David C. Wyld,
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

10th International Conference on Advances in Computing and Information Technology (ACITY 2020),
November 28~29, 2020, London, United Kingdom.

AIRCC Publishing Corporation

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Dhinaharan Nagamalai,
Wireilla Net Solutions, Australia
E-mail: dhinthia@yahoo.com

ISSN: 2231 - 5403
ISBN: 978-1-925953-29-9
DOI: 10.5121/csit.2020.101501- 10.5121/csit.2020.101521

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The 10th International Conference on Advances in Computing and Information Technology (ACITY 2020), November 28~29, 2020, London, United Kingdom, 10th International Conference on Digital Image Processing and Pattern Recognition (DPPR 2020), 11th International Conference on VLSI (VLSI 2020), 12th International Conference on Web services & Semantic Technology (WeST 2020), International Conference on Data Science and Applications (DSA 2020), 7th International Conference on Computer Networks & Data Communications (CNDC 2020), International Conference on Internet of Things & Embedded Systems (IoTE 2020), 10th International Conference on Artificial Intelligence, Soft Computing and Applications (AIAA 2020) and International Conference on NLP Techniques and Applications (NLPTA 2020) was collocated with 10th International Conference on Advances in Computing and Information Technology (ACITY 2020). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The ACITY 2020, DPPR 2020, VLSI 2020, WeST 2020, DSA 2020, CNDC 2020, IoTE 2020, AIAA 2020 and NLPTA 2020 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically.

In closing, ACITY 2020, DPPR 2020, VLSI 2020, WeST 2020, DSA 2020, CNDC 2020, IoTE 2020, AIAA 2020 and NLPTA 2020 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the ACITY 2020, DPPR 2020, VLSI 2020, WeST 2020, DSA 2020, CNDC 2020, IoTE 2020, AIAA 2020 and NLPTA 2020.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld,
Dhinaharan Nagamalai (Eds)

General Chair

Organization

David C. Wyld, Southeastern Louisiana University, USA
Dhinaharan Nagamalai, Wireilla Net Solutions, Australia

Program Committee Members

A K Daniel, M M M University of Technology, India
Abbas Baradaran, Azad Universities, Iran
Abdallah Makhoul, University of Bourgogne, France
Abhishek Shukla, A P J Abdul Kalam Technical University, India
Abirami, Kongu Engineering College, India
Adeyanju Sosimi, University of Lagos, Nigeria
Adiline Macriga, Sri Sairam Engineering College, India
Adrian Olaru, University Politehnica of Bucharest, Romania
Ahmed A. Elngar, Beni-Suef University, Egypt
Ahmed Farouk AbdelGawad, Zagazig University, Egypt
Akhil Gupta, Lovely Professional University, India
Akhlaq Ahmad, Umm Al Qura University, Saudi Arabia
Alessandro Massaro, Dyrecta Lab, Italy
Alia Karim Abdulhassan, University of Technology, Iraq
Amine Achouri, University of Tunis, Tunisia
Ammar A. Aldair, University of Basrah, Iraq
Anand Nayyar, Duy Tan University, Vietnam
Anandakumar, Sri Eshwar College of Engineering, India
Ankur Singh Bist, Signy Advanced Technology, India
Antoni B. Chan, City University of Hong Kong, Hong Kong
Arman Roohi, University of Central Florida, FL, USA
Asadollah Shahbahrami, University of Guilan, Iran
Ashish k, Sipna college of Engineering & Technology, India
Assia DJENOUHAT, University Badji Mokhtar Annaba, Algeria
Atena Abdi, Amirkabir University of Technology, Iran
Atika Qazi, University Brunei Darussalam, Brunei
Avishek Adhikari, Presidency University, India
Azeddine Wahbi, Hassan II University, Morocco
B. F. Oladejo, University of Ibadan, Nigeria
Barbaros Preveze, Cankaya University, Turkey
Basanta Joshi, Tribhuvan University, Nepal
bdullah, Adigrat University, Ethiopia
Beheshti A, Elect. Eng. School, Iran
Bouchra Marzak, Hassan II University, Morocco
Boukari Nassim, skikda university, Algeria
CH.Renu Madhavi, RVCE, India
Chandrasekar Vuppalapati, San Jose State University, USA
Chang-Yong Lee, Kongju National University, South Korea
Chikh Mohammed Amine, Tlemcen University, Algeria
Ching-Nung Yang, National Dong Hwa Univesity, Taiwan
Dadmehr Rahbari, University of Qom, Iran
Daniel Ekpenyong Asuquo, University of Uyo, Nigeria
Daniel Imanah, Zudan Electric Ltd, Nigeria

Khin Su Myat Moe,	Yangon Technological University, Myanmar
Kin-Choong Yow,	University of Regina, Canada
Laxmi Tripathy,	Siksha 'O' Anusandhan University, India
M.Kamala Kumari,	Adikavi Nannaya University, India
M.Suresh,	Kongu Engineering College, India
Mahdi Sabri,	Islamic Azad University, Iran
Maissa Hamouda,	University of Sousse, Tunisia
Malka N. Halgamuge,	The University of Melbourne, Australia
Manal Mostafa Ali,	Al-Azhar University, Cairo
Manish Kumar Mishra,	University of Gondar, Ethiopia
Manisha Singh,	Amity University, India
Manoj kumar,	NIT, India
Manoj Sahni,	Pandit Deendayal Petroleum University, India
Mansour Bader,	Al-Balqa Applied University, Jordan
Marco Anisett,	UniversitÃ degli Studi di Milano, Italy
María Hallo,	Escuela Politécnica Nacional, Ecuador
Mario Versaci,	DICEAM - University Mediterranea, Italy
Maryam Amiri,	Arak University, Iran
Mehdi Nezhadnaderi,	Islamic Azad University, Iran
Meriah Sidi Mohammed,	University of Tlemcen, Algeria
Moatsum Alawida,	USM university, Malaysia
Mohamad Al_laham,	Al-Balqa Applied University, Jordan
Mohamed Fakir,	Sultan Moulay Slimane University, Morocco
Mohamed Issa,	Zagazig University, Egypt
Mohamed Matoui,	Hassan II University, Morocco
Mohamed Sahbi Bellamine,	Carthage University, Tunisia
Mohammad Abido,	King Fahd University, Saudi Arabia
Mohammad Ashraf Ottom,	Yarmouk University, Jordon
Mohammed A. Salem,	Higher Technological Institute, Egypt
Mohammed Atif,	Arab Open University, Kuwait
Mohammed GH. AL Zamil,	Yarmouk University, Jordan
Mohankumar N,	Amrita Vishwa Vidyapeetham, India
Mueen Uddin,	Effat University Jeddah,Saudi Arabia
Muhammad Arif,	Guangzhou University, China
Muhammad Elrabaa,	KFUPM, Saudi Arabia
Muhammad Sarfraz,	Kuwait University, Kuwait
Muzhir Shaban Al-Ani,	University of Human Development, Iraq
Nadia Abd-Alsabour,	Cairo university, Egypt
Naoyuki Ishimura,	Chuo University, Japan
Natarajan Meghanathan,	Jackson State University, USA
Navaid Z. Rizvi,	Gautam Buddha University, India
Nawaf Omar N. Alsrehin,	Yarmouk University, Jordan
Nawres Khlifa,	University of Tunis El Manar, Tunisia
Nazmus Saquib,	University of Manitoba, Nazmus
Neeraj Chugh,	University of Petroleum and Energy Studies, India
Ngoc Hong Tran,	University College Dublin, Ireland
Nguyen Dinh Lau,	University of Danang, Vietnam
Nihar Athreyas,	University of Massachusetts, USA
Nikola Ivkovic,	University of Zagreb, Croatia
Niloofar Rastin,	Shiraz University, Iran
Olakanmi Oladayo O,	University of Ibadan, Nigeria
Paria Assari,	Islamic Azad University, Iran

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Digital Signal & Image Processing Community (DSIPC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

10th International Conference on Advances in Computing and Information Technology (ACITY 2020)

10th International Conference on Digital Image Processing and Pattern Recognition (DPPR 2020)

11th International Conference on VLSI (VLSI 2020)

12th International Conference on Web services & Semantic Technology (WeST 2020)

International Conference on Data Science and Applications (DSA 2020)

7th International Conference on Computer Networks & Data Communications (CNDC 2020)

International Conference on Internet of Things & Embedded Systems (IoTE 2020)

10th International Conference on Artificial Intelligence, Soft Computing and Applications (AIAA 2020)

International Conference on NLP Techniques and Applications (NLPTA 2020)

INTRUSION DETECTION IN COMPUTER SYSTEMS BY USING ARTIFICIAL NEURAL NETWORKS WITH DEEP LEARNING APPROACHES

Sergio Hidalgo-Espinoza, Kevin Chamorro-Cupuerán and Oscar Chang-Tortolero

School of Mathematics and Computer Science,
University of Yachay Tech, Ecuador

ABSTRACT

Intrusion detection into computer networks has become one of the most important issues in cybersecurity. Attackers keep on researching and coding to discover new vulnerabilities to penetrate information security system. In consequence computer systems must be daily upgraded using up-to-date techniques to keep hackers at bay. This paper focuses on the design and implementation of an intrusion detection system based on Deep Learning architectures. As a first step, a shallow network is trained with labelled log-in [into a computer network] data taken from the Dataset CICIDS2017. The internal behaviour of this network is carefully tracked and tuned by using plotting and exploring codes until it reaches a functional peak in intrusion prediction accuracy. As a second step, an autoencoder, trained with big unlabelled data, is used as a middle processor which feeds compressed information and abstract representation to the original shallow network. It is proven that the resultant deep architecture has a better performance than any version of the shallow network alone. The resultant functional code scripts, written in MATLAB, represent a re-trainable system which has been proved using real data, producing good precision and fast response.

KEYWORDS

Artificial Neural Networks, Information Security, Deep Learning, intrusion detection & hacking attacks

1. INTRODUCTION

Information security is, nowadays, one of the most important topics in computer science. This is due to the giant internet-connected networks and devices that increase day to day, making hacking activities to increase in the same proportion. The fight against these never end activities is complex because there exist lot of reasons why an information system becomes an attractive target. These goes from activism (called 'hacktivism'in this environment), people eager to show their hacking abilities or just for fun. Also, everyday hackers develop new abilities and techniques to infringe security of information systems [1]. Therefore, it is necessary to constantly design new tools that fight against malicious activities. A fertile alternative to develop hacking protecting systems are Artificial Neural Networks.

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 01-12, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101501

Artificial neural networks (ANN) are data structures dedicated to get results to non-deterministic problems, in a different approach to another conventional process. Important areas where ANNs are actively used are data classification, mapping, prediction and clustering [2], for which it is necessary to have a compilation of data (the dataset) to get the desired results. In general, there exists 3 phases to execute an ANN: training, validation and testing. Training is the phase in which weights are modified to get the optimal level of learning. This phase could be performed using several modifications in the components of the network to get varied weights to be tried and obtain the best weights which will return an optimal classification. This phase uses the major amount of data, it is recommended to use 80% of data approximately. Validation phase is used to test the different weights obtained in training phase to choose the best ones and sometimes to get weights more tuned than before. Finally, testing phase is the fireproof to the network. At this step, the error must be minimal and impossible to reduce. In case the network does not return convincing results, it is necessary to check and modify the parameters of the network. Usually the remaining 20% of data is distributed to perform validation and testing phases.

ANNs are systems based in the human brain functionality [3]: an external signal comes to the system as input data (external senses of the human body), this information is carried to the core of the neural network (human brain) to be interpreted by*synapses* [4] between artificial neurons (human brain neurons). Finally, the result is showed as output data (reactions in the human body). Generally, the results of an ANN are led by a series of calculations performed by steps into several iterations or epochs [5]. Some ANNs may be more advanced and solve problems with better results by the use of mixed ANNs into a nested one, these particular ANN's are called Deep Learning[6][7][8].

Deep Learning Architectures do not have a strict definition (like ANNs), it means does not exist a number of neurons, layers or models to work with these tools. They are just another kind of data processing structures which apply high levels of abstraction [6], using methods to analyse and get information from data deeper (hence its name) than other ANNs, turning Deep Learning architectures on “black boxes” and allows to get better results which means higher percentage of patterns classification.

As a result of applying these concepts, it has been designed and built a Deep Learning structure to process logins into a computer network, classifying them as hacking attacks or normal activity, obtaining high levels of accuracy.

2. RELATED WORKS

The importance of this work lies in the fact of create an Intrusion Detection System prepared to learn and improve its accuracy by using advanced techniques, trained and tested with data updated to current times which could provide a level of adaptationready to detect any attempted intrusion. To achieve this objective, some researches in the same field have been studied, giving a clear idea about the direction of this work.

2.1. Y. Liu, S. Liu and X. Zhao (2017)

Liu [9] and his team designed an intelligent system applied to detect intrusion using another kind of Deep Learning architecture: Convolutional Neural Network (CNN) and other different techniques, as factors of comparison, which are not necessarily related to ANNs. They used the 10% of a famous database of intrusion attacks: the KDD Cup 1999 assembled by DARPA consisting of more than one million of patterns and 41 parameters to each pattern, applying 22 types different hacking attacks. After processing these data into the network, they obtained 99.7%

of detection rate (capacity to perform a correct classification) using the CNN which was the better tool of that research. The problem of these results is, obviously, the age of the data, the world and computers do not work equal than the last century anymore. Testing this network with updated data will carry important changes and configurations into the same network, because new data will have new parameters to be processed. Also, this dataset has some problems, for which it is necessary perform pre-processing algorithms to clear the dataset, some of them are studied by McHugh [10].

2.2. Biswas (2018)

Biswas [11] performed a similar research than Liu, using different techniques for classification, such as Nave Bayes, Support Vector Machine, Decision Tree, Neural Network (no specified) and k-nearest neighbour algorithm (k-NN); in the other hand they are used feature selectors algorithms to take relevant characteristics of data, such as Correlation based Feature Selection method (CFS), Principal Component Analysis (PCA), Information Gain Ratio (IGR) and Minimum Redundancy Maximum Relevance. The difference of this research is the use of a modified version of KDD 99 dataset, the NSL-KDD. This improved version is based on the problems studied in the original dataset, detailed by Tavallaee [12] in 2009. The NSL-KDD consists of 100000 patterns of activity and more than 40 parameters, reduced to 10000 in order to avoid problems with computational cost.The combination of k-NN with IGR gives best results of experiments, reaching a 99.07% of accuracy. The problem is the same than before, data was created in 1999 which does not have compatible metrics with current data, despite it was filtered and improved.

2.3. Vinayakumar, Alazab, Soman, Poornachandran, Al-Nemrat and Venkatraman (2019)

In this work, developed by Vinayakumaret al. [8], it is used a Multi-layer Perceptron as processing architecture, the Feed Forward algorithm and several activation functions such as sigmoid, tangent,softmax and ReLU; they treat the compilation of these characteristics as a Deep Learning architecture. Their results are divided into a lot of tests with different datasets and different configurations of the network, obtaining high percentages of classification overcoming the 90% in almost all the experiments.

3. THE DATASET

The dataset is one of the main components for the training of the network. It consists in a lot of patterns previously generated by experiments or observation; each pattern has many parameters which characterize it. Each one of the patterns may be as random as the experiments be, we do not know or could predict its behaviour just its nature. The inputs are presented as a CSV (Comma Separated Values) file in which each pattern is a new entry or row and each parameter is a new column. Each row must have all columns filled, in case there exist any column empty, it is necessary to have a default value to fill it, but it is better to remove that row.

3.1. CICIDS2017 Dataset

For this case, it is used the **Intrusion Detection Evaluation Dataset (CICIDS2017)** developed in 2017 by the **Canadian Institute for Cybersecurity (CIC)**[13] in laboratories dedicated to information security. The idea to build this dataset is performing a massive amount of different controlled hacking attacks from different parts of the world to dedicated servers which measure and register the chosen parameters to be saved as files. Finally, the result is thousands of patterns

(hacking attacks or normal activity), which are the inputs patterns for this work, with 78 characterizing parameters and 1 target parameter (seen as CP and TP in Figure 2, respectively) which is the result to reach. Some important details of this dataset are studied by [14].

3.2. Data Normalization

Before data patterns could be processed by any ANN they must be processed by an algorithm of normalization, this is a process in which all patterns' values are placed in the same range, being [0-1] the most used. Normalization is important because major of data values are produced naturally in random ranges, it makes any ANN increase the effort to achieve optimal results or decrease the quality of them. Therefore, this process optimizes the capacity of the network to get the optimal classification for each pattern.

Figure 1 indicates how data does not change its behaviour after normalization but it takes a different shape in order to be easy to process by the network. It is just adjusting the numerical range of data between [0-1] instead of its natural range using the next equation for each one of the characterizing parameters (columns of the dataset):

$$n_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{1}$$

where:

x: the characterizing parameter in the dataset,
i: the number of each pattern,
n: the resultant column after normalization.

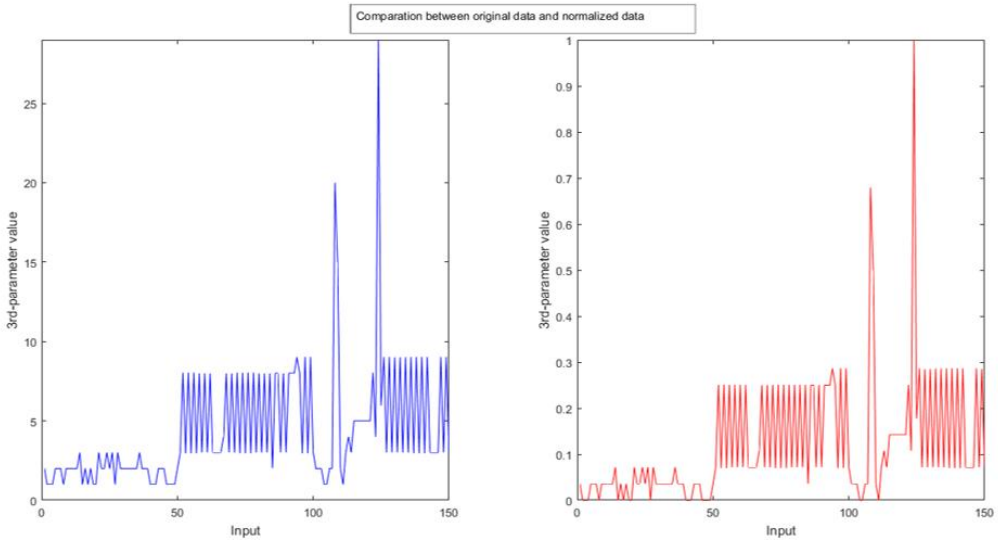


Figure 1. Data normalization for a single parameter using a 30-patterns dataset. Blue: Original data. Red: Normalized data between [0-1].

It is not necessary to normalize the target pattern because it is just a boolean output: 1 if the correspondent pattern is an attack, and 0 in the opposite case.

4. DEEP LEARNING ARCHITECTURE

In order to demonstrate the utility of this Deep Learning model, the experiments of this work are performed in two different ANN architectures: **a shallow network** which consists of a ***Back-propagation Algorithm*** **(BPA)** only, and a **deep network** formed by two single ANN architectures: a two-layered **auto encoder** and the BPA mentioned before. In this case, the autoencoder is a complement architecture dedicated to turn pure data into compress data easy to process by the BPA. Autoencoders usually have two parts: encoder and decoder, the encoder takes data and extract recondite information which may not be visible to the ANN. This process is performed by a set of equations and calculations (because autoencoders are a kind ANN such as) which leave as result the weights of the encoding and set of compressed data (the encoded data). In the other hand, the decoder takes encoded data and its respective weights to obtain the input data again. The relevance of the encoded data to work along the BP algorithm depends on the accurate when data is decoded back, it means the decoding error must be minimal to have useful encoded data.

The BPA used into the two architectures will be almost the same. For the first one (the shallow network), it has 3 layers: the**input layer**, the **hidden layer** and the **output layer**as showed in Figure 2. Every layer is formed by a number of neurons previously selected, in this work it was established the next distribution: 78 layers for the input layer, 11 for the hidden layer and one for the output layer which is compared with the target parameter to get the error between them. Finally, the result is obtained from this error after executing the number of proposed iterations.

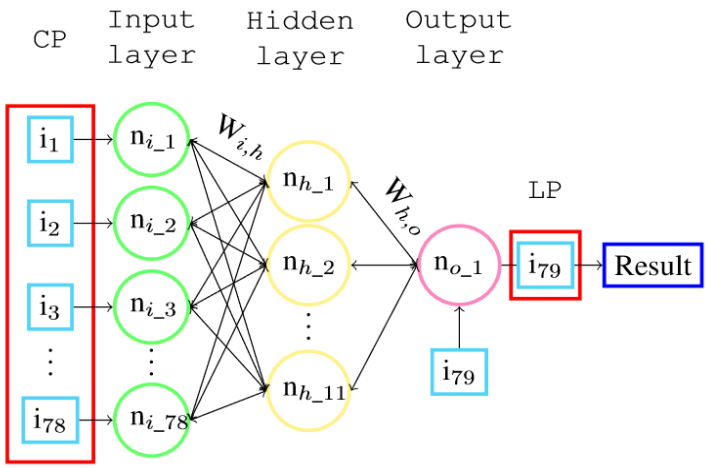


Figure 2. First ANN architecture: Shallow backpropagation topology. CP: Characterizing parameters. W: Weights. TP: Target parameter.

One of the most important part of the BPA are the **weights (W)**, which are data matrices whose values are calibrated in the process of ANN's learning. This calibration corresponds to “recording” the learning of ANN between pairs of layers (each layer with its adjacent layer) in each iteration of the ANN. The level of recording the learning in each iteration depends on a factor called the **learning rate (LR)** which determines the level of remembering what was learned in each iteration. For this research, it is used a value of 0.1 as LR and weights are initialized randomly between the selected range, it is achieved just using a random function:

$$W = rand([-1, 1])$$

Inside the tangled process of calibrating the weights between layers, there exist a function dedicated to calculate the performance of every neuron into the network layers, this performance is calculated from the output of the neuron, called the ***synaptic potential*** **(SP)**evaluated into an equation called **transfer function**. The output of this function is the real output of the neuron, consisting in an equation that could be selected depending on the applications of the network, even it could be designed by each developer, but in this research is used the so famous**logistic or sigmoidal function**, represented by:

$$g(h) = \frac{1}{1 + e^{-\beta h}} \tag{2}$$

where:

g: the transfer function,
h: the synaptic potential of the neuron,
β: is just a regulator of the SP variability, in this case it is 1, therefore it does not affect the results.

5. DEEP LEARNING IN ACTION

5.1. Feeding the network with data

The input patterns with their 78 characterizing parameters get into the input layer, one parameter per neuron (it means the number of neurons in the input layer is the same than the number of the characterizing parameters). Each pattern is provided only one time by iteration. In each iteration, every pattern is taken randomly, it allows cleaning the network from any **cyclic learning**, it is not an official term but sounds good to express a repetitive learning which could calibrate the weights in a monotonous way obtaining as result constant values for the matrix weights.

5.2. Processing data through the shallow network

Once into the input layer, input patterns are processed along with the input-hidden weights using the transfer function, these results will give rise to the hidden layer, which in turn are processed together the hidden-output weights, using the transfer function too. Then, when the process reaches the output layer it comes back from the output layer to the input layer optimizing the value of the weight's matrices. Then, the process goes back to the output layer, returning the output for this pattern in form of a probability to be or not a hacking attack. If this probability is greater than 50%, the pattern is considered an attack, in the opposite case it is not an attack (1 or 0). Finally, this result is compared with the target parameter to get the error using the squared error function:

$$E_t = \frac{1}{N}\sum_{i=1}^{N}(Y_i - Z_i)^2 \tag{3}$$

where:

E_t: the total error of the network,
N: the number of patterns,
Y: the output of the network,
Z: the target (label parameter in input patterns).

This process is repeated by every pattern in each iteration, performing a reiterative process the number of times (iterations) previously selected or when a minimum error is reached.

5.3. Processing data through the Deep Learning network

The process in this architecture is almost the same at the last one. The difference is a previous step before data introduction into the input layer. This step is the autoencoder, which replace the old input layer. Now, data is introduced directly into the autoencoder which will extract information digging deeper between the great amount of introduced data. Input data is introduced only once into the autoencoder, it means the outputs of the autoencoder will be the new inputs of the shallow network, whose number of neurons in the input layer has been decreased, because it does not take all inputs anymore but it takes the extracted data resultant from the encoder. In this network, the number of the input layer has been reduced to 19 and the other layers keep the initial shape. A nice visualization of the new and deep network is showed in Figure 3. The BP algorithm of the deep network works exactly the same as the shallow network.

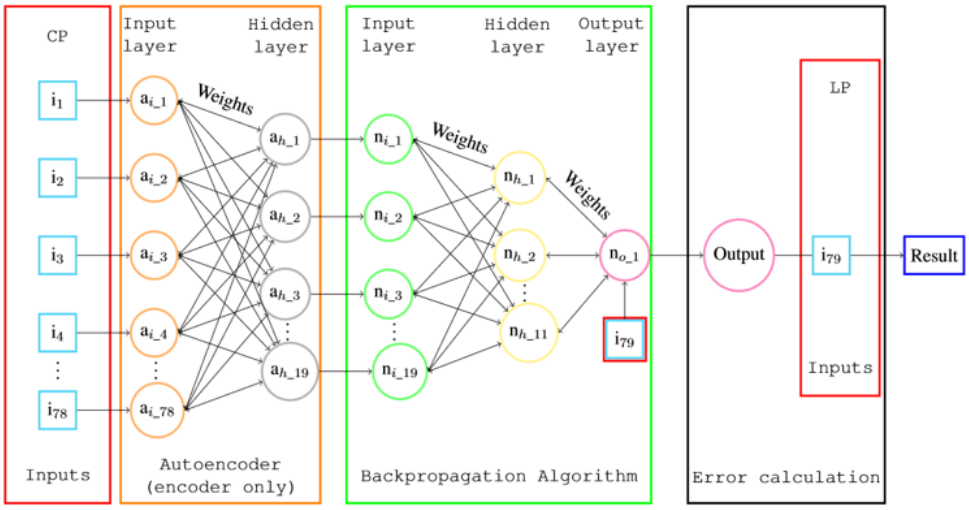


Figure 3. Ultimate visualization of the Deep Learning Network. Inputs have the same shape than before. Encoder is added. The shallow network (BPA) reduces its number of neurons.

5.4. Obtaining the final results

The most important results of this research lie in the fact of detecting hacking attacks in each one of the logs (input patterns) performed into processed data. In order to obtain the accuracy of the network, in both cases detecting an attack or discarding it, the testing phase results will pass new accuracy tests: **sensibility** and **specificity**, calculated by equations (4) and (5) respectively. Sensibility measures the proficiency to detect an attack when it is truly happening (true positives TP), a fail into the network will trigger a false positive (FP). Meanwhile, specificity takes care of cases which are not an attack but are normal activity (true negatives TN), a misclassification will lead a false negative (FN) which are more dangerous than FP because the non-detection of a real attack avoids a response against the attack.

$$Sensibility = \frac{TP}{TP + FN} \tag{4}$$

$$Specificity = \frac{TN}{TN + FP} \quad (5)$$

6. RESULTS

In order to show a good organization and prove the reliability and accuracy of the ultimate Deep Learning network, the results are shown in the same sequence than they have been performed: first using the shallow network and finally using the Deep Learning network. Every test use 1000 iterations for the autoencoder (do not confuse with the ANN's iterations), except for the last one, which uses 3500 iterations.

6.1. The Shallow Network

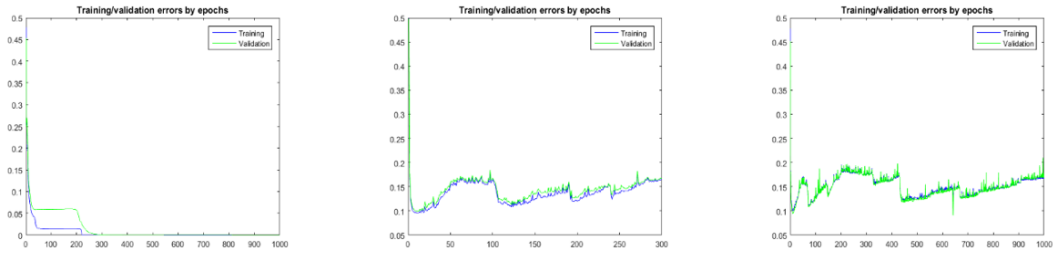
6.1.1. 150 inputs

At 1000 iterations/epochs it is reached an error of **0.0434** in the last iteration. It looks good because it means a performance of **95.66%** but 150 [inputs] is not a trustworthy number for an optimal training and Figure 4a is not exactly the objective to achieve in this research because these horrible fluctuations between iterations 50 and 250.

6.1.2. 6000 inputs

Let's use more data to work with the ANN. The first training with this dataset consists of 300 iterations resulting an error of **0.0961** and performance of **90.39%**, it does not look good anymore. Also, these peaks in Figure 4b shows a non-satisfactory training either, it evidences a low stability in the network given the extensive error variation between iterations which avoids a regular training.

Now, running 1000 iterations it is obtained an error of **0.0873** and a performance of 91.27**%** but Figure 4c shows the same behaviour than before which is not reliable: the poor stability of the network.



(a) 150 inputs, 1000 iterations. (b) 6000 inputs, 300 iterations. (c) 6000 inputs, 1000 iterations.

Figure 4. Training and validation results of the Deep Learning network.

6.2. The Deep Learning Network

6.2.1. 300 iterations

As a confidence prove to the final architecture, it has been selected directly 6000 inputs, number to be used in all experiments for this network. Using only 300 iterations it is reached an error of

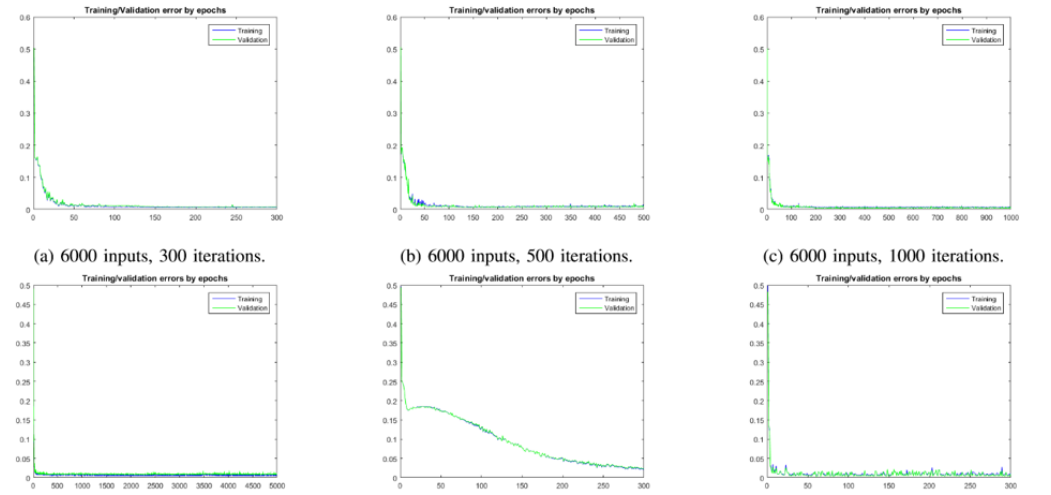
0.0084 which results in a performance of **99.16%**. It is a great percentage of accuracy and Figure 5a shows a behaviour better than any of the shallow network training.

6.2.2. 500 and 1000 iterations

Increasing the iterations to 500 results have improve, as expected, obtaining an error of **0.0068** corresponding to **99.32%** of performance, numbers which are obtained using 1000 iterations too, which demonstrate a stable behaviour in the network. The plotting of error by iterations does not change as seen in Figure 5b and5c.

6.2.3. 5000 iterations

It is just a checking to know whether the network will response in an unusual way given the large amount of iterations. The result is a performance better than any other one seen before in this research, reaching an error of **0.0060** equivalent to **99.40%** of performance. Obviously, the visualization is better too (Figure 5d). It is the better performance of the network to all experiments executed.



(a) 6000 inputs, 300 iterations. (b) 6000 inputs, 500 iterations. (c) 6000 inputs, 1000 iterations.

(d) 6000 inputs, 5000 iterations. (e) 6000 inputs, 300 iterations, LR = 0.01. (f) 6000 inputs, 1000 iterations, LR = 0.5.

Figure 5. Training and validation results of the Deep Learning network.

6.3. Other modifications in the Deep Learning Architecture

6.3.1. LR = 0.01

Decreasing ten times the LR selected to the last experiments gives as result a singular behaviour in the network, in which error is minimized in the last iterations. In the first iterations it is visualized a high error in respect to the last training. Also, it is appreciated a rise of the error which are not a good signal for the purposes of this work. Figure 5e shows all these details clearly. The error obtained **0.0165** and a performance of **98.35%**.

6.3.2. LR = 0.5

This experiment is the opposite case of which was made before: increasing the learning rate to 0.5. In this case the behaviour is similar to the experiments with LR=0.1 with an error of **0.0090** and a performance of **99.10%** but it is notorious the existent fluctuations in the error (Figure 5f)

which is derived in low stability. In any case, the LR=0.1 with 5000 iterations is still better given its performance and stability. A summarize of all results is visualized in Table 1.

Table 1. A summary of results from different architectures of the ANN.

N.	# inputs	Iterations	LR	Autoencoder	Error	Performance (%)	Stable
1	150	1000	0.1	No	0.0434	95.66	No
2	6000	300	0.1	No	0.0961	90.39	No
3	6000	1000	0.1	No	0.0873	91.27	No
4	6000	300	0.1	Yes	0.0084	99.16	Yes
5	6000	500	0.1	Yes	0.0068	99.32	Yes
6	6000	1000	0.1	Yes	0.0068	99.32	Yes
7	6000	5000	0.1	Yes	0.0060	99.40	Yes
8	6000	300	0.01	Yes	0.0165	98.35	Yes
9	6000	300	0.5	Yes	0.0090	99.10	No

6.4. False positives and false negatives

Using the best results of Table 1, the experiment number 7: 2000 new patterns totally unknown to the network and equations (4) and (5) it is obtained Table 2 which shows the accuracy of the network, it means, the capacity of the network to label a pattern as hacking or normal activity.

Table 2. FP and FN results: using the best architecture.

Total patterns	**Well-classified**	**False positive**
2000	1870	130
True Positive	**True Negative**	**False Negatives**
1000	870	0
Sensitivity	**Specificity**	**Accuracy**
1	0.87	93.5%

An ultimate experiment is performed, changing the number of iterations for the autoencoder to 3500, the network performs its respective training and validations steps and results looks so much better (Table 3).

Table 3. FP and FN: new results.

Total patterns	**Well-classified**	**False positive**
2000	1870	130
True Positive	**True Negative**	**False Negatives**
1000	870	0
Sensitivity	**Specificity**	**Accuracy**
1	0.915	95.75%

7. DISCUSSION

Several trial and error were required to obtain the good results presented in this work. It is shown that the algorithm works so much better using the Deep Learning architecture because the autoencoder is a powerful tool which extract information which maybe is hidden for the BPA only. These results are reflected in the poor stability of the BPA, in the other hand, the Deep Learning architecture has a high stability.

It is expected to do further research in this area using more variability in the configuration of the Deep Learning architecture and incorporate self-tuning techniques like agents and genetic algorithms. Testing the network using different datasets will be an important advance for this kind of researches or even the creation of an own dataset to have the total control of all characteristics of data.

8. CONCLUSION

This paper describes a Deep Learning environment where a tuned shallow network and an autoencoder trained with unlabelled big data, collaborate as to produce a reliable a trainable intrusion detection system, operating in access to web situations. The system uses the CICIDS2017 dataset and the autoencoder to produce abstract representation of the activities given into the computer systems. This compressed information is fed to a secondary shallow network that completes the possible attack entry prediction. It is proven that the composed Deep Learning network has a better performance than any other shallow network variation. The resultant functional MATLAB code behaves as a re-trainable, fast, reliable system proved with real data.

ACKNOWLEDGEMENTS

The authors thankfully acknowledge The Canadian Institute for Cybersecurity (CIC) since they freely provided their research data.

REFERENCES

[1] N. Garg, D. Kumar, Y. Khera, Sujay, and P. Jain, “Towards the impact ofhacking on cyber security,”*IIOAB Journal*, vol. 9, pp. 61–77, 05 2018.
[2] G. Jha, “Artificial neural networks and its applications,” 01 2019.
[3] M. Mijwel, “Artificial neural networks advantages and disadvantages,” 01 2018.
[4] H. Lodish, A. Berk, C. A. Kaiser, M. Krieger, A. Bretscher, H. Ploegh,A. Amon, and M. P. Scott,*Molecular Cell Biology*, 7th ed. W.H. Freemanand Company, 2013.
[5] R. Sathya and A. Abraham, “Comparison of supervised and unsupervised learning algorithms for pattern classification,”*International Journal ofAdvanced Research in Artificial Intelligence*, vol. 2, no. 2, 2013.[Online]. Available: http://dx.doi.org/10.14569/IJARAI.2013.020206
[6] Y. LeCun, Y. Bengio, and G. E. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available:https://doi.org/10.1038/nature14539
[7] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” *2017 International Conference on Engineering and Technology (ICET)*, Aug 2017, pp. 1–6.
[8] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, “Deep learning approach for intelligent intrusion detection system,” *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.
[9] Jia, F. & Kong, L.-Z. (2017). “Intrusion Detection Algorithm Based on Convolutional Neural Network,” *Beijing LigongDaxueXuebao/Transaction of Beijing Institute of Technology*. 37. 1271-1275. 10.15918/j.tbit1001-0645.2017.12.011.
[10] McHugh, John. (2000). “Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory,”*ACM Trans. Inf. Syst. Secur.*,3. 262-294. 10.1145/382912.382923.
[11] Biswas, Saroj. (2018). “Intrusion Detection Using Machine Learning: A Comparison Study,” *International Journal of Pure and Applied Mathematics*. 118. 101-114.
[12] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, 2009*, pp. 1-6, doi: 10.1109/CISDA.2009.5356528.

