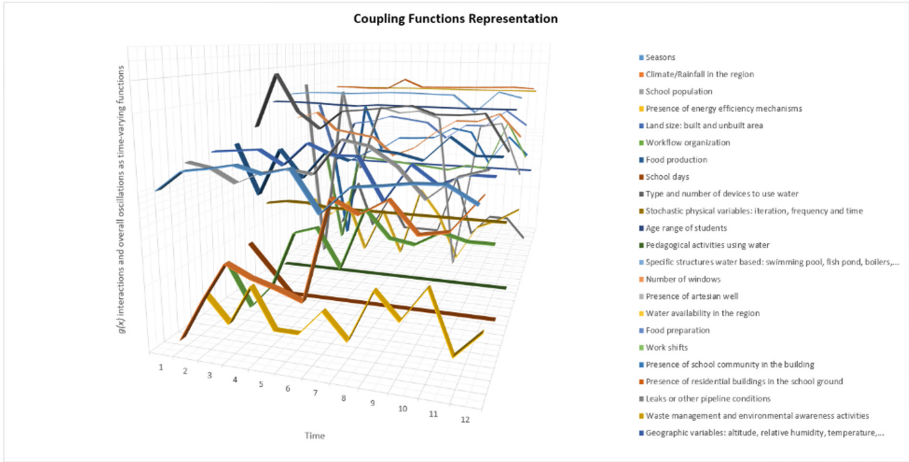
The behavior of variables of Fig. 4 can also be illustrated by Fig. 5, where  $g(x)$  can produce strings of variables defined heterogeneously within different time intervals, possibly generating nonlinear pattern formation through time lengths or at least, as this research an asymptotic stability effect towards the whole system as an objective pattern of complex configuration. Note that this figure was used to illustrate the observation of the phenomena with graphical aids.



**Fig. 5.** Oscillatory strings of complex variables and the coupling functions of  $f(x_{n+1})$  and  $g(x_{n+1})$ , represented by a stylized graph. In this ideal representation, for practical observation purposes, the strings for each  $g(x)$  iteration are assuming relatively harmonic and symmetric oscillations through time intervals, hence reflecting this behavior towards  $f(x)$  relative stability and  $f(g(x))$  fixed point convergence. The constant  $b$ , for visual reference, can be observed as a continuous string through oscillations, being composed by complex variables of  $g(x)$  and structured by  $f(g(x))_{x+1}$  dynamical fixed point convergence.

The scenario represented by Fig. 5, in the opposed direction can be viewed in Fig. 6 as representing asymptotic instability, therefore, reflecting the Fig. 3 phenomena and Fig. 4 variables, as an example of how the strings of variables under time lengths might assume oscillatory behavior being complicated to generate the same behavior as defined in Fig. 5.

Note that this same results approximation can be seen at Monte Carlo calculations, but differing in terms of finding how the mathematical expectation of the not *i.i.d.* complex variables iterations and interactions of  $g(x)$  might express as outcomes towards time and system own coupling functions dynamics [26], and thus not defining any visible weight to the probabilistic distributions oscillations (visible by a central limit theorem). This approach is, for example, different from the Tang [27] framework that identifies weights



**Fig. 6.** Representation of a strings of variables (as an example) that assume time-varying probability distributions and several possible synchronization states. In order to adjust the nonlinear system, it is necessary to understand a possible pattern formation of the variable's oscillations and promote modifications at the macroscopic dimension of the system, being it, depending on the considered system of analysis, composed of organic and inorganic components.

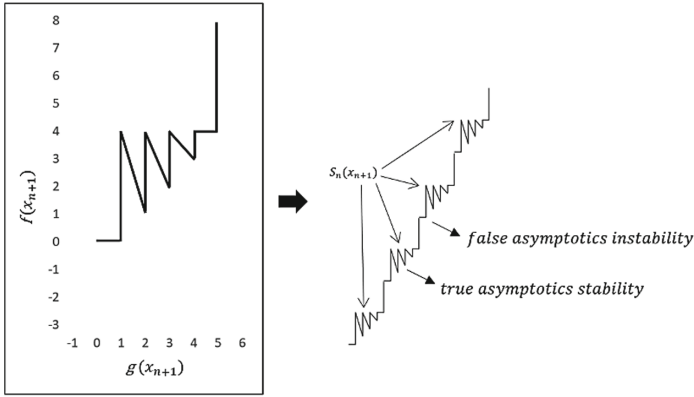
and the correlation of the system with the central limit theorem, thus resembling the uniqueness concept of stability.

Also, this interactive system and probabilistic distributions share not only physical components within the coupling relations, but there are biological agents [26, 28] that causes in many distinct aspects, influences over system functioning. In the view of modern scientific breakthrough, analyzing nonlinearity in the light of public administrative policy and infrastructure [29] are demands of investigations that can constitute a path to establish complex solutions for social systems that present nowadays high level of non-convergence and artificial oscillatory dynamics.

### 3.2 False Asymptotic Instability at Iterated Functions

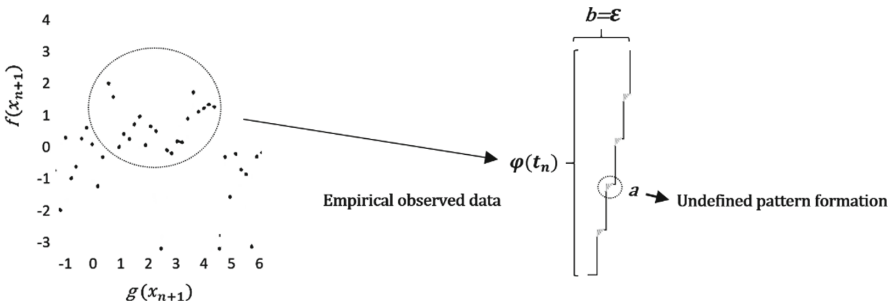
Despite of a false asymptotic instability effect might exist in an iterated function analysis, this research points out that since time lengths poses the true camouflage effect of the observed phenomena, it can be also used as a tool to design new results towards the quantitative aspect of physical, chemical or biological properties of reality [4, 8].

Considering all the nonlinear time series given in Fig. 3, the average largest and smallest distances of iterated functions from the dynamic fixed point in  $f(x)$  can be obtained by viewing the maximum distances between  $f(x)$  and  $g(x)$  by the exponential function defined as crescent towards  $b$  and two phase spaces can be defined within the available data of Fig. 3 as the smallest distances and frequencies of event whole iterations and interactions (phase I – Fig. 7), and depending on the empirical observation over position  $S_n$ , which increases to the time length of iterations and interactions of the event as a non-canonical phase space, the maximum exponential growth distance can be identified as the  $b$  constant (phase II – Fig. 7).



**Fig. 7.** Nonlinear time series whole representation of  $\sum_{S_n+1}^5 f(g(x))$  for a theoretical interval, identifying two defined phase spaces occurring, divided by phases I and II evidencing the distribution of the iterated phenomena between distinct functions  $f(x)$  and  $g(x)$  for a dynamics of true asymptotic stability and false asymptotic instability. Overall the event presents an exponential Lyapunov equilibrium, continuously iterating with the unobservable pattern formation of the stability property (Fig. 8).

Analyzing phase spaces in the iterated functions at a dynamic fixed point, the vectors variations that can be obtained by a fixed point dynamics in terms of time/space/other physical property observations. Representing in Fig. 8, a sequence of dynamic fixed points that occur as in Fig. 3 is presented, and it is possible to observe the iterated functions convergence illustration.

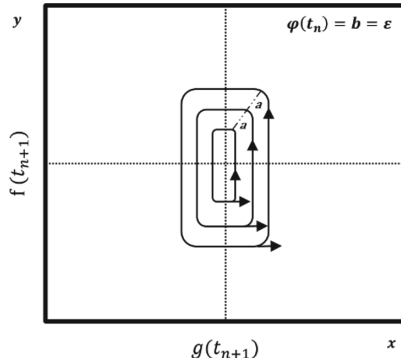


**Fig. 8.** Illustrating Fig. 2 and 3 information: false asymptotic instability behavior at iterating functions with Lyapunov  $a$  stability and  $b$  exponential stability in nonlinear time series.

By observing Fig. 3 properties, empirical object of this research, it is possible to realize that the dynamics between asymptotic true and false events remains as a defined phase space at Fig. 9.

It is not unusual asymptotic analysis are related to time aspects, leading to conclusions that confirm the asymptotic instability behavior rather than time effects over it. Empirical





**Fig. 9.** Phase portrait of the example shown in this research showing possibility of Lyapunov exponential stability formation for some empirical phenomena.

data observations can easily be interpreted as an asymptotic instability expression of the phenomena while time can be the true cause of phenomena expression of stability [4], making asymptotic analysis and conclusions rather a false observation of the experiment, simulation or natural observation.

## 4 Discussion

Globally speaking, the entire event presents a Lyapunov exponential stability due to the constant  $b$  and average orbits interaction growth towards the nominal value of constant  $b$ , defined by specific dynamical fixed points.

In several events involving iterations, such as physics, biology and chemistry phenomena and related knowledge, iteration events are manifestations of repetition in the order or disorder of elements that constitute an iteration [30, 31]. Although the cause of iterations as well as their behavior are key topics as the coupling functions [32] in the search for iterated events. An important aspect raised in this research reflects on the discretization of iterated events in terms of knowing how they behave and how to reproduce the same event under new experimental conditions. Empirically, many events are difficult to predict [4], which also affects the possibility of identifying a possible convergence in the proposed solutions of stability [33] as it can be seen in the empirical results and methods of Bastin *et al.* [34], Bingtuan *et al.* [35], Di Francesco *et al.* [36] and Maron *et al.* [37].

One important aspect not discussed at the methodology and results sections is that even if the system has influence of new variable's input, thus modifying system stability behavior, that was described in this research as a possible false instability effect, the global behavior through time assumes possibly a new pattern formation, variances that change time to time, thus characterizing the stability aspect defined in this research for a linear or nonlinear time series. Similar approaches were also investigated to cover nonlinear exponential function behavior rate through time, as defined in Chadwick *et al.* [38], but not defining time as an empirical constraints and database for the mathematical analysis of the event. Also, at Zhou [39], auto distance correlation function was defined

from Szekely et al. (2007), to measure nonlinear time series, however, not defining coupling functions and its physical topological properties in a broader framework of analysis.

If there is an initial definition of a maximum event with probabilities, the expected value of the frequency with which these probabilities occur will be limited to a maximum and a minimum of variation of  $f(g(S_n))$ , thus defining itself by the functions convergence to the given fixed point and its orbits proximity, being it an artificial or natural expression. This same approach, not considering probabilities measure, was performed by Dionisio *et al.* [40] for financial market (stock indexes) nonlinear time series, however, even with empirical constraints considered, and also notes on the time lengths limitations to the observation of the event, dynamic fixed points are considered as a dependent rate and expression of variables involved, hence defining the nonlinear phenomenon as stochastic in its finite analysis and predictable within a long term analysis (time lengths). However, the model considers the concept of dynamic fixed points without the evaluation and approaches of how the mechanisms in which coupling functions might assume in the given event, can be tracked or even manipulated. This might lead to the assumption that dependent variables are time-varying expressions in which the possibility of prediction is directly proportional to the use of a surrogate data, empirical values and autonomous observations, without the induction of dynamic fixed points existence and stability pattern formation.