[13] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating anew intrusion detection dataset and intrusion traffic characterization,”*4thInternational Conference on Information Systems Security and Privacy (ICISSP)*, Portugal, Jan. 2018

[14] A. Gharib, I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "An evaluation framework for intrusion detection dataset", *Proc. Int. Conf. Inf. Sci. Secur. (ICISS)*, pp. 1-6, Dec. 2016.

AUTHORS

Sergio Hidalgo-Espinoza is a teaching technician from University of Yachay Tech. He is a graduate of the same university in 2019 as Engineer in Information Technology. His main interests in researching are Information Security, Artificial Neural Networks, High-Performance Computing and Mathematical Modelling.

Kevin Chamorro-Cupuerán is a teaching technician from University of Yachay Tech. He is a graduate of the same university in 2019 as Mathematician. His main interests in researching are Artificial Neural Networks, Statistics, Mathematical Modelling and Data Mining.

Professor Dr. Oscar Chang-Tortolero is an experimented researcher in Artificial Neural Networks. He is actually full-professor at University of Yachay Tech.

AN OPTIMIZED CLEANING ROBOT PATH GENERATION AND EXECUTION SYSTEM USING CELLULAR REPRESENTATION OF WORKSPACE

Qile He[1] and Yu Sun[2]

[1]Webber Academy, Calgary, Alberta, Canada
[2]California State Polytechnic University, Pomona, USA

ABSTRACT

Many robot applications depend on solving the Complete Coverage Path Problem (CCPP). Specifically, robot vacuum cleaners have seen increased use in recent years, and some models offer room mapping capability using sensors such as LiDAR. With the addition of room mapping, applied robotic cleaning has begun to transition from random walk and heuristic path planning into an environment-aware approach. In this paper, a novel solution for pathfinding and navigation of indoor robot cleaners is proposed. The proposed solution plans a path from a priori cellular decomposition of the work environment. The planned path achieves complete coverage on the map and reduces duplicate coverage. The solution is implemented inside the ROS framework, and is validated with Gazebo simulation. Metrics to evaluate the performance of the proposed algorithm seek to evaluate the efficiency by speed, duplicate coverage and distance travelled.

KEYWORDS

Complete Coverage Path Planning, Mobile Robots, Graph Theory.

1. INTRODUCTION

As the robot vacuum cleaner technology [1] makes its way into the homes of millions, a need arises to further increase their efficiency. Current robot cleaners often adopt a random walk approach to area coverage [2]. Indeed, given enough time the monte-carlo approach will eventually achieve full coverage of the cleaning area, but it is certainly suboptimal in terms of coverage duplication, coverage speed and coverage completeness. With the recent emergence of cleaning robots equipped with Lidar, it is now possible for the robot to map its environment and plan its path accordingly, to optimize for certain performance metrics. For the application of robotic vacuum cleaners specifically, reasonable metrics to optimize along include speed, and coverage completeness [3].

In order to attack the complete coverage path problem, a representation of the robot's environment is required. Usually, the working space of the robot is divided into small square cells, each assigned a value that indicates how likely, or in our case, whether, an obstacle is present [4]. This approach, termed the occupancy grid, allows a graph representation of the workspace, and correspondingly, path finding and search algorithms can be applied. However, the occupancy grid model as-is is deeply flawed for tasks such as cleaning. The occupancy grid method is discrete, when in reality the set of possible robot positions is continuous. This

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 13-25, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101502

discretization is likely to produce variation in performance between a map with orthogonal features and one with features not aligned to the occupancy grid orthogonally. Most existing research avoids this issue because they use mostly orthogonal features in their performance evaluation [5][6]. In addition to lack of consideration for non-orthogonal features, the robot's footprint is not considered at the path generation level. Existing research adopts the method of obstacle inflation, as in marking the neighbouring areas of an obstacle as obstacles also [7][8]. This method addresses robot footprint potentially overlapping with obstacles in real life, but does not address potential duplicate coverage caused by the robot being larger than the occupancy grid cell size. On the evaluation stage, many existing explorations evaluate their algorithms within the occupancy grid model. These papers focus on whether creating the shortest path as represented by the occupancy grid. However, the shortest path approach does not guarantee optimized traversal time. For example, two paths that have identical length in the occupancy grid can have very different traversal time in real life, due to one requiring frequent acceleration and deceleration.

While existing solutions focus on a shortest distance complete coverage path, this paper proposes a more complete solution that addresses various problems that arise when paths represented in the occupancy grid are applied to real-life environments. The effectiveness of this approach is evaluated in lifelike computer simulation of a robot cleaning task scenario. The first problem is that paths on the occupancy grid representation are an ordered collection of poses. In many existing explorations, the robot's footprint is not considered. However, for tasks like room cleaning, the footprint of the robot is crucial to create a path that not only visits all the points on the occupancy grid, but also one that visits all the points in the physical working space with as little duplication as possible. In many cases the occupancy grid cells are smaller than the robot footprint, and the generated path may cause duplicate coverage if it visits all the cells in the occupancy grid. This path planning algorithm accounts for the footprint of the robot. The algorithm is a modified version of Depth First Search (DFS) [15]. Instead of marking each traversed node in the graph representation as visited, the algorithm marks all nodes within the robot's footprint on the occupancy grid as visited. The center node of the footprint is the midpoint of the wheel axis in a differential drive robot. The algorithm will only write the center node's position at each step of the traversal into the path. The proposed algorithm can then recursively visit cells neighboring the center cell and mark the entire robot footprint surrounding that cell as visited, as long as the robot's footprint is clear from visited points or obstacles. When the modified DFS visits a cell that has no available neighbors, it finds the nearest unvisited cell and uses Dijkstra [16] to find a path to that block. Thus, the occupancy grid's resolution does not have to match the robot's size, and can be smaller by the robot's diameter by an arbitrary (ideally integer) positive number of times. With the robot footprint accounted for at the path-generation level, the proposed algorithm significantly reduces duplicate coverage caused by occupancy grid resolution being incompatible with actual robot size, saving time and energy. In addition to optimizations to reduce duplicated coverage, the proposed algorithm also takes into account the effects of target pose distancing on navigation. Given that unit error in heading leads to error in position proportional to distance, and that embedded controllers often have limited memory space and processing capacity to handle a high density of target poses, the target poses generated from the algorithm requires post processing. By pruning points in a straight line, the proposed algorithm increases smoothness of motion and reduces computational load.

Using Gazebo with a cleaning application scenario with complex geometries such as oblique walls, small pillars and acute angle corners, the path generation algorithm's performance is evaluated. A simulated Turtlebot3 Burger completes the planned path in this evaluation. After the path generation algorithm is run, the sum of the number of times nodes are repeated is calculated. The duplication rate can then be calculated by dividing the number of instances of duplicated visits by the total number of points in the path. This happens before the post-processing, and is

therefore a relatively fine-grained measure of duplicated coverage. During the robot run, performance metrics are collected. Total distance traveled is measured by summing the Euclidean distance between the current ground truth pose and the last ground truth pose. Time is also measured. With these two metrics speeds can be calculated. Speed reduction rate measures the extent to which the robot is slowed down by having to decelerate and accelerate, as opposed to operating near the top speed most of the time. This is calculated by dividing the empirical average speed by the maximum speed parameter of Turtlebot3 Burger. These factors are crucial to the effectiveness and satisfaction of robot cleaning products.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the design, implementation and evaluation of the algorithm; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the evaluation of the path planner following by presenting related works in Section 5. Finally, Section 6 summarizes the results of this paper, as well as providing some insights on future works that could further enrich this paper.

2. CHALLENGES

To obtain a path generation and execution solution that is optimized for speed and coverage, a few challenges have been identified as follows.

2.1. Challenge 1: The Need of Optimized Path Post-Processing

DFS generates path points that are equally spaced apart. This creates many superfluous path points that lie on the same straight line. Removing these superfluous points creates significant savings in computation resources.

2.2. Challenge 2: The Complexity of Navigation Control

Traditional closed loop control creates curved trajectories for differential drive robots. This is problematic because the robot can run into obstacles unexpectedly. For maximum conformity to generate paths, the trajectory between any pair of path points needs to be a straight line. For equal angular speed, increased linear speed leads to increased disturbance in position. Therefore, the angular position control loop is scaled inverse to linear speed, such that deviation from the ideal trajectory is minimized.

2.3. Challenge 3: The Automation of the Occupancy Grid Transformation

The occupancy map model allows a high level of abstraction of the real-life environment such that existing algorithms can easily apply. However, translating the results of these abstracted algorithms into practice presents a set of problems. Firstly, gapping as implemented in ROS [12] has faulty coordinate transformation parameters. Therefore, manual calculation is required to set the parameters with which a transform between map coordinates and simulation world coordinates is conducted. Secondly, occupancy grid's limited resolution and poor representation of oblique lines and curves meant that it cannot be relied on for closely tracing the edges of obstacles. Instead, the obstacles as represented on occupancy grid must be inflated by the radius of the robot.

2.4. Challenge4: The Difference between Algorithm-Based Coverage and the Real Coverage

If coverage were to be measured by whether the robot has indeed visited all the points, as represented by the occupancy grid, it would almost always be 100%, by definition of a full coverage path. Therefore. measurements beyond the precision of the occupancy grid is required for a meaningful evaluation of robot performance. By approximating the robot trajectory as a series of short straight lines, the coverage area can be approximated as a series of quadrilaterals, with the four corners being the position of the left and right ends of the robot cleaning tool, at the beginning and the end of the straight line. As the sample rate of robot position increases, the accuracy of this approach increases. The four edges of each quadrilateral is represented by a linear inequality. This representation can then be operated on to calculate if an area has duplicate coverage.

3. SOLUTION

3.1. Overview

The proposed algorithm in this paper is implemented with Python. Before post-processing to reduce density of points in the path, the duplication rate of the path is measured. After a path is generated and pose-processed, it is stored in a YAML file. A Turtlebot3 Burger is simulated in Gazebo, integrated with ROS [13]. A ROS node serves as the navigation stack, by reading the path file and issuing velocity commands that take the robot to the current target pose by the shortest straight line before repeating the same process for the next.

3.2. Path Finding in Action

The simulations are carried out with a model of the Turtlebot3 Burger, a differential drive robot equipped with a laser ranger. Given that differential drive is commonly used in indoor cleaning robots, Turtlebot3 Burger is a good approximation of real-life hardware that relies on full coverage path planning.

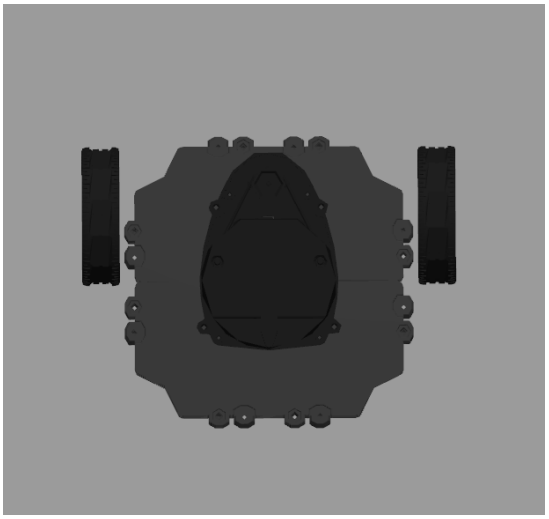


Figure 1: Turtlebot3 Burger

Robot Operation System, or ROS for short, is a framework that facilitates code reuse in robotics by packaging robot subsystem information in a distributed network. The network, or “ROS

runtime graph", consists of nodes broadcasting information, or "messages" about the robot, organized under different topics: heading, speed, sensor data, control commands, etc. Messages are shared through a subscription model, where a node broadcasts its messages independent of receivers. With ROS runtime network, robot subsystem code can be encapsulated, abstracted and reused. Within the ROS framework, a navigation node is written to carry out the planned paths.

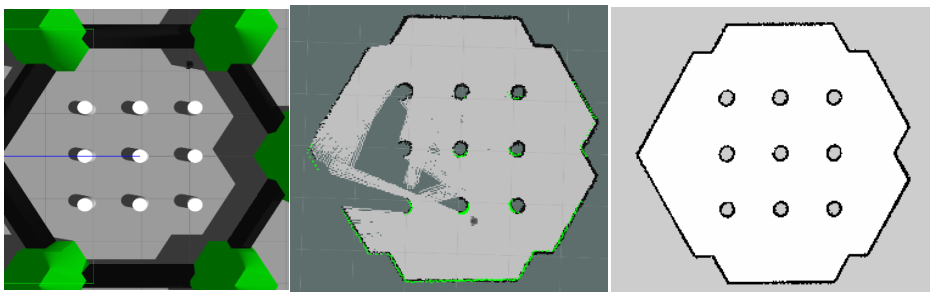


Figure 2: Robot workspace in Gazebo, Rviz LiDAR data view and occupancy grid view

While ROS provides a general framework of a robot system, Gazebo integrates with ROS to provide realistic simulations of real-life robot use cases. With realistic rigid-body dynamics, 3D graphics, and sensor noise generation, Gazebo is a reliable tool for evaluating robot software performance. Using Gazebo, Turtlebot3 Burger, and an example room, is simulated. Additionally, Gazebo messages provide a ground truth for robot position, with which many performance metrics, such as coverage completeness, coverage overlap or coverage time can be calculated.

Rviz is a ROS tool that visualizes ROS topics. Using Rviz, the actual path traveled by Turtlebot3 Burger is visualized.

In order to create a path, a priori knowledge of the robot's environment is required. Turtlebot3 Burger's laser rangefinder is used to carry out this task. When mapping the room, Turtlebot3 Burger travels in straight lines, and makes a turn at a random angle when its bumper sensors detect an obstacle. Given sufficient time, Turtlebot3 Burger can obtain a perspective on all areas in the room to provide a complete map. Meanwhile, odometry provides the relative position of the robot. With this relative position, as well as laser range sensor data, an occupancy grid is created using Gmapping from OpenSLAM. The occupancy grid is a 2D list of occupancy states (occupied, free or indeterminate), indexed by their positions.

3.3. Path Generation

Path planning algorithm is implemented in a python [14] script using Dijkstar library. Before generating the path, the script inflates obstacles and walls by a safety margin of 5 cells. After the inflation, the occupancy grid bmp file is read, and occupancy status of each tile stored in a 2-dimensional list. The robot is represented by a square block of tiles approximating its footprint. The center of this square block is the rotation axis of the differential drive chassis. A modified DFS is implemented, with the aim of having the center cell traverse the occupancy grid. At each move, the entire block of cells is checked for obstacles, such that every movement of the center cell is clear for the entire robot. The center cell's coordinates are then stored in a list, which would be the list of goal points. Meanwhile, the script marks the surrounding area of the visited center cell that is a radius of the block back in the goal points list as visited. Thus, each move of the center cell would not cause the robot footprint to intersect with a section already covered. When DFS runs into a block where no neighbors of the center cell may be added without running

into a visited cell, Dijkstra is used to find the nearest cell by Manhattan distance and generate a path to that cell. After a list of path points is generated, a function iterates through the list to check for collinear sections of goal points and returns a new goal points list that trims the excess points, while maintaining an arbitrary minimum density of goal points to prevent open loop behavior in the navigation stack. As the new goal points list is generated, the heading angle between each point in the path and the next point is calculated, and appended to the current point in the iteration. This list of goal poses is then formatted into YAML and stored.

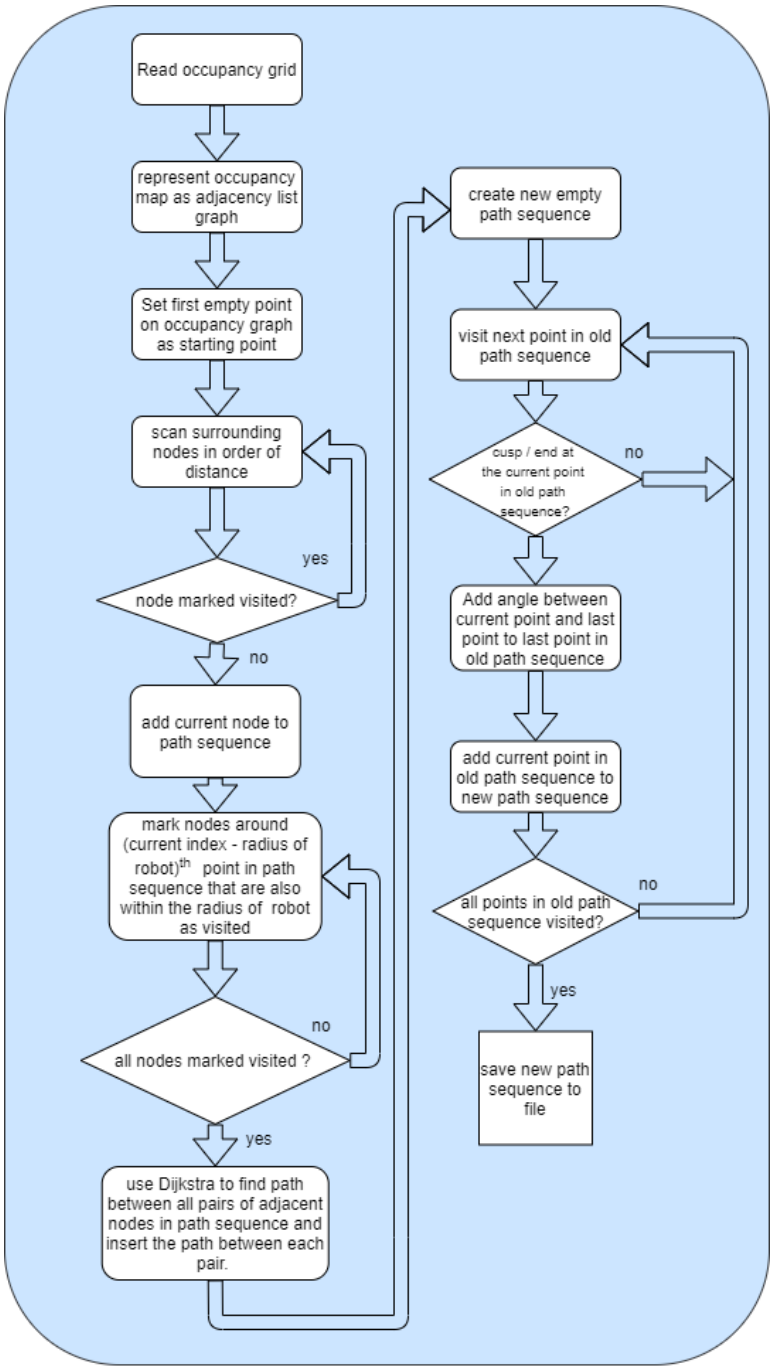


Figure 3: An Excerpt of the Path Generation and Cleaning Algorithm

3.4. Navigation

In order to determine how planned paths, perform despite imperfections induced by real-world physics, a navigation node is written to execute the planned paths in Gazebo. The navigation node reads the list of goal poses from the aforementioned YAML file. The node then calls a function that carries out the navigation between the robot's actual pose and the desired pose. When the robot reaches the vicinity of the goal pose, the navigation function exits and is called again with the next goal pose. Similar to real-life cleaning robots, absolute localization is carried out with odometry with the knowledge of robot initial pose. For each target the navigation node performs two operations. The first is to drive the robot from the initial position to the target, and the second is to rotate the robot to align with the orientation of the target pose. For the first step, the algorithm uses a proportional control loop to minimize two variables: heading error and position error. The target heading during this step is the angle between the initial position and the target position. When this parameter is minimized, only straight movement is required to reach the target. The angular speed in this step is proportional to heading error, and furthermore, to the inverse of linear speed. Because displacement for a given angular speed is proportional to linear speed, the heading adjustment angular speed is proportional to its inverse such that the heading adjustment behavior is consistent for varying distances between target and initial position. The linear speed is proportional to distance error raised to a power of larger than one, such that when the distance is below 1, the robot slows down and eventually stops.

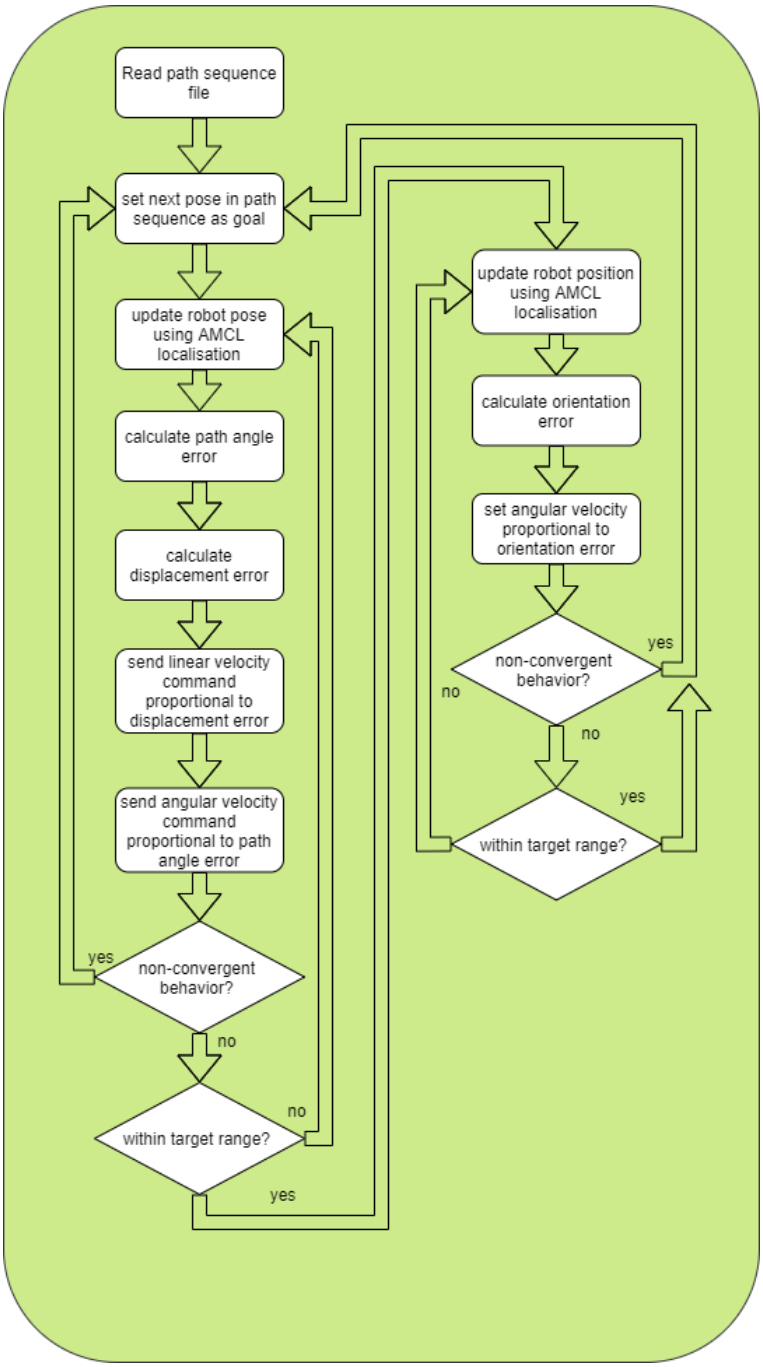


Figure 4: An Excerpt of the Navigation Algorithm

4. EXPERIMENT

The robot carries out the generated path within the Gazebo/ROS simulation environment. The navigation node takes in a desired pose and publishes commands to take the robot to the pose in a straight line maintained by closed loop control of heading. During the run, several pieces of data are calculated: area covered, area duplicated, distance travelled time taken, and poses not reached.

4.1. Occupancy Grid Efficiency

The most straightforward approach to measure the efficiency of the generated path is to utilize the generated points in path with the map dimensions. This shows a very accurate calculation on the coverage, as well as the duplicated points in the path. Even though the grid does not fully represent the actual coverage situation in reality, the efficiency result provides the evaluation from the algorithm foundation.

As shown in Table 1, the generated path only produces a 7.8% duplicated path while covering the 100% space.

Table 1: The Experiment Result of the Occupancy Grid Efficiency

Map	Total Pixels in Path	Duplication Instances	Duplication Rate
Indoor 1	9740	755	7.8%

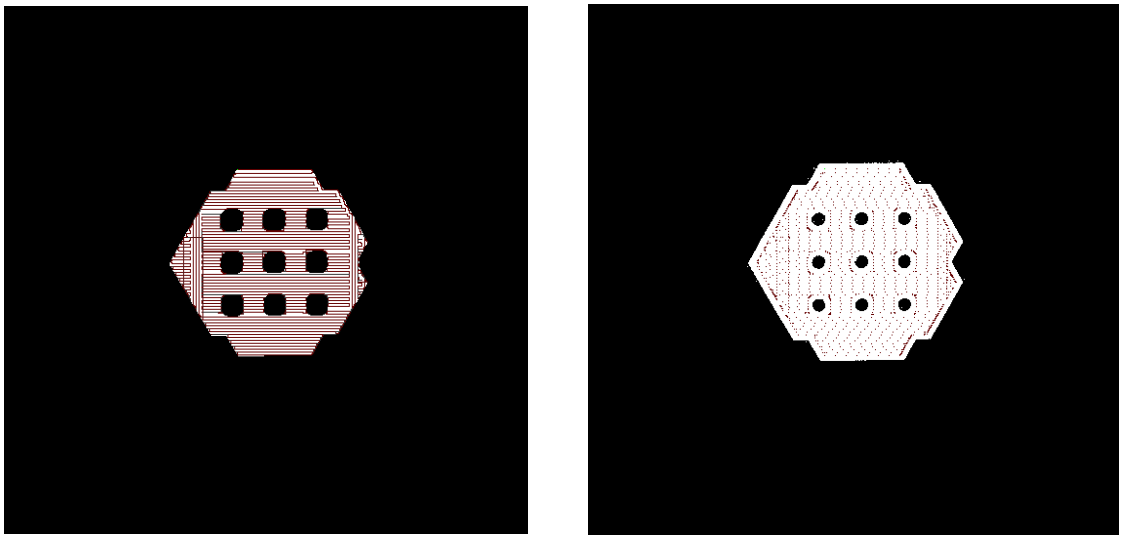


Figure 5: Generated path with duplicated blocks in gray (left) and target points after post-processing (right)

4.2. Simulation-Based Completeness

The area swept over subtracted by area duplicated yields the real covered area. The real covered area compared to the total free area on the map yields the real completeness of the coverage algorithm. This piece of data is obtained by a polygonal approximation of the area covered by the robot, and approaches a high accuracy as the robot position sample rate increases.

4.3. Simulation-Based Efficiency

The average speed of the robot is obtained by calculating the Euclidean distance over a traversal of points in the path sequence and dividing this distance by time taken. The average speed of the robot can be used to determine the amount of acceleration, deceleration and turns the robot goes through. The average speed as a percentage of the maximum speed represents how efficient the planned path is optimized for robot kinematics.

Table 2: The Experiment Result of the Simulation-based Efficiency

Map	Total Distance Traveled (m)	Total Time (s)	Average Speed (m/s)	Speed Reduction Rate
Indoor 1	241	3260	0.074	74%

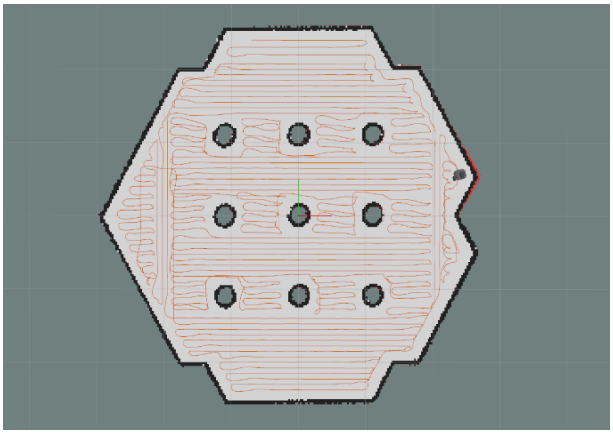


Figure 6: Robot trajectory during execution of generated path

5. DISCUSSION

5.1. Experimental Analysis

The generated path displays occupancy grid completeness by making sure that all points in the inflated occupancy grid are visited. It is also able to do this efficiently, with a duplication rate of 7.8%. It is worth noting that in real application, completeness can sometimes be sacrificed in favor of speed. Much of the duplicated coverage is caused by the robot travelling a long distance to visit one isolated cell. Given that it is difficult for humans to perceive the difference in cleaning results caused by the robot missing a number of small, isolated areas, coverage speed of this algorithm can be further improved by trimming points that are too costly to reach at little expense of completeness. The generated path follows a zig-zag scanning pattern at most places. The zig-zag pattern can be horizontal or vertical in orientation. More importantly, the proposed algorithm does not enforce scanning one row/column at a time, and is able to efficiently fill one local area bounded by obstacles for the most part and backtracking to another divided area. Given that our implementation of DFS weighs neighboring blocks on Manhattan distance, and selects the first unvisited block in the nearest equidistant set of neighbors, the zig-zag pattern is expected. If neighboring blocks during the search are weighted by other preferences, DFS may produce different patterns. For example, a wall- following spiral pattern can be encouraged by decreasing the cost of blocks that are near an obstacle.

The experimental validation shows that the proposed algorithm achieves complete coverage of the inflated occupancy grid even in environments that have a large number of oblique features. Using proportional control, the robot was able to follow the generated path to a precision such that there are no collisions throughout the test. Figure 6 shows the robot's trajectory in a full run. During this run, the robot averages a speed that is 74% of the maximum speed. While this number is subject to influences such as the particular robot's acceleration and deceleration capacity, it is true to Turtlebot3 specifically, due to Gazebo's ability to handle robot kinematic constraints according to manufacturer's specifications.

5.2. Related Works

Oh et al. attempts to create smoother movements by using a triangular cellular decomposition of the map [9]. We use the more traditional square cellular decomposition in the form of the occupancy grid, however, our methods of DFS path generation and duplicate coverage reduction does not depend on the shape of the cells, and can be extended to operate on an arbitrary connected graph as long as the geometric relationship between each node on the graph is defined (in a square cellular decomposition, neighboring cells are in the cardinal directions).

De Carvalho et. al uses a series of movement templates [10], including straight forward, turn and U-turns to carry out their planned paths. They seek to incorporate geometric constraints and kinematic characteristics into path planning for optimal speed. We incorporate the geometric constraint of the robot at the path planning stage by applying a robot shaped mask that marks its vicinity as visited during the path search. Additionally, linear approximation of the path point collection in this paper reduces unnecessary turning when covering and smooths out the path, thus reducing acceleration and deceleration.

Gonzalez et al. proposes Backtracking Spiral Algorithm, which generates a complete coverage path by spiraling from the perimeter of the environment inwards, and backtracking [5] to an unvisited region when the robot reaches the center of a local spiral. As DFS is a greedy algorithm, each path point is likely to be followed by its immediate neighbors on the occupancy grid. As a result, the proposed algorithm tends to continue in a straight line, and thus fills the map with zig-zag patterns. Whereas in the open space the spiral filling pattern is ostensibly better at reducing unnecessary acceleration, deceleration and turns, the superiority of the spiral filling pattern in this regard has not been established for all possible shapes of the workspace.

Khan et al. uses a zig-zag pattern to scan the map [6]. When the zig-zag scan leaves regions uncovered, the paper proposes two-way proximity search to backtrack to the border of the unscanned region. Whereas the above paper enforces scanning in an arbitrary direction, there is no enforced orientation of zig-zag filling in DFS.

Lee et. al improves upon the BSA approach by generating Bezier curves from the BSA path [11]. This is done to increase the smoothness of robot movement and reduce physical coverage time. This paper smooths out paths by constructing polylines from the collection of points in the path. Because DFS favors long straight paths, i.e. zig-zag patterns, the polyline method is more appropriate because it requires less angular acceleration for most of the path, whereas a Bezier curve would turn a set of points approximating a linear shape into a snaking path.

6. CONCLUSIONS AND FUTURE WORK

This paper proposes a novel path-motion planning solution for complete coverage in indoor differential robots. Cellular decomposition is implicitly applied when an occupancy grid is generated from SLAM. Using this cellular representation, an algorithm is used to visit all the cells in the occupancy grid. Beyond cell completeness, this paper seeks to maximize real coverage and reduce duplicate coverage by incorporating the robot's shape into path generation. The algorithm is tested in the ROS/Gazebo environment where the simulated robot carries out the path generated with the proposed algorithm. The robot carries out a test run in the test environment, during which data such as area covered, duplicate coverage and time are collected to validate the feasibility of this paper's proposal. The source of the completed implementation of the algorithm can be found at [17].

Some limitations exist in this paper's solution as-is. Firstly, this paper assumes complete a priori knowledge of the workspace, whereas real life applications may deal with unexpected obstacles. However, collisions can still be avoided using data from the cleaning robot's sensor suite. Moreover, real-time re-planning may be implemented simply by deleting inaccessible cells from the path sequence. Secondly, the a priori workspace knowledge must be obtained using specialized sensors, such as LiDAR or RGB-D stereo camera. Both sensors have seen limited application in indoor robots, but high price precludes these sensors from thoroughly penetrating the market of home cleaning robots. Thirdly, the current path is represented by a poly-line approximated from the cell visit sequence generated by the proposed algorithm. For an accurate execution of the poly-line representation, the robot must decelerate at each node in the poly-line to perform a real pivot turn. This may lead to extra time consumption. This problem can be partially solved by rounding the corners in the poly-line to reduce need for braking.

We would address the current solution's limited ability in dealing with unexpected obstacles by incorporating real-time re-planning capabilities. We would also direct our focus to further post-processing of the generated path for improved smoothness.

REFERENCES

[1] Robot vacuum cleaner, by T. Charles, A. Parker, S. Lau, E. Blair, A. Henginer, E. Ng, E. DiBernardo, R. Witman, M. Stout. (2006, Jan. 26). U.S. Patent 20060020369A1 [Online]. Available: https://assignment.uspto.gov/patent/index.html#/patent/search/resultAssignment?searchInput=20060020369&id=23224-384

[2] Random motion cleaner, by D. E. Gerber, K.L Thomas. (2002, Jun. 6). U.S. Patent 6571415B2 [Online]. Available: http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&l=50&d=PALL&s1=6571415.PN.

[3] W. Brockman and D. Klingbiel, "Rating the Performance of Robotic Coverage Tasks," Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles July 5-7, 2004 Lisboa, Portugal, vol. 22, no. 6, pp. 46-57, June 1989, doi: 10.1109/2.30720.

[4] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," in Computer, vol. 22, no. 6, pp. 46-57, June 1989, doi: 10.1109/2.30720.

[5] E. Gonzalez, O.Alvarez, Y.Diaz, C.Parra and C.Bustacara, "BSA: A Complete Coverage Algorithm," Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, pp. 2040-2044, doi: 10.1109/ROBOT.2005.1570413.

[6] A. Khan, I. Noreen, H. Ryu, N. L. Doh, and Z. Habib, "Online complete coverage path planning using two-way proximity search," Intel Serv Robotics 10, 229–240 (2017). doi: 10.1007/s11370-017-0223-z

[7] J. Connors and G. Elkaim, "Analysis of a Spline Based, Obstacle Avoiding Path Planning Algorithm," 2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring, Dublin, 2007, pp. 2565-2569, doi: 10.1109/VETECS.2007.528.

[8] Yuan, R., Zhang, F., Qu, J., Li, G. and Fu, Y., "A novel obstacle avoidance method based on multi-information inflation map," Industrial Robot, Vol. 47 No. 2, pp. 253-265. https://doi.org/10.1108/IR-05-2019-0114

[9] Joon Seop Oh, Yoon Ho Choi, Jin Bae Park and Y. F. Zheng, "Complete coverage navigation of cleaning robots using triangular-cell-based map," in IEEE Transactions on Industrial Electronics, vol. 51, no. 3, pp. 718-726, June 2004, doi: 10.1109/TIE.2004.825197.

[10] R. N. De Carvalho, H. A. Vidal, P. Vieira and M. I. Ribeiro, "Complete coverage path planning and guidance for cleaning robots," ISIE '97 Proceeding of the IEEE International Symposium on Industrial Electronics, Guimaraes, Portugal, 1997, pp. 677-682 vol.2, doi: 10.1109/ISIE.1997.649051.

[11] T. Lee, S. Baek, Y, Choi, and S. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," Robotics and Autonomous Systems, Vol. 59, No. 10, pp. 801-812. https://doi.org/10.1016/j.robot.2011.06.002

[12] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.

[13] Santos, Joao Machado, David Portugal, and Rui P. Rocha. "An evaluation of 2D SLAM techniques available in robot operating system." 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2013.
[14] Quigley, Morgan, Brian Gerkey, and William D. Smart. Programming Robots with ROS: a practical introduction to the Robot Operating System. " O'Reilly Media, Inc.", 2015.
[15] Korf, Richard E. "Depth-first iterative-deepening: An optimal admissible tree search." Artificial intelligence 27.1 (1985): 97-109.
[16] Jianya, Yue Yang Gong. "An efficient implementation of shortest path algorithm based on dijkstra algorithm [j]." Journal of Wuhan Technical University of Surveying and Mapping (Wtusm) 3.004 (1999).
[17] Project source code. https://github.com/Jackack/Turtlebot3_complete_coverage

TOWARDS A RISK ASSESSMENT MODEL FOR BIG DATA IN CLOUD COMPUTING ENVIRONMENT

Saadia Drissi, Soukaina Elhasnaoui, Hajar Iguer,
Siham Benhadou and Hicham Medromi

(EAS-LRI) Systems Architecture Team, ENSEM, Hassan II University
Pluridisciplinary Laboratory of Research & Innovation (LPRI), EMSI
Casablanca, Morocco

ABSTRACT

Cloud computing gives a relevant and adaptable support for Big Data by the ease of use, access to resources, low cost use of resources, and the use of strong equipment to process big data. Cloud and big data center on developing the value of a business while reducing capital costs.

Big data and cloud computing, both favor companies and by cause of their benefit, the use of big data growths extremely in the cloud. With this serious increase, there are several emerging risk security concerns. Big data has more vulnerabilities with the comparison to classical database, as this database are stored in servers owned by the cloud provider. The various usage of data make safety-related big data in the cloud intolerable with the traditional security measures.

The security of big data in the cloud needs to be looked at and discussed. In this current paper, my colleagues and me will present and discuss the risk assessment of big-data applications in cloud computing environments and present some ideas for assessing these risks.

KEYWORDS

Cloud computing, risk assessment, big data, security

1. INTRODUCTION

In the recent years, there has been an acceleration use of big data in cloud computing environment, in several domains such as science, health, finance and government [2]. Cloud computing provides an important support for big data by accessing to resources, low cost and the usage of robust container to process big data. As the product big data needs to be stored in the cloud, the cloud or the container of data needs many servers linked to each other to distribute the computing tasks by using internet.

Cloud computing helps their users to use resources, access servers and store data based on demand. Cloud computing provides rapid dynamic and cheap computing power. Big data in cloud reduces the cost of operations since the companies do not have to buy their own product or service and to manage servers in same time by them self.

Big Data is somewhat dependent on the cloud for the flexibility that it provides. The processing of Big Data tools is then facilitated in an environment that can be adapted at will to optimize analytical operations. In fact, the union between Cloud Computing and Big Data is becoming a

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 27-34, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101503

good practice in the management of IT systems in many companies. The evolution of Big Data in cloud computing naturally raises the question of data protection and privacy respect. This is the biggest drawback of Big Data in cloud and the biggest challenge.

Big data and cloud computing, both favor companies and by cause of their benefit, the use of big data generates extremely significant positive cash flows in the cloud. With this serious increase, there are several emerging risk security concerns. Big data has more vulnerabilities with the comparison to classical database, as this database are stored in servers owned by the cloud provider. The various usage of data make protecting big data in the cloud intolerable with the traditional security measures [1].

In order to build a comprehensive risk assessment methodology for big data in cloud computing, a large literature review was conducted to identify all risk factors that may affect big data in cloud computing and major related research**.** Literatures on risk assessment for big data in the cloud are missing. Thus, additional effort must be employed in developing security risk assessment of big data in cloud computing.

The remainder of this paper is organized as follows: Section 2 provides a global overview of big data and cloud computing. Section 3 discusses and presents how big data and cloud computing work well together. Section 4 discusses the several risks to address in cloud computing and big data systems. Section 5 discuss risk assessment regarding big data in cloud computing and Section 7 presents the discussion, conclusions and future work.

2. FUNDAMENTAL CONCEPTS

Actually, the use of latest technology is essential for diverse IT operations and various industries such as big data, cloud computing, machine learning and IoT (Internet of things), for many of their applications for efficient management of the company, several concepts and definitions are discussed below [3].

2.1. Big Data

The concept of Big Data comes from a set of electronic operations from multiple data sources. This technology requires very important processing power and high capacities for data analysis and processing [7]. The importance of big data is focused in the analytical use of data, which helps to generate a clear decision to deliver better and faster services [8].

Big data is becoming a major innovation force in the field of research. This paradigm can be considered as a way to get and get appropriate information from large data set, providing information on huge data loads. As such, companies see this paradigm as a tool for understanding their customers, getting closer to them, finding patterns, and predicting trends. Additionally, scientists see big data as a way to store and process huge sets of scientific data. This concept is a hot topic for researchers and is expected to continue to gain popularity in the years to come.

The five different aspects used to describe big data are Volume, Variety, Velocity, Value and Veracity [4]:

- Volume is the amount of data coming from multiple sources, which show the huge data in numbers, this huge data can be measured in Gigabytes is now measured in Zettabytes or even Yottabytes. The volume is one of most evident dimension in big data characteristics.

- Variety is the data types, with the increase the number of Internet users in the world, data has changed from structured data in databases to unstructured data that contains a large number of formats such as images, audio and video clips, SMS, and GPS data [9].
- Velocity is the speed of data process and accessibility from several sources. The huge increase in data volume and their frequency requires the need for a system that ensures super-speed data analysis.
- Veracity is the quality of data; it shows the correctness of the data and the confidence in the data content. The quality of the data stored can change greatly, which influences the correctness of analysis. Although there is large agreement on the potential value of big data, the data is almost meritless if it is not perfect [10].
- Value is the value of big data, i.e. it indicates the importance of data after analysis. This is because the data on its own is almost valueless. The value lies in aware analysis of the precise data, the information and ideas it provides. The fifth characteristics of big data is the final stage that can be provided after processing volume, velocity, variety, contrast, validity and visualization [11]

2.2. Cloud Computing

Cloud computing is the realization of utility computing where the service provider implements resources and the cloud costumer will pay as they use the resources. The user can access the cloud via a thin client. Cloud also provides memory for a large number of data to store and allows computation. Hence, many users can rely on a cloud as it reduces the infrastructure charge that the user needs to invest [5], [6].

Cloud providers offer three several basic services: Infrastructure as a Service (IaaS); Platform as a Service (PaaS); and Software as a Service (SaaS):

- **Software as a service (SAAS):** Cloud service providers offer different software applications to users who can use them directly without installation of software application on their computer. The user can adjust the settings and customizing the service as appropriate to his needs. SAAS helps big-data clients to perform data.

- **Platform as a service (PAAS):** Cloud service providers provide platforms, tools and other services to users, where the cloud service provider manages everything else, including the operating system and middleware, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated.

- **Infrastructure as a service (IAAS):** Cloud service providers provide infrastructure such as storage, computing capacity, etc. is a form of cloud computing that provides virtualized computing resources over the Internet , In an IaaS model, a third-party provider hosts hardware, software, servers, storage and other infrastructure components on behalf of its users [12][13].

- **DaaS** : It is the alternative cloud computing model, as it differs from traditional models like (SAAS, IAAS, PAAS) in providing data to users through the network, as data is considered the value of this model [14] in conjunction with cloud computing based on solving some of the challenges in managing a huge amount of data. For these reasons, DaaS is closely related to big data whose technologies must be utilized [15]. DaaS provides highly efficient methods of data distribution and processing. DaaS is closely related to SaaS (storage as a service) and SaaS (software as a service) which can be combined with one of these models or both of them [16].
must appear as close to their point of reference as satisfactory formatting of the final document permits.

3. BIG DATA IN CLOUD COMPUTING

In [19], the usage of big data and cloud computing have been studied from several important aspects, and the authors have concluded that the relationship between them is complementary. Because of this big link between them, a model was prepared to show the relationship between them, and how much the both is compatible. Cloud computing can be considered as an environment of flexible distributed resources that uses high techniques to process and manage of data and yet reduces the cost. All these characteristics show that cloud computing has an integrated and consolidated relationship with big data. The twice are moving towards rapid progress to keep pace with progress in technology requirements and users. Big Data is somewhat dependent on the cloud flexibility. The processing of Big Data tools is then facilitated in an environment that can be adapted at will to optimize analytical operations.

Cloud computing and big data each go with the other, the first reason is that big data could support a huge data storage capacity in the cloud system, the second reason is that cloud computing uses different computing resources and storage for data analysis and processing. Thus, with a Big Data application having calculation capacities, Big Data progresses and accelerates the development of cloud computing. Distributed storage technology in environmental computing helps manage big data [17]. In [18], the authors claimed that the use of big data in the cloud makes data easily exposed. Cloud, which consists of many servers connected to each other.

Cloud and Big Data complement each other; cloud-based systems delivers high bandwidth, immense amounts of memory, and scalable processing power to help Big Data applications with enhanced real-time processing, storing and analyzing big volumes of data. Because of this corresponding complementary of these two technologies, we need to couple each one to others for our better technology's future. Thus, big data and cloud computing are two compatible concepts as cloud enables big data to be available, scalable and fault tolerant.

4. RISK ASSESSMENT REGARDING CLOUD COMPUTING AND BIG DATA

In the literatures, there are several research paper, and the authors discuss risk assessment from various perspectives. In [20], a risk evaluation model is proposed to solve the dynamic and fuzzy of security evaluation and to solve the problem of expert evaluation. In [21], a security risk assessment algorithm is claimed to predict security risk level. In [22], risk assessment metrics used to assess risk and continuous assessment techniques to ensure cloud service customers' trust on the security and privacy assurances of cloud service providers. In [23], risk assessment is discussed to processes obtained with the deployment of different security controls to provide automatic assessment of costs and risk factors. In [24], a systematic analysis of threats and vulnerabilities are introduced in risk assessment to provide a better security. In [25], a classification is used to segregate information based on the importance and level of protection required to protect privacy. Finally, in [26] the trust and control are discussed to reduce risk of cloud adoption.

While big data is a Swiss army knife that solves many of the current issues around high data volumes, it is an ever-evolving field that is still developing and still has some problems. In this section, we present some risks associated with Big Data.

Cloud computing has more than twenty biggest risks for digital enterprises.

Table I. Cloud Risk's Classification [27]

Cloud risk's	***Description***
Lock-in	Relying strongly on the services of one provider can lead to severe difficulties in changing the provider.
Loss of governance	When using Cloud services, the CC necessarily cedes control to the CP on a number of issues which may affect security
Supply chain failure	A CP can outsource parts of its production chain to third parties, or even use other CPs as part of its service. This way, a potential for cascading failures is created
Conflicts between customer hardening procedures and cloud environment	Certain security measures of a CC may conflict with a CP's environment, making their implementation by the CC impossible.
Social engineering attacks	Social engineering is understood to mean the art of manipulating people into performing actions or divulging confidential information.
Resource exhaustion (under or over provisioning)	As Cloud services are on-demand services, there is the possibility that the CP won't be able to meet an increased demand in a certain shared resource, or to maintain a given service level.
Isolation failure	In shared environments, errors or attacks can lead to situations where one tenant has access to another tenant's resources or data.
Cloud provider malicious insider - abuse of high privilege roles	Malicious insiders at the CP can cause various kinds of damage to a CC's assets.
Management interface compromise (manipulation, availability of infrastructure)	The customer management interfaces of public cloud providers are Internet accessible and mediate access to larger sets of resources
Intercepting data in transit	Whenever data is transferred between different computers or sites, there is the possibility that the transfer can be intercepted
Insecure or ineffective deletion of data	Deleting data from Cloud storage does not in fact mean that the data is removed from the storage or eventual backup media.
Distributed denial of service (DDoS)	Distributed Denial of Service attacks aim at overloading a resource flooding it with requests from many sources distributed across a wide geographical or topological area
Economic denial of service (EDoS)	As a consequence of attacks, poor budget planning, or misconfigurations, the cost of a Cloud service can strain the financial resources of a CC to an extent that the service is no longer affordable.
Compromise of Service Engine	The service engine is a fundamental part of a Cloud service. A compromise of the service engine will give an attacker access to the data of all customers
Loss of Cryptographic Keys	The loss or compromise of cryptographic keys used for encryption, authentication or digital signatures can lead to data loss, denial of services, or financial damages
Non Cloud-Specific Network-Related Technical Failures or Attacks	Cloud services can be affected by a number of network-related technical failures that can also occur on classic IT settings.
Loss of Backups	The backups a CP makes of it's customers' data can get lost, damaged, or the physical media on which the backup is stored can get stolen.
Natural disasters	Natural disasters like flooding, earthquakes, tsunamis can affect the infrastructure of a CP.
Subpoena and e-discovery	Law enforcement authorities may ask operators of IT infrastructures to provide information pertaining to criminal cases, or information may

Cloud risk's	*Description*
	have to be provided during civil lawsuits.
Risk from changes of jurisdiction	When data is stored or processed in a data centre located in a country other than the CC's.
Data protection risks	Processing data in another country may incur difficulties regarding data protection legislation
Licensing Issues	Violating a software supplier's licensing agreements can result in significant financial penalties or disruptions of service.
Intellectual Property Issues	Both in the Cloud and when using certain software and service environments within the own infrastructure.

There are the five biggest risks that big data presents for digital enterprises.

Table II. Big Data Risk's Classification[28]

Big data risk's	*Description*
Unorganized data	Big data is highly versatile. It comes from number of sources and in number of forms.
Data storage and retention	This is one of the most obvious risks associated with big data.
Cost management	The process of storing, archiving, analyzing, reporting and managing big data involves costs.
Incompetent analytics	Without proper analytics, big data is just a pile of trash lying unnecessarily in your organization.
Data privacy	With big data, comes the biggest risk of data privacy.

Literatures on risk assessment for big data in the cloud computing are lacking. Thus, we need to put more efforts on it so that we can adopt and use the big data even though they are hosting in the cloud environment.

5. SYNTHESIS AND DISCUSSION

In the literature mentioned below, there are no research paper that presents a clear and detailed solution or model in risk assessment for big data in cloud computing.

Risk assessment for big data in cloud computing is one of a primary role in safety-critical business, in order to implement the control measures necessary to ensure an acceptable level of safety industries. However, it faces a series of general challenges, in part related to technology development and increasing needs. The corresponding models, theories, technologies and methods of traditional risk assessment have been hard to deal with the huge amount of data and information generated by the advancement of modern technologies. New related technologies need further research and improvement.

There is currently a need for a dynamic risk assessment to identify and analyze all potential events that may negatively affect the business environment. Therefore, dynamic risk assessment relies on experience, training and continuing education and the definition of techniques to process relevant data, which must be used with adaptable capacity to deal with unforeseen events and provide the right support to enable risk assessment. Through this work, we propose a risk assessment model for Big Data in cloud computing based on machine learning, in particular a deep learning model.

The proposed model is a machine learning software that can assess huge amounts of data using the most appropriate methods in mathematics for company objective. This solution assesses all the probable risk factors as well as their features and analyse each of them depending on the probability of occurrence so that companies can righty act to prevent any possible damage.

This model will help companies to deliver appropriate support for safety-related decision-making and propose the adapted security measures in an automatic and reactive way.

6. CONCLUSIONS

With the increase use of data on a daily base, big data systems have become one of major force of innovation that provides a way in managing information. Cloud environments strongly control big data solutions by providing adapted environments to big data systems.

Although big data in cloud computing is powerful systems that enable both enterprises and further research to develop and enhance, there are some concerns relating risk assessment that need further and real investigation and discussion. Additional effort must be employed in developing risk assessment security mechanisms for big data in cloud computing. Further research must be employed in near time to tackle this risk assessment security problem. Regarding this particular area, we are planning to use adaptable mechanisms in order to develop a solution for implementing a risk assessment model at several dimensions of big data systems running on cloud environments. In this current paper, we provided an overview of big data, cloud computing, big data in cloud environments, highlighting its complementary and its solid relationship and showing that both technologies work very well together but also presenting the challenges faced by the two technologies mainly in security risk assessment.

In the future work , we will present the risk assessment model for big data in cloud computing, the proposed model will based on deep learning to predict the different risks relating each environment and propose the adapted security measures in an automatic and reactive way.

REFERENCES

[1] M. Paryasto, A. Alamsyah, and B. Rahardjo, “Big-data security management issues,” 2014 2nd Int. Conf. Inf. Commun. Technol., pp. 59–63, 2014.

[2] Hashem, I.A.T. et al., 2014. The rise of “big data” on cloud computing: Review and open research issues. Information Systems, 47, pp.98–115.

[3] S. Drissi, S. El hasnaoui, H. Iguer, S. Benhadou and H. Medromi,, Integration of Cloud Computing Big Data AI and Internet of Things: Review and open research issues, DIGITECH2019 , At: Germany, 25/ 26 April 2019.

[4] Sakr, S. & Gaber, M.M., 2014. Large Scale and big data: Processing and Management Auerbach, ed.,

[5] S. Drissi, S. Benhadou and H. Medromi,"A new shared and comprehensive tool of cloud computing security assessment", In the Proceedings of the Springer, vol: 366 , Advances in Ubiquitous Networking, UNET 2015, Casablanca, Morocco, pp 155-167

[6] S. Drissi, S Elhasnaoui, H Iguer, S. Benhadou and H. Medromi," Security Risk Assessment of Multi-cloud System Adoption: Review and Open Research Issues", In the Proceedings of the Springer, International Conference on Big Data and Smart Digital Casablanca, Morocco, January 2019 DOI: 10.1007/978-3-030-12048-1_37

[7] Boyd, D., & Crawford, K. (2011, September). Six provocations for big data. In A decade in internet time: Symposium on the dynamics of the internet and society (Vol. 21). Oxford: Oxford Internet Institute.

[8] SHAN, Y. C., Chao, L. V., ZHANG, Q. Y., & TIAN, X. Y. (2017). Research on Mechanism of Early Warning of Health Management Based on Cloud Computing and Big Data. In Proceedings of the 23rd International Conference on Industrial Engineering and Engineering Management 2016 (pp. 291-294). Atlantis Press, Paris.

[9] Parvin Ahmadi Doval Amiri and Mina Rahbari Gavgani, 2016. A Review on Relationship and Challenges of Cloud Computing And Big Data: Methods of Analysis and Data Transfer. Asian Journal of Information Technology, 15: 2516-2525
[10] Chen, Min, et al. Big data: related technologies, challenges and future prospects. Heidelberg: Springer, 2014.
[11] Demchenko, Yuri, et al. "Big security for big data: Addressing security challenges for the big data infrastructure." Workshop on Secure Data Management. Springer, Cham, 2013.
[12] Vacca, J. R. (Ed.). (2016). Cloud Computing Security: Foundations and Challenges. CRC Press. ch-15.
[13] https://support.rackspace.com/how-to/understandingthe-cloud-computing-stack-saas-paas-iaas/
[14] Terzo, O., Ruiu, P., Bucci, E., & Xhafa, F. (2013, July). Data as a service (DaaS) for sharing and processing of large data collections in the cloud. In Complex, Intelligent, and Software Intensive Systems (CISIS), 2013 Seventh International Conference on (pp. 475-480). IEEE.
[15] Motahari-Nezhad, H. R., Stephenson, B., & Singhal, S. (2009). Outsourcing business to cloud computing services: Opportunities and challenges. IEEE Internet Computing, 10(4), 1-17.
[16] Rajesh Saturi, Data as a Service (Daas) in Cloud Computing [Data-As-A-Service in the Age of Data] Data as a Service Daas in Cloud Computing, Global Journal of Computer Science and Technology Cloud & Distributed Volume 12 Issue 11 ,2012.
[17] MALLICK, Pradeep Kumar (ed.). Research Advances in the Integration of Big Data and Smart Computing. IGI Global, 2015.
[18] Hazirah Bee bt Yusof Ali, Lili Marziana bt Abdullah, Systematic Literature Review of Risk Assessment for Big Data in Cloud Computing Environment: Security, Privacy and Trust, AICCC '18, December 21–23, 2018, Tokyo, Japan.
[19] Nabeel Zanoon, Abdullah Al-Haj, Sufian M Khwaldeh, Cloud Computing and Big Data is there a Relation between the Two: A Study, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 17 (2017) pp. 6970-6982
[20] D. Zong-you, Z. Wen-long, S. Yan-an, and W. Hai-tao, “The application of cloud matter #x2014; Element in information security risk assessment,” 2017 3rd Int. Conf. Inf. Manag., pp. 218–222, 2017.
[21] S. Deng, D. Yue, X. Fu, and A. Zhou, “Security risk assessment of cyber physical power system based on rough set and gene expression programming,” IEEE/CAA J. Autom. Sin., vol. 2, no. 4, pp. 431–439,2015.
[22] R. Trapero, J. Luna, and N. Suri, “Quantifiably Trusting the Cloud: Putting Metrics to Work,” IEEE Secur. Priv., vol. 14, no. 3, pp. 73–77, 2016.
[23] V. Bellandi, S. Cimato, E. Damiani, G. Gianini, and A. Zilli, “Toward economic-aware risk assessment on the cloud,” IEEE Secur. Priv., vol. 13, no. 6, pp. 30–37, 2015.
[24] Lulu Liang, Wang Ren, Jing Song, Huaming Hu, Qiang He, and Shuo Fang, “The state of the art of risk assessment and management for information systems,” 2013 9th Int. Conf. Inf. Assur. Secur., pp. 66–71, 2013.
[25] V. Agrawal, “A Framework for the Information Classification in ISO 27005 Standard,” Proc. - 4th IEEE Int. Conf. Cyber Secur. Cloud Comput. CSCloud 2017 3rd IEEE Int. Conf. Scalable Smart Cloud, SSC 2017, pp. 264– 269, 2017.
[26] A. Khosravani, “A case study analysis of risk, trust and control in cloud computing,”Conf. (SAI), 2013, pp. 879–887, 2013.
[27] Catteddu, D., Hogben, G.: Cloud Computing Information Assurance Framework. ENISA (2009)
[28] M. Paryasto, A. Alamsyah, and B. Rahardjo, “Big-data security management issues,” 2014 2nd Int. Conf. Inf. Commun. Technol., pp. 59–63, 2014

DIGIPRESCRIPTION: AN INTELLIGENT SYSTEM TO ENABLE PAPERLESS PRESCRIPTION USING MOBILE COMPUTING AND NATURAL-LANGUAGE PROCESSING

Richard Zhang[1], Mary Zhao[1], Yucheng Jiang[2],
Sophadeth Rithya[2] and Yu Sun[2]

[1]Irvine High School. 4321 Walnut Ave Irvine, CA 92604
[2]California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Through our app, it is aimed to teach and tell the patients how to use the drug properly taking off the chances of putting their lives in danger, especially the elderly. It is also efficient to give patients these instructions as well as saving lots of paper. Because of the law, every drug that is given from the pharmacy to the user includes a receipt that lists information of, patient's information, drug information, insurance information, directions on taking the medicine (black box warning issued by FDA), medication details on how it works, side effects, storage rules, and etc. These pieces of information are crucial to patients, where it tells them how to use the drug properly, but most people would throw these receipts away, which is a risk as well as a waste. Through using this app, the patient can efficiently get information on how to properly use the drug. This application is also helpful, where the user can choose to set reminders on when to eat this drug each week or month.

KEYWORDS

Reminder System, HTML, Python, Firebase, Bootstrap

1. INTRODUCTION

Visiting a doctor's office and getting a prescription for drugs is not a rare occurrence in the modern world countries. Many patients will go through the process of visiting a doctor, get a prescription, go to a drugstore, and get some medicine to treat their disease. Prescription, abbreviated Rx, is an instruction written by a doctor authorizing the provision of medicines or treatments to patients. The 2016 National Ambulatory Medical Care Survey shows that the total number of physician office visits in the United States is 883.7 million. In the doctor's office, the doctor collects the patient's information such as heart rate, weight, blood pressure, and height before reaching a conclusion of how to treat the patient. Many patients need to employ prescription drugs to treat their disease. To prescribe drugs for the patient, the doctor will write a handwritten prescription paper or send the prescription electronically to the pharmacy. In the case of handwritten prescriptions, the patient will receive a prescription paper guiding them on what medicine to buy and how to consume medicine. This prescription contains information including:

- The patient's name the date of birth, and the address of the patient.
- Name of the drug, amount to be consumed every time, and frequency of taking the drug.

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 35-43, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101504

- Amount of the drug that the pharmacy gives the patient and the number of refills.
- The signature of the doctor and physician identifies like NPI and DEA number.

The patient needs to bring the handwritten prescription paper with him to a pharmacy or drug store in order to get the prescription drugs recommended by the doctor. At the pharmacy, the pharmacist communicates with the patient on what medicine the patient can have and how much the drug will cost. The patient will decide which drug to choose depends on the doctor's recommendation. Once the decision has been made, the patient picks up the prescription drug at the pharmacy.

Handwritten prescription paper creates inconvenience for the patients and wastes paper resources. The standard size of the handwritten prescription paper is 4 inches by 5.5 inches. Such a size of paper is inconvenient to the patients. It does not fit into the pocket, and putting the paper in a bag might damage the prescription. The paper prescription also has the risk to be lost. When the patients damage or lose their prescription, they have to run back to the doctor's office for the same prescription. This causes unnecessary trouble for patients. Another issue of handwritten prescription is paper usage. An average of 4 billion retail prescriptions is filled every year in the United States. This enormous amount of paper used is very costly. The cost is both financially and environmentally. The doctor has to buy more when he runs out of prescription paper. To the environment, producing paper means deforestation, or cutting down trees. 93 percent of paper produced comes from trees, and 42% of trees harvested is used to make paper. Caused by deforestation, the future agricultural productivity in tropical regions is threatened by the increase in average temperature and related extreme temperatures, as well as the decrease in average rainfall and rainfall frequency.

We use an online database to store the prescription information and a mobile app for the patients to access. Utilizing the technologies of HTML [1] coding, fire store database, and flutter coding, we created an application software addressing the issues on the inconvenience and paper waste of handwritten prescription paper. Our app is composed of an online website for the doctor to use, a cloud database to store the data, and a mobile app for the patients to use. The website functions to receive the prescription information that the doctor enters rather than handwriting on paper. A QR code containing that information will be generated from the website and shown on the screen. The patient then uses the mobile app installed on their phone to scan the QR code that contains their prescription information. This information will be stored in our online database. If the patients sign in with their account, they could see their past scanned prescription information history. To obtain the prescription drug, the patients just need to carry their phones with them to the pharmacy. Our method differs from the traditional handwritten prescription paper by not having the patients carry the prescription paper with them, but only a mobile app to be installed on their phone. This approach is more convenient since most people today already carry a mobile phone with them wherever they travel. The patients do not have to worry about losing the prescription as they can access the stored information with their account. Also, our app helps to reduce paper waste because our app eliminates the usage of handwritten prescription paper.
The rest of this paper is organized as follows: Section 2 details the challenges we encountered in the process of experimenting and designing samples. Section 3 focuses on the details of the solutions corresponding to the challenges mentioned in Section 2. Section 4 introduces the relevant details of our experiments, and then Section 5 introduces related work. Finally, Section 6 gives conclusions and points out the future work of the project.

2. CHALLENGES

There are a few of challenges existing in the project. They will be discussed in this section.

2.1. Challenge 1: New Drugs Update

The system needs to know how to update itself due to the new drugs that come out each year. Through the research online, it shows that twenty to twenty-five new drugs are made per year for the past two decades. When each new drug comes out, the app needs to know how to update itself, possibly through another website online. It will be difficult for an individual managing the app, to keep updating the site throughout the year. For example, if a new drug was invented and tested that would help cure the COVID-19, the app would be able to add a new regulation with this drug by itself, possibly through another website's newly added information in its database, and add this information onto our app. When this new information is added to our app's database, the patient would be able to scan the QR Code, given from the pharmacy, and get the right information on how to use this drug correctly.

2.2. Challenge 2: Create QR Codes by Pharmacy

The Pharmacy needs to have our app's program in order to create QR codes. In order for the pharmacy to create a QR Code for the patient to scan, they would have to input our app's program into their computer. Our program has to also be compatible with the program the pharmacy is using in order for them to input their information of the drug, creating the QR Code, and sending it for the patient to use. A pharmacy always has its own system to work with a patient, giving them the right drugs. Our app has to be compatible with the pharmacy's software, be able to create a recognizable barcode and be able to print onto the label. This is a compatible challenge, where there are several pharmacies' software's which are used by different pharmacies. So, the real question is, how can we make our app compatible with all pharmacies' software?

2.3. Challenge 3: Check OTC Drugs

The app should be able to check OTC drugs. The OTC stands for "over the counter" which represents the people that buy drugs and medicine without the guidance of the doctor's prescriptions. If the patient is not buying a drug from the doctor's prescription, the app can tell the patient if the drug is safe or not to buy. When the patient goes to the store, he can scan the drug or input drug name, then the app would tell the patient if the drug is safe or not for the buyer to buy. The app would be able to tell the patient if the drug is compatible or not to the buyer's statistics of allergies, or problems. The app would also need to connect to the NDC system which stands for "the drug code of commercial". This is difficult to be completed due to the need to add more information into the database of the app. This may lead to us thinking of all the possibilities that could occur, depending on the patient's conditions.

3. SOLUTION

3.1. Overview of the Solution

Because of the law, each medicine purchased by the patient comes with a receipt with the patient's information, drug information, insurance information, directions on taking the medicine (black box warning issued by FDA), medication details on how it works, side effects, storage rules, and etc. Most people after receiving medicine from a pharmacy, would throw away the receipt, which could be a factor of putting their lives in danger, as well as wasting lots of paper. On the pharmacy's end, they can use the app to create a QR Code for the patient to scan, containing the long list of information that would be on the receipt. For the patients, they would use the app to scan the QR Code, giving them the list of all the medicine's information, as well as

a way to set reminders on when to take the medicine. There are several similar tools at present [12][13]. For this app, you would first need to sign up for an account. After you signed up, you have to sign in to your account. After you signed in, this page will include your profile, your drug prescription history, and the button to scan QR Codes. Each prescription listed in that main page will have the drug number, the drug name, and the date the drug was added. Each drug page will have the information from the receipt, with the choice of reading the information out loud, as well as the choice of adding reminders on when to take this drug.

3.2. Implementation of the App

The web end of our app is designed for the use of doctors and pharmacists. The prescriber is using our website can deliver their medical instructions to patients more conveniently. On our website, a form is provided for the prescriber to fill out with patient information. After entering the necessary information, the prescriber can click on the “submit” button to generate a QR code for the patient to scan. This QR code contains all the information entered by the doctor. Patients that scanned the QR code will be able to access the information anytime in our mobile app.

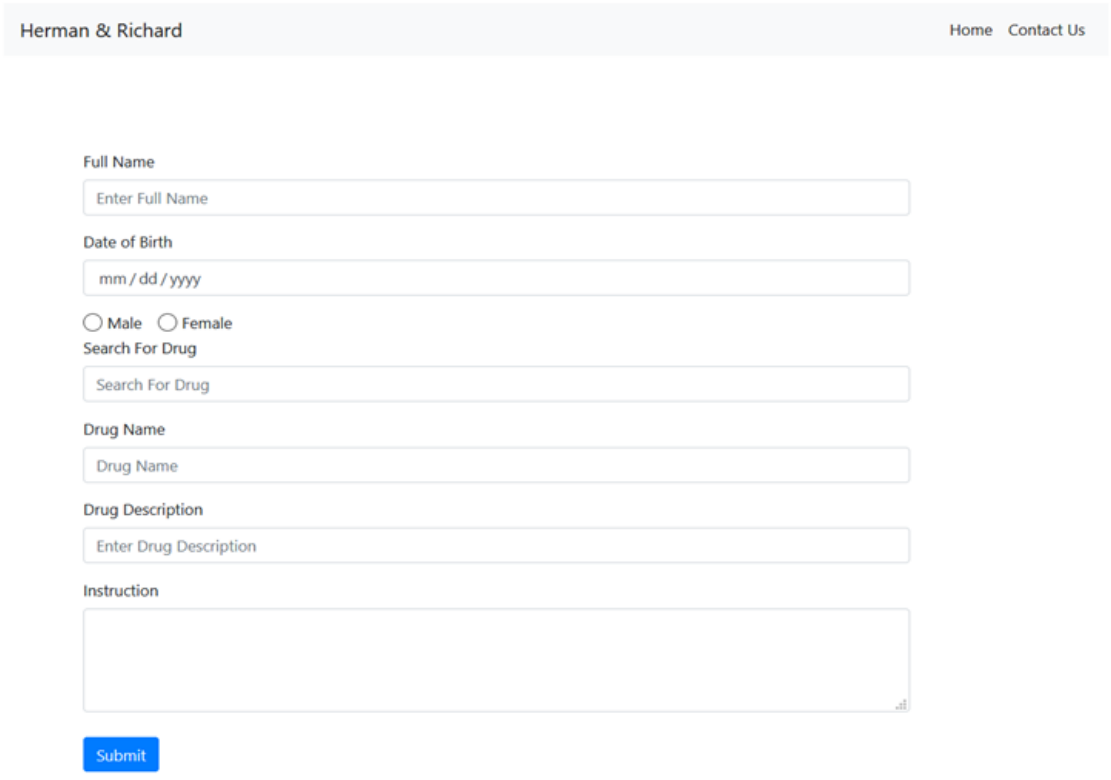


Figure 1: submission page for the app

We implemented this website using languages HTML and Python [2] [3]. At the top of our website is a navbar created from Bootstrap [7][8][9]. The main body of our website is a list of text fields for the prescriber to input patient information. We decided to incorporate the patient's full name, date of birth, gender, drug name, drug description, and doctor instruction in our form. Each input is labeled to guide doctors filling in the patient information properly. We limited input type to improve information accuracy. For example, the input type of date of birth is limited to date.

```
<div style="display:flex; just">
    <form id="dataForm">
      <div class="form-group">
        <label for="fullName">Full Name</label>
        <input type="text" class="form-control"
        id="fullName"  placeholder="Enter Full Name">
      </div>
      <div class="form-group">
        <label for="dob">Date of Birth</label>
        <input type="date" class="form-control"
        id="dob" placeholder="Enter DOB">
      </div>
```

Figure 2: HTML for the app

```
var data = {};
$("#dataForm").on("submit", function(event){
  data.name = $("#fullName").val();
  data.dateOfBirth = $("#dob").val();
  data.gender = $("input[name='gender']:checked").val
  ();
  data.searchForDrug = $("#searchForDrug").val();
  data.drugName = $("#drugName").val();
  data.drugDescription = $("#drugDescription").val();
  data.drugInstruction = $("#instruction").val();
  console.log(data.searchForDrug);
```

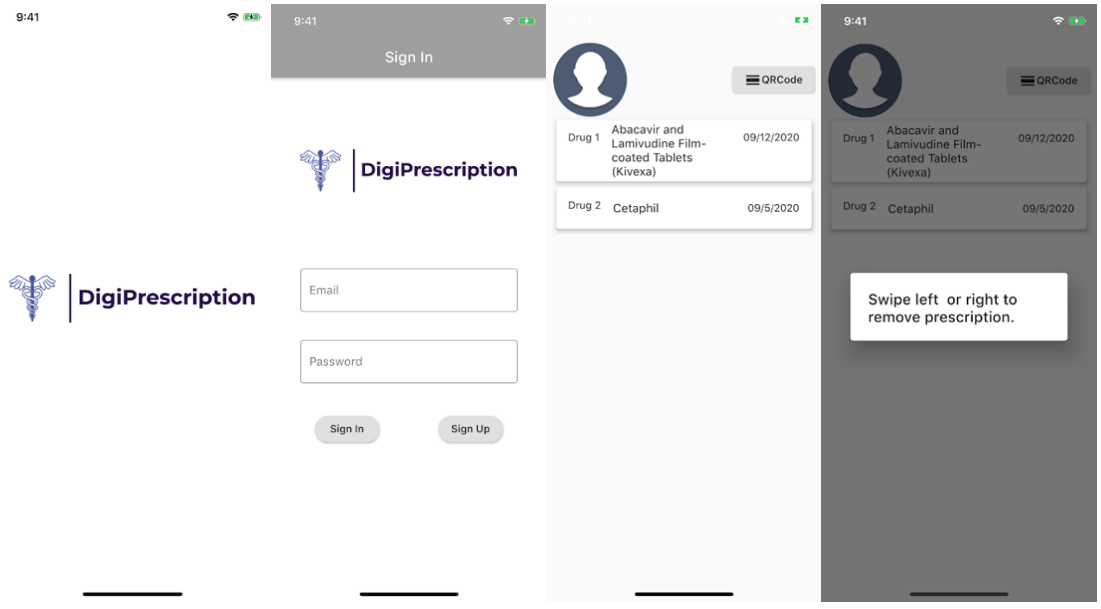
Figure 3: Data processing for the app

In the JS phase of our application, we created a variable database to contain all the information entered by the prescriber. The "submit" button in our website has an on-pressed function of collecting data. After the doctor filled in all the patient information and pressed the "submit" button, the string of data in each input will be collected, named, and stored in the database.

To improve the website's user friendliness, we added the auto complete function for the prescriber to search for a medicine. Suggestions of existing drugs pop out as the prescriber enters the drug name letters. This function operates by matching the entered drug name with existing drugs information stored in our second database.

Our drug information in the second database is obtained by the method web scraping. Beautiful Soup and html parser were used during our web scraping. Using Python, we targeted specific tag "li" of data from our source website. All the pieces of data including the targeted tag on the source website will be detected and printed out to our second database. After web scraping is finished, our second database contains names and information links of many existing drugs.