These limits define that no matter how much empirically the number of elements in the process increases or assume an ordered arrangement, the maximum relative frequency of convergence does not exceed its limits since the initial function is already defined at a fixed point, that is, of the rate of convergence. That aspect of engineering solutions to complex systems has great importance for contemporary science in many fields of knowledge. This phenomenon can be very related to weaker and strong law of large numbers properties where initial conditions of phenomenon can present constant instability expression due to short time lengths in observation, that reflects as well the number of observations in a given phenomenon. As pointed at methodology section, analogously, the weak and strong law of large numbers if considered only theoretically, it does not express the same empirical aspect of convergence that does necessarily imply time dependency to prove overall exponential or not exponential stability of a given phenomenon studied.

For all interactions that may not have an initial solution given and have a high rate of instability, the empirical aspect can express the tendency of the event to balance oscillations as it does not reach infinite randomness or present great difficulty to predict. But in the other hand, events with low rate of convergence, affect the definition of the oscillation convergence more, because time allows the expression of variations of possible outcomes or unsolvable solutions that intensify through time. This asymptotic instability effect can't be observed theoretically with traditional mathematical definitions since theoretical procedures can't be time accelerated with empirical domain constraints and excluding this alternate dimension of experiment, it leads to a prove by simple deduction that the phenomena property of false asymptotic instability can't exist.

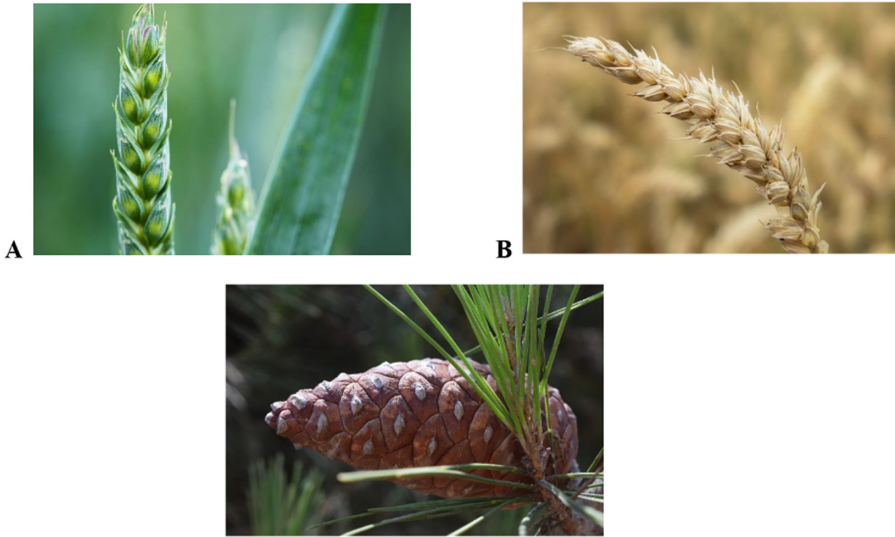
In the research done of Thomas et al [41], experimentally, was proved to the contrary of the dynamic fixed points concept. However, with new empirical expressions

of these same events, the asymptotic instability effect might be observable within the noises encountered among empirical and theoretical definitions with solutions not yet proposed for this apparent instability detected due to empirical constraints and the type of experimentations of the study and this should be treated as far as new empirical experimentations, simulations or frameworks can be understood, within a new form of analysis.

If we consider, for example, atmospheric events, in a temperature range between distinct air masses, to be considered as a set A cold mass and a set B hot mass, the interaction between these events necessarily generates a region of instability C, caused mostly by iteration properties. This example helps to illustrate that the dynamics of continuously iterating atmospheric particles, according to the methodological definitions of this research, will express behavior in which regions of the physical space under which, it creates a dynamic fixed point, express symmetric sets A and B, in turn modifying themselves, generating densities of iterations and interactions over space, formed to the prior evolution of the event, by a growth or decrease move of all dynamical fixed points over the time. Dynamical fixed point reveals that regions of space in which particles will have the highest and lowest time length of interaction and iteration to achieve exponential stability, will affect the physical expression effect of particles in physicochemical aspects. The duration of the time length of iterations and interactions in different regions of space allows both sets A and B (hot and cold air mass) to express higher or lower time influence on the event with expressions of greater or lesser influence on the linear and nonlinear dynamics of the physical and chemical phenomena effects of the event.

One last important aspect of this research resembles on the  $x$  nominal value considered for analysis of convergence. In respect to the plural form of the physical world, not only the time, as used in this research, could resemble connectivity and stability property of a given phenomenon. It means the physical space as well as other physical properties or nominal values of nonphysical parameters can be used as a tool to identify the same research aspect of convergence. One way of empirically observing the expression of iterated functions in connectivity on time, similar to the descriptions presented in this research, would be in Fig. 10. In the left image [42, 43] a chemical event of wheat is represented in which the time length of iterations and interactions has a false asymptotic instability at the beginning of the event A (higher time length of iterations that gives a certain physicochemical property), and asymptotic stability after the initial phase of event B (the loss of physicochemical properties that give rigidity to the material, which is understood to be an asymptotic stability of the event). And in the image at bottom, the closed pine cone biological structure in its fractals and shape [44] can be observed for its asymptotic stability (regular time lengths) at the center of the structure, obtaining maximum interactions and determined time length of iterations, and in the final portion of the structure (tapered) it is possible to observe the false asymptotic instability (higher expression of a given function on time) in which the interactions between the functions become smaller and, consequently, the frequencies on time with which each iterated function expresses become larger.

Note that there might be some variations on the fractal and the form of the closed pine cone as well as rigidity or bendiness of wheat structure due chemical or biological



**Fig. 10.** Representation of an event with a possible Lyapunov exponential stability and its stable and unstable regions of convergence.

factors such as defective proteins, genes or other environmental conditions. But it does not change the mathematical modeling of describing in this research.

## 5 Conclusion

One first conclusion of the theoretical research conducted, was to demonstrate that in nonlinear time series analysis the asymptotic instability observed with empirical data sets, if time accelerated, this feature reveals itself as an asymptotic stability behavior oriented by an attractive distribution. One technical conclusion arising from it and the results section is about when investigating the empirical behavior of complex events, many conducted experiments can also fail when trying to identify asymptotic stability behavior on events, mainly defining this asymptotic stability behavior as unstable, unpredictable and without any prior pattern formation.

Also, deriving from this, the mathematical observation through theoretical point of view can fail in identifying the false instability effect due methodology constraint caused by the lack of inductive inference. This point leads the methodology proposed in this article as an observation to most scientific data concerning fixed point stability that is disconnected of empirical data sets.

A second conclusion states that considering that iterated complex variables can possibly not present a pattern formation for short time lengths, the average distance between fixed points is also followed by a low rate of convergence (instability), that in terms of scientific empirical interpretation can be understood as the phenomenon not presenting a high level of interactions expected to be in convergence to any given fixed point, but oriented to the maximum and minimum range of metric spaces defined within

it (stability). Though, these interactions are existent in the extent of time of empirical observation be enough greater than the periodic time in which complex variables interact itself resulting in many infinite oscillation expression patterns, also leading to the concept of dynamical fixed point existence. This could lead to the statement that if a phenomenon presents a very high level of complexity, the outcomes of convergence also present higher time periods to be able to expand it into a new degree of pattern formation. This time-varying patterns of convergence and non-convergence at specific time lengths can also be confirmed theoretically and empirically by many researches performed by Aneta Stefanovska and PVE McClintock, as well as the axioms of the Shannon's theory of communication, the law of great numbers physical properties and monotone functions as well.

Also another aspect of detecting convergence or non-convergence, distinct manifolds of a given phenomenon can present constrained expression over time lengths, leading the observer to the use of other physical/chemical/biological exponential oriented constant (nonautonomous differential equations).

**Funding.** This research received no external funding.

**Conflicts of Interest.** The author declares no conflict of interest.

## References

1. Langa, J.A., Robinson, J.C., Suárez, A.: Stability, instability, and bifurcation phenomena in non-autonomous differential equations. *Nonlinearity* **15**(3), 887 (2002)
2. de Bruijn, N.G.: An asymptotic problem on iterated functions. In: *Indagationes Mathematicae (Proceedings)*, North-Holland, 1 January 1979, vol. 82, no. 1, pp. 105–110 (1979)
3. Crutchfield, J.P.: The calculi of emergence: computation, dynamics and induction. *Phys. D: Nonlinear Phenom.* **75**(1–3), 11–54 (1994)
4. Newman, J., Lucas, M., Stefanovska, A.: Limitations of the asymptotic approach to dynamics. arXiv preprint [arXiv:1810.04071](https://arxiv.org/abs/1810.04071) (2018)
5. Grujic, L., Siljak, D.: Asymptotic stability and instability of large-scale systems. *IEEE Trans. Autom. Control* **18**(6), 636–645 (1973)
6. Ignatyev, A.O., Ignatyev, O.A., Soliman, A.A.: Asymptotic stability and instability of the solutions of systems with impulse action. *Matematicheskie Zametki* **80**(4), 516–525 (2006)
7. Stankovski, T., Duggento, A., McClintock, P.V.E., Stefanovska, A.: Inference of time-evolving coupled dynamical systems in the presence of noise. *Phys. Rev. Lett.* **109**(2), 024101 (2012)
8. Lucas, M., Newman, J., Stefanovska, A.: Stabilization of dynamics of oscillatory systems by nonautonomous perturbation. *Phys. Rev. E* **97**(4), 042209 (2018)
9. Lundberg, A.: On iterated functions with asymptotic conditions at a fixpoint. *Arkiv för Matematik.* **5**(3), 193–206 (1964)
10. Williamson, D.: Dynamically scaled fixed point arithmetic. In: *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing Conference*, 9 May 1991, pp. 315–318. IEEE (1991)
11. Yadeta, Z.: Lyapunov's second method for estimating region of asymptotic stability. *Open Sci. Repos. Math. Online (open-access)*, e70081944 (2013). <http://www.open-science-repository.com/lyapunovs-second-method-for-estimating-region-of-asymptotic-stability.html>
12. Matetski, K., Quastel, J., Remenik, D.: The KPZ fixed point. arXiv preprint [arXiv:1701.00018](https://arxiv.org/abs/1701.00018) (2016)