3.3. Functionalities of the App

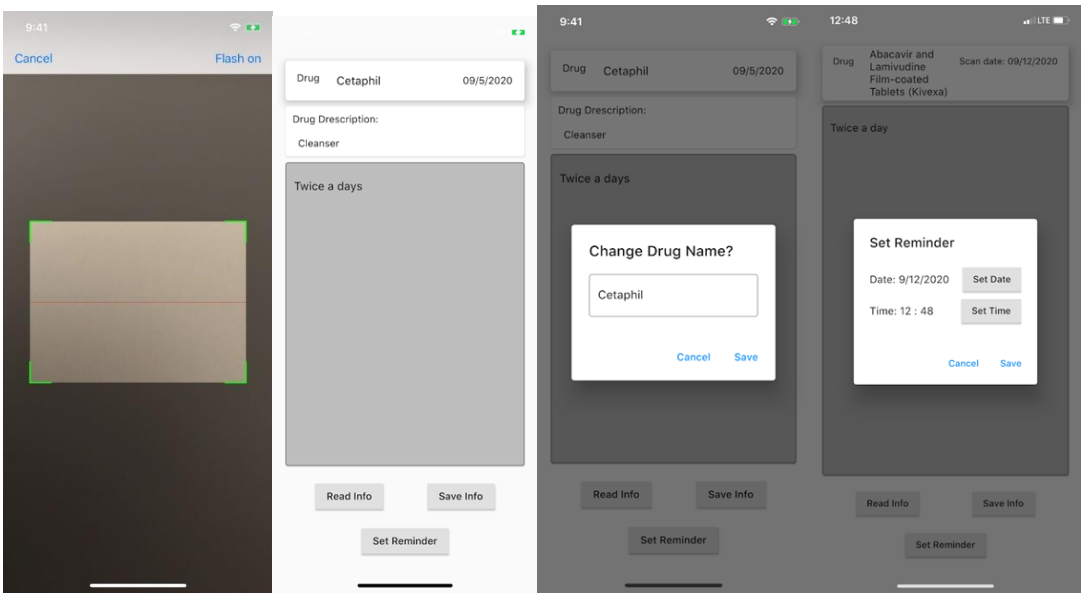


(a) (b) (c) (d)

Figure 4: screenshot of the app

Figure 4 (a) is the title page, which includes the logo I created, and the sign-in and sign-up page Figure 4 (b). If you haven't created an account, then you would press the Sign-Up page and then you would have to re-sign in. I linked my app to firebase [4][5][6] so that when a user creates an account, the information of the user would be stored there.

After you created an account for the app, this page would show up, displaying your profile with all your previous prescriptions. It would also have an option to scan QR Codes to add more prescriptions to the list. For each drug prescription displayed Figure 4 (c), there would be the drug number, the drug name, and the date when this drug prescription was added to your account. When a drug prescription is no longer needed or when you are done using the drug, you can swipe right or left to delete the prescription Figure 4 (d).



(a) (b) (c) (d)

Figure 5: screenshot of the app

With this feature, you can scan the barcode provided on your drug bottle. After you scan the QR Code, the Prescription Details page will show up, giving you all the information from the receipt.

3.4. Prescription Details

After you have scanned the QR Code on the drug bottle, you will get this page with all the information from the receipt. This will include the patient's information, drug information, insurance information, directions on taking the medicine (black box warning issued by FDA), medication details on how it works, side effects, storage rules, and etc. This page will also have the ability to read the information out loud as well as setting reminders of when to take the medicine.

In the Prescription Details Page, other than having the information from the receipt, there is also the choice of changing the name of the drug prescription as well as setting the date of reminders. For setting the dates of taking the medicine, you can choose the time each day and the days per month.

4. EXPERIMENTS

To evaluate the effectiveness of our app in helping patients, we collected user evaluation and feedback from 30 patients who used the app.

We collected evaluations from the participants by asking them to list out the things that they liked and/or disliked about the app. Some things that the patients liked about the app were features such as the reminder and prescription details. Some participants reported that they can retrieve information such as drug information, insurance information, directions on taking the medicine, side effects, storage rules anytime they want.

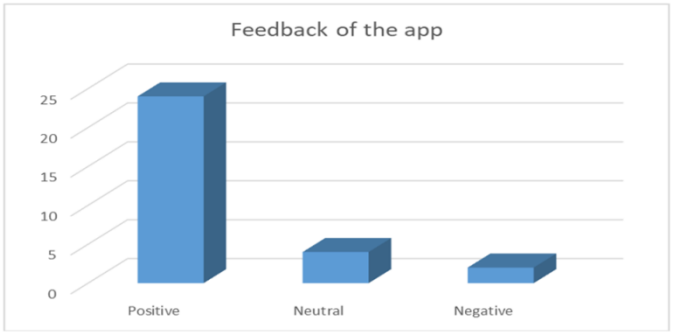


Figure 6: feedback collected from patients

Lastly, we asked the patients to give any feedback or suggestions for the app in terms of improvements of the features in the future. We believe that all of these suggestions are extremely valuable to the future improvements of the app.

5. RELATED WORK

A medication reminder mobile app: does it work for different age ranges? Mina Fallah, Mobin Yasini designed an Android based software that reminds patients with chronic disease to consume their routine medication. The app was developed and evaluated based on its user-friendliness, efficacy, and usability. An inquiry of 2 efficiency questions and 10 usability questions was created to assess the app. Through this app, medication consumers, app users, and patient relatives are joined together. Patient's adherence to the medication improves due to the connection between doctor and patients. Our app differs from this app in the employment of QR code. Doctors enter medication information on our web page to generate a QR code for the patients to scan with their phone. This process of making a reminder is more convenient.

Medication Reminder & Healthcare is an android application, developed by Deepti Ameta, Kalpana Mudaliar and Palak Patel [10], which provides various functions for user patients. Their app allows interaction between doctors and patients. Patients can search for a doctor and make appointment disease wise or based on their location. Patients are also able to search for disease information through their app. Our app varies from this app with our QR code scan and online database. Patients' scanned information will be stored online, and they are able to access it by logging in with their account.

Slagle, J.M., Gordon, J.S., Harris, C.E., Davison, C.L., Culpepper, D.K., Scott P. and Johnson, K.B., (2011) "MyMediHealth – Designing a next generation system for child-centered medication management" [11]. This medication reminding app was developed for children, allowing parents to create an intuitive 24 hours day planner, which meals and medications can be

instructed in each time interval. This app contains advanced features such as auto-population function (allows medication to be taken multiple times in a day) and medication-specific business logic (safety limit on the amount of medication taken in a period of time). Our app differs from this app with the applicant of QR code and ability to read drug description. After entering the drug description, users can tap on the “read info” button to have the app read description of the drug.

6. CONCLUSION AND FUTURE WORK

Through creating this application, lots of good effects have come out of this. We have saved a lot of paper but also making it easier for patients to know the instructions on how to take the drug properly without putting their lives in danger. The patient can also have the choice to set reminders on when they are going to take the medicines per week and per month, which is also very efficient.

A limitation within this app is translation. Even though the app can read the information from the receipt out loud for minorities, it is a problem for people that don’t know English. Another limitation is that for people that are elders, the app might be hard for them to use the app due to technological advancements. It is going to be hard for them to know how to use this app. The last limitation is that the application should know when the drug is used up due to the time when the patient added the drug to the list. This function will significantly improve compliance which will have a good outcome on medication therapy and clinical results.

For translating, and adding a solution for compliance problems in pharmacy, we can add more features to this application for this to happen. For the elders that don’t know how to use this app, we will try to do experiments and surveys seeing how to make this app easier to use through the data we receive.

REFERENCES

[1] Lemay, Laura, and Arman Revised By-Danesh. Teach yourself web publishing with HTML 4 in a week. Sams, 1997.

[2] Grinberg, Miguel. Flask web development: developing web applications with python. " O'Reilly Media, Inc.", 2018.

[3] Dolgert, A., L. Gibbons, and V. Kuznetsov. "Rapid web development using AJAX and Python." In Journal of Physics: Conference Series, vol. 119, no. 4, p. 042011. IOP Publishing, 2008.

[4] Moroney, Laurence, Moroney, and Anglin. Definitive Guide to Firebase. Apress, 2017.

[5] Moroney, Laurence. "Firebase Cloud Messaging." In The Definitive Guide to Firebase, pp. 163-188. Apress, Berkeley, CA, 2017.

[6] Stonehem, Bill. Google Android Firebase: Learning the Basics. Vol. 1. First Rank Publishing, 2016.

[7] Spurlock, Jake. Bootstrap: Responsive Web Development. " O'Reilly Media, Inc.", 2013.

[8] Shenoy, Aravind, and Ulrich Sossou. Learning Bootstrap. Packt Publishing Ltd, 2014.

[9] Radford, Stephen. Learning Web Development with Bootstrap and AngularJS. Packt Publishing Ltd, 2015.

[10] Ameta, Deepti, Kalpana Mudaliar, and Palak Patel. "Medication reminder and healthcare-an android application." International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT) Vol 6 (2015).

[11] Slagle, Jason M., Jeffry S. Gordon, Christopher E. Harris, Coda L. Davison, DeMoyne K. Culpepper, Patti Scott, and Kevin B. Johnson. "MyMediHealth–Designing a next generation system for child-centered medication management." Journal of biomedical informatics 43, no. 5 (2010): S27-S31.

[12] Herrmann, James M., Gerald S. Indorf, and Sunway R. Wang. "Interactive medication reminder/dispenser device." U.S. Patent 5,805,051, issued September 8, 1998.

[13] Hawley, Tarwa L., and Jon H. Kirk Jr. "Medication reminder." U.S. Patent 6,325,534, issued December 4, 2001.

IMAGE WAFER INSPECTION BASED ON TEMPLATE MATCHING

Massimiliano Barone

STMicroelectronics, Agrate Brianza, Milano, Italy

ABSTRACT

This paper presents a template matching technique for detecting defects in VLSI wafer images. This method is based on traditional techniques of image analysis and image registration, but it combines the prior art of image wafer inspection in a new way, using prior knowledge like the design layout of VLSI wafer manufacturing process. This technique requires a golden template of the patterned wafer image under inspection which is obtained from the wafer image itself mixed to the layout design schemes. First a mapping between physical space and pixel space is needed. Then a template matching is applied for a more accurate alignment between wafer device and template. Finally, a segmented comparison is used for finding out possible defects. Results of the proposed method are presented in terms of visual quality of defect detection, any misalignment at topology level and number of correctly detected defective devices.

KEYWORDS

Wafer inspection, template matching, image registration, pattern recognition, VLSI wafer images, Golden template, segmented comparison, space mapping

1. INTRODUCTION

Defects inspection plays a very important role in the very large integrated circuits (VLSI) manufacturing, because it can ensure the correctness during design and production, as well as the reliability of the microchip. The automatic inspection systems, despite of the traditional manual inspection, is mandatory for VLSI manufacturing processes due to high device number per wafer, up to 100.000 devices. Besides, a manual inspection is typically less accurate. The automatic inspection system usually consists on image acquisition and image processing. Image processing and analysis are the key to the automatic inspection system. In this paper, the surface image of VLSI wafer acquisition is made by acoustic scansion microscope, called SAM. A mapping between physical space and pixel space based on contour extraction and radon space is needed before to apply registration. The image registration algorithm based on template matching was used to achieve the alignment of the template image and the detected image in the pixel spatial position. Segmentation processing was applied to obtain the specific defects image. This method was tested by using some real image wafer on a PC.

1.1. Background

Wafer inspection systems are composed by two main stages. The first one is responsible for image registration, while the second one is for the defect detection inside the non-repetitive area found after the registration phase. The first stage can be solved in several different manners. It is possible to distinguish two different approaches: design-rule checking or image-to-image reference [10]. A design-rule system checks for the violation of a set of generic rules everywhere

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 45-56, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101505

on the wafer sub-images. A design-rule-based PDI prototype system has been developed by NanYang Technical University, Singapore [7] and [6]. Image-to-image-reference approach compares every pixel in the digital image under inspection with the corresponding pixel in the reference image, therefore a very accurate registration is required. Another strategy is optical spatial filtering [4] used in defect detection on masks and patterns. This method is very fast, but a major disadvantage is that small defects cannot be recognised. Wavelet technique [17] has also been used in wafer inspection. A complete review of the related literature may be found in Babian [2], Newman et al. [11], Moganti et al. [12], [15] and [16]. In addition, there are some techniques without prior knowledge like above mentioned methods. In fact, it is possible to extract the reference image directly from the wafer image. A self-reference technique was developed by Dom et al. [3], in which the comparison is made using the repeating cells in the image. This method was further developed by Khalaj et al. [8] by proposing a technique to extract the building block of repeating patterns from the acquired image. In particular, the ESPRIT algorithm [1], [5] and [9] is used in estimating the frequency components. The last stage can detect and classify defects inside the non-repetitive area with many different algorithms like described in [19-22]. In general, the detection is based on reference image or no referential approaches. Morphological processing [23, 24] and neural network [25] are without references, while methods [26, 27] need the standard image. Finally, image registration based on the SURF algorithm is another way to align the detected image and the standard image. The advantage of this algorithm is that image scaling, rotation and even affine transformation remain invariant [28].

1.2. The Structure of this Paper

In Section 2.1 the context where this solution was applied is described, with some examples of wafer images acquired and defects inside device. The full algorithm is presented in Section 2.2 including all stages of the pipeline. In Section 3, some results are shown. Finally, in Section 4, comments and conclusions about this work are given.

2. TEMPLATE MATCHING

As shown in [12] the template matching is one of possible ways for image wafer inspection. This technique is one of more robust in terms of alignment and minor different details, so it is the best for our options, see inspection algorithm taxonomy Fig. 1.

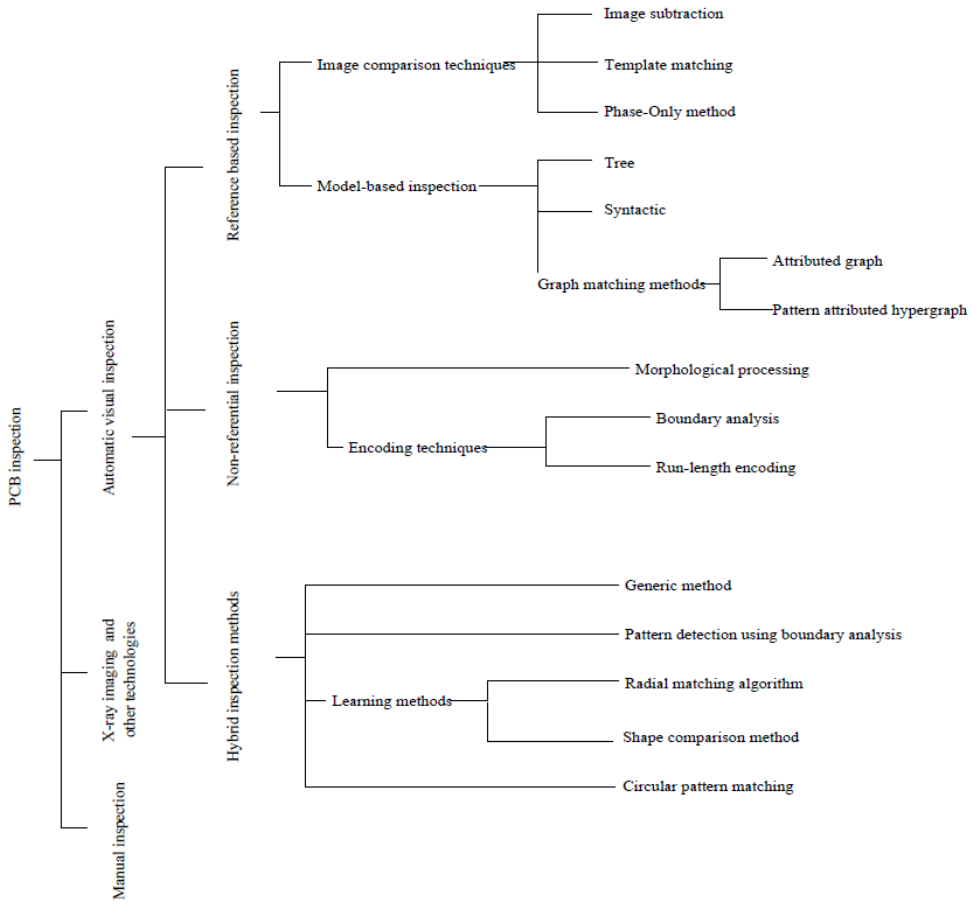


Figure 1. Wafer image inspection algorithm taxonomy.

Template Matching is just the core of this solution; the full method is composed by several stages that will be described in the sections below. The first step is defining the context where it is applied.

2.1. Context

Wafer Inspection in semiconductors field applications is applied in critical steps of the manufacturing processes and it is based on electrical test, mechanics test and image test. Image analysis is typically applied before dicing stage, see Fig. 2. Visual Inspection Technology is based on different physical principles; in particular, image processing is applicable to all following instruments: Infrared sensors, x-rays beam, electronic beam (SEM), laser beam, optical analysis and scansion acoustic microscope (SAM) beam. Machines Inspection of manufacturing processes provides images. The solution described in this paper was applied to SAM images. The devices tested were MEMS bonded; SAM inspection is used to check bonding interface integrity.

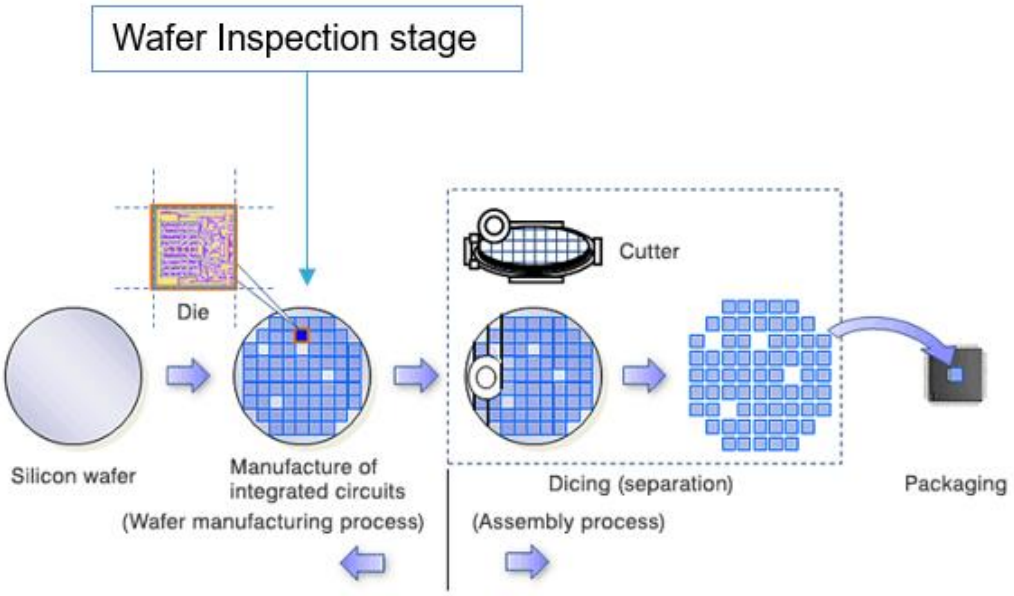


Figure 2. Wafer semiconductor process.

The goal of inspection is to detect the wrong area inside the image wafer. The defect could be large, and it could cover a lot of devices, see Fig. 3 or it could be inside each single device, see Fig. 4.

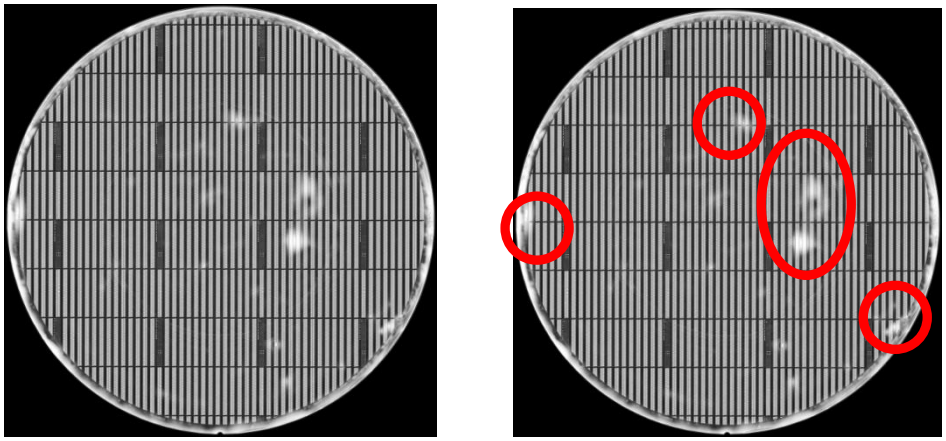


Figure 3. Image Wafer by SAM with large defects.

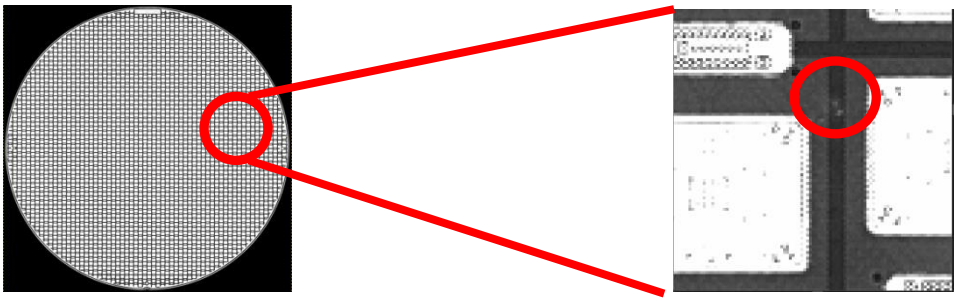


Figure 4. Image Wafer by SAM and defect device zoom.

As we can see, the device could have a different internal shape or alternate orientation and different size like the wafer sizes. Besides the brightness of the whole wafer or of each device could be different. It means to find out a method that is invariant about all possible changes, not only about the type of defect. This requirement needs a solution more complex than a simple template matching. For this reason, there is a pipeline of stages and each of them solves a specific problem.

2.2. Overview of Template Matching Method

The solution is composed by three main stages, see Fig. 5. The first stage is the topology one, it acquires the wafer image containing all devices, then it receives the list of centre device in micron space and the image of template device. Topology is responsible for mapping the list of centre device in micron into pixel space. The next stage is about the segmentation of internal device area. The last stage applies the specific defect test for each device.

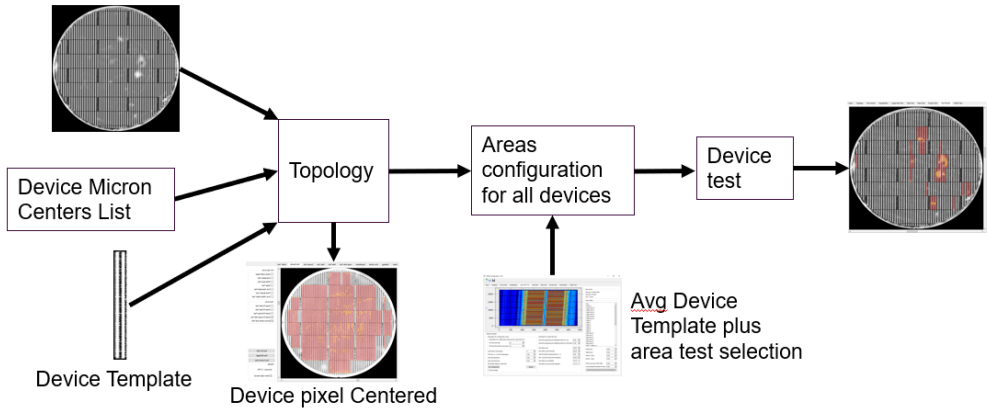


Figure 5. Template Matching Overview.

2.3. Topology

The matrix transform estimation to map centres from micron space to pixel space requires different steps, see Fig. 6. Centre and radius are estimated by Circle Fit (Taubin method) [29] based on pixel detection circumference to find the circle equation with minimum square error, so pixel circle centre and pixel radius are found. The size in micron of radius wafer is known so the scale factor micron-pixel is evaluated. The matrix transform matrix needs also the rotation estimation. Rotation angle is estimated by Radon transform (similar to Hough transform [30]) that uses vertical and horizontal pixel lines (lines detection based on Sobel filter) to count the lines with same slope rate, so the angle is found.

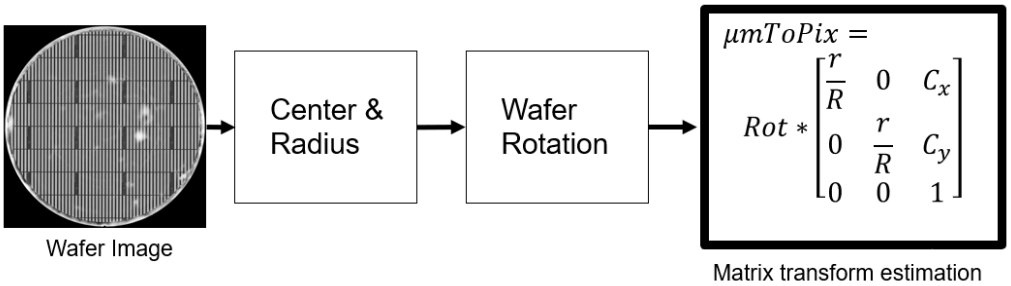


Figure 6. Matrix transform estimation.

The prior art strategy is different. In fact, it is based only on image wafer input and it extracts the frequency of repetitive pattern device to estimate centre device in pixel space. It does not use other information, but image wafer is the result of a design layout, so some information is known like wafer size in micron and centre device in micron, hence it is possible to re-use this information to simplify the problem.

After matrix estimation, centre device in pixel have been found, but they are not accurate. The image distortion, low resolution and previous estimations are not precise enough, so a refinement is needed. A template matching is applied starting from a sub-set of all device pixel centre using a window search large like the maximum pixel imprecision, see Fig. 7. To save computation due to thousands of devices inside wafer, just a sub-set is taken. There are a lot of possible cost functions used to calculate the similarity between template device and image device. A good cost function is the cross correlation because it does not depend on shape and absolute luminance values. In fact, the template device comes from the layout design and it is a binary image (black/white). It is not the extraction of a golden template from wafer image. In this way, more precise shape device is obtained, no defects and, thanks to scale factor previously estimated, the size in pixel space is gained. The subset pixel centre obtained is more accurate.

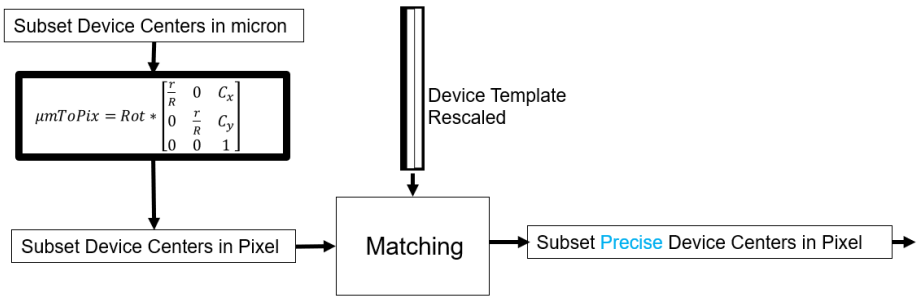


Figure 7. Template matching core.

The sub-set of new pixel centre device and sub-set micron centre device are selected, so a Matrix transform refinement based on Jacobi SVD, singular value decomposition method, is applied. The new matrix transform is more precise, but it is possible to increase precision re-using all device and a better template device for a second template matching, see Fig. 8. The second matching starts from pixel center more precise, so a smaller window search is enough. The new template comes from the average of all sub-set device, so the luminance is very similar to wafer luminance and the values are several levels of gray, no more a simple binary image. Before to calculate the average, all outliers are removed from it to have a gold template without defects. The image device with bad matching and distance between starting center and matched one are discarded. After the second matching, all pixel centers are very accurate.

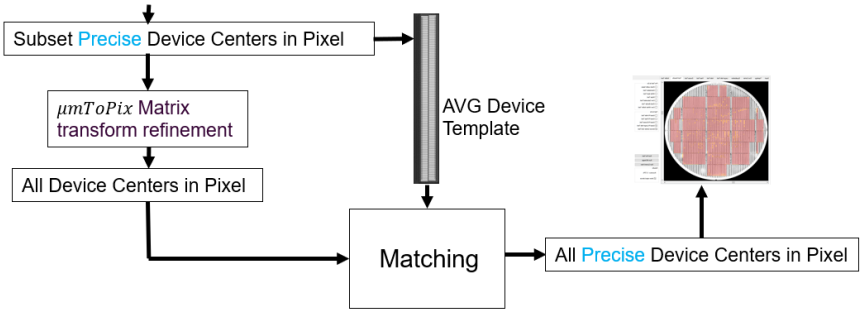


Figure 8. Second full template matching.

The image is acquired by SAM with grey levels, while template coming from design layout, so the image template is binary (black or white). A simulation of grey levels starting from a design layout would be better. A future development to estimate the grey level from design layout requires to know material property of an acoustic scansion and 3D surface shape, but it is not the focus of this paper.

2.4. Segmentation

The next stage is a manual segmentation to explore the device internally in order to find a specific defect. The whole area could be useless or not testable. The manual segmentation is applied to the average image template found. Fig. 9 is an example of manual segmentation. The tool used for inspection allows to define by user the area where to apply the test inside the device (red and blue image). The black area of the created mask isn't used to discover wrong defects and grey levels are proportional to the weights used for defects inside these positions.

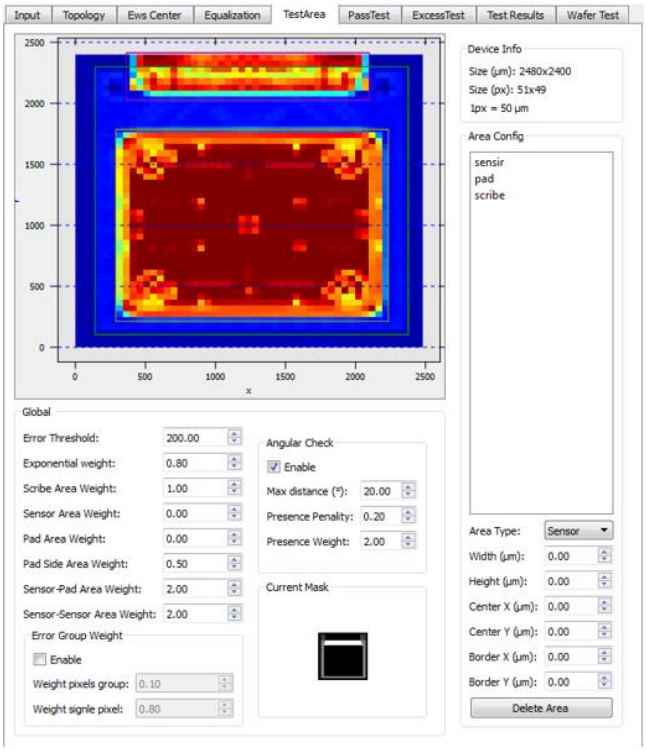


Figure 9. Manual segmentation: inside red circle (black area is ignored by device test).

The goal of above test is to find all blobs whose brightness is excessive in dark areas (blue colour in Fig. 9). For this reason, the almost white part (red colour) of the device is skipped from test.

2.5. Device Test

Before applying a specific test, it is necessary to equalize each device to use the same thresholds test for all devices, see Fig. 10. Equalization is based on a square root minimization of polynomial mapping of luminance.

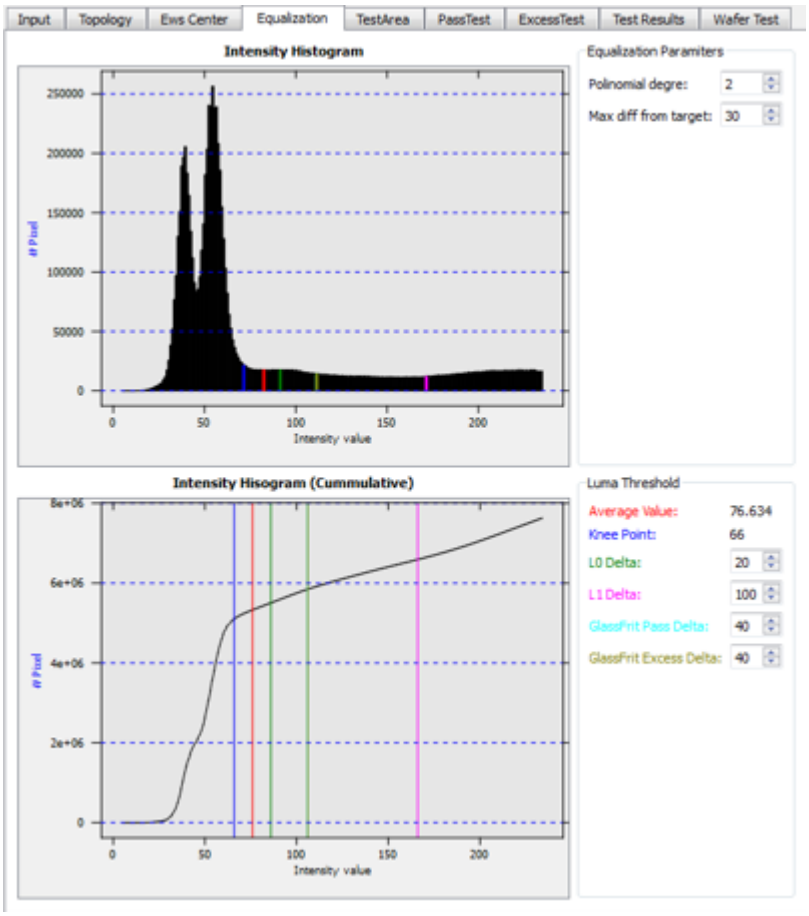


Figure 10. Histogram and cumulation of wafer luminance of all devices with knee point (blue).

Each device pixel colour is re-mapped to be like the Median template image obtained by the median of all devices well matched. The pixel colours of median template are put into a B array vector, while the associated colours of the current device, that are not too different from median one, are computed by power of N degree. So, a rectangular A matrix composed by all N power of all pixel colours of the current device associated is built. The solution consists on finding the common polyonomy of N degree that gives A*X=B. Where X contains all polynomial coefficients. The final solution to minimize the square root difference between re-mapped colours and median colours is X obtained by QR decomposition of A.

There are different types of defect, so for each type a specific test is implemented. In this paper, the focus is on searching too bright blobs inside dark area. The knee of cumulated luminance of all devices is taken as reference value, see cumulation Fig. 10. It is the boundary between dark and bright values. A device is wrong if the weighted sum of bad pixels is above the threshold defectivity level, to do that it is necessary to normalize all values into a range [knee+L0, knee+L1], see equation (1). If a single pixel value is more than knee+L0 then the pixel is classified as bad. After normalization, the weighted values are multiplied by the manual segmentation mask grey levels and finally they are summed.

$$f(x) = \begin{cases} 0 \; if \; x < KneePoint + L0Delta \\ \dfrac{x - KneePoint + L0Delta}{L1Delta - L0Delta} \\ 1 \; if \; x > KneePoint + L1Delta \end{cases} \quad if \; KneePoint + L0Delta < x < KneePoint + L1Delta \tag{1}$$

3. RESULTS

The results of this method are shown here. As we can see, the results are good as expected. In Fig. 11, the devices with wrong bright colors, inside the manual segmented device area, are detected and marked with a red mask. The tool analysis gives the total device accuracy or wafer yield. In this case was selected the specific test mentioned into above section about wrong bright areas. Fig. 12 and 13 show the details of test result of some wrong devices.

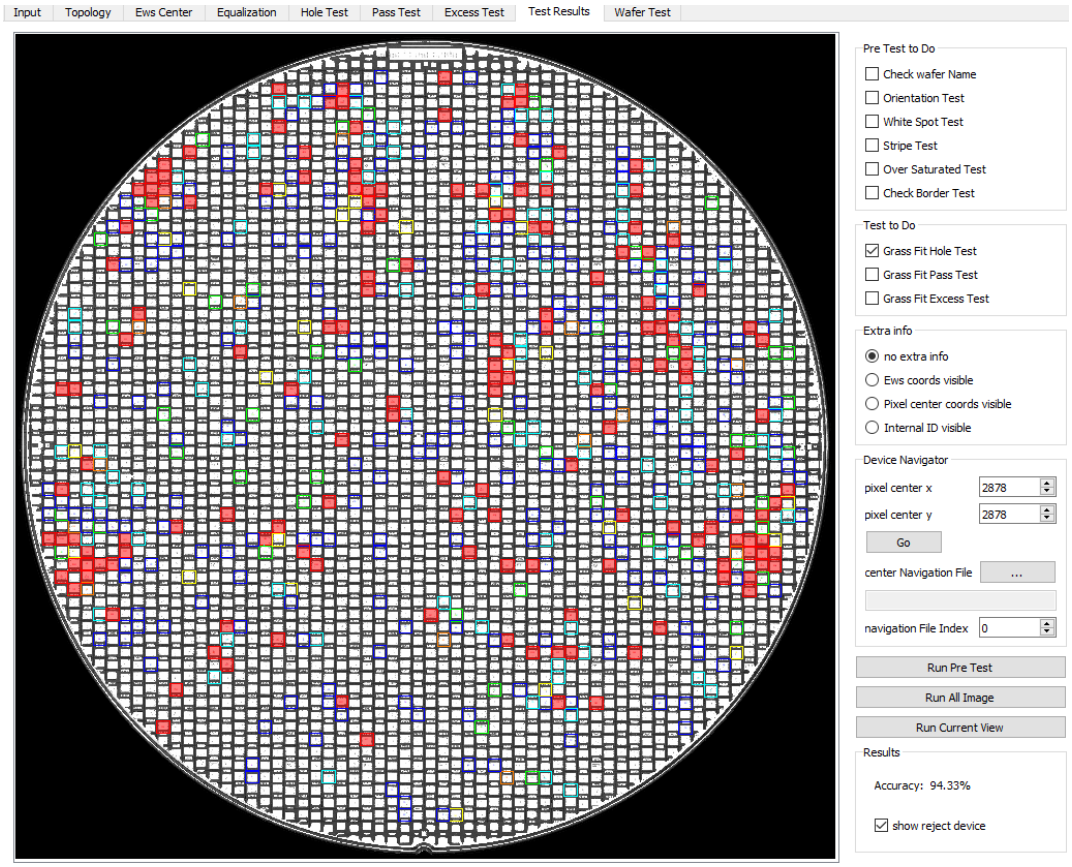


Figure 11. Test result of the whole wafer, red devices are wrong.

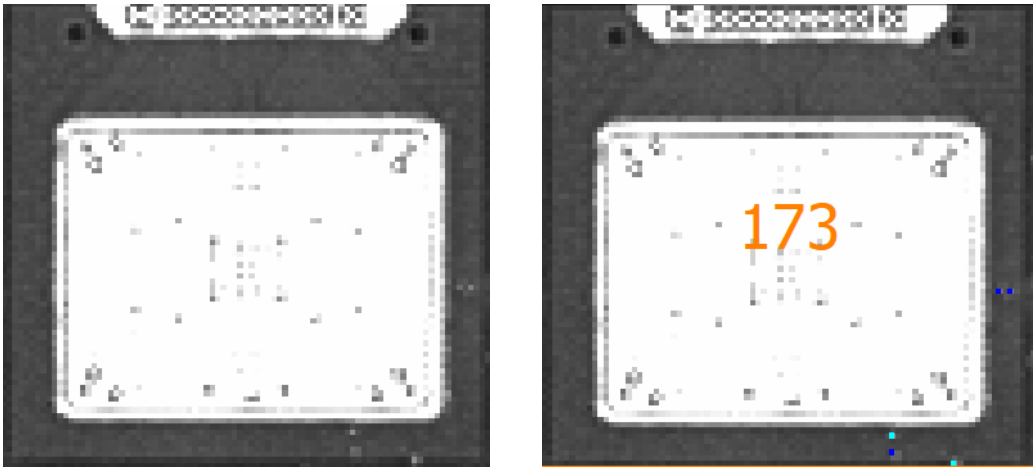


Figure 12. Left: device before test. Right: test result not enough bad (< 200) with defectivity level printed.

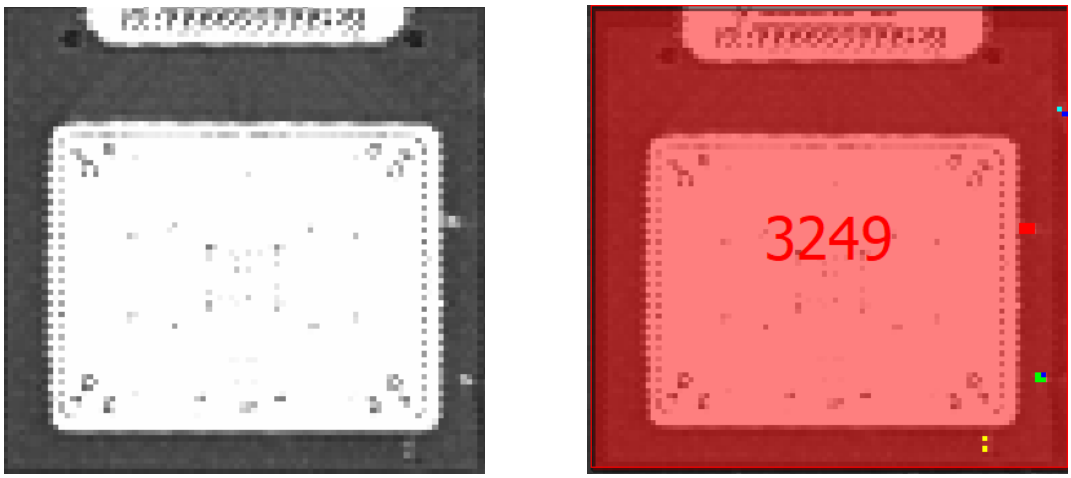


Figure 13. Left: device before test. Right: test result bad (> 200) with defectivity level printed.

In this test, the threshold of defectivity level is set to 200 and only devices with higher values are recognized as wrong. It is possible to see also single bad pixels with coloured pixels, even if the sum is under the threshold.

4. CONCLUSIONS

The worst case for wafer inspection is an image registration failure that moves all centres into a wrong position, so that also manual segmentation is misaligned. If this case happens, all devices may seem defective, but this is not the reality. Tested devices did not show this type of failure, but the shape of all verified device types is very similar, so it is not possible to demonstrate the method robustness for any kind of device shape. In addition, a thorough comparison with other image classification methods is not possible either, because a common data set of wafer images does not exist and because of confidentiality issues in the semiconductors industry. The effectiveness of the proposed method is not easy measurable, because a ground true for each tested wafer does not exist. The method is validated with human visual inspection. A small data set of tens wafers was extracted and all devices were fully manually inspected to have a ground true. Each wafer contains thousands of devices. Thousands of wafers were then analysed without a full device human inspection: only hundreds of randomly sampled devices were manually tested. The method was able to find out several small defects that human inspection could not see. Only 5% of devices were false positive and negative compared to ground true data set, but failures cases were closed to thresholds, so they can easily be classified as ambiguous cases, not real failures. State of art usually measures false positives and negatives in a binary way, but, in this type of defect analysis, the level of confidence about each single defect classification, defined as the distance from thresholds, could be a good indicator as well. If continuous values are not appropriate for failure representation, then a good quantization could be at least based on three states: false/true positive, false/true negative and ambiguous cases, where a specific metric, like the absolute value of distance from thresholds normalised by the threshold as an example, defines the range of ambiguous cases.

ACKNOWLEDGEMENTS

The topology was fully developed and conceived by Pierluigi Taddei, PhD of Polytechnic of Milan.

REFERENCES

[1] Paulraj A., Roy R., Kailath T. (1985) Estimation of signal parameters by rotational invariance techniques (ESPRIT). Proc. of 19th Asilomar Conference on Circuits, Systems and Comp.

[2] Babian F. (1986) Optical defect detection limits in semiconductor wafers and msks. PhD thesis, Stanford University, Stanford, Calif.

[3] Dom B.E., Brecher V.H., Bonner R., Batchelder J.S., Jaffe R.S. (1988) The P300: A system for automatic patterned wafer inspection, Machine Vision and Applications

[4] Chin R. T. (1988) "Survey Automated Visual Inspection: 1981 to 1987". Computer Vision, Graphics and Image Processing, 41: 346-381

[5] Roy R., Kailath T. (1989) ESPRIT: Estimation of signal parameters via rotational Invariance Techniques. IEEE Trans. On ASSP, 37(7):984-995

[6] Mital D.P., Khwang T.E. (1991) Microcomputer based low cost vision system for wafer inspection. Intell. Robotics Proceedings of the International Symposium Bangalore, India SPIE 1571:200-214

[7] Meisburger W, Brodie A, Desai A (1992) Low-voltage electronic-optical system for the high-speed inspection of integrated circuits. J. Vacuum Sci. Technol. B

[8] Khalaj B.H., Aghajan H.K., Kailath T. (1993) Digital image processing techniques for patterned wafer inspection. SPIE 1926

[9] Khalaj B. H., Aghajan H. K., Kailath T. (1994) Patterned wafer inspection by high resolution spectral estimation techniques. Machine vision and applications, 7:178-185

[10] Dom B.E., Brecher V. (1995) Recent advances in the automatic inspection of integrated circuits for pattern defects. Machine Vision and Applications 8:5-19

[11] Newman T.S., Jain A.K. (1995) A survey of automated visual inspection, Computer Vision, Graphics, Image Processing, 61(2): 231-262

[12] Moganti M., Ercal F., Dagli C.H., Tsunekawa S. (1996) Automatic PCB inspection algorithms: a survey, Computer Vision Image Understanding, 63(2): 287-313

[13] Chou P.B., Rao A.R.., Sturzenbecker M.C., Wu F.Y., Brecher V.H. (1997) Automatic defect classification for semiconductor manufacturing. Machine Vision and Applications, 9:201-214.

[14] Porat B. (1997) "A Course in Digital Signal Processing". John Wiley & Sons, Inc., Chapter 4.4, pp.104-107

[15] Moganti M. Ercal F. (1998a) A subpattern level inspection system for printed circuit boards, Computer Vision Image Understanding, 70(1): 51-62

[16] Moganti M. Ercal F. (1998b) Segmentation of printed circuit board images into basic patterns, Computer Vision Image Understanding, 70(1): 74-86

[17] Chen C. H., Cheng T. H., Wu W. T., Driscoll S. (1998) "Machine Vision Algorithms for Semiconductor Wafer Inspection: A Project with Inspex", Proceedings of SPIE, 3521:221-228

[18] Guan S.U., Xie P. (1999) "A golden block self-generating scheme for continuous patterned wafer inspections", Proceedings of the 10th International Conference on Image Analysis and Processing, Sep. 1999, pp. 436-441

[19] Serra, Jean. "Image Analysis and Mathematical Morphology",Image analysis and mathematical morphology , Academic Press,pp.536,1982.

[20] William K. Pratt. "Image detection and registration", Digital Image Processing, pp. 551–566, Wiley–Interscience, New York, 1978.

[21] Chou, Paul B, MC Sturzenbecker,VH Brecher. "Automatic defect classification for integrated circuits", IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology, vol.1907,pp.95-103,1993.

[22] Y J Chen, C Y Fan, K H Chang. "Manufacturing intelligence for reducing false alarm of defect classification by integrating similarity matching approach in CMOS image sensor manufacturing",Computers & Industrial Engineering,vol. 99,pp. 465-473,2016.

[23] Hou A, Zhou W, and Cui G. "Study on defect detection of IC wafer based on morphology", Electronic Imaging and Multimedia Technology V. International Society for Optics and Photonics,683334-683334-6. 2007.

[24] Qu G, Wood S L, Teh C. "Wafer Defect Detection Using Directional Morphological Gradient Techniques", Eurasip Journal on Advances in Signal Processing, vol.7,pp.1-18,2002.

[25] Abedini, Arsham, and M. Ehsanian. "Defect detection on IC wafers based on neural network", International Conference on Microelectronics pp.1-4,2017.

[26] D. M. Tsai and C. H. Yang. “A quantile-quantile plot-based pattern matching for defect detection,” Pattern Recognition Lett., vol. 26, pp.1948–1962, Oct. 2005.
[27] H Liu, W Zhou,Q Kuang, L Cao,and B Gao. “Defect detection of IC wafer based on two-dimension wavelet transform”, IEEE Transactions on Semiconductor Manufacturing, vol.41,pp.171-177,2010.
[28] Bay H, Ess A, Tuytelaars T, Van Gool L. “Speeded-Up Robust Features”,Computer Vision & Image Understanding, vol.3,pp.404-417,2008.
[29] G. Taubin in article "Estimation of Planar Curves, Surfaces and Nonplanar Space Curves Defined by Implicit Equations, With Applications to Edge and Range Image Segmentation", IEEE Trans. PAMI, Vol. 13, pages 1115-1138, (1991).
[30] Duda, R.O.; Hart, P. E. (January 1972). "Use of the Hough Transformation to Detect Lines and Curves in Pictures".

AUTHORS

Massimiliano Barone

Degreed at of Polytechnic of Milan in 1997, He works for STMicroelectronics since 2000. He developed projects about computer graphics, computer vision and image processing. He attended Cammi and Made4in European projects.

AN EFFICIENT LANGUAGE-INDEPENDENT MULTI-FONT OCR FOR ARABIC SCRIPT

Hussein Osman, Karim Zaghw, Mostafa Hazem and Seifeldin Elsehely

Computer Engineering Department, Faculty of Engineering,
Cairo University, Egypt

ABSTRACT

Optical Character Recognition (OCR) is the process of extracting digitized text from images of scanned documents. While OCR systems have already matured in many languages, they still have shortcomings in cursive languages with overlapping letters such as the Arabic language. This paper proposes a complete Arabic OCR system that takes a scanned image of Arabic Naskh script as an input and generates a corresponding digital document. Our Arabic OCR system consists of the following modules: Pre-processing, Word-level Feature Extraction, Character Segmentation, Character Recognition, and Post-processing. This paper also proposes an improved font-independent character segmentation algorithm that outperforms the state-of-the-art segmentation algorithms. Lastly, the paper proposes a neural network model for the character recognition task. The system has experimented on several open Arabic corpora datasets with an average character segmentation accuracy 98.06%, character recognition accuracy 99.89%, and overall system accuracy 97.94% achieving outstanding results compared to the state-of-the-art Arabic OCR systems.

KEYWORDS

Arabic OCR, Word Segmentation, Character Segmentation, Character Recognition, Neural Network

1. INTRODUCTION

The problem of Optical Character Recognition (OCR) has been the scope of research for many years [1]–[3] due to the need for an efficient method to digitize printed documents, prevent their loss and gradual unavoidable wear, as well as increase their accessibility and portability. The challenges that face Arabic OCR systems stem from the cursive and continuous nature of Arabic scripts. The presence of a semi-continuous baseline in Arabic text prevents the use of segmentation techniques proposed for other OCR systems. Moreover, the vertical overlapping of characters caused by ligatures means that segmenting a word along a single horizontal line will not achieve perfect segmentation. Furthermore, the challenges introduced by the nature of the Arabic script do not only affect character segmentation. The recognition of Arabic characters requires a huge training set since each character can have a different shape depending on its position in the word. Also, cases of constantly misclassifying a character as another specific one are frequent due to the presence of characters that are only told apart through the number of dots.
In this paper, we propose a complete OCR pipeline for Arabic text that is language-independent and supports multiple fonts. Our system takes a scanned image of an Arabic document as an input and outputs a digitized text document containing the predicted text. The input image is first preprocessed where binarization, denoising, and deskewing are carried out, followed by line and word segmentation. Character segmentation is then performed based on the extracted word-level features, followed by cut filtration based on the wide rules-set we have defined. The segmented

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 57-71, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101506

characters are then fed into our character recognition neural network model which classifies the given character into one of the 29 Arabic characters. Furthermore, post-processing is applied to the predicted characters and conveniently concatenates them into a comprehensible text document.

Our main contribution in this paper lies in the development of an efficient light-weight system that outperforms current state-of-the-art accuracy in the field of Arabic OCR. We achieve outstanding runtime results without trading off our system accuracy through efficient denoising of documents, vectorized implementation of all segmentation stages, and finely tuning our recognition model's complexity. In addition, we maintain our overall system accuracy by improving the character segmentation using the proposed improved cut filtration algorithm. The algorithm is robust against structural, morphological, and topological similarities between letters. We also propose a neural network model to learn the underlying features of input characters and build a character classification model. We finally eliminate the use of any lexical analysis to maintain the language independence of the system.

The rest of the paper is organized as follows: Section II describes the state-of-art related work. Section III discusses in detail the proposed OCR system. Section IV describes the datasets used for training the neural network and for evaluating the overall system performance, then discusses the results with comprehensive comparisons with other related algorithms and methods. Section V discusses limitations in our proposed system and proposals on future work in this area. Finally, Section VI presents the paper's conclusion.

2. RELATED WORK

There has been a variety of techniques proposed in the area of OCR for Arabic text. In this section, we will review the different approaches in the literature for character segmentation, feature extraction, and character recognition in OCR systems.

A significant number of approaches for the Arabic character segmentation task have been proposed in the literature. Mohamed et al. [4] proposed the use of contour extraction to facilitate the character segmentation phase followed by cut index identification based on the contour of a word. Their method achieved significant results in character segmentation; however, its performance degraded in the case of small-sized Arabic fonts or noisy documents. The scale-space method utilized by El Makhfi et al. [5] represents another direction in Arabic character segmentation. This method works well for scanned documents containing high random noise since blobs are retrieved from characters and are then detected to recover the appropriate cut positions in the image. Although this approach has been used in several computer vision applications since its first proposal, its use in Arabic character segmentation is still widely unexplored.

Inspired by NLP applications, Alkhateeb et al. [6] and Radwan et al. [8] proposed the use of a sliding window approach for segmentation. A Convolutional Neural Network (CNN) is used to determine the likelihood of a given sliding window consisting of several segments to be an actual character. Subsequently, the segments that qualify as characters are fed into their recognition model. This segmentation approach has shown very high performance on a single font but failed to maintain this high performance when tested on multiple fonts and font sizes. Lots of efforts [8]–[10] in Arabic character segmentation have been based on histogram analysis of words.
It is important to mention that Qaroush et al. [8] also proposes a very effective word segmentation methodology that achieves state-of-the-art accuracy by implementing cut identification and filtration through gap length. Their proposed method handles multiple fonts and font sizes.

Although not being as challenging as character segmentation, feature extraction is one of the keys to boosting the accuracy of any OCR system. Rashad and Semary [11] adopted a simple approach that manually extracts basic features from each character such as height, width, number of black pixels, number of horizontal transitions, number of vertical transitions, and other similar features. Rashid et al. [12] proposed a multi-dimensional Recurrent Neural Network (RNN) for their recognition system that achieved outstanding character recognition rates. However, the effects of using this complex approach on the runtime were not properly investigated. The use of deep learning approaches is highly efficient when developing Arabic OCR systems that operate on unconstrained scene text and video text, not scanned documents [13].

Dahi et al. [14] adopted a similar approach by manually selecting features from a noise-free and pre-segmented character input. They added a font recognition module before the feature extraction, to include the font as a feature for the character recognition, alongside other slightly complex features such as ratios between black pixel count per region and the statistical centre of mass for each character proposed by [15]. The overall system of [14] achieved very high accuracy for Arabic character recognition.

It is worth mentioning that the OCR system of [14] did not include a character segmentation module as it worked only on a pre-segmented character input. Additionally, the OCR architecture of Dahi et al. [14] failed to scale up and recognize Arabic characters in other fonts that were not supported by the font recognition module. A convenient middle ground between ineffective manual extraction [11], [14], and the computationally expensive use of deep learning [12], [13] is presented by the use of Principal Component Analysis (PCA) for automatic feature extraction.
As proposed by Shayegan and Aghabozorgi [16], PCA provides an efficient and effective solution for the problem of feature extraction in recognizing Arabic numerals. However, applying PCA becomes computationally infeasible for the huge datasets needed for training Arabic character recognizers.

As for character recognition, implementing this module without the use of machine learning has been deemed obsolete; because of the outstanding results achieved by machine learning recognition models. Hence, we will only consider the techniques for character recognition that are based on machine learning for review in the subsequent paragraphs. Shahin [17] proposed using linear and ellipse regression to generate codes from the segmented characters. This approach of codebook generation and code matching showed average results for character recognition. Additionally, it suffered from the same problem of not being able to generalize to other Arabic fonts as [14]. The use of the holistic approach in [18] emerged from the difficulty of the segmentation phase as we mentioned. This word-level recognition technique skips all the inaccuracy produced by segmentation errors but creates the need for post-recognition lexical analysis, thus resulting in a language-dependent system that relies on a look-up data base and semantic checks after recognition.

Often paired with the use of a sliding window for segmentation, the use of a Hidden Markov Model (HMM) for character recognition was adopted by [18]–[20]. By using a model that mimicked the architecture of an Automatic Speech Recognition system and an HMM, Rashwan et al. [19] managed to overcome the challenges presented by the presence of ligatures. Many OCR systems use other classical machine learning techniques in their character recognition module; such as random forests [11], [14], K-Nearest Neighbor [11], shallow neural networks [21]. The neural network model used by Al-Jarrah et al. [21] yielded the best results, compared to other classical machine learning techniques [11], [14], [19], [20], and was able to generalize over different fonts.

3. METHOD PROPOSED

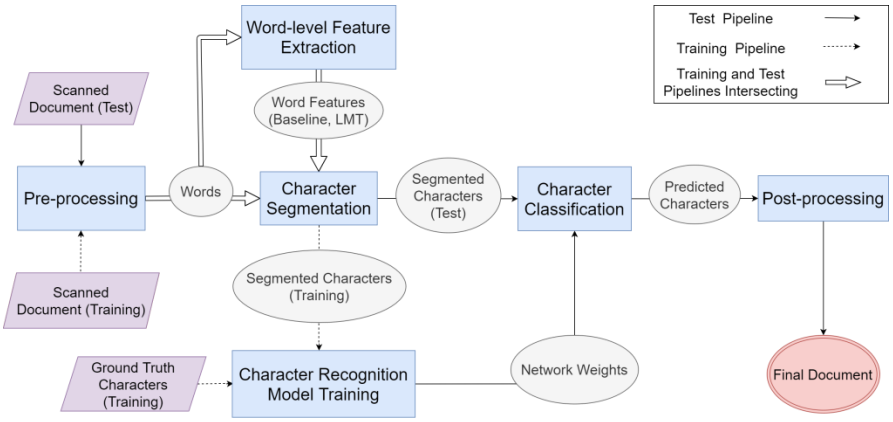
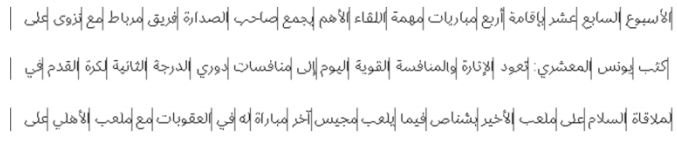


Figure 1. System Architecture: A block diagram representing the system modules and both training and testing (prediction) pipelines.

As shown in Figure 1, the input to the OCR system is expected to be a number of scanned images of computerized Arabic documents. In the Pre-processing stage, we apply image preprocessing through the filtering, deskewing, and denoising of the input images, followed by line and word segmentation. In the Word-Level Feature Extraction stage, we generate statistical, structural, and topological features for every word. In the Character Segmentation stage, we apply the Excessive Cut Creation and Improved Cut Filtration algorithms to segment the word into individual characters. These segmented characters, together with the associated ground truth labels, represent the dataset that our Character Recognition Model uses for training. We train an Artificial Neural Network (ANN) to classify each segmented character into one of the 29 possible characters in the Arabic language. Finally, we aggregate the segmented characters into words and generate the output of our OCR system.

3.1. Pre-processing

a)



b)

Figure 2. Examples from Preprocessing Stages: a) cut indices for line segmentation, on document b) cut indices for word segmentation, on lines

The preprocessing module consists of four main steps: raw image filtering, document deskewing, line segmentation, and word segmentation. Initially, we start by converting the input image to

grayscale, then binarizing it by applying Adaptive Gaussian Thresholding. The document deskewing is carried out by rotating the document about its geometric centre with a specific angle calculated through obtaining the orientation of the text's minimum bounding rectangle. This is followed by another round of binary thresholding to set binary values for the pixels that have been interpolated due to the previous rotation. The line segmentation step is performed by blurring the image then applying horizontal histogram projection of black pixels. Local minima of this histogram indicate positions of separations between lines. The goal of blurring is to avoid generating segmented lines containing dots only (e.g. the dots of the 'yaa' letter at the end of the word) or containing special Arabic diacritics only (e.g. 'hamza' or 'shadda'). For word segmentation, we applied thinning to the image instead of blurring. We propose thinning as a solution to enhance the fine details within and between words. This eliminates the overlapping pixels between any two words (if one of them ends with a curved letter) and results in standardized gap lengths between words. We subsequently implemented [8]'s algorithm for cut identification and filtration based on gap lengths.

3.2. Word-level Feature Extraction

The word-level feature extraction algorithm takes the segmented words as an input and generates for each word several geometric features. These features are essential for the character segmentation algorithm to be able to identify individual characters and segment them accordingly. We discuss the generated features for each segmented word in this subsection.

3.2.1. Baseline

The baseline is an imaginary horizontal line that connects all of the letters in an Arabic word [8]. In order to detect the baseline for every word, we search for the row of pixels with the greatest number of black pixels by applying horizontal histogram projection and finding the global maximum.

3.2.2. Line of Maximum Transitions (LMT)

We define a transition as a change in pixel value from 0 (black) to 1 (white) or vice versa. An important feature of the Arabic script is that a transition above the baseline is always due to a character being drawn. The Line of Maximum Transitions (LMT) is the line that cuts through the greatest number of these transitions (i.e. the row of pixels in which the number of transitions from black to white and white to black pixels is greatest) [8]. For better estimation of the baseline and LMT, we propose that both features should be derived from the whole line of text rather than from each word.

3.2.3.Potential Cut Region (PCR)

The LMT's key characteristic is that it passes through all potential characters in an Arabic word and is therefore essential in separating the word into its individual character components. A cut is defined as an imaginary line that separates two characters, and a Potential Cut Region is the area where a cut may exist. Since we cannot at first determine which character-intersections with the LMT belong to the same character and which are the result of a new character being written, we assume that each intersection represents a distinct character and therefore a PCR exists between any two successive intersections.

In order to determine the start and end indices of PCRs, we traverse the LMT from right to left. Each black pixel followed by a white pixel is defined as a start index of a PCR and each white

pixel followed by a black pixel is defined as an end index. Furthermore, the column of pixels that is chosen as the location of a cut is called a cut index.

Figure 3. An Arabic word with the baseline highlighted in red, the LMT highlighted in green, the PCR start indices highlighted in dark blue, and the PCR end indices highlighted in light blue.

3.3. Character Segmentation

To solve the problem of over-segmentation of characters, our character segmentation algorithm consists of two main steps: Excessive Cut Creation (ECC), which generates excessive potential cuts, and Improved Cut Filtration (ICF), which filters the false cuts from these potential cuts and outputs a set of valid cuts only. The Improved Cut Filtration algorithm is considered an improvement over the Cut Filtration algorithm proposed by Qaroush et al. [8].

3.3.1. Excessive Cut Creation Algorithm (ECC)

In the Arabic script, characters are either connected through the baseline or separated by a single space. Therefore, there are two different methods used in the Excessive Cut Creation algorithm: Finding Baseline Cuts and Finding Separation Cuts. The former deals with separating baseline-connected characters and some space-separated characters while the latter addresses the remaining space-separated characters.

I. Baseline Cuts: In order to identify a baseline cut, we inspect each column of pixels in a PCR, starting from the end index to the start index. We count the total number of black pixels above and below the baseline and, for every PCR, we propose the position of the cut index to be the first column where the count is zero, i.e. where the only black pixel allowed is the baseline. This approach is useful as a preliminary step for separating baseline-connected characters and also helps in separating some space-separated characters; e.g. the ‘aleph’, ‘daal’ or ‘thaal’ (ـا، ـد ، ـذ) followed by another letter.

II. Separation Cuts: While baseline cuts succeed in separating some space-separated characters, it will not place a cut whenever a black pixel exists below the baseline. This introduces a real challenge for letters that have curves which dip below the baseline such as ‘reh’ and ‘zeen’ (ـر، ـز) since the entire PCR may contain black pixels below the baseline. To solve this problem, we place a separation cut whenever the pixels at the left and right indices of a PCR are not connected by an uninterrupted path of black pixels. This ensures that a cut will be placed wherever there are two space-separated characters even if one of them happens to dip below the baseline. We choose the separation cut index to be the middle of this PCR.

As illustrated in Algorithm 1, we search every PCR for a baseline cut. If we find a baseline cut, then we add this cut to the set of cut Indices. If we fail to find a baseline cut, we search for a

separation cut. If we find a separation cut, then we add this cut to the set of cut Indices. If we find neither a baseline cut nor a separation cut, we claim that this PCR contains a part of a character that should not be cut.

Algorithm 1. ECC Algorithm

```
Input: PCRArray

ExcessiveCutIndices = Φ
for all PCR in PCRArray do
    CutFound = False
    for all PixelColumn in PCR do
        if ProjAboveBaseline = 0 and ProjBelowBaseline = 0 then
            CutFound = True
            ExcessiveCutIndices.Add(PixelColumn.Index)
            break
    if CutFound = False and PCR.IsConnected = False then
        SeparationCutIndex = (PCR.LeftIndex + PCR.RightIndex) / 2
        ExcessiveCutIndices.Add(SeparationCutIndex)

Output: ExcessiveCutIndices
```

3.3.2. Improved Cut Filtration Algorithm (ICF)

After generating a relatively large number of potential cut indices in the ECC stage, we begin inspecting each Potential Character (PC), where a PC is defined as any region that exists between two successive cuts. The goal of the cut filtration stage is to determine which of the cut indices are excessive false cuts. We identify the Arabic letters that usually cause false cuts in the cut filtration algorithm in the paper written by [8] and other character segmentation algorithms and we propose an improved algorithm to detect all of these letters. To the best of our knowledge, there is no single algorithm that can perfectly segment all Arabic letters, and hence we introduce our solution as the new state-of-the-art. We will discuss these challenging cases, and how ICF handles each of them accordingly.

I. **‘Seen’ and ‘Sheen’ case (س، ش):** The most notable causes of excessive false cuts are the letters ‘seen’ and ‘sheen’ (س، ش). These two letters are composed of three successive strokes, with three dots above the second stroke in the case of ‘sheen’. A stroke is a PC that represents a part of a character having a one-pixel thickness. Because each stroke passes through the LMT, the ECC algorithm generates three cut indices, instead of one. In order to filter these cuts, we define a seen-stroke as a stroke with no dots above or below the baseline and no hole. A seen-stroke is also characterized by having a small peak above the baseline (i.e. is relatively short) and a flat structure near the baseline (does not dip below the baseline). Also, we define a sheen-stroke as a seen-stroke with dots above the baseline.

   Based on the above definitions, the false cuts in the ‘seen’ or ‘sheen’ letters in the start and middle of the word (ـسـ، ـشـ) will be filtered by detecting three successive seen-strokes for the ‘seen’ case or two seen-strokes with one sheen-stroke in between for the ‘sheen’ case. Nevertheless, this method filters the false cuts of the two letters in the start and the middle of the word, but it fails to detect false cuts when they appear at the end of the word.
   We propose an improvement over this algorithm by taking into consideration the unique characteristic of the ‘seen’ and ‘sheen’ letters when they exist at the end of the word. This characteristic is the bowl shape seen at the end of these letters (ـس ، ـش). We define a bowl as (1) a PC with no dots above or below the baseline (2) a PC such that its right cut index

must have at least one black pixel and its left cut index must contain no black pixels. This means that the bowl must be directly connected through the baseline to a PC before it and must not be connected to any PC after it. (3) a PC such that there must exist a region where the baseline vanishes within but resurfaces again (i.e. there exists a dip below the baseline). (4) a PC with a relatively small peak above the baseline.

Based on this improved algorithm, the false cuts (the first two cut indices) in the 'seen' letter will be filtered when the IFC algorithm detects three successive seen-strokes or two successive seen-strokes followed by a bowl, whereas the false cuts (the first two cut indices) in the 'sheen' letter will be filtered when the IFC algorithm detects two seen-strokes separated by a sheen-stroke or a seen-stroke followed by a sheen-stroke and a bowl.

II. **'Saad' and 'Daad' case (ص، ض، صـ، ضـ):** The second set of characters that result in an additional false cut is 'saad' and 'daad' (ص، ض، صـ، ضـ). To filter the false cuts in these two letters, we define a hole as a PC containing a rounded area of white pixels (a hole) enclosed by a larger rounded area of black pixels. The two letters consist of a hole followed by a seen-stroke when they appear at the start or the middle of a word, or a hole followed by a bowl when they appear at the end of a word. Initially, we wrote our cut filtration algorithm so that whenever it encountered such a case, it merged the two PCs into one by removing the cut index in-between.

However, an interesting case that generated confusion with this definition was a 'meem' or a 'faa' followed by a 'daal' (فد ، مد). Both cases would always be misinterpreted as a 'saad' or 'daad' and the cut filtration algorithm would falsely merge the 'daal' and the preceding character into one.

Therefore, we defined a saad-stroke to differentiate between the 'daal' stroke and the strokes of 'saad' and 'daad'. The saad-stroke is a seen-stroke with the additional condition of being surrounded by cut indices that have at least one black pixel each. The goal of this extra specification is to ensure that the saad-stroke is connected to the baseline from both sides and is not followed by a space, as is the case with 'daal'. As such, the false cuts in 'saad' and 'daad' can be filtered when the cut filtration algorithm finds a hole followed by a saad-stroke or when it finds a hole followed by a bowl.

III. **'Baa', 'Taa', 'Thaa' and 'Faa' case:** There is also a difficult case where a 'baa', 'taa', 'thaa' or 'faa' (ب، ت، ث، ف) at the end of a word may cause a false cut. This occurs when the stroke at the end of the aforementioned characters is tall enough to intersect with the LMT. In this case, the ECC algorithm will generate a false cut at this stroke position which will need to be filtered.

In order to filter this extra false cut, we define an end stroke as a 'seen' stroke that has additional restrictions. Firstly, an end stroke must be followed by a cut index that does not intersect the baseline and preceded by a cut index that intersects the baseline. Furthermore, we locate the leftmost and the uppermost black pixels of the PC and we calculate the horizontal distance d between these two pixels. If d is measured to be less than or equal to 2 pixels, the ICF algorithm identifies the PC as an end stroke and removes the preceding cut.

It is worth noting that the extra restriction on the horizontal distance between the top-leftmost and the uppermost black pixels is essential in order not to incorrectly identify the 'daal' letter (ـد) as an end stroke because the stroke in the 'daal' letter has geometrical features that resemble the second stroke in the 'taa' and 'thaa' letters.

Algorithm 2. ICF Algorithm

```
Input: PCArray

FilteredCharacterArray = Φ
for all PC in PCArray do
    if PC is SeenStroke then
        if NextPC is SeenStroke or NextPC is SheenStroke then
            if AfterNextPC is SeenStroke or AfterNextPC is Bowl then
                    PCArray.Merge(PC, NextPC, AfterNextPC)
for all PC in PCArray do
    if PC is Saad/DaadStroke or PC is Bowl then
        PCArray.Merge(PreviousPC, PC)
for all PC in PCArray do
    if PC is EndStroke with D ≤ 2 then
        PCArray.Merge(PreviousPC, PC)
FilteredCharacterArray = PCArray

Output: FilteredCharacterArray
```

Although the previous cut filtration cases may seem largely independent of each other, the filtration order can greatly affect the character segmentation performance. For instance, executing the saad/daad-case before the seen/sheen-case may cause the algorithm to confuse the first stroke of the ‘seen’ letter as a ‘saad’ or a ‘daad’ stroke, and falsely merge it with the preceding character. As shown in Algorithm 2, our ICF algorithm filters all PCs according to the order of filtration cases mentioned in this paper.

It is worth mentioning that our character segmentation algorithm does not depend on any linguistic or statistical patterns in the Arabic language and is well-equipped to segment any sequence of Arabic letters. The previous work of [8] relied on their cut filtration algorithm’s assumption that some letters such as ‘saad’ and ‘daad’ are never followed by ‘seen’ or ‘sheen’. On the other hand, our ICF algorithm is general enough to segment any Arabic word regardless of the arrangement of the letters in the word.

Our character segmentation algorithm provides several improvements over the work of [8]. As opposed to the algorithm proposed by [8], the ECC algorithm does not generate a cut at positions where there are black pixels above or below the baseline, and this improvement reduces the number of false cuts that the cut filtration algorithm has to filter. The second improvement in our ICF is filtering each PC based on clear and specific structural features of Arabic letters such as ‘seen-stroke’, ‘sheen-stroke’, ‘saad-stroke’, ‘end-stroke’, ‘bowl’ and ‘hole’. These features are essential for the ICF algorithm in order to not accidentally remove any valid cut. Previous work in character segmentation [8] did not provide a correct exact definition of a bowl and, as a result, the characters having a bowl-shape in their structure such as ‘noon’, ‘qaaf’, ‘yaa’, ‘raa’, and ‘zaay’ letters (ن، ق، ي، ر، ز) were confused with the bowl part of the ‘saad’ and ‘daad’ letters (ص، ض) and were falsely merged with the preceding character.

The third improvement is filtering the false cuts in the case of the letters ‘baa’, ‘taa’, ‘thaa’, and ‘faa’ based on the leftmost black pixel and the uppermost black pixel instead of the top-leftmost black pixel only. The aforementioned false cuts are detected in [8]’s algorithm if the top-leftmost black pixel has a relatively small height when compared to the highest black pixel in the line. As a result, almost all letters that occur at the end of a word will be falsely merged with the preceding character. Only the ‘alef’ (أ) character will not be falsely merged when written at the end of a word since its top-leftmost black pixel is relatively tall, while every other character in

the Arabic alphabet will have a relatively low top-leftmost black pixel when written at the end of a word.

Finally, our segmentation algorithm solves the challenging case of the ‘seen’ (س) and ‘sheen’ (ش) letters at the end of the word, which was not handled by the algorithm proposed by [8] and resulted in a considerable number of incorrect segmentations. We conclude that our algorithm improves on the one proposed by Qaroush et al. [8] and addresses many of its shortcomings in the character segmentation problem.

3.3.3. Character Recognition Model

We propose feeding the images of the segmented characters to an artificial neural network since this will yield better performance than choosing features manually. The architecture of the ANN used in this research is a multilayered feed-forward network architecture with four layers. This neural network learns highly complex nonlinear features by training on a sufficiently large training set of Arabic characters together with their ground-truth labels.

We will generate our own training set by comparing the number of segmented letters generated from a word-using our proposed algorithms- with the number of letters in the ground truth word. If the numbers match, we associate each letter with its corresponding label. If the numbers do not match, then we skip and discard this word. It is worth noting that this over/under-segmentation problem arises if the scanned document was too noisy or blurry. However, we are cautious about the training set and prefer not to risk accidentally training with false characters.