13. Arnold, L., Schmalfuss, B.: Lyapunov's second method for random dynamical systems. *J. Diff. Equ.* **177**(1), 235–265 (2001)
14. Natoli, C.: *Fractals as fixed points of iterated function systems*. University of Chicago, 26 August 2012
15. Bharucha-Reid, A.T.: Fixed point theorems in probabilistic analysis. *Bull. Am. Math. Soc.* **82**(5), 641–657 (1976)
16. Gupta, A., Jain, R., Glynn, P.: Probabilistic contraction analysis of iterated random operators (2018). arXiv preprint [arXiv:1804.01195](https://arxiv.org/abs/1804.01195). Submitted to *Ann. Appl. Probab.*
17. Murray, R.M.: *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton (2017)
18. Jamison, B.: Asymptotic behavior of successive iterates of continuous functions under a Markov operator. *J. Math. Anal. Appl.* **9**(2), 203–214 (1964)
19. Zou, C.: *Lecture Notes on Asymptotic Statistics*
20. Kiel, L.D.: The evolution of nonlinear dynamics in political science and public administration: methods, modeling and momentum. *Discrete Dyn. Nat. Soc.* **5**(4), 265–279 (2000)
21. Kantz, H., Schreiber, T.: *Nonlinear Time Series Analysis*, vol. 7. Cambridge University Press, Cambridge (2004)
22. Zdun, M.C.: The embedding problem in iteration theory. *ESAIM: Proc. Surv.* **1**(46), 86–97 (2014)
23. Arab, I., Oliveira, P.: Asymptotic results for certain weak dependent variables. *Theory Prob. Math. Stat.* **2**(99), 19–36 (2018)
24. Darwin, C.: *On the Origin of Species 1859*. Routledge, Abingdon (2004)
25. Telles, C.R., et al.: Supplementary material: excel datasheet of water usage metrics at Secretary of State for Education and Sport of Paraná, Brazil, Curitiba, November 2019. <https://doi.org/10.13140/rg.2.2.15499.85285>
26. Hagos, Z., Stankovski, T., Newman, J., Pereira, T., McClintock, P.V.E., Stefanovska, A.: Synchronization transitions caused by time-varying coupling functions. *Philos. Trans. R. Soc. A* **377**(2160), 20190275 (2019)
27. Tang, X.: Some strong laws of large numbers for weighted sums of asymptotically almost negatively associated random variables. *J. Inequal. Appl.* **2013**(1), 4 (2013)
28. Telles, C.R.: A mathematical modelling for workflows. *J. Math.* **2019**, 23 (2019)
29. Delorme, R.: *Deep Complexity and the Social Sciences*. Edward Elgar, Cheltenham (2010)
30. Harmati, I., Hatwagner, M.F., Kóczy, L.: On the existence and uniqueness of fixed points of fuzzy cognitive maps. In: *Proceedings of the Information Processing and Management of Uncertainty in Knowledge Based Systems. Theory and Foundations - 17th International Conference, IPMU 2018. Communications in Computer and Information Science*, pp. 490–500. Springer (2018)
31. Fromm, J.: On engineering and emergence. arXiv preprint [arXiv:nlin/0601002](https://arxiv.org/abs/0601002), 3 January 2006
32. Stankovski, T., Pereira, T., McClintock, P.V.E., Stefanovska, A.: Coupling functions: dynamical interaction mechanisms in the physical, biological and social sciences. *Philos. Trans. Ser. A Math. Phys. Eng. Sci.* **377**(2160), 20190039 (2019)
33. Wu, M., He, Y., She, J.-H., Liu, G.-P.: Delay-dependent criteria for robust stability of time-varying delay systems. *Automatica* **40**(8), 1435–1439 (2004)
34. Bastin, G., Gevers, M.R.: Stable adaptive observers for nonlinear time-varying systems. *IEEE Trans. Autom. Control* **33**(7), 650–658 (1988)
35. Li, B., Wolkowicz, G.S.K., Kuang, Y.: Global asymptotic behavior of a chemostat model with two perfectly complementary resources and distributed delay. *SIAM J. Appl. Math.* **60**(6), 2058–2086 (2000)

36. Di Francesco, M., Lorz, A., Markowich, P.: Chemotaxis-fluid coupled model for swimming bacteria with nonlinear diffusion: global existence and asymptotic behavior. *Discrete Contin. Dyn. Syst. Ser. A* **28**(4), 1437–1453 (2010)
37. Maron, J.L., Horvitz, C.C., Williams, J.L.: Using experiments, demography and population models to estimate interaction strength based on transient and asymptotic dynamics. *J. Ecol.* **98**(2), 290–301 (2010)
38. Chadwick, E., Hatam, A., Kazem, S.: Exponential function method for solving nonlinear ordinary differential equations with constant coefficients on a semi-infinite domain. *Proc.-Math. Sci.* **126**(1), 79–97 (2016)
39. Zhou, Z.: Measuring nonlinear dependence in time-series, a distance correlation approach. *J. Time Ser. Anal.* **33**(3), 438–457 (2012)
40. Dionisio, A., Menezes, R., Mendes, D.A.: Mutual information: a measure of dependency for nonlinear time series. *Phys. A: Stat. Mech. Appl.* **344**(1–2), 326–329 (2004)
41. Thomas, S.C., Jasienski, M., Bazzaz, F.A.: Early vs. asymptotic growth responses of herbaceous plants to elevated CO<sub>2</sub>. *Ecology* **80**(5), 1552–1567 (1999)
42. Pixabay. Didgeman's image: bended old wheat, 13 July 2019. <https://pixabay.com/photos/wheat-cereals-grain-cornfield-4335863/>
43. Pixabay. Public domain pictures: green wheat, 5 June 2009. <https://pixabay.com/photos/agriculture-background-cereal-corn-2229/>
44. Pixabay. Ulleo's image: closed pine cone, 7 June 2016. <https://pixabay.com/photos/pine-tap-pine-cones-nature-closed-1457849/>



# The Influence of Methodological Tools on the Diagnosed Level of Intellectual Competence in Older Adolescents

Sipovskaya Yana Ivanovna<sup>(✉)</sup>

Institute of Psychology, Russian Academy of Sciences, Moscow State Psychological and Pedagogical University, St. Yaroslavskaya, 13, 129366 Moscow, Russia  
syai@mail.ru

**Abstract.** Adolescence presents dramatic qualitative changes in the physical, intellectual, personal and spiritual aspects. Intellectual development is the period determined by the maximum resolution due to the maturation of conceptual and metacognitive abilities, which, in turn, are necessary for productive actions in a specific subject area - intellectual competencies. The level of formation of constructive intellectual reliability was measured by two similar methodological guidelines - “Composing on a free topic” and “Interpretation of the moral dilemma”. However, the results were not. Study participants: 180 students of secondary schools in Moscow over the age of 15 years. Attributes: “essay on a free topic” and “interpretation of a moral dilemma” Thus, it should be noted that the structure of the numbers of manifestations of intellectual competence associated with the initiation of energy support, support for intellectual activity, a low degree of differentiation of the formation of this construct.

**Keywords:** Intellectual competence · Methodological tools · Adolescence

## 1 Introduction

Scientists often encounter a lack of existing teaching methods when conducting empirical studies of manifestations of intellectual competence. Often new tests, questionnaires, and other work tools are developed. So we did, defining competency as a systemically organized mental experience [2, 6, 7], which makes it possible to achieve high practical results in any particular subject area [3, 4, 12]. Moreover, in our studies, intellectual competence is singled out as a basic one, which provides the opportunity to develop professional, more specific types of competence. We have developed 2 methods - “Composing on a free topic” and “Interpretation”, each of which measures the level of formation of intellectual competence [10, 11, 13]. However, the question of the specificity of the influence of each of these methods on the results was not raised and, accordingly, the answer to it was not received.

In the matter of choosing the age of the study participants, we settled on the older adolescent in view of its criticality for many areas of human life, including the mental one.

Thus, the variables in this study were manifestations of intellectual competence.



## 1.1 Research Questions

Theoretical hypothesis of the study:

- Indicators of intellectual competence, measured by means of the “Composing on a Free Topic” methodology, and indicators of intellectual competence, measured by means of the “Interpretation” methodology, differ slightly.

Research hypotheses:

- There are no significant differences between older adolescents in the level of severity of the indicator of intellectual competence (in terms of narrative activity), measured using the methodology “Composing on a free topic” and the methodology “Interpretation”.

## 1.2 Purpose of the Study

Purpose: To reveal the specifics of the influence of teaching methods on the results of measuring intellectual competence in schoolchildren of older adolescents.

The objective of the study: to determine the degree of influence of the methodological specificity factor on the results of measuring intellectual competence in older adolescents.

Thus, the subject of the study is the manifestations of intellectual competence, measured by two different methods. The object of study is school children of older adolescents.

## 2 Study Participants

Study participants: 180 schoolchildren (91 girls and 89 boys) aged 15 years.

## 3 Research Methods

Methodological tools: “Interpretation” [9] and an essay on a free theme [8]. At the same time, 90 participants in the study performed tasks according to the Composing methodology - index 1, while 90 older adolescents - according to the Interpretation technique - index 2.

### 3.1 Methodology for Assessing Intellectual Competence “Composition” [8]

The work reveals the features of structuring data on a particular subject area, and also allows you to identify the features of the conversion of this data when generating a new text. The qualitative characteristics of the essay were considered by us as a manifestation of the student’s intellectual competence in the context of real educational activity, in which students are included.

To write an essay, students were provided with two white sheets of A4 format, on which it was proposed to write an essay on one of the three proposed topics or on any other randomly chosen topic. The topics proposed were:

- “The Great Patriotic War” (war as a phenomenon of the historical development of mankind; war: reasons and results; war through the eyes of a character, for example, war through the eyes of a soldier of the Soviet army);
- “Russia at the beginning of the 20th century”;
- “Correlation of Divine Providence and the evolutionary theory of C. Darwin”;
- “Worlds in the novel of Leo Tolstoy: peace as a truce, the temporary absence of war and peace as a description of life and mores of various segments of the population of Russia in 1812.”

The proposed essay themes were given in an extremely general wording, so as not to create strict attitudes in students, to give them the opportunity to reveal their personal preferences. If the students did not like any of the three proposed topics, then they could independently formulate a topic and write an essay on it. The students were not told about any standards for the size of the essay; they were only informed that they needed to write as much text as they deemed necessary to reveal the topic.

Participants in the study chose the theme of World War II because it corresponded to the theme of the school subject “Domestic History”, which students studied at the time of the empirical study.

The measure of complexity of the generated text was evaluated according to the following criteria: 0 points - the absence of a written essay; 1 point - a formally written essay that describes descriptive judgments and does not express its own point of view; 2 points - an essay with the establishment of causal relationships; 3 points - expressing one’s own attitude to the problem in the presence of causal relationships; 4 points - writing two essays on the same topic.

Since the essay as a generated text was considered as a manifestation of intellectual competence, a more detailed analysis of the texts of the essays was undertaken. The unit of the analysis was sentences as units of text. In the classification of proposals were highlighted:

- 1) propositions of a factual type (description/statement of facts; Fact; for example, “The Battle of Moscow took place in 1941”; “It will overgrow with grass”);
- 2) proposals of a systematizing type (allocation of a general category, construction of a hierarchical sentence; System; for example, “The battle of Stalingrad consisted of 3 stages: 1st stage: defensive; 2nd stage: fighting for the city; 3rd stage: counteroffensive”);
- 3) suggestions of an argumentative type (explanation, argumentation of any statement; Argum; for example, “He has not eaten for several days: in Leningrad, hunger”);
- 4) sentences of interrogative type (sentences-questions; Question; for example, “Could I go into battle?”; “Will I be able to find him tomorrow?”);
- 5) interpretative sentences (withdrawal into an alternative or more general context; Inter.; for example, “But if we had not defeated Napoleon, the whole world would have been under French rule, which is also not good”);
- 6) sentences of an emotionally-evaluative content type (impersonal assessment in a broad category; Content.; for example, “So let’s remember heroism!”; “It was a terrible time”);

- 7) suggestions of an emotionally-evaluative personality type (statement of a personal attitude to the described events; Personal; for example, “I believe that this is real heroism”; “I cry when I watch movies about the war”).

An individual protocol counted the number of sentences of each type.

Indicators: 1) a measure of the complexity of the text of an essay as an indicator of intellectual competence, in points; 2) the number of offers of different types.

### 3.2 Methodology “Interpretation of the Moral Dilemma” for Assessing Intellectual Competence [9]

Interpretation (essay) on the topic of one of the moral dilemmas of A.I. Podolsky and O.A. Karabanova ([1, p. 57–61]) is a type of text that, along with the characteristics of its content and features of its argumentation, reflects a motivational and personal attitude to the situation [5, 14]. Qualitative characteristics of the text were considered as a manifestation of the intellectual competence of the student.

To write an essay, students were provided with one A4 white sheet on which they were asked to write as much text as they saw fit to interpret the following moral dilemma (the moral dilemma was presented orally to the study participants):

“In the summer, Kolya and Petya worked in the garden - picking strawberries. Kolya wanted to buy sports watches with the money he had earned, which he had been eyeing for a long time. Kolya is from a low-income family, so parents cannot buy him such a watch. Petya wants to improve his computer with the money he earned.

Kolya is significantly inferior to Petya in strength and dexterity, and he rests more often, so Petya collected much more strawberries. In the evening, the foreman came to pay the guys for the work done. Counted the strawberry crates collected by both guys. He counted out the amount they earned and asked, turning to Petya: “Well, guys, pay evenly, or did someone collect more, is he supposed to get more?”

A measure of the complexity of the generated text was evaluated according to the following criteria: 0 points - the absence of a written essay; 1 point - formally written essay, which featured descriptive judgments and not expressed point of view; 2 points - an essay with the establishment of causal relationships; 3 points - expressing one’s own attitude to the problem and/or applying an analogy from another context in the presence of causal relationships.

Along with a general assessment of the complexity of the text, G., a more thorough analysis of the texts was undertaken. The unit of the analysis was sentences as units of text. The same types of sentences were highlighted as in the analysis of the text of the essay:

- 1) offers of a factual type (description/statement of facts; F; for example, “The boys worked together”, “Kolya cannot buy a watch because he is from a low-income family”);
- 2) proposals of a systematizing type (allocation of a general category, construction of a hierarchical sentence; S; for example, “There are several options: divide the money equally, tell the truth, and then divide the money equally or just say it as it is”);

- 3) suggestions of an argumentative type (explanation, argumentation of any statement; A.; for example, “You need to divide the money equally, because they are friends”, “You cannot indulge Kolya’s laziness, otherwise he will get used to it and will think that everyone owes him to help”);
- 4) sentences of interrogative type (sentences, questions; Q.; for example, “But will it benefit him?”, “Can they be friends and generally communicate with each other after that?”);
- 5) suggestions of an interpretative type (withdrawal to an alternative or more general context; I.; for example, “Friendship is like a fraternity, quite another, above money”, “We must try to establish good relations with all people, not just friends”);
- 6) sentences of an emotionally-evaluative content type (impersonal assessment in a broad category; C.; for example, “Friendship is the greatest value!”, “Friends must be valued, otherwise you will lose and never return”);
- 7) suggestions of an emotionally-evaluative personality type (statement of personal attitude to the described events; P; for example, “I’m sure that you can’t indulge laziness”, “I would share everything equally, otherwise I won’t be able to sleep peacefully after that”).

In each individual protocol, the number of sentences of each type was counted.

Indicators: 1) overall score as an indicator of the level of intellectual competence; 2) the number of offers of different types.

## 4 Results

Statistical processing consisted of the “Many Traits, Many Methods” method (MTMM) within the framework of the “R Studio” package, the results of which are presented in Table 1:

Notes: the validity index for measurements sentences of an emotionally-evaluative content type is 0.10, suggestions of an emotionally-evaluative personality type with indexes ‘1’ and ‘2’ – 0,03 and 0,07 respectively and the validity index for suggestions of an interpretative type ‘2’ is 0,31; the color markers of the table are decrypted as follows:

	Validity Indicators
	Reliability indicators
	Indicators of features of the monomethod ("Composition")
	Monomethod traits (“Interpretation of the moral dilemma”)
	Indicators of different traits - different methods ("Composition")
	Indicators of different traits - different methods ("Interpretation of the moral dilemma")

Based on the results presented in Table 1, there are reasons to draw a number of conclusions regarding the correlation of the results of intellectual competence in older adolescents by different methods:

**Table 1.** The correlation of measurements of intellectual competence in older adolescents with two different methods - “Composition” (“1”) and “Interpretation of the moral dilemma” (“2”).

	G1	G2	F1	F2	S1	S2	Ar1	Ar2	Q1	Q2	I1	I2	P1	P2
G1	1													
G2	0,28	1												
F1	0,79	0,26	1											
F2	0,20	0,09	0,08	1										
S1	0,42	0,20	0,68	0,06	1									
S2	0,14	0,11	0,21	0,02	0,08	1								
Ar1	0,66	0,14	0,72	0,11	0,48	0,18	1							
Ar2	0,13	0,51	0,13	0,07	0,07	0,22	0,09	1						
Q1	0,43	0,04	0,16	0,14	0,01	0,05	0,18	0,02	1					
Q2	0,26	0,19	0,11	0,07	0,01	0,04	0,29	0,08	0,08	1				
I1	0,80	0,20	0,89	0,08	0,61	0,14	0,78	0,07	0,31	0,16	1			
I2	0,30	0,74	0,23	0,02	0,12	0,19	0,26	0,58	0,06	0,31	0,20	1		
P1	0,87	0,28	0,88	0,10	0,59	0,16	0,69	0,15	0,39	0,15	0,88	0,31	1	
P2	0,12	0,31	0,04	0,19	0,03	0,25	0,09	0,19	0,12	0,33	0,04	0,34	0,03	1
C1	0,73	0,18	0,39	0,23	0,08	0,11	0,48	0,15	0,46	0,25	0,44	0,30	0,56	0,07
C2	0,09	0,57	0,13	0,23	0,12	0,17	0,15	0,36	0,13	0,33	0,08	0,55	0,14	0,19

1. indicators of reliability-consistency of measurements of intellectual competence in older adolescents by different methods are statistically significant, and accordingly, intellectual competence can be measured in high school students by both the “Composing” method and the “Interpretation of the moral dilemma” method;
2. validity indicators demonstrate the fact that the results of measurements of intellectual competence in older adolescents by different methods are well comparable in schools only if the total score for the methods and interpretative narratives is taken into account. Assessments of validity as a suitability for comparing data from other manifestations of intellectual competence do not reach a level of significance;
3. indicators of traits of monomethods (measurements of intellectual competence in older adolescents by different methods) are large enough to conclude that the method takes precedence over the construct;
4. indicators of different traits - different methods (linguistic school and sports school) are small, but still in some cases exceeding the validity of indicators of different traits - different methods, which may indicate a lack of differentiation and integration of a number of indicators of intellectual competence associated with initiation and energy support, support of intellectual activity.

Accordingly, it was concluded that the methodological impact of measuring private indicators of intellectual competence in older adolescents is significant. So, it should be

pointed out that the structure of a number of manifestations of intellectual competence associated with initiation and energy support, support of intellectual activity is characterized by low differentiation, which can be argued for by the insufficient degree of formation of the construct.

Thus, depending on the specifics of the methodological apparatus used in the study, there is reason to talk about the differences between older adolescents in terms of the severity of manifestations of intellectual competence.

It should be noted that if the “Composition” methodology gives the task in the broadest possible way: neither the topic, nor the standards of volume, style of presentation were indicated, then the “Interpretation of the moral dilemma” methodology implies more specificity - the moral dilemma is asked, which is asked to be interpreted. As was proved in our previous studies [10, 11], the structure of narratives varies depending on the features of the techniques. Probably, in the case of “Composing”, the participants in the study, found themselves in the conditions of search, research intellectual activity, have to use not only their verbal, but also some other abilities. So, older adolescents find themselves on an equal footing, while the greater specificity of the “Interpretation of the moral dilemma” methodology simplifies the intellectual task.

Thus, in the presented study, the significance of the methodological influence on the measured level of expression of intellectual competence in older adolescents is argued.

## 5 Findings

The results obtained allow us to conclude that it is necessary to take into account the methodological influence on the manifestations of intellectual competence in adolescents of older adolescence.

Thus, it can be concluded, that the hypothesis that the indicators of intellectual competence, measured by the methodology “Composing on a free topic”, and the indicators of intellectual competence, measured by the methodology “Interpretation”, are slightly different. When registering a general indicator of intellectual competence, the differences due to the use of various methodological tools are small, but when taking into account the private manifestations of intellectual competence, this is different statistically large.