We begin by resizing all the images generated from the character segmentation algorithm to be 24x24 pixels and then flattening them to be 576-dimensional vectors. In addition, we perform dimensionality reduction on these 576-dimensional vectors using Incremental Principal Component Analysis (IPCA), which -as illustrated in Figure 4- can represent the original vectors using only 200 principal components while retaining 99% of the total variance in the data. These 200-dimensional vectors are then fed to the neural network that consists of two hidden layers and a softmax output layer of sizes 150, 70, and 29 respectively. The softmax output layer assigns a likelihood value for each character of the 29 characters. This value represents the probability that this character is the correct classification for the input image. The inference phase of the model then classifies the input character as the character with the highest likelihood of being the correct prediction.

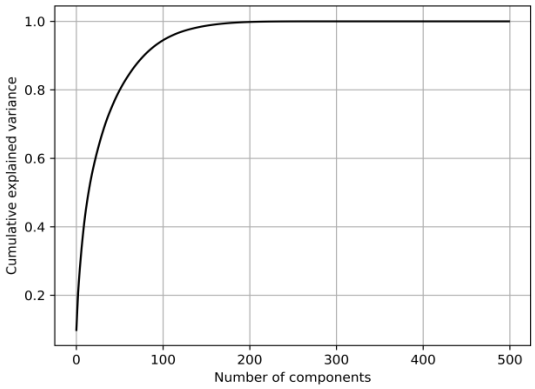


Figure 4. PCA Dimensionality Graph

We trained the model with our generated dataset that consists of 1,200,000 images of perfectly segmented Arabic letters in Naskh script using mini-batch training of batch size 8192 for 80 epochs. Categorical cross-entropy loss is calculated in the output layer and optimized using Adam optimizer [22] with hyper-parameters β1 and β2 being 0.9 and 0.999 respectively. For the hidden layers, the Rectified Linear Unit (ReLU) is used as a non-linear activation function followed by a 10% dropout layer to ensure elimination of overfitting as much as possible. A learning rate of 0.001 with no decay factor is used. The network is initialized using He initialization [23] and shuffled per epoch. We used a validation set of 12,000 images, representing 1% of the training set.

3.3.4. Post-processing

The character recognition model outputs a predicted class for each of the character images. However, a final step of post-processing is necessary to aggregate these characters into words and separate them with spaces to generate a meaningful text document. The previously predicted letters are produced consecutively with no spaces until we encounter an activated End of Word (EOW) flag. Every character has an EOW flag value; values for this flag are obtained in the character segmentation phase by setting the EOW flag to true (activated) for the final segment of the word that is being segmented. In the post-processing step, whenever we encounter a character with an activated EOW flag, indicating the end of a word, we place a space directly after this character. This step ensures the production of a comprehensible document.

4. EXPERIMENTAL RESULTS

4.1. Datasets

There are several datasets for Arabic character recognition [24]. We used the open dataset AlWatan corpus [25] for training our neural network. The dataset includes very large Arabic vocabulary, with different font types, sizes, and styles and is rich with separable characters, overlapping characters, and ligatures. The scanned images in the dataset are 72 dpi resolution images. For training the model, we randomly selected 550 documents/images of plain Arabic Naskh from different topics, approximately 282,000 words, or 1,200,000 characters. For validation and testing, we randomly chose 6 and 10 documents/images of Arabic Naskh script as a validation set and a test set respectively, of sizes 3300 words (12000 characters) and 5500 words (100,500 characters).

We further experimented with the system with different test sets of plain Arabic fonts; Naskh, Transparent Arabic, Simplified Arabic, M. Unicode Sara, Tahoma, Times New Roman, and Arial with different font sizes; 10, 12, 14, and 16. We tested the system on the APTI dataset (Arabic Printed Text Image) [26], which is a standard benchmarking dataset for Arabic OCR tasks. We used Keras [27] for training our model and we ran our experiments on a core i7 5820K 3.3 GHz machine, with 32 GB RAM, Ubuntu OS 16.04, and GPU NVIDIA RTX 2070 with 8 GB memory and 2560 cores.

4.2. Results and Evaluation

This section evaluates our word segmentation algorithm, character segmentation algorithm, character recognition model, as well as the overall system accuracy. We used both Watan-2004 and a subset of the APTI datasets for evaluating our system. We also compare our segmentation algorithms to the work of Anwar et al. [28], Radwan et al. [29], and Mousa et al. [30]. We

compare the overall system performance when our improved segmentation algorithm is used against when the segmentation algorithm of Qaroush et al. [8] is used.

4.2.1. Word Segmentation

We define the word segmentation accuracy as the number of correctly segmented words divided by the total number of actual words in the document. Our method achieves an average word segmentation accuracy of 99.94% for different fonts, as indicated in Table 1, where we also compare our results for each font with Qaroush et al. [8].

Table 1. Word Segmentation Comparison with Qaroush et al. [8]

Font	Method Proposed		Qaroush et al. [8]	
	Input Words	Accuracy	Input Words	Accuracy
Tahoma	25,928	99.96%	2,319	99.4%
Naskh	29,169	99.95%	2,921	96.1%
Simplified Arabic	22,687	99.94%	2,884	98.8%
Transparent Arabic	21,055	99.90%	2,860	99.1%

4.2.2. Character Segmentation

We first segment each word and compare the number of segmented characters with the number of actual characters in the word. Then we count their absolute difference as incorrectly segmented characters. We finally define the character segmentation accuracy as the total number of correctly segmented characters divided by the total number of actual characters. Our method achieves an average character segmentation accuracy of 98.23% for different fonts, as indicated in Table 2, where we also compare our results for each font with [8]. Comparing our results with the results of [28] and [30], as shown in Table 3, we note that our system outperforms all other character segmentation algorithms in the character segmentation task.

Table 2. Character Segmentation Comparison with Qaroush et al. [8]

Font	Method Proposed		Qaroush et al. [8]	
	Input Characters	Accuracy	Input Characters	Accuracy
Tahoma	114,080	97.80%	12,262	97.00%
Naskh	131,260	98.66%	12,585	94.52%
Simplified Arabic	100,957	99.06%	13,572	96.10%
Transparent Arabic	89,483	97.40%	13,120	96.26%

Table 3. Character Segmentation Results

Method Tested	Dataset	Font Sizes	Font Styles	Segmentation Accuracy
Anwar et al. [28]	Self-Generated	70pt	Traditional Arabic	97.55%
Mousa et al. [30]	Self-Generated	Not reported	Not reported	98.00%
Qaroush et al. [8]	[26]: 100,000 characters	10, 12, 14, 16, 18, and 24	Font Set A*	97.50%
Method Proposed	[25] + [26]: 1,000,000+characters	10, 12, 14, and 16	Font Set B†	98.23%

*Font Set A: Naskh, Transparent Arabic, Simplified Arabic, M Unicode Sara, Tahoma, and Advertising Bold

†Font Set B: Font Set A, Andalus, Diwani, Thuluth

4.2.3. Overall System Performance

The neural network achieved average recognition accuracy of 99.89%. The overall system accuracy is measured by calculating the Levenshtein edit distance between the generated document and the actual document. We show that our system achieves an overall accuracy of 97.94%. Therefore, and to the best of our knowledge, we propose that our Arabic OCR system is superior to all other segmentation based Arabic OCR in terms of accuracy and running time. It is also worth mentioning that our proposed system was developed and trained using a very large dataset, which is rich with Arabic text in different font types and sizes. Table 4 shows the evaluation of our system in comparison to [31]'s system and [8]'s segmentation method followed by an ANN for recognition.

Table 4. Overall System Evaluation

Method Tested	Dataset	Number of Words	System Accuracy	Avg. Run Time / 550 Words (sec)
Qaroush et al. [8] + ANN	[25]	3300	94.95%	3.42
Touj et al. [31]	1500 words	1500	97.00%	NA
Method Proposed	[25] + [26]	3300	97.94%	1.49

5. FUTURE WORK

Much like other OCR systems, our system's performance decreases when operating on documents with high noise. This creates the demand to work on more elaborate preprocessing methods in the future without sabotaging our system's remarkable runtime. Also, our recognition model is currently limited to the fonts that it inferred from the training set; as a result of that, we aim to enrich the training set to contain more fonts and possibly all fonts without ligatures.
Apart from the low-level additions to our system, potential work on this system would include integrating it into larger applications such as image-speech systems. Such applications -where instant results are crucial- will be a perfect fit for our method because of the very low runtime we provide.

6. CONCLUSION

Arabic Optical Character Recognition introduces many challenges in the character segmentation and recognition phase. This paper proposes a complete language-independent Arabic OCR pipeline with an improved character segmentation algorithm based on word-level features and a bio-inspired character recognition model based on neural networks. We proposed a highly accurate and efficient system. The overall system architecture consists of: a simple yet effective pre-processing module, an enhanced reliable module for character segmentation, an artificial neural network for the recognition of segmented characters, and finally a post-processing module that formulates the output of our system into a digitized document. We evaluated our system on different datasets with high variability in font types and sizes. The experimental results show that our system outperforms the current state-of-the-art algorithms for word segmentation, character segmentation, and character recognition. We evaluated the overall performance of the system and concluded that our system achieves outstanding results in accuracy and running time.

REFERENCES

[1] M. Namysl and I. Konya, "Efficient, lexicon-free ocr using deep learning," arXiv preprint arXiv:1906.01969, 2019.

[2] S. Dixit, M. Bharath, Y. Amith, M. Goutham, K. Ayappa, and D. Harshitha, "Optical recognition of digital characters using machine learning," International Journal of Research Studies in Computer Science and Engineering, vol. 5, no. 1, pp. 9–16, 2018.

[3] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multidigit number recognition from street view imagery using deep convolutional neural networks," arXiv preprint arXiv:1312.6082, 2013.

[4] K. Mohammad, A. Qaroush, M. Ayesh, M. Washha, A. Alsadeh, and S. Agaian, "Contour-based character segmentation for printed arabic text with diacritics," Journal of Electronic Imaging, vol. 28, no. 4, p. 043030, 2019.

[5] N. El Makhfi and O. El Bannay, "Scale-space approach for character segmentation in scanned images of arabic documents," Journal of Theoretical and Applied Information Technology, vol. 94, no. 2, p. 444, 2016.

[6] J. H. AlKhateeb, J. Ren, J. Jiang, and H. Al-Muhtaseb, "Offline handwritten arabic cursive text recognition using hidden markov models and re-ranking," Pattern Recognition Letters, vol. 32, no. 8, pp. 1081– 1088, 2011.

[7] M. A. Radwan, M. I. Khalil, and H. M. Abbas, "Neural networks pipeline for offline machine printed arabic ocr," Neural Processing Letters, vol. 48, no. 2, pp. 769–787, 2018.

[8] A. Qaroush, B. Jaber, K. Mohammad, M. Washaha, E. Maali, and N. Nayef, "An efficient, font independent word and character segmentation algorithm for printed arabic text," Journal of King Saud University- Computer and Information Sciences, 2019.

[9] L. Zheng, A. H. Hassin, and X. Tang, "A new algorithm for machine printed arabic character segmentation," Pattern Recognition Letters, vol. 25, no. 15, pp. 1723–1729, 2004.

[10] M. Amara, K. Zidi, K. Ghedira, and S. Zidi, "New rules to enhance the performances of histogram projection for segmenting small-sized Arabic words," in International Conference on Hybrid Intelligent Systems. Springer, 2016, pp. 167–176.

[11] M. Rashad and N. A. Semary, "Isolated printed arabic character recognition using knn and random forest tree classifiers," in International Conference on Advanced Machine Learning Technologies and Applications. Springer, 2014, pp. 11–17.

[12] S. F. Rashid, M.-P. Schambach, J. Rottland, and S. von der N¨ull, "Low resolution Arabic recognition with multidimensional recurrent neural networks," in Proceedings of the 4th International Workshop on Multilingual OCR, 2013, pp. 1–5.

[13] M. Jain, M. Mathew, and C. Jawahar, "Unconstrained scene text and video text recognition for arabic script," in 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR). IEEE, 2017, pp. 26–30.

[14] M. Dahi, N. A. Semary, and M. M. Hadhoud, "Primitive printed arabic optical character recognition using statistical features," in 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS). IEEE, 2015, pp. 567–571.

[15] A. Rosenberg and N. Dershowitz, Using SIFT descriptors for OCR of printed Arabic. Tel Aviv University, 2012.

[16] M. A. Shayegan and S. Aghabozorgi, "A new dataset size reduction approach for pca-based classification in ocr application," Mathematical Problems in Engineering, vol. 2014, 2014.

[17] A. A. Shahin, "Printed arabic text recognition using linear and nonlinear regression," arXiv preprint arXiv:1702.01444, 2017.

[18] F. Nashwan, M. A. Rashwan, H. M. Al-Barhamtoshy, S. M. Abdou, and A. M. Moussa, "A holistic technique for an arabic ocr system," Journal of Imaging, vol. 4, no. 1, p. 6, 2018.

[19] M. Rashwan, M. Fakhr, M. Attia, and M. El-Mahallawy, "Arabic ocr system analogous to hmm-based asr systems; implementation and evaluation," Journal of Engineering and Applied Science-Cairo, vol. 54, no. 6, p. 653, 2007.

[20] N. B. Amor and N. E. B. Amara, "Multifont arabic characters recognition using houghtransform and hmm/ann classification." Journal of multimedia, vol. 1, no. 2, pp. 50–54, 2006.

[21] O. Al-Jarrah, S. Al-Kiswany, B. Al-Gharaibeh, M. Fraiwan, and H. Khasawneh, "A new algorithm for arabic optical character recognition." WSEAS Transactions on Information Science and Applications, vol. 3, no. 4, pp. 832–845, 2006.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.
[24] A. Lawgali, M. Angelova, and A. Bouridane, "Hacdb: Handwritten Arabic characters database for automatic character recognition," in European Workshop on Visual Information Processing (EUVIP). IEEE, 2013, pp. 255–259.
[25] M. Abbas, "Alwatan," https://sites.google.com/site/mouradabbas9/corpora, 2004.
[26] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert, "A new arabic printed text image database and evaluation protocols," in 2009 10th International Conference on Document Analysis and Recognition. IEEE, 2009, pp. 946–950.
[27] F. Chollet et al., "Keras," https://keras.io, 2015.
[28] K. Anwar, H. Nugroho et al., "A segmentation scheme of arabic words with harakat," in 2015 IEEE International Conference on Communication, Networks and Satellite (COMNESTAT). IEEE, 2015, pp. 111–114.
[29] M. A. Radwan, M. I. Khalil, and H. M. Abbas, "Predictive segmentation using multichannel neural networks in arabic ocr system," in IAPR Workshop on Artificial Neural Networks in Pattern Recognition. Springer, 2016, pp. 233–245.
[30] M. A. Mousa, M. S. Sayed, and M. I. Abdalla, "Arabic character segmentation using projection based approach with profile's amplitude filter," arXiv preprint arXiv:1707.00800, 2017.
[31] S. Touj, N. E. B. Amara, and H. Amiri, "Generalized hough transform for arabic printed optical character recognition." Int. Arab J. Inf. Technol., vol. 2, no. 4, pp. 326–333, 2005.

FPGA ROUTING ACCELERATION BY EXTRACTING UNSATISFIABLE SUBFORMULAS

Zhang Jianmin, Li Tiejun and Ma Kefan

College of Computer, National University of Defense Technology, Changsha, China

ABSTRACT

Explaining the causes of infeasibility of Boolean formulas has practical applications in various fields. A small unsatisfiable subset can provide a succinct explanation of infeasibility and is valuable for applications, such as FPGA routing. The Boolean-based FPGA detailed routing formulation expresses the routing constraints as a Boolean function which is satisfiable if and only if the layout is routable. The unsatisfiable subformulas can help the FPGA routing tool to diagnose and eliminate the causes of unroutable. For this typical application, a resolution-based local search algorithm to extract unsatisfiable subformulas is integrated into Boolean-based FPGA routing method. The fastest algorithm of deriving minimum unsatisfiable subformulas, called the branch-and-bound algorithm, is adopted to compare with the local search algorithm. On the standard FPGA routing benchmark, the results show that the local search algorithm outperforms the branch-and-bound algorithm on runtime. It is also concluded that the unsatisfiable subformulas play a very important role in FPGA routing real applications.

KEYWORDS

FPGA routing, Boolean satisfiability, unsatisfiable subformula, local search.

1. INTRODUCTION

Many real-world problems, arising in electronic design, equivalence checking, property verification, automatic placement and routing and Auto Test Pattern Generation (ATPG), can be formulated as constraint satisfaction problems, which can be translated into Boolean formulas in conjunctive normal form (CNF). Modern Boolean satisfiability (SAT) solvers, such as Chaff [1] and MiniSAT [2], which implement enhanced versions of the Davis-Putnam-Logemann-Loveland(DPLL) backtrack-search algorithm, are usually able to determine whether a large formula is satisfiable or not. When a formula is unsatisfiable, it is often required to find an unsatisfiable subformula, that is, a small unsatisfiable subset of the original formula. Localizing a small unsatisfiable subformula is necessary to determine the underlying reasons for the failure.
Explaining the causes of unsatisfiability of Boolean formulas is an essential requirement in various fields, such as electronic design automation and formal verification of hardware. A typical paradigm is fixing wire routing in FPGAs, where an unsatisfiable subformula implies that the channel is unroutable. Furthermore, we are usually interested in a small explanation of infeasibility that excludes irrelevant information. There have been considerable researches in deriving the unsatisfiable subformulas. Most of previous works are complete search approaches, essentially on the basis of enhanced versions of the DPLL backtrack-search algorithm. However, existing studies have very little concern regarding unsatisfiable subformulas extraction using incomplete local search method.

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 73-81, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101507

The unsatisfiable sub formulas solver plays a very important role in the automatic routing tool for deciding the routability of FPGA devices. The Boolean-based FPGA detailed routing formulation expresses the routing constraints as a Boolean function which is satisfiable if and only if the layout is routable. The Boolean-based routers have two unique features: One is simultaneous embedding of all nets regardless of net ordering; the other is ability to demonstrate routing infeasibility by proving the unsatisfiability of the generated routing constraint Boolean function. The unsatisfiable subformulas can help the FPGA routing tool to diagnose and eliminate the causes of unroutable. In general, the unsatisfiable subformulas are extracted faster, the tool completes the FPGA routing process more efficiently. Therefore, a fast algorithm of deriving the unsatisfiable subformulas, called the resolution-based local search algorithm [3], is integrated into Boolean-based FPGA routing method. We have compared two optimal algorithms of computing unsatisfiable subformulas, respectively called branch-and-bound algorithm [4] and the resolution-based local search algorithm, on the standard FPGA routing benchmark. The evaluation results show that the local search algorithm strongly outperforms the branch-and-bound algorithm on runtime. It is also shown that the unsatisfiable subformulas can help the tool to quickly diagnose the root causes of unroutability problem and eliminate the fail nets.

The paper is organized as follows. The next section gives the basic definitions and notations of unsatisfiable subformula used throughout the paper. Section 3 surveys the related work on computing unsatisfiable subformulas. Section 4 introduces the principles of the Boolean-based FPGA routing algorithm. Section 5 describes the algorithms of computing the unsatisfiable subformulas. Section 6 shows and analyzes experimental results of two algorithms on the FPGA routing benchmark. Finally, Section 6 concludes this paper.

2. PRELIMINARIES

Resolution is a proof system for CNF (Conjunctive Normal Form) formulas with the following rule:

$$\frac{(L_0 \vee a) \wedge (L_1 \vee \neg a)}{(L_0 \vee L_1)}, \tag{1}$$

where L_0, L_1 are disjunctions of literals. The clauses $(L_0 \vee a)$ and $(L_1 \vee \neg a)$ are the resolving clauses, and $(L_0 \vee L_1)$ is the resolvent. The resolvent of the clauses (a) and $(\neg a)$ is the empty clause ($\perp$). Each application of the resolution rule is called a resolution step. The above resolution step is represented as $((L_0 \vee a) \wedge (L_1 \vee \neg a)) \models (L_0 \vee L_1)$. A sequence of resolution steps, each one uses the result of the previous step or the clauses of the original formula as the resolving clauses of the current step, is called a resolution sequence.

Definition 1. (Boolean Satisfiability) Given a CNF formula $\varphi(V)$, where V is the set of variables, and a Boolean function $F(V)$: $\{0,1\}^n \rightarrow \{0,1\}$, the Boolean satisfiability problem consists of identifying a set of assignments M_v to the variables, such that $F(M_v)=1$, or proving that no such assignment exists.

Lemma 1 A CNF formula φ is unsatisfiable if and only if there exists a finite sequence of resolution steps ending with the empty clause.

It is well-known that a Boolean formula in CNF is unsatisfiable if it is possible to generate an empty clause by resolution sequence from the original clauses. The set of original clauses involved in the derivation of the empty clause is referred to as the unsatisfiable subformula.

Definition 2 (Unsatisfiable subformula) Given a formula φ, ϕ is an unsatisfiable subformula for φ if and only if ϕ is an unsatisfiable formula and $\phi \subseteq \varphi$.

Observe that an unsatisfiable subformula can be defined as any subset, which is infeasible, of the original formula. Consequently, there may exist many different unsatisfiable subformulas, with different number of clauses, for the same problem instance, such that some of these subformulas are subsets of others.

Lemma 2. The set of original clauses involved in the derivation of an empty clause is referred to as the unsatisfiable subformula.

Definition 3 (Minimal Unsatisfiable Subformula) Given an unsatisfiable subformula ϕ for a formula φ, ϕ is a minimal unsatisfiable subformula if and only if removing any clause $\omega \in \phi$ from ϕ implies that $\phi - \{\omega\}$ is satisfiable.

For Boolean formulas in CNF, an unsatisfiable subformula is minimal if it becomes satisfiable whenever any of its clauses is removed. According to the definition, a minimal unsatisfiable subformula has two features: one is unsatisfiable, the other is irreducible, in other words, all of its proper subsets are satisfiable.

Definition 4 (Minimum Unsatisfiable Subformula) Consider a formula φ and the set of all unsatisfiable subformulas for φ: $\{\phi_1, \phi_2, \ldots, \phi_n\}$. Then, $\phi_k \in \{\phi_1, \phi_2, \ldots, \phi_n\}$ is a minimum unsatisfiable subformula iff $\forall \phi_i \in \{\phi_1, \phi_2, \ldots, \phi_n\}$, $1 \le i \le n$: $|\phi_k| \le |\phi_i|$.

According to the definition, a minimum unsatisfiable subformula has the smallest cardinality of all unsatisfiable subsets of a formula. From the above definition, one may conclude that any unsatisfiable formula has at least one minimum unsatisfiable subformula.

We may observed that, the clauses, contained in the intersection of a resolution trace and the original formula, belong to some unsatisfiable subformula. From the lemmas, it is concluded that a refutation proof contains the explanation of infeasibility of the formula. In other words, the causes of unsatisfiability can be derived from the resolution sequence in the sense that removing them will correct the infeasibility. Then we illustrate the process of extracting unsatisfiable subformulas from a Boolean formula according to Lemma 1 and Lemma 2. For example, a CNF formula is

$$\varphi = (\neg a_0) \wedge (a_1) \wedge (a_0 \vee \neg a_1) \wedge (a_0 \vee \neg a_2) \wedge (\neg a_1 \vee a_2) \tag{2}$$

The above formula is refuted by a series of resolution steps ending with the empty clause. Two refutation sequences to affirm the infeasibility of the formula φ are shown as follows:

$$\Gamma_1 = \left\{ \frac{(\neg a_0) \wedge (a_0 \vee \neg a_1)}{(\neg a_1)}, \frac{(\neg a_1) \wedge (a_1)}{\bot} \right\} \tag{3}$$

$$\Gamma_2 = \left\{ \frac{(\neg a_0) \wedge (a_0 \vee \neg a_2)}{(\neg a_2)}, \frac{(a_1) \wedge (\neg a_1 \vee a_2)}{(a_2)}, \frac{(\neg a_2) \wedge (a_2)}{\bot} \right\} \tag{4}$$

From Γ_1, the resolvent $(\neg a_1)$ of the first resolution step serves as one of the resolving clauses of the second step, and the result of the second resolution step is the empty clause. Similarly, the other sequence Γ_2 of resolution steps also arrives at the empty clause. According to Lemma 2, the

original clauses included in the proof of infeasibility belong to the unsatisfiable subformula. More specifically, two unsatisfiable subformulas respectively corresponding to the refutation proofs Γ_1 and Γ_2 are

$$\phi_1 = (\neg a_0) \wedge (a_0 \vee \neg a_1) \wedge (a_1) \quad (5)$$

$$\phi_2 = (\neg a_0) \wedge (a_0 \vee \neg a_2) \wedge (a_1) \wedge (\neg a_1 \vee a_2) \quad (6)$$

In summary, this Boolean formula example demonstrates that the resolution-based local search algorithm to extract the unsatisfiable subformulas is essentially based on Lemma 1 and Lemma 2.

3. BOOLEAN-BASED FPGA ROUTING ALGORITHM

In electronic design automation fields, one of the motivating applications is a Boolean-based FPGA detailed routing formulation on island-style FPGA architecture. The Boolean-based router expresses the routing constraints as a Boolean function which is satisfiable if and only if the layout is routable. The Boolean-based routers have two unique features: One is simultaneous embedding of all nets regardless of net ordering; the other is ability to demonstrate routing infeasibility by proving the unsatisfiability of the generated routing constraint Boolean function.

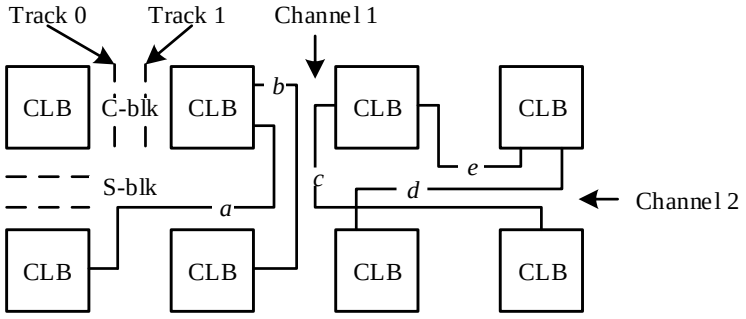


Figure 1. An FPGA routing example

Consider the small FPGA routing problem shown in Figure 1. As depicted in Figure 1, an island-style FPGA consists of a two-dimensional array of Configurable Logic Blocks (CLB), Connection Blocks (C-blk), and Switching Blocks (S-blk). In this figure, there are five nets named *a* through *e* to be routed over an FPGA fabric that has two tracks numbered 0 and 1 in each routing channel. For instance, a net *c* is modeled by two binary variables c_0 and c_1 that indicate its track assignment: $c_i = 1$ expresses that net *C* is assigned to track *i*. With this way of encoding, the routing requirements are now formulated as a set of CNF clauses that fall into one of two categories: Liveness constraints are to ensure that each net is assigned to at least one routing track; Exclusivity constraints are to guarantee that each track is assigned to at most one net.

The resulting set of constraints is listed in Table 1 along with four minimal unsatisfiable subformulas. There are five liveness constraints, one per net, and twelve exclusivity constraints, six for routing channel 1 and six for routing channel 2. The minimal unsatisfiable subformula 1 and minimal unsatisfiable subformula 2 respectively correspond to routing channels 1 and 2. As a means of diagnosing the causes of unroutability, the minimal unsatisfiable subformula 1 pinpoints the conflicting requirements of trying to route three nets *a*, *b* and *c* in a 2-track channel. Similarly, the minimal unsatisfiable subformula 2 indicates the impossibility of three routing nets *c*, *d* and *e* in channel 2. Beyond pointing out a routing channel whose capacity is exceeded, these

minimal unsatisfiable subformulas are able to find a crucial net, namely *c*, which contributes to the unroutability of both channels. This can be located by noting that the *c* variables occur more frequently in these unsatisfiable subformulas than all other variables. This simple example shows that the unsatisfiable subformulas might play a very important role in diagnosing and eliminating the causes of failure.

Table 1. The constraints and minimal unsatisfiable subformulas

Constraints and MUSes		**Expressions in CNF**
Liveness constraints		$L_0=(a_0 \vee a_1)$, $L_1=(b_0 \vee b_1)$, $L_2=(c_0 \vee c_1)$, $L_3=(d_0 \vee d_1)$, $L_4=(e_0 \vee e_1)$
Exclusivity constraints	Channel 1	$E_0=(\neg a_0 \vee \neg b_0)$, $E_1=(\neg a_0 \vee \neg c_0)$, $E_2=(\neg b_0 \vee \neg c_0)$, $E_3=(\neg a_1 \vee \neg c_1)$, $E_4=(\neg a_1 \vee \neg c_1)$, $E_5=(\neg b_1 \vee \neg c_1)$
	Channel 2	$E_6=(\neg c_0 \vee \neg d_0)$, $E_7=(\neg c_0 \vee \neg e_0)$, $E_8=(\neg d_0 \vee \neg e_0)$, $E_9=(\neg c_1 \vee \neg d_1)$, $E_{10}=(\neg c_1 \vee \neg e_1)$, $E_{11}=(\neg d_1 \vee \neg e_1)$
Minimal unsatisfiable subformula 1		$L_0 \vee L_1 \vee L_2 \vee E_0 \vee E_1 \vee E_2 \vee E_3 \vee E_4 \vee E_5$
Minimal unsatisfiable subformula 2		$L_2 \vee L_3 \vee L_4 \vee E_6 \vee E_7 \vee E_8 \vee E_9 \vee E_{10} \vee E_{11}$
Minimal unsatisfiable subformula 3		$L_0 \vee L_1 \vee L_2 \vee L_3 \vee L_4 \vee E_0 \vee E_4 \vee E_5 \vee E_6 \vee E_7 \vee E_{11}$
Minimal unsatisfiable subformula 4		$L_0 \vee L_1 \vee L_2 \vee L_3 \vee L_4 \vee E_1 \vee E_2 \vee E_3 \vee E_8 \vee E_9 \vee E_{10}$

4. ALGORITHMS OF EXTRACTING UNSATISFIABLE SUBFORMULA

There have been many different contributions to research on unsatisfiable subformulas extraction in the last few years, owing to the increasing importance in numerous practical applications. Some research works, based on a relationship between maximal satisfiability and minimal unsatisfiability, have developed some sound techniques for finding a minimum unsatisfiable subformula [4], or all minimal unsatisfiable subformulas [5].

CoreTimmer [6] iterates over each internal node that consumes a large number of clauses and attempts to prove them without these clauses. In [7], the authors presented the algorithms which tracks minimal unsatisfiable subformulas according to the trace of a failed local search run for consistency checking. Two new resolution-based algorithms are proposed in [8]. These algorithms are used to compute a minimal unsatisfiable subformula or, if time-out encountered, a small non-minimal unsatisfiable subformula. Based on these algorithms, seven improvements are proposed in [9], and the experiments have shown the reduction of 55% in run time and 73% in the size of the resulting subformula.

In [10], the authors also proposed two algorithms, one is to optimize the number of calls to a SAT solver, the other is to employ a new technique named recursive model rotation. An improvement to model rotation called eager rotation [11] is integrated in resolution-based minimal unsatisfiable subformulas algorithm. Belov et al. [12] have proposed some techniques to trim CNF formulas using clausal proofs. A new algorithm [13] is to exploit the minimal unsatisfiable subformulas (mus) and minimal correction sets connection in order to compute a single mus and to incrementally compute all muses. An algorithm [14] is proposed to improve over previous methods for finding multiple muses by computing its muses incrementally. The authors [15] aimed to explore the parallelization of partial MUS enumeration. The evaluation results show that the full parallelization of the entire enumeration algorithm scales well.

A technique [16] implements a model rotation paradigm that allows the set of minimal correction subsets to be computed in a heuristically efficient way. The authors [17] proposed a new algorithm for extracting minimal unsatisfiable subformulas and correction sets simultaneously. Liu et al. [18] introduced an algorithm for extracting all MUSes for formulas in the field of

propositional logic and the function-free and equality-free fragment of first-order logic. Luo et al. [19] proposed a method for accelerating the enumeration of MUSes based on inconsistency graph partitioning. A novel algorithm [20] is presented for computing the union of the clauses included in some MUSes, by developing a refined recursive enumeration of MUSes based on powerful pruning techniques.

Bendik et al. [21] firstly approximated MUS counting procedure called AMUSIC, combining the technique of universal hashing with advances in QBF solvers along with a novel usage of union and intersection of MUSes to achieve runtime efficiency. They proposed a novel maximal satisfiable subsets enumeration algorithm called RIME [22]. The experimental results showed that RIME is several times faster than existing tools. In [23], they focused on the enumeration of MUSes, and introduced a domain agnostic tool called MUST. This tool outperforms other existing domain agnostic tools and is even competitive to fully domain specific solutions.

In recent years, the complete methods have made great progress in solving many real life problems including Boolean satisfiability, but they usually cannot scale well owing to the extreme size of the search space. One way to solve the combinatorial explosion problem is to sacrifice completeness, thus some of the best known methods using this incomplete strategy are local search algorithms. In general, the local search strategy starts from an initial solution, which may be randomly or heuristically generated. Then the search moves to a better neighbor according to the objective function, and terminates if the goal is achieved or no better solution can be found. Local search methods are underlying some of the best-performing algorithms for certain types of problem instances, both from an empirical as well as from a theoretical point of view. Consequently, this stochastic strategy is adopted to tackle the problem of finding unsatisfiable subformulas, and in general it has better performance than DPLL-based complete algorithms, especially on 2-SAT and 3-SAT problem instances. According to the rules described in Section 3, the FPGA detailed routing problem can be translated to the Boolean formulas with a number of 2-literal and 3-literal clauses. Therefore, we integrated a resolution-based local search algorithm [3] into the Boolean-based FPGA detailed routing method. The local search algorithm to extract the unsatisfiable subformulas from the Boolean formulas is based on the Lemma 1 and Lemma 2 introduced in Section 2.

5. EXPERIMENTAL RESULTS AND ANALYSIS

To experimentally evaluate the efficiency between two algorithms of computing unsatisfiable subformulas on FPGA routing, we have selected a suit of typical paradigm of the Boolean-based FPGA routing problem. As described above, the FPGA routing benchmark suite is derived from the problem of Boolean-based FPGA detailed routing formulation on island-style FPGA architecture, which is one of the typical applications for unsatisfiable subformulas. The Boolean-based router expresses the routing constraints as a CNF formula which is unsatisfiable if and only if the layout is unroutable. The benchmark includes 10 instances. We have compared the resolution-based local search algorithm with the branch-and-bound algorithm, which is the fastest tool to compute an exactly minimum unsatisfiable subformula. The experiments were conducted on a 2.5 GHz Athlon*2 machine having 2 GB memory and running the Linux operating system. The limit time was 1800 seconds.

The experimental results of the branch-and-bound algorithm and the resolution-based local search algorithm on the 10 formulas of FPGA routing problem are listed in Table 2. Table 2 shows the number of variables (vars) and the number of clauses (clas) for each Boolean formula. The fourth column gives the total number of minimal unsatisfiable subformulas contained in every formula (MUSes). For generating all minimal unsatisfiable subsets we use the CAMUS algorithm [5]. However there are five instances which run out of time, and we mark them with time out in the

table. Table 2 also provides the runtime in seconds (BaBA time) of branch-and-bound algorithm, and the number of clauses (BaBA size) in the derived minimum unsatisfiable subformula. The next three columns present the runtime of the resolution-based local search algorithm in seconds (RbLSA time), and the memory consumption in MB (RbLSA mem), and the size of the unsatisfiable subformula (RbLSA size). The last column shows the percentage of clauses in the unsatisfiable subformula occupying the original formula (Per %).

Table 2. Experimental results on FPGA routing benchmark

Benchmarks	**vars**	**clas**	**MUSes No.**	**BaBA**		**RbLSA**			**Per** (%)
				time	size	time	mem	size	
fpga_routing1	10	17	4	<0.001	9	<0.001	0.12	9	52.9
fpga_routing2	14	25	11	0.02	9	<0.001	0.29	9	36.0
fpga_routing3	18	33	26	0.18	9	0.08	0.63	9	27.3
fpga_routing4	22	41	57	2.63	9	1.2	1.15	9	21.9
fpga_routing5	26	49	120	62.7	9	27.6	3.25	9	18.4
fpga_routing6	30	57	time out	time out		182.5	10.6	9	15.8
fpga_routing7	34	65	time out	time out		358.0	17.5	9	13.8
fpga_routing8	38	73	time out	time out		617.0	23.4	9	12.3
fpga_routing9	42	81	time out	time out		1040.1	28.0	9	11.1
fpga_routing10	46	89	time out	time out		1690.0	36.5	9	10.1

From Table 2, we may observe the following. The resolution-based local search algorithm outperform the branch-and-bound algorithm for all of 10 formulas. For the instances of fpga_routing6 through fpga_routing10, the branch-and-bound algorithm failed to extract the unsatisfiable subformula within the timeout, but the resolution-based local search algorithm succeeded in obtaining it. Moreover, the local search algorithms find the minimum unsatisfiable subformula for each formula of the FPGA routing benchmark suite.

Therefore, the following conclusion can be reached that the runtime of resolution-based local search algorithm strongly exceeds the branch-and-bound algorithm, although the local search algorithm cannot guarantee obtaining the exact minimum unsatisfiable subformulas. The causes include three aspects: The first is that the function of deriving unsatisfiable subformula is coupled tightly with the satisfiability checking procedure of the formula. While the resolution is proceeding, the refutation is recorded, and the parsing tree is constructed simultaneously, then the unsatisfiable subformula is computed very efficiently. The second reason is that the decision of satisfiability is implemented simply and performs many more moves per second. The third cause is there are many powerful heuristics in the local search algorithm, especially for unit clauses and binary clauses. The formulas translated by the Boolean-based FPGA routing problem contain many 2-literal clauses.

From the last column of Table 2, we may observe the following. For 10 formulas, the percentage of clauses in the unsatisfiable subformulas is quite small, in most cases from 10% to 30%. In general, the unsatisfiable subformulas can provide more succinct explanations of infeasibility, and is very valuable for a variety of practical applications. In the automatic routing tool of FPGA chips, the unsatisfiable subformulas can help the tool to quickly diagnose the root causes of unroutability problem, and eliminate the fail nets, and then finish the FPGA routing process much more efficiently.

6. CONCLUSIONS

An unsatisfiable subformula generally provides the most accurate explanation of infeasibility and is valuable for FPGA routing application. The automatic routing process of FPGA devices is very difficult and time-consuming. Therefore, two algorithms of deriving unsatisfiable subformulas, respectively called the resolution-based local search algorithm and the branch-and-bound algorithm, are employed to accelerate the FPGA routing tool. The standard FPGA routing problem instances are adopted as the benchmark. The results show that the resolution-based local search algorithm outperforms the branch-and-bound algorithm on runtime. We have also analyzed that the unsatisfiable subformulas play a very important role in automatic routing tool of FPGA chips.

ACKNOWLEDGEMENTS

The authors would like to thank all peer reviewers for their valuable comments and suggestions. This work is supported by the National Key Research and Development Program of China under grant No. 2018YFB0204301, and the National Natural Science Foundation of China under grant No. 62072464 and U19A2062.

REFERENCES

[1] M.W. Moskewicz, C.F. Madigan, Y. Zhao, L. Zhang & S. Malik, (2001) "Chaff: engineering an efficient SAT solver", *Proc. of the 38th Design Automation Conf.*, Las Vegas: ACM Press, pp530-535.

[2] N. Een & N. Sorensson, (2003) "An extensible SAT-solver", *Proc. of the 6th Intl. Conf. on Theory and Applications of Satisfiability Testing*, LNCS 2919, Heidelberg: Springer-Verlag, pp502-518.

[3] J. Zhang , S. Shen , & S. Li, (2009) "Tracking Unsatisfiable Subformulas from Reduced Refutation Proof", *Journal of Software*, Vol. 4, No. 1, pp42-49.

[4] M.H. Liffiton, M.N. Mneimneh, I. Lynce, Z.S. Andraus, J.P. Marques-Silva, & K.A. Sakallah, (2009) "A branch and bound algorithm for extracting smallest minimal unsatisfiable formulas", *Constraints*, Vol. 14, No. 4, pp415-442.

[5] M.H. Liffiton, & K.A. Sakallah, (2008) "Algorithms for computing minimal unsatisfiable subsets of constraints", *Journal of Automated Reasoning*, Vol. 40, pp1-30.

[6] R. Gershman, M. Koifman, & O. Strichman, (2008) "An approach for extracting a small unsatisfiable core", *Formal Methods in System Design*, Vol. 33, No. 1, pp1-27.

[7] E. Gregoire, B. Mazuer, & C. Piette, (2009) "Using local search to find MSSes and MUSes", *European Journal of Operational Research*, Vol. 199, No. 3, pp640-646.

[8] A. Nadel, (2010) "Boosting minimal unsatisfiable core extraction", *Proc. of 10th Intl. Conf. Formal Methods in Computer Aided Design*, pp221-229.

[9] V. Ryvchin & O. Strichman, (2011) "Faster extraction of high-level minimal unsatisfiable cores", *Proc. of 14th Intl. Conf. Theory and Applications of Satisfiability Testing*, pp174-187.

[10] A. Belov, I. Lynce, & J. Marques-Silva, (2012) "Towards efficient MUS extraction", *Journal AI Communications*, Vol. 25, No. 2, pp97-116.

[11] A. Nadel, V. Ryvchin, & O. Strichman, (2013) "Efficient MUS extraction with resolution", *Proc. of 13th Intl. Conf. Formal Methods in Computer Aided Design*, pp197-200.

[12] Anton Belov, Marijn Heule, & Joao Marques-Silva. (2014) "MUS extraction using clausal proofs", *Proc. of 17th Intl. Conf. on Theory and Applications of Satisfiability Testing*, pp48-57.

[13] F. Bacchus & G. Katsirelos, (2015) "Using minimal correction sets to more efficiently compute minimal unsatisfiable sets", *Proc. of the 27th Intl. Conf. on Computer Aided Verification*, pp70-86.

[14] F. Bacchus & G. Katsirelos, (2016) "Finding a collection of MUSes incrementally", *Proc. of 13th Intl. Conf. on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pp35-44.

[15] W. Zhao & M. H. Liffiton, (2016) "Parallelizing partial MUS enumeration", *Proc. of IEEE 28th Intl. Conf. on Tools with Artificial Intelligence*, pp464-471.

[16] E. Gregoire & Y. Izza, (2018) "Boosting MCSes Enumeration", *Proc. of the 27th Intl. Joint Conf. on Artificial Intelligence*, pp1309-1315.
[17] N. Narodytska, N. Bjorner, M.C. Marinescu, & M. Sagiv, (2018) "Core-guided minimal correction set and core enumeration", *Proc. of the 27th Intl. Joint Conf. on Artificial Intelligence*, pp1353-1361.
[18] S. Liu & J. Luo, (2018) "FMUS2: An efficient algorithm to compute minimal unsatisfiable subsets", *Proc. of 2018 Intl. Conf. on Artificial Intelligence and Symbolic Computation*, pp104-118.
[19] J. Luo & S. Liu, (2019) "Accelerating MUS enumeration by inconsistency graph partitioning", *Science China Information Sciences*, Vol. 62, pp212104.
[20] C. Mencia, O. Kullmann, A. Ignatiev, & J. Marques-Silva, (2019) "On computing the union of MUSes", *Proc. of 22nd Intl. Conf. on Theory and Applications of Satisfiability Testing*, pp211-221.
[21] J. Bendik & M.S. Kuldeep, (2020) "Approximate counting of minimal unsatisfiable subsets," *Proc. of the 32nd Intl. Conf. on Computer Aided Verification*, pp1-23.
[22] J. Bendik & C. Ivana, (2020) "Rotation based MSS/MCS enumeration", *Proc. of the 23rd Intl. Conf. on Logic for Programming, Artificial Intelligence and Reasoning*, pp120-137.
[23] J. Bendik & C. Ivana, (2020) "MUST: minimal unsatisfiable subsets enumeration tool", *Proc. of the 26th Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, pp135-152.

AUTHORS

Zhang Jianmin was born in China in 1979. He is an associate professor in computer science at National University of Defense Technology. He received Ph. D. degree in computer science from National University of Defense Technology, Changsha, China, in 2008. His major fields of study include SAT-based formal verification and FPGA acceleration algorithms.

Li Tiejun was born in China in 1977. He is a professor in computer science at National University of Defense Technology. He received Ph. D. degree in computer science from National University of Defense Technology, Changsha, China, in 2005. His research interests include functional verification and high performance CPU design.

Ma Kefan was born in China in 1985. He is a Ph.D in computer science at National University of Defense Technology. He received Ph. D. degree in computer science from National University of Defense Technology, Changsha, China, in 2019. His research interests include FPGA accelerated SAT solver and formal verification.

INCREMENTAL AUTOMATIC CORRECTION FOR DIGITAL VLSI CIRCUITS

Lamya Gaber[1], Aziza I. Hussein[2] and Mohammed Moness[1]

[1]Department of Computers and Systems Engineering,
Minia University, Minia, Egypt
[2]Department of Electrical and Computer Engineering,
Effat University, Jeddah, KSA

ABSTRACT

The impact of the recent exponential increase in complexity of digital VLSI circuits has heavily affected verification methodologies. Many advances toward verification and debugging techniques of digital VLSI circuits have relied on Computer Aided Design (CAD). Existing techniques are highly dependent on specialized test patterns with specific numbers increased by the rising complexity of VLSI circuits. A second problem arises in the form of large sizes of injecting circuits for correction and large number of SAT solver calls with a negative impact on the resultant running time. Three goals arise: first, diminishing dependence on a given test pattern by incrementally generating compact test patterns corresponding to design errors during the rectification process. Second, to reduce the size of in-circuit mutation circuit for error-fixing process. Finally, distribution of test patterns can be performed in parallel with a positive impact on digital VLSI circuits with large numbers of inputs and outputs. The experimental results illustrate that the proposed incremental correction algorithm can fix design bugs of type gate replacements in several digital VLSI circuits from ISCAS'85 with high speed and full accuracy. The speed of proposed Auto-correction mechanism outperforms the latest existing methods around 4.8x using ISCAS'85 benchmarks. The parallel distribution of test patterns on digital VLSI circuits during generating new compact test patterns achieves speed around 1.2x compared to latest methods.

KEYWORDS

Auto-correction, ATPG, Fault detection, Verification.

1. INTRODUCTION

Recently, integrated circuits (ICs) known as microchips have become omnipresent in all aspects of our lives from automobiles to smartphones. Their impact on our lives has gradually evolved to be the most crucial chips as they are considered the heart of modern computing devices. These compact ICs contain billions of transistors which perform billions of computations compared with vacuum tubes or single transistor preceded them. Therefore, the complexity of ICs design flow has increased over the last three decades with a highly increasing list of requirements in terms of functional, power, performance and physical size. Therefore, the IC design is not a push-button process. Instead, it is a highly complicated task as it needs a full understanding of the IC restrictions, specifications and all the required EDA tools. Several procedures should be considered in the IC design flow which are highly error-prone as many translations should be performed on various levels of IC representations (e.g. RTL description and gate-level netlist). Therefore, the uselessness rate of silicon slabs has been increased by increasing the number of undetected errors that might propagate from the RTL description to the fabricated chip. Hence,

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 83-93, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101508

several recent contributions of developing verification and testing methodologies have been emerged to avoid the consumed cost and time for achieving the desired design with its required specifications. And by the time, the procedure of verifying, testing, debugging and correcting digital circuits have been contributing up to 70% of the entire time for designing the aspired IC [1]. On top of that, the functional verification has become the most significant issue of the IC design as the fact proved in [2] that the main reason for unsuccessful Application Specific Integrated Circuits (ASIC), around 60% of them, is the existence of undetected functional errors (Not because of high power consumptions and timing issues). Consequently, up to 46% of the whole time of IC design is consumed in the functional verification step [3]. Therefore, three main processes are implemented following each conversion from high level to more detailed level of abstraction in the pre-silicon design, termed Verification, Debugging and Correction. These processes are defined as follows:

Definition 1: the process of searching for inconsistencies between two levels of circuit abstraction in the pre-silicon design is called "*Verification*".

Definition 2: In case of functional inconsistencies intended for diagnosing and detecting potential bug locations in an erroneous circuit is called "*Debugging*". Therefore, it is also often termed *Bug Localization.*

Definition 3: the phase responsible for modifying components causing errors discovered by debuggers is known as "*Correction*", so it can rectify the desired circuit to behave in its intended manner. The output of this phase is the correct design that can match the desired specification (the previous abstraction).

By increasing the design size, the uselessness rate of silicon slabs has been grown by increasing the number of undetected errors that might propagate from the RTL description to the fabricated chip. In addition, the proportion of losing time and increasing the non-recurring engineering (NRE) cost is gradually extended by taking the blind assuming of "synthesis tools make no mistakes". Therefore, several recent contributions of developing verification/testing, debugging and correction methodologies have been emerged to avoid the cost and time consumed during achieving the desired design that should be matched with the behavioral specifications. And by the time, the procedures of verifying/testing, debugging and correcting digital circuits have been contributing up to 70% of the entire time for designing the aspired IC [1]. As the growth of the complexity of digital integrated circuits and reduced time-to-market budget, debugging mechanisms with auto-correction in a digital design flow have become more and more challenging task, contributing on average 60% of the verification process in case of observing some failures in a given design. Although, many contributions have been devoted on the verification and debugging steps, few efforts have been dedicated on fixing the detected errors which are left to the creativities of the designers to manually correct them. Also, studies report that the main reasons for failures existed in a large portion of first tap-out are simply functional errors which could escape from the earlier correction step [1]. So, the correction procedure is considered the expensive and complicated task because other phases of IC design directly depend on "how efficiently the correction procedure works". In this paper, we focus on improving the process of auto-correction using partial test patterns in the gate-level representation as it is a core factor of reducing the ad-hoc manual effort, time and possibilities of undetected bugs in the manufactured chip.

Gate-level auto-correction problem has been addressed by many researches proposing different algorithms for auto-debugging and/or suggesting the possible modification for achieving the correct design. Also, most of error correction techniques are focused on gate-level netlist because it can give the designers different reasons of occurring bugs such as incorrect individual

connection wires or incorrect individual gate types that can NOT be provided in a higher level of abstraction. The conceivable idea of fixing errors by re-synthesis is not an efficient way for most of today's digital circuits as errors can be caused by the synthesis tool itself. Also, some physical optimization that should be previously performed might Not be validated.

In this paper, we proposed a new efficient automatic correction (ACM-CTV) method using partial test patterns to generate both a corrected digital circuit and compact test patterns by the following advantages:

1- The injecting circuit is reduced to be 3-to-1 multiplexers instead of 6-to-1 multiplexers as it used in [4].
2- One of two solutions that are generated in every iteration is exploited to reduce the search space (*Find_OneSol* Algorithm).
3- Parallel distribution of test patterns (*GPU_UC* procedure) is performed that can give a high performance in case of large number of inputs and outputs of faulty digital circuit.

2. RELATED WORK

Recently, many extensive problems in digital very-large-scale-integration problems can be addressed within a Boolean satisfiability framework as it offers various solutions. In [5-7], the main idea is mapping the diagnosis problem to SAT problem in a compatible form (as CNF) to be solved using SAT solvers. These SAT-based methods outperform the traditional methods as it is implemented with high performance.

In [8], the bug localization problem is addressed by replacing every possibly buggy gate with 2-to-1 multiplexers, then converting to a conjunction normal form (CNF) formula. After that, a new constraint is added to the CNF instance to represent the initial test patterns and passing all the final formula to SAT solver to return a satisfiability model of CNF instance that can detect the exact locations.

Authors in [6] have proposed a debugging method of gate-level circuits using partial maximum satisfiability (MAX-SAT) for detecting spatial and temporal bug locations.

In [9], authors proposed a bug localization of gate-level circuits using level Quantified Boolean Formula (QBF) that is equivalent process to repetitive calls of normal SAT solvers. All the previous methods attempt to exactly detect logic bugs in order to finally rectify the digital circuit to satisfy specifications.

On the other hand, the auto-correction methods have been addressed using both the bug locations and test patterns for the corresponding logic bugs. In [10], a mutation-based correction mechanism is proposed by adding 6-to-1 multiplexers into the place of every potential bug to quickly check all possible gates instead of the faulty one. In [11], the rectification process is employed by injecting 3-to-1 multiplexers instead of 6-1 multiplexers in order to reduce the CNF instance passed to SAT engine for shrinking the search space. The previous two methods are based on knowing exact bug locations and test patterns of the correct digital circuit in order to automatically correct circuits. In [12], a new correction method is proposed based on [10] in order to incrementally correcting a given circuit by generating new test patterns. The generation of compact test patterns is performed by finding two different solutions in every iteration, so it can guarantee that the returned rectified circuit is completely corrected for all test patterns.

In [4], authors improved the previous automatic correction method by exploiting the two solutions generated in every iteration not only in finding a new test pattern but also, in shrinking the search space for the next iterations. Therefore, the proposed rectification (ACM-CTV) mechanism exploit advantages of methods proposed in [4, 11] in order to reducing search space and running time for correction.

3. BACKGROUND

3.1. Design Errors

A component among a design implementation such as RTL or gate-level representation or behavioural specifications such as testbench and assertions that produces functional inconsistencies between the two representation is known as an error or a bug. Therefore, failures (defined in Definition 4) can be occurred in design implementation or behavioral specification as a result of bugs or errors. There are different categories of bugs or errors that can be found in digital VLSI circuits but they can be divided into three main types: design errors, verification errors and manufacturing errors. Design errors also can be divided into: functional errors and electrical errors or circuit bugs. Functional errors are the functional inconsistences (or functional mismatches) in a design implementation as a result of incorrect wire connections or functional misbehavior of some gate elements. The main reason of occurring design errors is usually a designer interference during a synthesis phase in order to reach to a specific level of a system optimization. On the other hand, any bugs occurred in a behavioral specification is known as verification errors.

Definition 4: If the same primary input stimulus is applied to observe IC design with the same initial condition and at one or more observation points, there is an inconsistency between a design implementation (RTL abstraction) and its specification, it can be called as a *failure*.

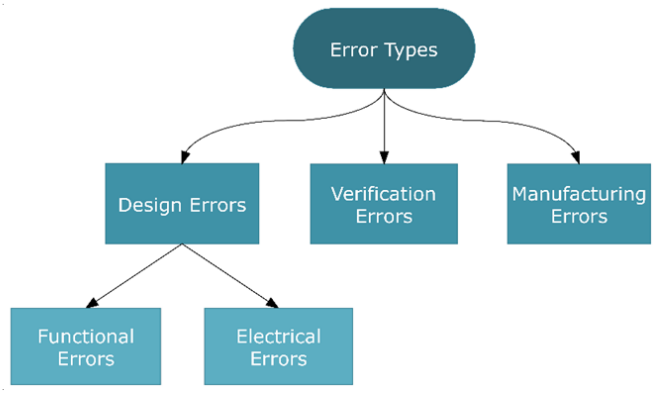


Figure. 1. Main Types of Errors or Bugs.

3.2. Bug Localization and Auto-Correction processes

In Bug localization process, counterexamples generated in logic verification phase are exploited to quickly detect potential locations of observed bugs. The efficient mechanism is identifying fault candidates by adding extra logic to faulty circuit and converting to a suitable formula to be handled by recent SAT solvers. Therefore, most debuggers take test patterns (logic assignments of inputs and their expected output of circuits) and buggy digital circuit then passed to four main stages as shown in figure 2 to finally detect fault candidates. The CNF formula [13, 14] of the

complete design can be easily refined and replicated to every test pattern and every time frame then passing to SAT solver to find error suspects

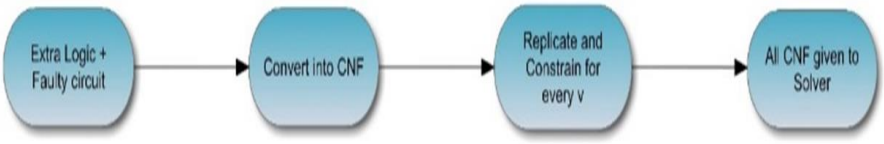
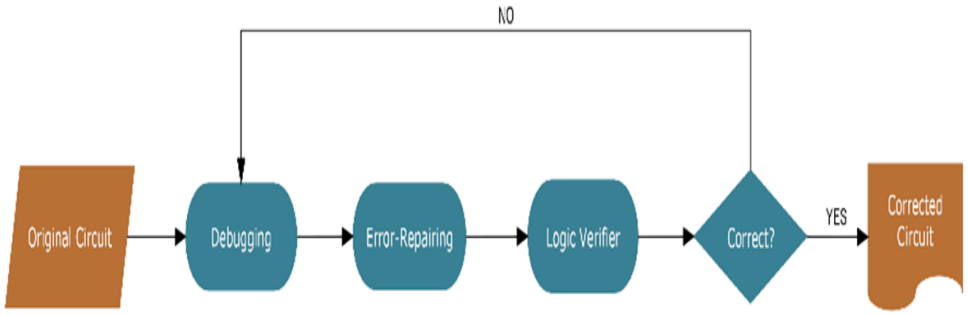


Figure 2. Main Steps of SAT-based Debugging Approach

In [15], a diagnosis technique is proposed to find potential bug locations using minimal correction subsets of UNSAT formula represented a faulty circuit. Also, the reasons of errors can be generated as a minimal unsatisfiable formulas (MUS) which can give designers which clauses combined together and produce the logic fault.

In most SAT-based auto-correction routines, the faulty circuit is automatically modified in order to be satisfy the given test pattern. Therefore, the accuracy and number of test patterns give to rectification algorithm have a high impact on the performance and accuracy of the rectified circuit. Therefore, the algorithms of auto-correction are classified into two main categories. Figure 3 demonstrates the main differences between those two categories. The first group of approaches as shown in figure (3.a) [16] focuses on the accuracy of corrected circuit by combining the debugger and correction algorithms together in order to achieve the equivalence between the actual circuit and the golden test patterns (as a black box of specification). Some techniques can be utilized to improve these error-fixing categories by abstracting and refining processes [17] for reducing the repetitive checking process. But these methods are not practical in a large-sized digital circuit as it has high consumed time and low resolution.

The second category is shown in figure 3.b. In this mechanism, the repetitive calling of logic debuggers and verifiers can be dispensed by fixing upon reduced counterexamples and using the original gate-level circuits [18]. Therefore, the performance of these techniques for rectifying design errors is improved but the accuracy is dependent on how many test patterns utilized.



a. First Mechanism

b. Second Mechanism

Figure 3. Categories of Debugging and Correction strategies.

4. THE PROPOSED RECTIFICATION ALGORITHM

In this section, the proposed automatic correction algorithm with generating compact test patterns (ACM-CTV) is described in more details.

Algorithm 1 illustrates the pseudo code of the proposed algorithm for generating both the Correct Circuit (CC) and Compact Test Vectors (CTV). The main inputs of this algorithm are:

1- Buggy circuit.
2- Buggy location.
3- Specification of the expected correct circuit.
4- Initial Test pattern (always one test pattern)

First of all, the Boolean values of inputs and outputs of buggy component should be defined. So, if it's one of the circuit's inputs and outputs, there is no need to call SAT solver. Otherwise, the buggy circuit with the initial test pattern are passed to SAT solver in CNF form in order to detect the Boolean values of inputs and outputs of the faulty location. Defining these assignments have a great impact on reducing the size of search space for generating the correct circuit and the compact TV. This process performed in lines 1-9 at Algorithm 1. Next, If the faulty gates have one input so the faulty circuit is enriched by 2-1 MUX. Otherwise, the faulty circuit is injected with 3-1 MUX. The inputs of 3-1 MUX are determined according to the detected Boolean values of inputs and output of faulty gate. These steps for generating CNF formula of MUX is performed in lines 10-16. After that lines 17-38, the steps for expecting the correct circuit with compact Tv are performed using the enriched circuit and the initial Tv. This process is a loop of generating two possible solutions (sol1, sol2), mitering between them, generating a new TV, pushing it into compact TV and retaining the possible correct solution as sol1 between two solutions for the next iteration. The loop stops when there is no second solution. Therefore, the sol1 is the correct component and the generated TV is the compact TV.

Algorithm 1: Pseudo code of Proposed ACM-CTV Method

Input: ECA_CTV ($Specs, CNF_b, BUGloc, Init_{tv}$):
Output: $CNF_c, Tv_{compact}$

1. $(TV_{Compact}, iter) \leftarrow (TV_{Init}, 1)$
2. $(Err_{gate}, Err_{Inp}) \leftarrow Find\_ErrorG(CNF_b, BUGloc)$
3. $(Prt, count) \leftarrow Check\_IO(Err_{gate})$
4. **if** $(Prt == 1 \;\&\&\; count = 2)$ **then**
5. $(I, O_{Correct}) \leftarrow Extract\_IOvalues(Init_{tv})$
6. **else**
7. $(I, O_{fault}) \leftarrow SAT\_Solver(CNF_b, Init\_tv)$
8. $O_{Correct} \leftarrow fliping\_func(O_{fault})$
9. **end-if**
10. **if** $(count == 1)$ **then** // NOT, BUFF or DFF gates
11. $MUX\_eq \leftarrow GEN\_2To1MUX(Err_{Inp})$
12. **else**
13. $PossMat \leftarrow Produce_{3DMatrix}()$
14. $Poss\_Gates \leftarrow Extract\_3Poss(I, O_{correct}, PossMat)$
15. $MUX\_eq \leftarrow Produce\_3To1\_MUX(Poss\_Gates)$
16. **end-if**
17. $CNF_e \leftarrow Enriched\_Func(CNF_b, MUX\_eq)$
18. $(cir1, sol1) \leftarrow GEN\_PossCorr(Poss\_Gates[0])$
19. **do**
20. **if** $(iter = 1)$ **then**
21. $(cir2, sol2) \leftarrow GEN\_PossCorr(Poss\_Gates[iter])$
22. **else**
23. $(cir2, sol2) \leftarrow Find\_OneSol(CNF_e, Sol1, Tv_{compact})$
24. **end-if**
25. $TV_{New} \leftarrow GEN\_Newtv(cir1, cir2, Specs)$
26. **if** $(Tv_{New} \neq \emptyset \;\&\&\; Tv_{New} \nexists Tv_{Compact})$ **then**
27. $Tv_{compact} \leftarrow Tv_{compact} \cup Tv_{New}$
28. **if** $(whichOne(cir1, Tv_{New}) == 0)$ **then** // cir1 is not compatible with Tv_{New}
29. $(cir1, sol1) \leftarrow (cir2, sol2)$
30. **end-if**
31. **else**
32. $CNF_e \leftarrow Block\_sol2(sol2)$
33. **end-if**
34. $iter++$
35. **while** $(sol2 = \emptyset)$
36. **end-do-while**
37. $CNF_c \leftarrow cir1$
38. ***return*** CNF_c

Algorithm 2 illustrates the pseudo code of the procedure of $Find\_OneSol$. This main goal of this process is to detect a probable second corrected circuit that can satisfy the generated compact test patterns and can be another solution than sol1. This process can be performed by duplicating the enriched circuit according to the current number (n) of compact test patterns then adding new constraints for:

1- representing compact test patterns
2- blocking solution 1.

Therefore, the final CNF instance can be a combination of duplicated circuits

$CNF_d = \sum_{i=0}^{n} CNF_e$, blocked clause $(cl_{blocked})$ and unit clauses for $Tv_{compact}$ as shown in equation 1, where :$n = Num\ of\ Tv_{compact}$

$$\varphi = cl_{blocked} \vee \sum_{i=0}^{n} CNF_e \vee UC_i$$

After creating equation 1, the solver is called by this instance to find a second solution (Sol2), the assigned values of MUX' selectors that can be used to produce cir2 which is used to find a new test pattern if it exists (line 25 in algorithm 1).

Algorithm 2: Find_OneSol Method

Input: $Find\_OneSol\ (CNF_e, Sol1, Tv_{compact})$:
Output: $cir, Tv_{compact}$

1. $CNF_d \leftarrow DUP\_Func(CNF_e, Err_{Inp})$
2. **for** $\forall tv_i \in Tv_{compact}$ **do**
3. $UC_i \leftarrow GPU\_UC(IMPd, tv_i)$
4. $CNF_d \leftarrow CombineCNF(CNF_d, UC_i)$
5. **end-for**
6. **if** $(Sol1 \neq \emptyset)$ **then**
7. $CNF_d \leftarrow Block\_sol1(CNF_d, sol1)$
8. **end-if**
9. $(st, \mu) \leftarrow Parallel\_Solver(CNF_d)$
10. **if** (st) **then**
11. $sol \leftarrow Extract\_Sol(\mu)$
12. **end-if**
13. $cir \leftarrow Gen\_Posscir\ (CNF_e, sol)$
14. ***return*** cir

Algorithm 3 illustrates the pseudo code of GEN_NewTv method that is used to generate new compact test pattern during searching for the correct circuit. This procedure has three main inputs:

1- Specification of the expected correct circuit.
2- The probable first circuit.
3- The probable second circuit.

This procedure is dependent on finding the Boolean values of inputs that can represent a difference between outputs of cir1 and cir2. For finding these values, a CNF instance represented a miter circuit between cir1 and cir2 is performed using $GEN_{miter}(cir1, cir2)$ (line 2). The miter circuit is nothing but XORs between the outputs of the circuit and OR between all XORs. After that, a new constraint is added to CNF_{mit} in order to reduce the search space that make the output of miter circuit as 1 (line 3). After that, this instance is passed to SAT solver. If the CNF formula is satisfied, the model returned from solver can be used to find the assigned values of inputs (Inp_{Tv})(line 6). Then, the expected correct output (Out_{Tv}) corresponding to Inp_{Tv} is generated using specification which can be combined with Inp_{Tv} (line 7) in one CNF formula and passed to SAT solver (line 8). Otherwise ($st = 0$), there is no new test pattern $Tv_{New} = \emptyset$. This means that Sol2 is a spurious solution (line 31-33 in algorithm1) that should be blocked in the enriched circuit before going to the next iteration in algorithm 1.

Algorithm 3: *GEN_NewTv* Method

Input: $GEN\_NewTv\ (cir1, cir2, Specs)$:
Output: Tv_{New}

1. $Tv_{New} \leftarrow \emptyset$
2. $CNF_{miter} \leftarrow GEN_{miter}(cir1, cir2)$
3. $CNF_{miter} \leftarrow Add\_Constraint(CNF_{miter}. output)$
4. $(st, \mu) \leftarrow Parallel\_Solver(CNF_{miter})$
5. **if** (st) **then**
6. $Inp_{Tv} \leftarrow Extract\_Inputs(\mu)$
7. $CNF_{specs} \leftarrow GEN\_fun(Specs, Inp_{Tv})$
8. $(st, \mu) \leftarrow Parallel\_Solver(CNF_{specs})$
9. $Out_{Tv} \leftarrow Extract\_Outputs(\mu)$
10. $Tv_{New} \leftarrow (Inp_{Tv}, Out_{Tv})$
11. **end-if**
12. ***return*** Tv_{New}

5. PERFORMANCE EVALUATION

In this section, a comparison between our proposed ACM-CTV algorithm with serial and parallel distribution and the previous proposed algorithm in [4] in terms of running time is proposed in table 1. The implementation is performed using ISCAS'85 benchmark [19] of SAT CNF instances (6 combinational circuits). Two algorithms were implemented in C++ and executed on Intel core i7-4510U working at 2.6 GHz with 8 GB system memory. As it proposed in compared algorithm in [4], every bug has been injected randomly in the Verilog module of a given circuit as a logic design error. Also, the SAT instance of every faulty digital circuit of ISCAS'85 is generated using a serial version of SAT encoder proposed in [13]. Also, the solving process is performed using Parallel CUD@SAT DPLL engine in two algorithms. Parallel subroutines are implemented in CUDA C and executed on NIVIDIA Geforce.

Table 1 illustrates the consumed time in seconds of previous correction process proposed in [4] and the proposed ACM-CTV correction algorithm. After analysis, the proposed ACM-CTV algorithm can rectify faulty digital circuit with full accuracy by generating compact test patterns during correction process in order to guarantee that there is no new test pattern can prove inconsistence. The proposed correction algorithm for single design errors in digital VLSI circuits delivers about 4.8x average speed compared to the latest existed correction method proposed in [4]. Also, the parallel version of test pattern distribution has a good benefit in case of using digital circuit with large ports as c432. Therefore, the maximum speed of parallel distribution of test patterns are 1.2x comparing to serial distribution proposed in [4].

Table 1. Comparison between running time (s) of previous Correction algorithm and proposed ACM-CTV algorithm.

CNF Type	#gates	#Inputs	#outputs	Time (s) of Correction Algorithm in [4]	Time(ms) of proposed ACM-CTV with serial dis.	Time (ms) of proposed ACM-CTV with parallel dis.
C17	6	5	2	0.420	0.018	0.469
C432	160	36	7	3.127	1.668	1.529
C499	202	41	32	3.195	2.654	3.038
C880	383	60	26	5.931	5.953	5.537
C1908	880	33	25	17.683	15.698	15.843
C3540	1699	50	22	40.389	31.356	34.445

6. CONCLUSIONS

In this paper, we propose an incremental auto-correction algorithm with generating compact test patterns. The main advantages of the proposed ACM-CTV algorithm are avoiding the dependency of the given test patterns by incrementally generating compact patterns and reducing the search space by injecting small size of in-circuit mutation (MUX 3x1 instead of MUX 6x1). By combining two enhanced methods, the proposed algorithm significantly outperforms the previous algorithm in [4] in terms of run time, delivering 4.8x average speed. Also, it shrinks the search space using small size of injected circuit and one test pattern in case of single faults. Therefore, the shrinking rate of search space is 6x compared to previous methods in [4, 11].

REFERENCES

[1] P. Rashinkar, P. Paterson, and L. Singh, *System-on-a-chip verification: methodology and techniques*: Springer Science & Business Media, 2007.
[2] J. Jaeger, "Virtually every ASIC ends up an FPGA," *EE Times*, 2007.
[3] N. Eén and A. Biere, "Effective preprocessing in SAT through variable and clause elimination," in *International conference on theory and applications of satisfiability testing*, 2005, pp. 61-75.
[4] B. Alizadeh and Y. Abadi, "Incremental SAT-based Correction of Gate Level Circuits by Reusing Partially Corrected Circuits," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2020.
[5] A. Suelflow, G. Fey, R. Bloem, and R. Drechsler, "Using unsatisfiable cores to debug multiple design errors," in *Proceedings of the 18th ACM Great Lakes symposium on VLSI*, 2008, pp. 77-82.
[6] Y. Chen, S. Safarpour, J. Marques-Silva, and A. Veneris, "Automated design debugging with maximum satisfiability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, pp. 1804-1817, 2010.
[7] L. G. Ali, A. I. Hussein, and H. M. Ali, "An efficient computation of minimal correction subformulas for SAT-based ATPG of digital circuits," in *Computer Engineering and Systems (ICCES), 2017 12th International Conference on*, 2017, pp. 383-389.
[8] A. Smith, A. Veneris, M. F. Ali, and A. Viglas, "Fault diagnosis and logic debugging using Boolean satisfiability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, pp. 1606-1621, 2005.
[9] K.-H. Chang, I. Wagner, I. Markov, and V. Bertacco, "Automatic error diagnosis and correction for RTL designs," ed: Google Patents, 2013.
[10] P. Behnam and B. Alizadeh, "In-circuit mutation-based automatic correction of certain design errors using SAT mechanisms," in *2015 IEEE 24th Asian Test Symposium (ATS)*, 2015, pp. 199-204.
[11] L. Gaber, A. I. Hussein, and M. Moness, "Improved Automatic Correction for Digital VLSI Circuits," in *2019 31st International Conference on Microelectronics (ICM)*, 2019, pp. 18-22.
[12] B. Alizadeh and S. R. Sharafinejad, "Incremental SAT-Based Accurate Auto-Correction of Sequential Circuits Through Automatic Test Pattern Generation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, pp. 245-252, 2018.
[13] M. Osama, L. Gaber, A. I. Hussein, and H. Mahmoud, "An Efficient SAT-Based Test Generation Algorithm with GPU Accelerator," *Journal of Electronic Testing*, vol. 34, pp. 511-527, 2018.
[14] L. G. Ali, A. I. Hussein, and H. M. Ali, "Parallelization of unit propagation algorithm for SAT-based ATPG of digital circuits," in *Microelectronics (ICM), 2016 28th International Conference on*, 2016, pp. 184-188.
[15] L. Gaber, A. I. Hussein, H. Mahmoud, M. M. Mabrook, and M. Moness, "Computation of minimal unsatisfiable subformulas for SAT-based digital circuit error diagnosis," *Journal of Ambient Intelligence and Humanized Computing*, 2020/06/29 2020.
[16] K.-H. Chang, I. L. Markov, and V. Bertacco, "Fixing design errors with counterexamples and resynthesis," in *Proceedings of the 2007 Asia and South Pacific Design Automation Conference*, 2007, pp. 944-949.
[17] S. Safarpour and A. Veneris, "Automated design debugging with abstraction and refinement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, pp. 1597-1608, 2009.

[18] A. Sulflow, G. Fey, C. Braunstein, U. Kuhne, and R. Drechsler, "Increasing the accuracy of SAT-based debugging," in *2009 Design, Automation & Test in Europe Conference & Exhibition*, 2009, pp. 1326-1331.

[19] D. Bryan, "The ISCAS'85 benchmark circuits and netlist format," *North Carolina State University*, vol. 25, 1985.

U-MENTALISMUTILITY PATENT: AN OVERVIEW

Luís Homem

Centro de Filosofia das Ciências da Universidade de Lisboa, Portugal

ABSTRACT

This paper discloses in synthesis a super-computation computer architecture (CA) model, presently a provisional Patent Application at INPI (nº 116408). The outline is focused on a method to perform computation at or near the speed of light, resorting to an inversion of the Princeton CA. It expands from isomorphic binary/RGB (typical) digital "images", in a network of (UTM)s over Turing-machines (M)s. From the binary/RGB code, an arithmetic theory of (typical) digital images permits fully synchronous/orthogonal calculus in parallelism, wherefrom an exponential surplus is achieved. One such architecture depends on any "cell"-like exponential-prone basis such as the "pixel", or rather the RGB "octet-byte", limited as it may be, once it is congruent with any wave-particle duality principle in observable objects under the electromagnetic spectrum and reprogrammable designed. Well-ordered instructions in binary/RGB modules are, further, programming composed to alter the structure of the Internet, in virtual/virtuous eternal recursion/recurrence, under man-machine/machine-machine communication ontology.

KEYWORDS

U-Mentalism, Super-computation, Computer Architecture, Cybernetics, Programming Languages Design.