**Acknowledgments.** The study was carried out by a grant from the Russian Science Foundation (project 19-013-00294).

## References

1. Asmolov, A.G.: Formation of universal educational actions in primary school: from action to thought, pp. 57–61. M.: Enlightenment (2010)
2. Bolotov, V.A., Serikov, V.V.: Competence model: from idea to educational program. *Pedagogy* (10), 8–14 (2003)
3. Danilova, N.N.: *Psychophysiology: Textbook for Universities*. Aspect Press, London (2004)
4. Kholodnaya, M.A.: *Psychology of intelligence: the paradoxes of St. Petersburg research: Peter* (2002)

5. Korchemny, P.A.: The content characteristic of the basic concepts of the competency-based approach in education (acmeological component). *Akmeology* **2**(58), 30–38 (2016)
6. Libin, A.V.: Differential psychology: at the intersection of European, Russian and American traditions. *Psychology for the student*. M.: Sense (1999)
7. Melnichuk, A.S.: A multidimensional approach to the analysis of subjective strategies for the development of professional competencies. *Acmeology* **42**(2), 23–30 (2014)
8. Sipovskaya, Ya.I.: Conceptual, metacognitive and intentional descriptors of intellectual competence in older adolescents. *Bull. St. Petersburg State Univ.* **12**(4), 22–31 (2017)
9. Sipovskaya, Ya.I.: Features of measuring intentional abilities in older adolescents. *Int. Sci. Sch. Psychol. Pedagog.* **9**(17), 37–42 (2015)
10. Sternberg, R.J.: The theory of successful intelligence. *Revista Interamericana de Psicología/Interam. J. Psychol.* **39**(2), 189–202 (2005)
11. Sultanova, L.B.: The Problem of Implicit Knowledge in Science. Publishing House of Ural State Technical University, Ufa (2004)
12. Vecker, L.M.: *The Psychological Processes*. Publishing House of Leningrad State University, St. Petersburg (1976)
13. Winter, I.A.: Key competencies - a new paradigm of the result of education. *High. Educ. Today/Acmeology* **5**, 34–42 (2003)
14. Yadrovskaya, E.R.: The development of the interpretative activity of the reader-student in the process of literary education (grades 5–11). (Doctoral dissertation). Retrieved from Higher Attestation Commission from The Dissertation Department of the Russian State Library (2012)



# The Automated Solar Activity Prediction System (ASAP) Update Based on Optimization of a Machine Learning Approach

Ali K. Abed<sup>(✉)</sup> and Rami Qahwaji

Bradford University, Bradford, UK

{a.k.abed, r.s.r.qahwaji}@bradford.ac.uk

**Abstract.** Quite recently, considerable attention has been paid to solar flare prediction because extreme solar eruptions could affect our daily life activities and on different technologies. Therefore, this paper presents a novel method of the development of improved second-generation of the Automated Solar Activity Prediction system (ASAP). The suggested algorithm improves the ASAP system by expanding a period of training vector and generating new machine learning rules to be more successful. Two neural networks are responsible for determining whether the sunspots group will release flare as well as determining if the flare is an M-class or X-class. Several measurement criteria are applied to determine the extent of system performance also all results are provided in this paper. Furthermore, the quadratic score (QR) is used as a metric criterion to compare between the prediction of the proposed algorithm with the Space Weather Prediction Center (SWPC) between 2012 and 2013. The results exhibit that the proposed algorithm outperforms the old ASAP system. Keywords: Solar flares, Machine Learning, Neural network, Space, Prediction, weather.

**Keywords:** Neural networks · Automated Solar Activity Prediction · Sunspot · McIntosh classifications

## 1 Introduction

Nowadays, space weather prediction in real-time is an important issue for many countries because of extreme solar eruptions could influence our daily life activities and affect various technologies. Wherefore, the U.S. National Space Weather Program (NSWP) defined space weather as “conditions on the Sun and in the solar wind, magnetosphere, ionosphere, and thermosphere that can influence the performance and reliability of spaceborne and ground-based technological systems and can endanger human life or health” [1]. Wherefore, the significance of studying space weather is increasing.

Solar flares and Coronal Mass Ejections (CMEs) are solar events that have a significant impact on daily life and on different technologies at Earth [2]. As a result, these solar activities can spew vast quantities of radiation and charged particles into space this causes extreme ultraviolet and X-ray flux from flares reacted with the ionosphere making widespread blackout situations for High-Frequency radio communications [3].



In spite of the fact that the event of these activities cannot be stopped. However, predicting when these solar activities are possible to occur could reduce possible damage to industries, for example, space agencies, power generation and distribution industry, oil, satellite operators and gas industry, and thus lead to a lowering in their economic effect. Solar flare study has confirmed that Solar flares and Coronal Mass Ejections (CMEs) are generally associated to active regions and sunspots [3–5].

There are predictive science researchers and different research organizations scattered all over the world are involving in solar prediction and analysis. These predictions usually rely mainly on experts with high knowledge in space weather, which may lead to a discrepancy in space weather forecasting. To solve these problems, objective computerized analysis of images surface of the sun can supply automated processing and consistent execution by applying the enormous computational abilities of computers with high-speed processing to analyse and compare huge amounts of new and historical data. Thus, there is still a need for design high-performance forecasting system.

There are many challenges facing the scientists of space weather forecasting. We can face those challenges by building an effective computer system has the ability to the automated determination, classification, and representation of solar features and the creation of a perfect correlation between these features and the appearance of solar activities. In order to propose an effective system for space weather forecasting, we need to apply real-time, high-quality space weather data and processing techniques to forecast solar activities (Wang et al. 2003). The launch new satellites for space weather, for example, the Solar Dynamics Observatory (SDO) has helped to provide accurate data this helped a solar events observation. Therefore, that requires prediction methods which conformity, and to benefit from, additional information in the data. Many predictions systems for instance, those depend on sunspot detection models, classification, machine learning algorithms, time-series analysis, and many more that have been suggested [6].

Previous studies indicate that there have been automated systems that can provide real-time forecasting of significant solar flares that may influence our Earth. The performance of the former automated systems that designed still will hope a better prediction than subjective analysis. For example, The University of Alabama in Huntsville developed new a technique called (MAG4) system of forecasting an active region's rate of production of greater flares in magnetic energy [7]. This system prediction depended on applying magnetogram data for the Sun. The principal work of this system is using the McIntosh active-region (AR) classes to prediction Solar Proton Events (SPE), CMEs, and M and X class flares.

Hong et al. [8] designed a system called Automatic Solar Synoptic Analyzer (ASSA) the main function of this system is identifying coronal holes, sunspot groups, and filament channels that are three properties responsible the space weather. This system is built depends on an artificial neural network technique with the ASSA coronal hole data archive of the period from 1997 to 2013. In addition, this system-applied image of SOHO EIT 195 and SDO AIA 193 used for morphological recognition and thereafter SOHO MDI Magnetograms and SDO HMI Magnetograms applied for quantitative verification. This system has the ability to predict three classes of solar flares are C, M, and X-flare.

We updated the Automated Solar Activity Prediction system (ASAP). It is an automated space weather forecasting system that contents from advanced image processing

and machine learning techniques with solar physics. The update process included the following steps: Increase the data used in the neural network training by increasing the time period. We have used the data from Dec 1918 to June 2017. In addition, a new training strategy was used in this paper. We will mention all the update details in the next part of this paper.

The remainder of the paper is organized as follows: Sect. 2 outlines the images processing method that is responsible for the automated detection and classification of sunspots. The machine learning models that are trained on solar flares and historical sunspot data are presented in Sect. 3; Sect. 4 discusses applying, integration, performance, and evaluation for Machine Learning system. The concluding and suggestions for future work are presented in Sect. 5 concludes the paper.

## 2 Detection and Classification for Sunspot

In 2008 Colak and Qahwaji [9] proposed a computer system that can automatically detect, group, and classify sunspots depend on the McIntosh classification. SDO/HMI Continuum and Magnetogram images use in this system as input to reveal sunspot regions and extract their features including their McIntosh classifications. In this study, this system working by integrated with a machine learning-based method to supply real-time forecasting for the probable occurrence of significant flares like type-X or type- M, as described in the next part in this report.

### 2.1 SDO HMI Images

The Solar Dynamics Observatory (SDO) supplies 13 various wavelengths of the sun. Two instruments have been used are the Helioseismic and Magnetic Imager (HMI) and the Atmospheric Imaging Assembly (AIA) instrument. In this study, we used two types of images are HMI Continuum and HMI Magnetogram. HMI Continuum provided images of the solar surface, incorporating a broad range of visible light for Solar Region Photosphere. On the other hand, HMI Magnetograms show maps of the magnetic field on the sun's surface, with black and white. the black showing magnetic field lines pointing away from Earth, On the contrary, the white showing magnetic field lines coming toward Earth for Solar Region Photosphere [10].

### 2.2 Sunspot Detection and Classification Algorithms in Colak and Qahwaji System

In this part of the report, we provide a brief about the worked algorithms of the system Colak and Qahwaji. These algorithms include different functions such as sunspot detection, grouping, and classification. The following steps for these algorithms:

#### 1) Pre-processing of HMI images

- Continuum and magnetogram images used together to determine the solar disk, radius and centre, make a mask and remove any information for example date and direction from the image, calculate the Julian date and solar coordinates.

- Magnetogram images only use in this algorithm. Map the magnetogram image from Heliocentric-Cartesian coordinates to Carrington Heliographic coordinates. Centre, radius and solar coordinates of the continuum image use a replacement of centre, radius and solar coordinates of the magnetogram image of re-map to Heliocentric-Cartesian coordinate.

## 2) Sunspot grouping

- MDI continuum images using to detect sunspot candidates by applying limited intensity thresholding.
- MDI magnetogram images used to detect active region candidates by applying morphological image processing algorithms such as intensity filtering, dilation, and erosion.
- Use region growing approach to combine active region with sunspot candidates. More details about this technique are described in [9].
- Apply neural networks to combine regions of opposite polarities so as to determine the boundaries of sunspot groups.
- Sign the discovered sunspot groups.

## 3) McIntosh-based Classification

- Applying neural networks and image processing in order to detect local features from each sunspot in each group.
- Apply image processing in order to detect features from every sunspot group these features are largest spot, distribution, length, and polarity.
- Used a decision tree approach to determine McIntosh classification by used the extracted features as input for a decision tree.

# 3 Apply Machine Learning for Solar Flare Prediction

Many experts in space weather contend has shown that flares exceedingly related to active regions and sunspots [3–5]. A study of the solar physics literature characterizing the association between sunspots and flares was introduced in [11]. In this system, all information used as input to machine learning provided from historical data such as solar catalogues and converted it in computerized learning rules that allow computers system to analyse current solar data and supply solar flare forecasts.

## 3.1 Knowledge Representation of Solar Catalogues

The National Geophysical Data Centre (NGDC) provides sunspots groups catalogue and solar flares catalogue. These catalogues are publicly available on the (NGDC) website. The NGDC sunspot catalogue contains the following data the date, time, location, physical properties, and classification of sunspot groups, and the National Oceanic and Atmospheric Administration (NOAA) number, on the flip side, the NGDC flare catalogue contains the following data dates, starting and ending times for flare eruptions,

location, x-ray classification, and the National Oceanic and Atmospheric Administration (NOAA) number. The association is done through Atmospheric Administration (NOAA) number for the flare that is associated with the active region detected. In this section, a C++ platform created to automatically associate between sunspots and flares. NGDC has sunspot data from various observatories and sometimes contains different observations of the same sunspot group for various times of the same day. Flares and sunspot catalogues examined to associate sunspot groups with the solar flares, which are consider the main cause of these flares. All the recorded flares and sunspots for the periods from 1st December 1981 until 30th June 2017, which includes 71475 solar flares (43147 C-class, 5435 M-class and 417 X-class) and 271883 sunspot groups, are tested using the association algorithm described in [11].

The association algorithm is based on the following conditions:

- A sunspot and solar flare are associated if it has the same NOAA number. Whereas, sunspots and flares does not have NOAA numbers are filtered out from the lists.
- A solar flare catalogue should be listed after a sunspot within a predefined time window. Four-time windows were used: 6, 12, 24, and 48 h.
- If more than one sunspot associated with the same solar flare, only the nearest sunspot in time to the flare is considered as related, and the all remnant of sunspots are deleted.

**Table 1.** Illustrate the final association results for a sunspot and solar flare by the association algorithm.

Time window	C	M	X	Associated (A)	Not Associated (NA)	A + NA	Ignored
6	17210	2877	252	20339	53540	234,944	36,939
12	21615	3686	316	25617	24162	49779	185165
24	20011	3558	313	23882	8290	32172	224554
48	20234	3583	318	24135	3878	28013	228713

Table 1 represents the final association results between flare and sunspot for the periods from 1st December 1981 until 30th June 2017.

### 3.2 The Flare Prediction System

We applied the area of sunspot groups together with the McIntosh classes as the input for solar flare forecast algorithm in order to produce forecasts for the M-class and X-class flares. Our solar flare forecast algorithm is composed of two Neural Network (NN) illustrated in Fig. 1. McIntosh classifications and sunspot numbers for everyday applied as input to the neural network algorithms. Furthermore, we used tools like the Jack-knife method [12] to evaluate the training and generalization efficacies of the neural network algorithms. The first NN accepts four inputs. These inputs are the three McIntosh classes

and the sunspot area. On the other hand, this neural network provides one output in the range of 0.1 to 0.9 in the next 6, 12, 24 and 48 h and a threshold of 0.5 is used to categorize the generated outputs as 0.9 if  $>0.5$  or 0.1 if  $<0.5$ , which represents flare or no-flare respectively. It produces the probability that this sunspot group will produce a solar flare in the next 6, 12, 24 and 48 h. Therefore, this first NN is trained using sunspot regions from the NGDC sunspot catalogue and solar flare from the NGDC flare catalogue associations as described in the previous section. The training vector includes four inputs numerical values and their corresponding single output value (Flare = 0.9 or No Flare = 0.1) as shown in Table 2. For instance, if there is a sunspot region with a McIntosh classification of EFI and an area of 875 in millionths of solar hemisphere that is related with solar flare then the training vector will be [0.7, 0.9, 0.5, 0.35; 0.9].

**Table 2.** Illustrate numerical values for the first neural network representing the McIntosh classes and sunspot area as the inputs and their corresponding target.

Inputs			Output
McIntosh classes		Normalized (with 2500) sunspot area	Flare= 0.9 No flare = 0.1
A = 0.10	X = 0	X = 0	
H = 0.15	R = 0.10	O = 0.10	
B = 0.20	S = 0.30	I = 0.50	
C = 0.35	A = 0.50	C = 0.90	
D = 0.60	H = 0.70		
E = 0.75	K = 0.90		
F = 0.90			

The second NN is worked to determine the forecasted flare is going to be M-class and/or X-class flare. The second NN trained using a training set that includes only the three McIntosh classes for sunspot groups that related to M-class and X-class flares. Therefore, the second NN includes three inputs and two outputs. The first and second outputs represent the M-class and X-class flares, respectively. This neural network works as follows:

- If the sunspot group is related only with an M-class flare, then the first output will be more than or equal to 0.5 otherwise it will be less than 0.5.
- If the sunspot group is related only with an X-class flare, then the second output will be less than or equal to 0.5 otherwise it will be less than 0.5.
- If the sunspot group is related to M-class and X-class of a solar flare, all the corresponding outputs will be greater than or equal to 0.5 otherwise it will be less than 0.5.

*For instance, if there is a McIntosh classification of FKI that is related only with M-class and X-class solar flares during the same time then the training vector for this example will be [0.9, 0.9, 0.5; 0.77, 0.56].*

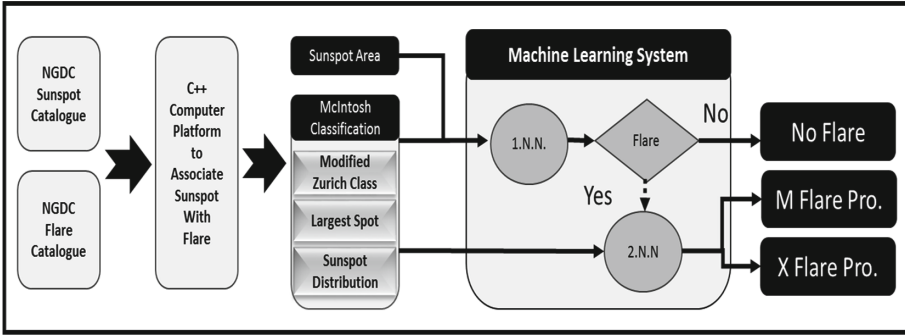


Fig. 1. Solar flare prediction algorithm with training and testing data.

### 3.3 Optimization of the Neutral Networks

The two neural networks are optimized by comparing the forecast outputs from the two neural networks against the actual outputs. From the comparison, the following measures are calculated first: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). For this report, the significance of these measures is described in this section below:

- TP represents the number of cases when a sunspot group is related with an actual flare and a flare forecast is produced then this prediction is true.
- FP represents the number of cases when a sunspot group is related with an actual flare, but no flare forecast is produced then this prediction is wrong.
- TN represents the number of cases when a sunspot group is not related with any actual flare and no flare forecast is produced then this prediction is true.
- FN represents the number of cases when a sunspot group is not related with any actual flare, but no flare forecast is produced then this prediction is wrong.

We are using the measures above to calculating the prediction performance of the learning algorithm. These forecasting measures are:

- True Positive Rate (TPR), represents the possibility sunspot group finds that are successfully forecasted as flaring. Higher TPR represents a better prediction performance. This value is calculated by applying Eq. (1).

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

- False Positive Rate (FPR), represents the possibility sunspot group finds that are unsuccessfully forecasted as flaring. Minimum FPR represents better prediction performance. This value is calculated by applying Eq. (2).

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

- True Negative Rate (TNR), represents the possibility of non-flaring sunspots group finds that are unsuccessfully forecasted as non-flaring. Maximum TNR represents a better prediction performance. This value is calculated by applying Eq. (3).

$$TNP = \frac{TN}{FP + TN} \tag{3}$$

- False Negative Rate (FNR), represents the possibility of flaring sunspot group finds that are unsuccessfully forecasted as non-flaring. Minimum FNR represents better prediction performance. This value is calculated by applying Eq. (4).

$$FNP = \frac{FN}{TP + FN} \tag{4}$$

- False Alarm Rate (FAR), represents the possibility of false flare forecasts. Minimum FAR represents better prediction performance. This value is calculated using Eq. (5).

$$FAR = \frac{FP}{FP + TP} \tag{5}$$

- Mean Squared Error (MSE), this value represents the average of the squares of the difference between the predicted flare cases and the actual flare cases for all sunspot group detections. A minimum MSE value represents better prediction performance. This value is calculated by applying Eq. (6).

$$MSE = \frac{1}{n} \sum_{i=1}^n (p_i - r_i)^2 \tag{6}$$

Where,  $p_i$  is the value of all output for the inputs given in the training vector, and  $r_i$  is the actual output given in the training vector,  $n$  is the whole number of symbols in the training vector.

- Accuracy (ACC), this value represents how close the overall forecast produces are to the actual values. Maximum ACC rates represent a better prediction performance. This value is calculated by applying Eq. (7).

$$ACC = \frac{TR + TN}{(TP + FN) + (FP + TN)} \tag{7}$$

- Heidke Skill Score (HSS), this value represents the chance factor of predicting. The value of HSS is between  $-1$  to  $1$ . Negative values represent the prediction is based on chance,  $0$  shows no-skill, and positive values represent perfect forecasting.

$$HSS = \frac{2 \times ((TP \times TN) - (FP \times FN))}{((TP + FN) \times (FN + TN) + ((TP + FP) \times (FP + TN)))} \tag{8}$$

More details on these measures can get from [13]. Several training experiments are carried out while changing the number of nodes in the hidden layer as follows.

### 3.4 Optimization Strategies

For each association time window (6, 12, 24, and 48), the training and testing methods for the first neural network was as follows:

- 10-time training experiments are carried out while changing the number of nodes in the hidden layer from 1 to 15.
- The average and standard deviation to the forecasting measures were calculated for different decision thresholds (0.5, 0.45, 0.4, and 0.35) and different time-window (6, 12, 24, and 48) for each experiment (Table 3).

**Table 3.** Shows the best number of nodes used in the hidden layer for the first neural network to different decision thresholds (0.5, 0.45, 0.4, and 0.35) and different time-window (6, 12, 24, and 48).

Threshold	Time window	Nodes	Run	TPR	FPR	FNR	TNR	FAR	ACC	SPC	HSS	MCC	TSS
0.35	6	6	Mean	0.4340	0.0218	0.5660	0.9782	0.4597	0.9481	0.9782	0.4526	0.4565	0.4121
0.45	12	9	Mean	0.6635	0.0506	0.3365	0.9494	0.3124	0.9090	0.9494	0.6207	0.6220	0.6128
0.35	24	3	Mean	0.8669	0.1105	0.1331	0.8895	0.2350	0.8828	0.8895	0.7277	0.7311	0.7564
0.5	48	2	Mean	0.8705	0.0823	0.1295	0.9177	0.1020	0.8962	0.9177	0.7901	0.7906	0.7882

Four sets of decision rules for the neural network were created by retraining the full dataset of associations between sunspots and flares: flare\_6.dat, flare\_12.dat, flare\_24.dat, and flare\_48.dat.

For each association time window (6, 12, 24, and 48), the training and testing methods for the second neural network was handled as follows:



- The number of hidden nodes for the second neural network was varied from 1 to 20.
- The only one measure that used to measure the second neural network performance was the MSE.
- From the former table (Table 4), it was found the best neural network structure for the second neural network are: 20 hidden nodes for 6 h forecast window, 18 hidden nodes for 12 h forecast window, 19 hidden nodes for 24 h forecast window, and 20 hidden nodes for 48 h forecast window.
- Four sets of decision rules were created: intensity\_6.dat, intensity\_12.dat, intensity\_24.dat, and intensity\_48.dat.

**Table 4.** Shows the best number of nodes applied in the hidden layer for the second neural network to different time-window (6, 12, 24, and 48).

Time	MSE	Number of Hidden Nodes																				
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
6	M	0.0234	0.0213	0.0213	0.0214	0.0214	0.0314	0.0127	0.0079	0.0171	0.0068	0.0336	0.0117	0.0063	0.0039	0.0111	0.0037	0.0039	0.0064	0.0051		
	X	0.0217	0.0206	0.0206	0.0206	0.0206	0.0367	0.0180	0.0259	0.0204	0.0133	0.0332	0.0103	0.0128	0.0096	0.0181	0.0097	0.0067	0.0119	0.0068		
12	M	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242
	X	0.0174	0.0145	0.0168	0.0168	0.0168	0.0145	0.0196	0.0159	0.0107	0.0160	0.0171	0.0171	0.0126	0.0064	0.0105	0.0108	0.0038	0.0082	0.0065		
24	M	0.0239	0.0218	0.0218	0.0219	0.0218	0.0159	0.0186	0.0158	0.0070	0.0048	0.0223	0.0073	0.0143	0.0128	0.0046	0.0049	0.0050	0.0043	0.0035	0.0051	
	X	0.0170	0.0159	0.0159	0.0159	0.0159	0.0167	0.0164	0.0173	0.0090	0.0065	0.0310	0.0077	0.0142	0.0154	0.0053	0.0072	0.0056	0.0066	0.0029	0.0090	
48	M	0.0240	0.0216	0.0216	0.0216	0.0216	0.0121	0.0121	0.0149	0.0132	0.0275	0.0091	0.0219	0.0073	0.0059	0.0096	0.1734	0.0094	0.0054	0.0054	0.0052	
	X	0.0178	0.0167	0.0167	0.0167	0.0153	0.0092	0.0162	0.0297	0.0269	0.0091	0.0219	0.0082	0.0107	0.0117	0.3186	0.0188	0.0101	0.0118	0.0109		

## 4 Actual Implementation and Evaluation of Updates ASAP’s System

The main principles work of ASAP’s system is integrating the imaging and machine learning systems for the hybrid solar flare forecast. This system is shown in Fig. 2. The inputs of this system are HIM images from SDO. The system starts its real-time working by deals with SDO/HIM continuum and magnetogram images in the method illustrated in Sect. 2 to issue automated McIntosh classifications for the detected sunspots group. After that, the McIntosh classified sunspots and sunspot area are fed to the machine learning system(first and second neural network) explained in Sect. 3 which is trained with 37 years of data after applying the association algorithm. Based on the embedded learning rules the system predicts if a solar flare is going to occur or not without time windows were used (6, 12, 24, and 48 h). If a significant solar flare is forecasted then the probability of this solar flare to be M-class or X-class flare is also predicted. The entire system is implemented in C++.

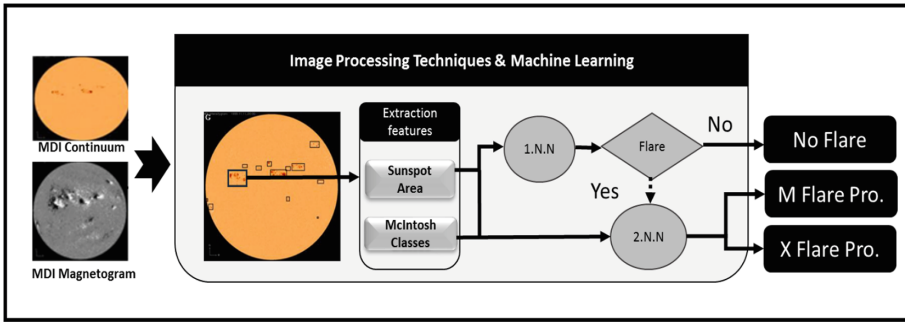


Fig. 2. The final updated ASAP system

#### 4.1 Evaluation of the Updated ASAP System

The update ASAP's system was tested on solar SDO HMI continuum images from January 1, 2012, to December 31, 2013. Furthermore, the performance of the new system was evaluated by comparing the produced forecasts with the actual flares as registered by NOAA SWPC1 in the NGDC X-ray solar flare catalogue and with old ASAP. There were 67787 HIM continuum images available during this period at a cadence of 96 images per day. These HIM continuum images and their corresponding 67787 MDI magnetogram images were processed using the updated ASAP System and four sunspot catalogues was generated which we refer to as ASAPDATABASE\_06, ASAPDATABASE\_12, ASAPDATABASE\_24, and ASAPDATABASE\_48. Parts of the ASAPDATABASE\_06.

With a view to linking the sunspot groups detected by our system together with x-ray solar flares registered in the NGDC catalogue, we had to update the association algorithm, suggested in (Qahwaji and Colak, 2007), the main details of this algorithm as follows:

- A special version of the association algorithm updated to find the associations between the sunspots reported by new ASAP system and the flares reported in the NGDC flares catalogues. These associations were found based on the availability of flare locations and the possibility of having a flare event within the used association time window (6, 12, 24, or 48 h).
  - Read all the sunspot groups and their solar flare forecast as written in ASAP-DATABASE\_06, ASAPDATABASE\_12, ASAPDATABASE\_24, and ASAP-DATABASE\_48 files.
  - Read all the actual M and X-class flares as written in the NGDC solar flare catalogue.
- Carry out an search to link all actual solar flare with its corresponding sunspot group, The location based association is found according to the condition that the distance between a sunspot group detected by ASAP and a solar flare registered by NGDC is less than one  $10^\circ$  radius of corrected the sunspot location as calculated by ASAP.

Furthermore, the difference in time between the detected sunspot group and its associated flare must be less than 6, 12, 24, and 48 h, depending on the forecast lead time window objective.

- X-class flares that are registered without a location in the NGDC catalogue are linked according to time. In this case, if this flare linking with sunspots groups that are found from the same image and same time, only the sunspot group that provides the X-Flarity is evaluated as linked with the reported solar flare.
- The association algorithm updated calculates the values TP, TN, FP, and FN for every single registered in ASAPDATABASE\_06, ASAPDATABASE\_12, ASAPDATABASE\_24, and ASAPDATABASE\_48 files according to the following decision thresholds (Table 5):

**Table 5.** The various thresholds for testing the results of the proposed solar flare prediction system.

Time window	Flarity $P_F * (0.9 - 0.1) + 0.1$	M flarity $P_M * (0.9 - 0.1) + 0.1$	X flarity $P_X * (0.9 - 0.1) + 0.1$
6	0.208	0.431	0.229
12	0.348	0.2291	0.476
24	0.467	0.454	0.53
48	0.695	0.673	0.737

- $P_F = A / (A + NA)$  Percentage of the associated cases to the total number of cases.
- $P_M = M$  and above flares /all cases associated with flares.
- $P_X = X$  flares /all cases associated with flares.

The output of the association algorithm for ASAP system performance is evaluated by applying different measures as explained below.

#### 4.2 Verification the New ASAP’s System

The new ASAP system produces forecasts in numerical format, between 0.0 and 1.0, as shown in Table 3. We used different measures for evaluating the forecasts of the new ASAP system. This different measure divided into two types, the first type requires forecast probabilities. So, the system directly converting them to percentages. For instance, if the flaring output of the ASAP system is 0.35, it is supposed that the sunspot group has a 35% flaring probability. On the other hand, the second type of categorical forecasts (Yes/No). The system used a threshold value of 0.5 (50%) for deciding the final forecasts. The first neural network includes output values 0.1 (10%) non-flaring and 0.9 (90%) flaring to sunspot groups. The value of the hybrid system output is determined according to the value of the threshold used if the resulting value is greater than the threshold, there is a possibility of a flare is predicted to occur. Nevertheless, if the value less than the threshold, there is not a possibility of a flare. The second neural network includes two output values. The first output for M-Flare probability and the second output for X-Flare

probability. The M-Flare probability and X-Flare probability are determined according to the value of the threshold used if the resulting value is greater than the threshold, there is a possibility of an M-Flare probability is predicted to occur. However, if the value less than the threshold, there is not a possibility of an M-Flare probability. The same method determines the probability of occurrence X-Flare.

With a view to calculate the success of the produced forecasts the association results are investigated using four criteria are TP, FP, TN, and FN. We referred to these criteria in the previous section. The solar flares in the NGDC catalogues during the verification period (January 1, 2012, to December 31, 2013) are compared with 309535 sunspot groups that were detected from 67,787 MDI image pairs and recorded in ASAP-DATABASE\_06.txt, ASAPDATABASE\_12.txt, ASAPDATABASE\_24.txt, and ASAP-DATABASE\_48.txt for old and new ASAP system. The forecast outputs are compared for different time windows: 6, 12, 24, 48 h. Different prediction verification measures are applied to evaluate our new output system and old ASAP system for each time window as shown in Tables 6 and 7. These measures are arranged in tables as follows (Tables 6 and 7): Probability of Detection (POD), False Alarm Rate (FAR), Percent Correct (PC), Heidke Skill Score (HSS) and Quadratic Score (QR). More details about these measures in a recent paper by [13].

**Table 6.** Results of the proposed solar flare prediction system.

Time	Type	POD	FAR	PC	QR	HSS
6	Falrity	0.854406	0.499251	0.919021	0.0810	0.63015
	M-flarity	0.738095	0.85514	0.988418	0.0116	0.242014
	X-flarity	0.4	0.9	0.999247	0.0008	0.159978
12	Falrity	0.906371	0.27322	0.868493	0.1315	0.805872
	M-flarity	0.864865	0.614458	0.980164	0.0198	0.533176
	X-flarity	0.5	0.857143	0.998734	0.0013	0.222207
24	Falrity	0.935178	0.076503	0.79621	0.2038	0.928934
	M-flarity	0.888889	0.552795	0.966566	0.0334	0.594895
	X-flarity	0.6	0.769231	0.997706	0.0023	0.333318
48	Falrity	0.942724	0.0417	0.709635	0.2904	0.950158
	M-flarity	0.873239	0.639535	0.946229	0.0538	0.51012
	X-flarity	0.777778	0.5625	0.996265	0.0037	0.559984

POD: the main function of this vector measures the probability of actual solar flares being forecasted true by the ASAP system. The best results for this vector in 48-h time window because the value of this vector is expected that POD would rise with time since there are many significant flares happens in 48-time windows. In new ASAP system for 24 h' time difference POD demonstrates slightly improve in the whole solar flares 93.5%, 88.8% of the M-class flares, and 6.0% of the X-class flares are forecast correctly.

**Table 7.** Results of the old ASAP system.

Time	Type	POD	FAR	PC	QR	HSS
6	Falrity	0.839884	0.600139	0.937993	2.68E+12	0.535244
	M-flarity	0.531915	0.857143	0.990831	2.68E+12	0.224392
	X-flarity	0.571429	0.973333	0.997467	2.68E+12	0.050773
12	Falrity	0.898763	0.440868	0.906802	2.68E+12	0.683951
	M-flarity	0.625	0.78125	0.985088	2.68E+12	0.323255
	X-flarity	0	1	0.998877	2.68E+12	-4.99E-05
24	Falrity	0.932456	0.280298	0.858383	2.68E+12	0.808277
	M-flarity	0.822917	0.594872	0.975718	2.68E+12	0.542088
	X-flarity	0	1	0.997971	2.68E+12	-5.00E-05
48	Falrity	0.952738	0.188498	0.794965	2.68E+12	0.873182
	M-flarity	0.938967	0.514563	0.961771	2.68E+12	0.638486
	X-flarity	0.916667	0.947494	0.991637	2.68E+12	9.87E-02

Otherwise in old ASAP, the whole solar flares 93.2%, 82.2% of the M-class flares and 0% of the X-class flares.

FAR represents measures the proportion of the ASAP system predicting a solar flare that in effect does not happen. The data shows that FAR produced from new ASAP system improved by a reduced rate of this vector for all time window. False alarm rate has to be reduced in order to improve the reliability of the system.

PC can be defined as a term measure the true forecasts rate of the ASAP system that is the ratio of successful flare and no flare forecasts generated by ASAP system. The data shows that for 24 h' time window difference, 85.8% of whole the forecasts (flare or no flare), 96.6% of M-class forecasts and 99.7% of X-class forecast.

In spite of the fact that PC rates for three-time window are extremely high, that means if the ASAP system supplies just one output, which is no- flare, these PC rates would still be big.

HSS: We have defined this term in the previous part of this report. However, this is a very useful measure when occurrences of the solar flare events to be forecasted very rare. Therefore, HSS is a very significant measure for evaluating forecast of ASAP system. HSS results show that new ASAP forecasts are much more than chance especially for flaring and M-class and X-class flare forecasts. Furthermore, HSS term can be used to optimize the ASAP system by selecting different thresholds value (0.5, 0.45, 0.4, 0.35, and 0.3 were used for our experiences as described earlier).

QR: The quadratic score (QR) represents the mean square error (MSE) of the probabilities provided by the ASAP system. QR is used to calculate the accuracy in probability predictions. In the previous part of this report, we showed the importance of calculating and how to calculate MSE. We compared the results of the new ASAP system with NOAA Space Weather Prediction Centre (SWPC) for the same years from 2012 to 2013

and the 24 h and 48 h prediction results as shown in Table 8. In addition, the average QR (or mean square error) between 2012 and 2013 are also calculated. The results of the comparison showed that ASAP provides better accuracy in predictions than SWPC for M-class and X-class flare predictions.

**Table 8.** The comparison between the proposed solar flare prediction system with SWPC for QR factor.

Date	Class	24	48
2012–2013 (ASAP)	M	0.0334	0.0538
	X	0.0023	0.0037
2012 (SWPC)	M	0.15	0.15
	X	0.022	0.017
2013	M	0.043	0.12
	X	0.02	0.024
Average	M	0.0965	0.135
	X	0.021	0.0205

## 5 Conclusions and Future Research

In this paper, we have updated a fully automated hybrid system called Automated Solar Activity Prediction (ASAP) which integrates image processing techniques and machine learning approaches with solar physics. The main aim of this system is predicting automatically whether detected a sunspot group will produce a solar flare and whether this flare will be a C-class, M-class or X-class flare.

The results obtained in this research (HSS, POD, PC, and QR measures) very good compared to the old system, especially when forecasting that a significant solar flare is going to erupt. Particularly, HSS is quite hopeful which displays that new system forecasts are much better than chance. On the other hand, the FAR measure was not good. This is a problem that has to be tackled we are planning to find solutions to this problem in the future. Furthermore, a comparison of QR results of the new system, old ASAP and SWPC showed that the new system provides a better forecast than the ASAP system and SWPC.

Future work will focus on finding solutions to the geometric impact on MDI images near to the limb which considers one of the reasons to changes the classification of sunspot groups that would lead to the wrong prediction. The lack of graphical details for the limb is the major problem blocking us from getting accurate classifications.

## References

1. Koskinen, H., et al.: Space weather effects catalogue. ESA Space Weather Study (ESWS) (2001)

2. Pick, M., Lathuillere, C., Liliensten, J.: ESA space weather programme feasibility studies. Alcatel-LPCE Consortium (2001)
3. Lenz, D.: Understanding and predicting space weather. *Ind. Phys.* **9**(6), 18–21 (2004)
4. Zirin, H., Liggett, M.A.: Delta spots and great flares. *Sol. Phys.* **113**(1–2), 267–283 (1987)
5. Shi, Z., Wang, J.: Delta-sunspots and X-class flares. *Sol. Phys.* **149**(1), 105–118 (1994)
6. Raboonik, A., et al.: Prediction of solar flares using unique signatures of magnetic field images. *Astrophys. J.* **834**(1), 11 (2016)
7. Falconer, D.A., et al.: MAG4 versus alternative techniques for forecasting active region flare productivity. *Space Weather* **12**(5), 306–317 (2014)
8. Hong, S., et al.: The automatic solar synoptic analyzer and solar wind prediction. In: AGU Fall Meeting Abstracts (2014)
9. Colak, T., Qahwaji, R.: Automated McIntosh-based classification of sunspot groups using MDI images. *Sol. Phys.* **248**(2), 277–296 (2008)
10. Zell, H.: *How SDO Sees the Sun* (2017)
11. Qahwaji, R., Colak, T.: Automatic short-term solar flare prediction using machine learning and sunspot associations. *Sol. Phys.* **241**(1), 195–211 (2007)
12. Fukunaga, R.: *Statistical Pattern Recognition*. Academic Press, Cambridge (1990)
13. Balch, C.C.: Updated verification of the space weather prediction center’s solar energetic particle prediction model. *Space Weather* **6**(1) (2008)

# Author Index

## A

Abbadi, Mohammad A., 391  
Abbasi, Qammer H., 26  
Abed, Ali K., 702  
Abeywardana, Hasini, 446  
Abuomar, O., 142  
Abuzitar, Raed, 278  
Ahmad, Jawad, 26  
Alabi, Sunday, 380  
Al-Afandi, Jalal, 1  
Alamleh, Dalia, 198  
Alamleh, Hosam, 198  
Al-Ardhi, Saleh, 185  
Al-Bustanji, Ahmed M., 391  
Al-Daraiseh, Ahmad, 278  
Aledort, Sabina, 644  
Al-kasassbeh, Mouhammd, 391  
Al-Muhammed, Muhammed Jassem, 278  
Alqahtani, Ali Abdullah S., 198  
Andriyanov, N. A., 652  
Assiri, Sareh, 361, 494  
Atoum, Jalal Omer, 322

## B

Bansal, Madhur, 161  
Basuhail, Abdullah, 185  
Becciani, Ugo, 598  
Bekaulova, Zh. M., 622  
Beloff, Natalia, 380  
Bhoir, Deepak, 150  
Bobbert, Yuri, 336  
Bogatu, Ana Maria, 613  
Bonakdari, Hossein, 77  
Booher, D. Duane, 361  
Booher, Duane, 302

Bordiu, Cristobal, 598  
Brieger, Flemming, 37  
Brown, Ric, 665  
Buchanan, William, 26  
Bufano, Filomena, 598

## C

Calanducci, Antonio, 598  
Cambou, Bertrand, 302, 361, 494  
Carabas, Mihai, 244  
Caramihai, M., 613  
Ceasar Aguma, J., 348  
Chandrasekara, Pranieth, 446  
Cheah, Madeline, 581  
Costa, Alessandro, 598

## D

Daineko, Y. A., 622  
Daneshkhah, Alireza, 581  
Dave, Mayank, 161  
Demeri, Anthony, 204  
Diehl, William, 204  
Duzbayev, N. T., 622

## E

Ebtehaj, Isa, 77

## F

Famador, Sandra Mae W., 97  
Finger, Holger, 37  
Füllhase, Sonja, 37

## G

Gao, Zhan, 420  
Garg, Vibhu, 161



Gegov, Alexander, 12  
 Ghafarian, Ahmad, 431  
 Gharabaghi, Bahram, 77  
 Giri, Naresh Kumar, 545

**H**

Hadi, Ali, 322  
 Harie, Yojiro, 570  
 Hely, David, 494  
 Heras, Robert, 224  
 Horváth, András, 1  
 Hriez, Raghda Fawzey, 322  
 Huang, Meng, 420

**I**

Ignatenko, Vera, 560  
 Imran, Ahmed, 607  
 Ipalakova, M. T., 622  
 Iskandarani, Mahmoud Zaki, 51  
 Islam, Mohammad Amanul, 255  
 Ivanovna, Sipovskaya Yana, 694

**J**

Jackson, David, 665  
 Jackson, Karen Moran, 665  
 Jafari, Raheleh, 12  
 Jaskolka, Jason, 511

**K**

Kanarachos, Stratis, 581  
 Karova, Milena, 628  
 Keskin, Deniz, 431  
 Khan, Jan Sher, 26  
 Klemsa, Jakub, 404  
 Koltcov, Sergei, 560  
 Kong, Joonho, 545  
 Kozhaly, K. B., 622  
 Kucharska, Katarzyna, 174  
 Kumar, Nitin, 532

**L**

Lewis, Amari N., 471  
 Li, Tao, 420  
 Liu, Xiaojie, 420

**M**

Manisha, 532  
 Marinov, Daniel, 628  
 McMillin, Bruce, 348  
 Meriah, Mohamed, 132  
 Michalik, Krzysztof, 174  
 Miraoui, Abdelfettah, 132  
 Mohamed, Abduljalil, 116  
 Mohamed, Amer, 116

Mohammadi, Mohammad, 302  
 Mohammadinodoushan, Mohammad, 361  
 Morose, Boris, 644  
 Munir, Arslan, 482, 545  
 Mustafa, Yasir, 116

**N**

Nalgozhina, N. Zh., 622  
 Negoita, Ovidiu, 244  
 Novotný, Martin, 404

**O**

Otasowie, Owolafe, 61  
 Ozkanli, Nese, 336

**P**

Palade, Vasile, 581  
 Papa, Rosemary, 665  
 Parameshwaran, Sanjeevan, 446  
 Pashakhin, Sergei, 560  
 Patterson, F., 142  
 Penev, Ivaylo, 628  
 Perez-Pons, Alexander, 224  
 Philabaum, Christopher, 302  
 Pipa, Gordon, 37  
 Prabhu, Nandana, 150  
 Prabhu, R. K., 142

**Q**

Qahwaji, Rami, 702

**R**

Raciti, Mario, 598  
 Rajapaksha, Sammani, 446  
 Rao, Uma, 150  
 Razvarz, Sina, 12  
 Regan, Amelia, 348  
 Regan, Amelia C., 471  
 Riggi, Simone, 598  
 Russell, Gordon, 26

**S**

Salman, Ahmad, 204  
 Salman, Ammar S., 459  
 Salman, Odai S., 459  
 Sanjana, A., 161  
 Sciacca, Eva, 598  
 Severin, Irina, 613  
 Shabani, Neda, 482  
 Shah, Syed Aziz, 26  
 Shanbhag, Nita, 150  
 Sharifi, Ali, 77  
 Sharshova, R. N., 622  
 Sidi, Lotfi Merad, 132

Spooner, James, [581](#)  
Sütfeld, Leon René, [37](#)  
Szentannai, Kálmán, [1](#)

**T**

Tahir, Ahsen, [26](#)  
Telles, Charles Roberto, [673](#)  
Thayananthan, Vijey, [185](#)  
Tjahjadi, Tardi, [97](#)

**V**

Vitello, Fabio, [598](#)

**W**

Wasaki, Katsumi, [570](#)  
White, Martin, [380](#)

**Y**

Yapa Abeywardana, Kavinga, [446](#)

**Z**

Zaidman, Gal, [644](#)  
Zhao, Hui, [420](#)  
Zhu, Yuxuan, [494](#)