1. INTRODUCTION

This document is intended to serve as white paper to describe in the most possible composed details and in anticipation the technology of U-Mentalism. As referred beforehand "U-Mentalism is a philosophical and programming idea that proposes a singular (one only and *individual*, *intensional*) and universal (all and wholly *comprehensive*, *extensional*) programming language which is, simultaneously, an inverted scheme of all the established computer architectures (…)"[1], with this meaning a common ever-evolving Assembly Programming Language, giving rise to a semantic explosion of programming languages, all throughout what can be described as an inversion of the so called von Neumann or Princeton CA in network cybernetic fashion. Protected as it may be by a provisional Patent Application at INPI (Portuguese Institute of Industrial Property) (nº 116408), a fairly elaborated disclosure can be eloquent enough as to describe its most basic settings. Although fundamentally expanded in technical computational terms, it should always be attained that one such implementational, informatic and informational method [U-Mentalism and the "C" approach in computation] is inextricable from a metaphysical naturalistic method [U-Mentalism and the "O" approach in ontology]. In addition to this, it is also to remember that the shades of relativistic and possibly technical contentious matter are all related to the perdurable problem of the context and contingent matter of technology's state-of-the-art, yet never to the very core of the new utility general-purpose application or, better said, the invention's original idea. Lastly, it is worth mentioning that the technical drawing of the CA herein disposed can also be found in the following divulging website: *www.u-mentalism.com* .

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 95-108, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101509

2.1. The imagetic frame of reference of U-Mentalism in relation to the "O(ntological)" and "C(omputational)" approaches

We shall begin by stating that the "C" method is to be performed by UTM(s) with controlled, orthogonal and synchronous, camera-like digital images RGB sensing/processing and computing binary isomorphic processors, thus with (typical) digital images "scanner" (impression) and "printer" (emission) abilities. Universal Turing Machines are described in .6 "The universal computing machine" of Alan Turing´s 1936 seminal paper *On Computable Numbers, with an Application to the Entscheidungsproblem* [2]. In this paragraph the machine $\mathcal{U}$ is supplied with a tape where is ahead written the S. D. (standard description)for any other machine $\mathcal{M}$. In such manner, $\mathcal{U}$ will compute exactly the same sequence as the machine $\mathcal{M}$. The novelty herein in U-Mentalism is that the U-machine or UTM(s) are camera image sensing (impression) and processing (emission) RGB/binary isomorphic processing computers with likewise symbols and *m*-configurations, and because UTM(s) are also $\mathcal{M}$(s), every UTM is also able to compute any other UTM(s)' computable sequence, and in such a way that any such computable sequence in a network of information, or the Internet, is equally likely to be computed. In UTM(s) the graphical camera display is the forefront processor, and RGB, although synchronous to the binary code, is the primeval symbolic feedback, in exact opposition to classical computation in M(s), wherein control and communication is wholly set on the binary code.

Now, in U-Mentalism under the "O" approach, what is relevant is the constitution of the physicalist most differentiated quanta of spacetime and observables, each of which to bear all possible viewpoint "images" for every possible and most differentiated quanta of spacetime and observables, wherein the latter observables are themselves included, as well as every all other, thus conceivably measured, in every possible viewpoint "images" in spacetime, infinitely and recursively. All in all, one such cogitation is produced by a pure imagetic frame of reference ofspacetime with bijective transformations of the state spacetime common to the different observers, and wherein the proper metric/imagetic/recursive and observable state spacetime is the observational reference frame of spacetime itself. All in all, in an analogy argument it goes as if spacetime, not affected by the indeterminacy principle (Heisenberg) and in full entanglement, could infinitely observe itself with the frame of reference being any constituted chosen metric and noematic image. If, for example, we could, non-contradictorily, in inter-*noumenal* or inter-*monadic* fashion, observe every viewpoint of every photon for the most differentiated quanta of spacetime observables, recursively in a standard description (S.D.) frame of reference of imagetic nature, herein presumed the pure imagetic viewpoint of photons, we are more inclined to understand not only the observables incompleteness (roughly expanding Gödel´s theorems from logic to "O") and undefinability (roughly expanding Tarski´s theorem from logic to "O"), but also, more conveniently, the consistent and effective passible intermediate states if the set of boundaries or limits are much less forceful. One such case is, most definitely, the"(typical) digital image", a conveniently neutral physicalist viewpoint, further permitting the classes of computable expressions and functions, such as the binary code at its core, to be mapped onto images, as expressions of the RGB codomain. We choose the S.D. frame of reference to be, according to the state-of-the-art technology, the 8K (≈8000 Pixels) 60 Frames per Second (FPS), 24 bits in depth images, choosing pixels-per-inch (PPI) as the standard resolution pattern, in which case we are exhibiting a basic setting for U-Mentalism under the "C" approach. In this wise, although far away from the philosophical *crux* of U-Mentalism "O" - "every possible image in every possible spacetime composed in every possible mind and *n*-dimensionally by *perceived* photons of light" [1] -we are resolutely bridging the chiasm by means of the presentation of a simple object, i.e., an $\mathcal{U}$.

2.2. U-Mentalism as a method of (typical) digital images non-standard positional numeration base with isomorphic many-bijective modules recursive powers

Considering the general analogy between the "byte" and the "pixel", with 2^8 (0-255) (256 tonal/chromatic values per each byte in the pixel with 3 pixels), it is known that a single RGB pixel holds, according to the formula $2(^{8x3})$, i.e., 16.777.216 tonal values or colours, which is basically the same number of bytes overall combination. This value corresponds to 2^{24} in prime factorization, which is a measure very appurtenant due to the imperative of cryptography in the system. The provisory value of the pixel (2^{24} colour/bytes combinations) is now our minimal symbolic unit, in correlation with the standard binary code, also an (observable noematic) wavelength impression. Now, the density of the pixel equation in agreement with the number of total pixels provides the image resolution in PPI=$\frac{d_p}{d_i}$[diagonal resolution in pixels=(d_p);diagonal size in inches = (d_i])]after the diagonal pixel resolution found through the use of the Pythagorean theorem:

$$d_p = \sqrt{w_p^2 + h_p^2}$$

[diagonal resolution in pixels =(d_p); width resolution in pixels =(w_p);height resolution in pixels=(h_p)]. Needless to say, we are envisaging any possible variations of measures in the overall structural and functional method.

Also relevant, both physically and symbolically, is the fact that the system in UTM(s) has invariance by synchronicity in all positively-defined and non-accelerating frames of reference (herein "Frames Per Second" = FPS) of the "(typical) digital image" in the system, and likewise the speed of light in the vacuum is the invariant *non plus ultra* limit of the technology. Therefore, under one such assertion, an 8K Ultra Full HD (7680*4320) has 33.177.600 pixels disposed (in a 16:9 ratio, i.e., 2^4: 3^2 in prime factorization). In other terms, this means that each 8K RGB digital colour (FPS) image has 796.262.400 bits, or 99.532.800 bytes.

As we are referring to an RGB/binary synthesis within an isomorphic and bijective model, we are most surely asserting a presumed less-to-the-furthest well-ordering recollection of typical digital images, arriving either by general image sensing or general image processing to the UTM(s), from a pool of very different kinds of typical digital images most generally found on the Internet: photos, URL(s) and Web pages, all Turing-machine Frames (FPS) including e-mails and instant messaging, kernel and system logs, OS environment FPS "films", digital TV FPS "films", and every other sort of Turing-machines graphical interface FPS like outdoors and consoles, ATM(s) and GPS(s), CCTV, camera drones, mobiles and tablets, ubiquitous computing things, etc.

Before anything else, the well-ordering of the RGB/binary graphical/digital coeval isomorphic code should be preliminarily understood. Accordingly, below is shown a table of partial well-ordering (16; 0-15) in a positional numeration base, with inherent many-bijective modules recursive powers. The table is the correspondent to $\frac{256}{16}$ bytes or colours, i.e., $\frac{1}{16}$ of the whole symbolic power of one pixel only under the standard description of the 8K model which holds 33.177.600 pixels. It is to be noticed in the table below that the symbolic manipulation under this pixelized part is, hence, only affecting the Blue Byte (in truth, rounded off to even numbers, roughly only $\frac{7680*3}{16}$ i.e., $\frac{1}{1440}$ bytes parts of the total in one width or horizontal pixelized line with 7680 pixels, in turn intersected with 4320 pixels in height or vertical lines, in a 16:9 ratio, which sums up a total for each FPS, or "(typical) digital image", of≈33.000.000 pixels and, therefore, of ≈99.000.000 bytes.

This is so due to the patterns of the RGB code industry convention, wherein WHITE is 255*65536+255*256+255 = #FFFFFF, and so, by order, RED is 255*65536+0*256+0 = #FF0000, GREEN is 0*65536+255*256+0 = #00FF00, and BLUE is 0*65536+0*256+255 = #0000FF. However, any other suitable well-ordering complies and falls as predicted and logically accommodable in the technology.

Table 1. RGB 24 Bits Colour Calculus, RGB Binary, Hexa, and Ordinal RGB modules

RGB 24 Bits Colour Calculus	**RGB Binary**	**Hexa**	**Ordinal module and Decimal**
(0*65536)+(0*256)+Blue	(00000000,00000000,00000000)	#000000 = 0	1^{st} = 0
(0*65536)+(0*256)+Blue	(00000000,00000000,00000001)	#000001 = 1	$2^{nd,}$ =1
(0*65536)+(0*256)+Blue	(00000000,00000000,00000010)	#000002 = 2	3^{rd} = 2
(0*65536)+(0*256)+Blue	(00000000,00000000,00000011)	#000003 = 3	4^{th} =3
(0*65536)+(0*256)+Blue	(00000000,00000000,00000100)	#000004 = 4	5^{th} = 4
(0*65536)+(0*256)+Blue	(00000000,00000000,00000101)	#000005 = 5	6^{th} = 5
(0*65536)+(0*256)+Blue	(00000000,00000000,00000110)	#000006 = 6	7^{th} = 6
(0*65536)+(0*256)+Blue	(00000000,00000000,00000111)	#000007 = 7	8^{th} = 7
(0*65536)+(0*256)+Blue	(00000000,00000000,00001000)	#000008 = 8	9^{th} =8
(0*65536)+(0*256)+Blue	(00000000,00000000,00001001)	#000009 = 9	10^{th} = 9
(0*65536)+(0*256)+Blue	(00000000,00000000,00001010)	#00000a = 10	11^{th} = 10
(0*65536)+(0*256)+Blue	(00000000,00000000,00001011)	#00000b = 11	12^{th} = 11
(0*65536)+(0*256)+Blue	(00000000,00000000,00001100)	#00000c = 12	13^{th} = 12
(0*65536)+(0*256)+Blue	(00000000,00000000,00001101)	#00000d = 13	14^{th} = 13
(0*65536)+(0*256)+Blue	(00000000,00000000,00001110)	#00000e = 14	15^{th} = 14
(0*65536)+(0*256)+Blue	(00000000,00000000,00001111)	#00000f = 15	16^{th} = 15
(0*65536)+(0*256)+Blue	(00000000,00000000,00010000)	#000010 = 16	17^{th} = 16

The well-ordering herein disposed constitutes itself a major resemblance feature with the fundamental theorem of arithmetic, but this time in graphic/digital format, and fundamentally with inherent many-bijective recursive powers under super-computation, greatly emancipating the products of the faculty of imagination, with much greater power of synthesis and scope. Because it is not only analytical, but dialectic, or better said dynamic, it is not only arithmetic, but essentially algorithmic.

Indeed, the unique-prime factorization theorem in arithmetic progression, as well as any chosen unique or non-unique progression containing composites (not necessarily through 8 bits modules) is *spontaneously* an algorithm of the system, moreover with canonical or non-canonical operations and functions, either cognitive (man-machine noematic-representing) or practical (man-machine evaluation-apt and choice-expressing), where from mathematical and philosophical noemas and judgements are predicated in relation with UTM(s).

In what regards the positional system, enough is said if we declare that the binary radix of the system, congruent with any other numeral system, works with the “octets-bytes”, or rather “3 octets – 3 bytes”, of the “pixel” itself as placed (RGB bytes and colour) value notations, with width and height, and ahead time-valued and inter placed-FPS combined index positions, constituting any possible number or algorithm. Let us notice that by programming itself, non-*autonomously* but *spontaneously*, a non-standard positional numeral system, in synthesis (presumably also under the unity of apperception in man-machine communication) and synchronicity with a place and time combined valued notations, is present, even if the system is a standard positional numeral system.

At this point, with the value at hand of 99.532.800 bytes in an 8K Ultra HD digital image, this value corresponds to the total isomorphic RGB digital image-to-binary code ready to be processed by image sensors (impression), and also ahead processors (emission), being in this way clear that the system has only to differentiate 2^8 per byte/per pixel, or 2^{8*3} per pixel locally, instead of the much harsher demand of the linear-dependent CPU, or non-linear CPU-dependent GPGPU general amount of 99.532.800 bytes per FPS, in one "(typical) digital image".

The next step is, thus, the assessment of the chosen value of 60 FPS (Frames per Second) in conjunction with the present invention, a pretty conservative value to take into account, especially if we consider that the INRS research team has, with the T-CUP, overpassed the threshold of 10 trillion FPS, invading the femtosecond scale, i.e., $\frac{1}{1*10^{-15}}$ of one second (or quadrillionth of a second). Now, 60 (2^2*3*5 in prime factorization) FPS, as soon as it meets the second FPS or frame, is defined as a "movement-image" or "film", pointing out to a value of

60 (FPS) * 60 (´´) * 60(´) * 99.532.800 bytes

21.499.085.000.000 = 2.1499085e+13 bytes per hour in one film only in the technology herein presented. The correspondent and conveniently converted value of bytes per hour in one film only is, hence, 2.1499085e+25 Terabytes, or 21.499085*10^{15}Zettabyes.

It is pertinent to contend that this value is a dense discrete metric measure and, although it can be put forward in synthesis in one film only, in programming algorithmic technical terms it might eventually have been formed by the concurrence of many permuted and/or combined, rather than composed, "films", or as it might be FPS tunnels of "pixels" as wavelength impressions/emissions all throughout every bit and at the full length of the movement-image per hour in one film only under the technology.

Confronting anew with Alan Turing´s *On Computable Numbers, with an Application to the Entscheidungsproblem*, definition – "The machine is supplied with a 'tape' (the analogue of paper) running through it, and divided into sections (called 'squares') each capable of bearing a 'symbol'"[2] – the shift to U-Mentalism in the "C" approach is easy to follow if we declare that the "tape" is now film, "squares" are now (FPS) frames, and the "symbol" the graphical/digital movement-image encompassing the necessary RGB/binary "*r*-th bearing of the symbol" in network distributed in as many as possible partial computing UTM(s).

Attention should also be called to the fact that IDC and Seagate forecast that the global datasphere, which was of 33 Zettabytes in 2018, will grow to 175 Zettabytes ($175 * 10^{21}$ bytes) by the coming year of 2025 [3]. In other words, the figure found of 21.499085*10^{15} Zettabyes for U-Mentalism one hour of one film only of processing power return from data is, on its own, 1.2285191e+14 times more than the expected global data for 2025. In point of fact, it would have to elapse 120 years, with each year equally with 2025's 175 Zettabytes + a 100% growth rate for each year, so that the data sphere would approximate the return result of processing power of the technology for one only film of one hour only thereof encapsulated. The growth rate of data is hereof paramount, as considering if not, figures are that 17.500 years, each with equal 175 Zettabytes of data, would have to pass by to meet one hour only of U-Mentalism, with very conservative parameters for the processing power. In view of this, and on the other way, assuming the quadratic nature of quantum computing worst-case complexity in confront with classical computation, if we envisage the system´s complexity complemented with quantum computing complexity, and most specially, counting with the accelerator factor of U-Mentalism on the production of data, the inception in years might be dramatically shorter. What is more,

supposing that the technology and CA would work on the full 175 Zettabytes of global data for the year 2025, the super-computation involved would assume 48.611.111 Terabytes per second ($4.8611111 * 10^{19}$ bytes per second), which sums up $2^{65.39792}$ per second, already above the capacity of a 64 bits architecture. In truth, in one such case, the technology is imminently conjectured to improve above 1 Exbibyte (EiB) = 2^{60} or 1024^6.

Because the S.D. of the technology can be represented by a set forming the symmetric group of the set, which is a bijection from S.D. to itself, and for which every placed octet-byte (binary) element occurs exactly once as the corresponding placed (RGB) image value, $S.D._n$ is the symmetric group under permutation, a broad relation and it is also a function composition under group theory. For the reason of implementation of less-to-the-furthest well-ordered recollection (large numbers arithmetic) and further forward well-ordered collection (at large algorithmic) of typical digital images, onward to be run by as many as possible UTM(s) in a network of UTM(s) on the Internet, a calculus of permutations is needed and, complementarily, it is imperative to calculate a fair assessment of U-Mentalism computational time complexity. With this procedure we are already asseverating the inclusion of crucial demands of the system, such as the use of AI & cryptography general image processing of typical digital images, apart from general image sensing of typical digital images, as well as the run time complexity in relation with the data memory involved.

3.1. U-Mentalism (typical) digital images permutations in partial and distributed UTM(s) in a network, on the Internet

Once the 8K resolution Ultra Full HD (typical) digital image (FPS) bears $(7680 * 4320)$ pixels, which sums up 33.177.600 pixels (width * height), what follows is the application of the formula of permutation having in mind the measurement in pixels

$$P(n,r)\frac{n!}{(n-r)!}$$

Thus, $n^?$ is, really, the number of colors per pixel, which is 256^3 and the correspondent to the combination range, which equals 16.777.216 color combinations per pixel, while $r^?$ is, really, the previous value of 8K Ultra Full HD (width * height), i.e., 33.177.600 pixels. It is very easy to appreciate that the break-up of the two orders of factorials points out to unmanageable numbers, directing both to countable infinite numbers, and $O(n!)$ non-assessing time complexity, regardless of the CA, recursion power or machine. One such calculus is, nevertheless, judiciously desirable by cause of the intrinsically $O(n!)$ factorial time complexity exposition of the system, dragging $O(2\text{^}n)$ exponential time, and O(n^2) quadratic time hardness lines, as well as, in middle-way and by order, $O(n)$ linear and $O(log\ n)$ logarithmic times -herein $O(n!)$ grows faster as it abridges a constant exponential base 2 -, but on the side of U-Mentalism overlapping solvability, not of classical computation. Simply, the algorithms of U-Mentalism are yet unknown, and the infimum complexity that solves a class of problems is of the same complexity as that of the problem. One such assessment comes even beforehand newly fine-grained analysis, defining the possible class of problems as the set of computational problems of related resource-based complexity, given that time, processing, memory, and more so the relation between them is radically different in U-Mentalism, however included in Turing-machines computability *Application to the Entscheidungsproblem* [2].

On the edge, by nature of the intrinsic arithmetic system in U-Mentalism, we could even consider each frame a large number image, reducing the composites factor in 8K of 33.177.600 pixels to one FPS and, consequently, to each one (FPS) large number linear arithmetic progression image,

which and in turn transforms the calculus of the combination of colors per pixel per FPS, to one onlylarge number arithmetic mirror per FPS. One such FPS large number image progression would meet a factorial (FPS) table itself of $O(n)$ linear time complexity, that is made exactly a "C"*imago mundi* for the "movement-image" recursion in the system. It is true that the immediate next polynomial running times (quadratic, cubic, n^c, etc) hold important classes of algorithms to discern an unequivocal well-ordered (FPS) large numbers arithmetic progression mirrored (RGB) images, in recollection (composition) and collection (permutations/combinations) of typical digital images. However, U-Mentalism is not a system to solve one such linear progression FPS problem, but instead to solve ahead any algorithm class problems newly defined by the system itself, with inherent new space and time complexity powers, insofar one such linear FPS progression is being expanded in the system.

It is important, under this context, to remember that decidability is based on the localistdecidability (even working with non-localist quantum computation by chance) of the pixel isomorphic RGB image of every binary octet. Accordingly, the large number arithmetic (RGB) image mirrors of the system have already exceedingly computing power, all throughout a system where data and processing are positively "C" entangled: data capacity returns processing power, and processing power returns data capacity.

All in all, and luckily, steadily pacedperformance up to constant verifiable factors is all that is needed under the U-Mentalism system, in a deep and low-level performance requirement RGB/binary enhancement only, wherein, unambiguously, arithmetic progression in FPS follows locally each digit power of two in binary, once decidability in terms of a machine $\mathcal{M}$ or $\mathcal{U}$ is said to be a decidable problem if there exists a corresponding $\mathcal{M}$ or $\mathcal{U}$ which halts on every input with either 0 or 1, thus low-level feeding the FPS arithmetic progression of large number mirror RGB images. This is, besides all, what makes it not constructive at all, and indeed counter-productive, any glimpse whatsoever over a hypothetical solution based on (typical) digital image decreasing measure overall pixel/bytes reconversion. An aforesaid presumptive choice of lesser resolution-say, maintaining the $16{:}9 = (4^2{:}3^2)$ ratio, $500 * 281.25$ (width * height) in pixels -would naturally decrease the computational power of the technology and, ergo, the overall scalability of the technology in relation to the cybernetic network on the Internet, under which a minor convolution of data and processing power altogether would impend on time complexity solvability.

What happens is exactly the contrary: the datasphere is too tiny when confronted with the power of U-Mentalism, to the point where well-ordered recollection and collection are pivotal not only to operability, but also to the progress of the technology.

More importantly, in the localist decidability of the pixel isomorphic RGB image of every binary octet resides the fundamental criterion of difference and repetition at which underlies the XOR or$Exclusive\ OR$ argument at the root of progression of binary numbers or, indeed, $mod\ 2$ addition.

If noticed, the progression $(00,01,10,11)$ corresponds to binary addition, after which completion the next two bits on the left are triggered to shift by half-addition, the same is saying, the double of the previous elements of the series of progression.

In other terms, it corresponds to a not equivalence NEQ difference and repetition operation in binary/RGB isomorphic arithmetic progression, wherein the proper R(ed), G(reen) & B(lue) are module operations. In this fashion, the whole (FPS) (typical) digital image becomes a truth-table,

for the reason that the lines and the columns (width * height) are themselves a sum operation, needing only a carry-out color/bit to the left when the progression (00,01,10,11) ends.

This would equate having in the binary logic and image sensing (and processing) unit, presumably, a two-color/bits *XOR* and *AND* adder. Naturally, because of the need of a full adder circuit for the entire (FPS) (typical) digital image, the binary/RGB isomorphic nature of the UTM(s) would rather, again presumably, be prone to use a two-bits/color *XOR*, *AND* and *OR*. Inasmuch as, for instance, in propositional calculus, laying the foundational bedrock of logic since Aristotle – considering the four different types of categorical propositions in the square of opposition, withstanding the syllogism theory - there are three propositions for each place-valued syllogism figure out of possible four. Thereupon, the possible total number of syllogism modes is four times that number, i.e., 256 logically possible distinct types. Because 256 is the same number of module 8 bits per RGB color in the (FPS) (typical) digital image, what this signifies is that, for the sake of the argument and hypothetically, an (FPS) (typical) digital image is a polysyllogism and a *calculus ratiocinator* $\mathcal{M}$.

However, and fundamentally, in U-Mentalism the CA is inverse and, thereupon, it is not built on binary/RGB, but instead RGB/binary. Without this judgement, it comes not to be transcendental. It shall produce wavelength colors and forms synthesis, just like φύσις (nature).

Due to the localist nature of both the pixel and of the whole (FPS) (typical) digital image, the RGB/binary isomorphic nature processed in the UTM(s) will be prone to use, not quite an equivalent color summands adder, but instead an every *n*-arycolor/bits RGB *imagetic* relations instant mirror, filter and mixer, always remembering that in between different modules and (FPS) films the exactly same holds true. What this means is that U-Mentalism is, at each UTM processing, constant metric localist, either in a pixel, a module, or the entire FPS, with equal time-dependent "film" computing power on the previous synchronous and orthogonal base for each (typical) digital image, *n*-arymodules or bytes.

In truth, the RGB/binary relation in the CA is always affected by a special bottleneck related with the communication with classic computation, as far as other much less grievous than the von Neumann bottleneck, which basically corresponds to the arithmetic logic unit binary mirroring of the RGB image codomain in case only of U-Mentalism scanner-to-printer or Eye-to-Brain, but except for machine learning, not in the case of U-Mentalism printer-to-scanner or Brain-to-Eye. If any bottleneck in the system exists that is worth mentioning it is, inevitably, what we choose to designate the unfolding "O" and "C" philosophy of (time) history bottleneck. The reason behind so is that computational means and resources up to this point of the "C" state-of-the-artdo not produce a reasonable amount of data as to test match the system, which is test halting (time) history itself. Before having a chance of escalation from one hour of one only film in the technology to a film of several hours, years, and even synthesis of the image metric distance, in the "C" movement-image, in light-seconds, (time) history "O" has, simply, to elapse. In contrast, as seen before, ["C"] computational time complexity in U-Mentalism has equally to elapse, although much more tied with bondless mathematical and dynamical limits, precisely on the grounds of the ["O"]constitutive transcendence on ["C"].

It is, ultimately, by virtue of this assertion that it is more appropriate to make mention of a general U-Mentalism "C"-"O" bottleneck, which is, by and on itself, a rectification of the classical von Neumann bottleneck. To alleviate any remaining doubts, it should be elucidated that the system is an inversion of the von Neumann CA, not only because of the RGB graphical primeval symbolic precedence, but also, amongst other aspects, of the inherent entanglement of data and processing power, which drives high latency, on a U-Mentalism turn, from being orderly unavoidable to well-orderly avoidable.

U-Mentalism, in this fashion, especially due to the diagonal method of computability beforehand, and of well-ordering collection also on the basis of the diagonal method of cryptography and permutation/combination, has a nature of transfinite (FPS) typical digital images, denumerable "image-movement" sets, with inherent cardinals and ordinals equipollence. For the moment it suffices to say that the nature of computable and definable numbers, as for the rest composite or prime numbers in U-Mentalism as "3 octets – 3 bytes" modules, typical digital images, or "films" is, intrinsically, a bijection of the well-ordered set of all finite ordinals in the system w_0 to cardinality $\aleph_0$. Therefore, an algorithm for an well-ordered collection of typical digital images in the system could easily resort to a typical diagonal on the binary basis and exceptionally reductionist.

Inquisitively, one can picture also, in an *Imitation Game* [4] register, or in a Turing Test flair, a different *dialogical* test. We shall call it for now both the U-Mentalist "O" inquiry and the "C" test. On the grounds that any (human or machine) synthesis of electromagnetic wave-like physical and symbolical impressions, susceptible of being, in turn, emitted in any body or technology, are to be, in quantized electromagnetic wave-like impressions in continuous spacetime, indiscernible in nature, the following questions arise:

Under U-Mentalism "O", we inquire if it is possible to be an observer without photons or any observable frame of reference in spacetime.

Under U-Mentalism "C" and likewise, we shall test if ever the prior impressed and, thus, emitted UTM(s) wave-like synthesis or images, in spite of the foreign face-to-face relation within a body of a presence in front of it, can be made discernible from the frame of reference of the observer.

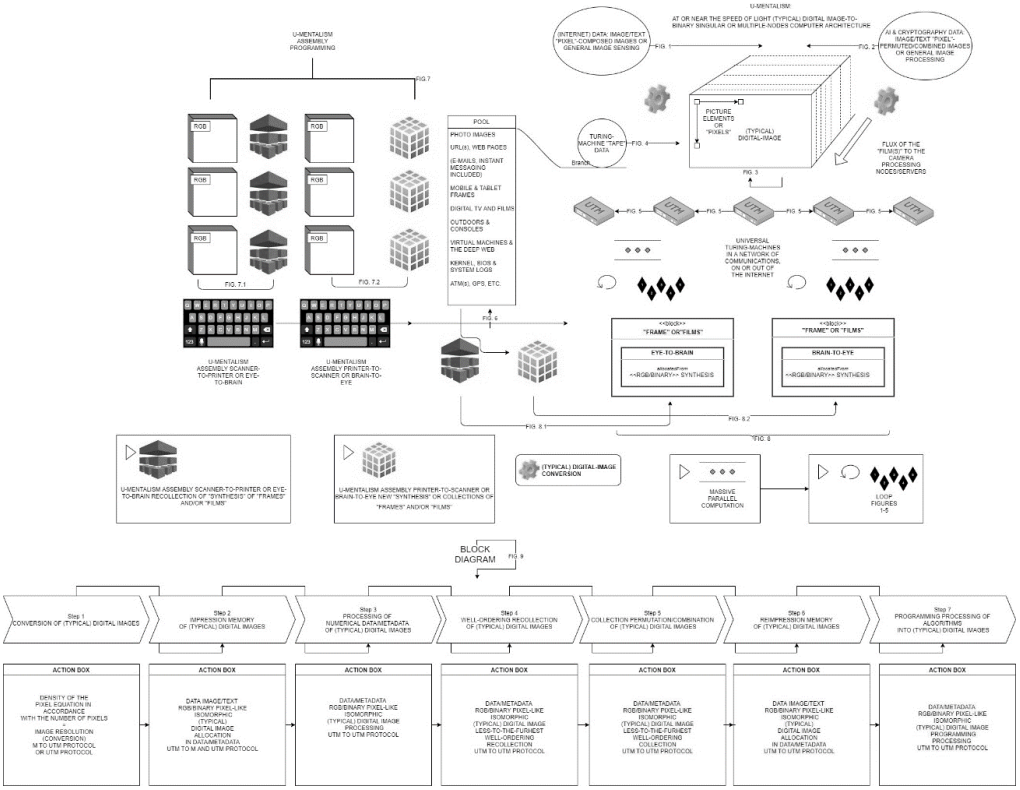


Figure 1. U-Mentalism Computer Architecture Design

3.2. U-Mentalism UTM(s) network analogy with current data nº1 TOP500 Linpack

Super-computation is often measured or estimated according to the floating-point (FLOPS) (additions and multiplications) computing power in the $\mathcal{M}$, under the numeric and scientific Fortran-derived linear equations in the LINPACK Benchmarks, taken as 64-bit floating-point peak performance. Besides being quite a multi-dimensional problem to address, and essentially a non-distributed supercomputer system ranking, it is the most reliable source for high-performance computing so to have a basis of comparison with the U-Mentalism CA, either distributed or non-distributed, in or out of a network, on or outside of the Internet. This is important to refer, once the CA is developed to be on distributed massive parallel computation with the most possible advanced cryptography methods on each UTM server/node (like open-source Blockchain), in a network, on the Internet. Any contrary application of the CA is envisaged as jeopardizing and, indeed, potentially very harming if ill-fated. It shall, hence, be laid open – *patere* (wherefrom the word patent derives) – for both public inspection and public policies, laws and interests of governments and the people. Once these aspects bring in multi-variables of which values is difficult to know, although being much easier to acknowledge approximations, and most specially orientated-guise measures of central tendency, normal distributions probability, and deterministic ranges, our method shall simply attain to the plausible measure for the intensity of the required memory per unit of performance, along the standard of FLOPS and bytes per FLOP (B/F).

Currently, the nº1 position of the 55^{th} TOP 500 following the LINPACK benchmarks suite (June 2020) is the Fugakupetascale (10^{15} floating-point operations per second = 1 petaFLOPS, i.e., a thousand million millions 64 bits operations per second) supercomputer. The Fugaku holds 415 petaFLOPS with a 158,976 (two types of) nodes Fujitsu TofuD, 6D mesh/torus Interconnect, in a A64FX CPU (48+4 core) per node CA, with a second-generation High Bandwidth Memory (HBM2) of 32 GiB/node.

Our next step is, thus, by multiplying Fugaku´s instance of cores * nodes (52*158,976 =8.266.752) find the equivalent processing power of the very same number of UTM(s) cameras/computer processing nodes/servers and later, having in mind that the Internet has around 50.000.000.000 nodes, well above the 10^{12} to 10^{24} FLOPS of all the existent computers (2015), at the end reasonably cut the latter figure by reason of factors such as entrance in the industry, price and energy, besides any hindering variables, thus obtaining a fair value for the CA implementation in the network of networks, i.e., on the Internet.

First off, the value of $21.499085 * 10^{15}$ Zettabytes of one only "film" of one hour in U-Mentalism divided by the number of cores * nodes (52*158,976 = 8.266.752), equals $2.6006689e + 15$ Zettabyes, which converted is 2.6006689e+15 Petabytes. Therefore, assuming a 64 bits operation, we have roughly 325083615.064 PetaFLOPS per each server/node out of 8.266.752 in the technology, if U-Mentalism was to have the same number of nodes/servers on the Internet, presumably settled for a start on an Internet with an even much bigger number of nodes. And even if there would never be the same number of servers as nodes, in reality and at present the Internet detains around 50.000.000.000 nodes.

In abstract, the value divided by the computer performance of the Fugaku supercomputer indicates that one only hour of one only "film" in the technology would equate to 783334.012203 times the nº1 position of the TOP 500 Linpack benchmark as of October 2020. In reverse manner, we could affirm that the 415 overall petaFLOPS of the Fugaku supercomputer, compared with the 597196805556 Zettabytes/per second (or 597196.805556

Petabytes/per second) in U-Mentalism, stipulates that a presumed value in 64 bits (74649.6006944) petaFLOPS/per second in U-Mentalism is, in itself, 179.87855589 times more than the overall performance of the Sugaku supercomputer.

The extreme low latency of the invention is best shown if we divide the overall abstract performance of U-Mentalism of petaFLOPS/per hour (2.6006689e+15) by the actual number of Internet nodes (50.000.000.000). The result is 52013.378 petaFLOPS per each one out of 50.000.000.000 Internet nodes, when indeed, chances are not only that the divisor will be much larger, but essentially that the number of typical digital images reaching the system will be exponentially wider. Confronting with the Sugaku supercomputer, this would mean that at each one of these 50.000.000.000 nodes, it would be instantiated 125.333440964 times more the performance peak of the Fugaku supercomputer in PetaFLOPS.

Nevertheless, because the hindering variables are numerous and immense, even though we are experimenting values with greatly sub-optimized inherent values (pixel resolution, FPS, Hz, and subordinate processing-time of the technology to one-hour only), we shall now, thinking ahead the barriers to the entrance in the industry in terms of price, energy, etc., cut the preliminary values to around 20%. Thus, the result at hand is, under one such 20% cut under the very same parameters, of65016.7225 PetaFLOPSfor each one out of 50.000.000.000 Internet nodes, which equates to 100.266752771 times more the performance peak of the Fugaku supercomputer, measured in PetaFLOPS for the very same value of each one out of the 50.000.000.000 Internet nodes.

One such technology shall be exclusively scientifically-driven. In fact, in terms of the stored-program concept, we can designate it, in in differentiable manner, (φύσις) science, if granted that the overall feedback and cybernetic loop mechanism, that we choose to call an "algorithmitron", i.e., an accelerating mechanism for all classes of algorithms, and thus a procedure on its own, is itself well-ordered within the *extensional* and *intensional* self-image of man and the cosmos that is (φύσις) nature. The extensional and intensional philosophical synthesis and programing regulative idea shall be explored in recurrence and recursively.

Very concretely, the actual example provided by the current use of the nº1 TOP 500 Linpack benchmark Fugaku Japanese supercomputer, which has presently been used for COVID-19 research, and the nº2 TOP 500 North-American supercomputer Summit, whose current work with scientific impact is on various levels (deep learning for human systems biology, plasma fusion simulation, combustion in turbulent environments, stellar astrophysics nuclear burning, cancer treatment and surveillance planning, high-temperature superconductors) are nothing but just a pale *coup d'oeil* of what can be, at a greater extent, achieved with the forthcoming fabrication. By all means, U-Mentalism participative and all-engaging cultural-scientific accelerator, social and technological, financial and political, inter-dependability and transparency, shall act as new measures for the human. We have to remember, for that purpose, that a world with a 24 hours "film" in the technology, however $1.2285191e+14$ times the value of 175 Zettabytes (estimation of global data for 2025) with the necessary equivalent data input, thus dependent on the U-Mentalism "C"-"O" bottleneck, would grant the CA with a processing power of $5.1597804e+17$ Zettabyes ($21.499085 * 24 * 10^{15}$). The same is valid for "films" of several years, and even of the "image"-distance in light-seconds, most assuredly in prospective proper physicalist-reductionist underpinnings, rather than merely technological.

Yet, more frequently than not in dialectic terms, time and history itself naturally supersede and are transcendent in relation to any object (such as technology). In fact, the author suggests that the contemporary crisis in philosophy of science (φύσις) points out to the extreme of that reality. If any peak in civilization was possible to be found, it definitely was the period from the birth of

Classical Greece up to the end of the Hellenistic period, from classical Athens to the Hellenistic Alexandria. It worked as a natural philosophy (φύσις) explosion or radiation, all throughout two millennia with tergiversated and fragile-weaved, often sinuous paths on the edge of eradication, as epiphenomenon's echoes throughout the Roman and Byzantine, Islamic and Indian Empires, before the turning from Medieval times to the European Renaissance, and thereupon Modernity and Contemporaneity, that was afterwards, arguably, to see its dimmest resonate and last hour in XXth century Vienna. This serves to explain the non-mutual relation between progress and technology. More so, sometimes the value of a technology is best evaluated if tested against the worst demeanors and actions known to the history of civilizations. For example, in substitution of a colossal computing and processing power, the technology of U-Mentalism could benefit more from improvements in cryptography, or primarily human-agent decisions.

Coming to think of it, and bearing in mind imaging and sensing technology latent in U-Mentalism - metal–oxide–semiconductor (MOS) based charged-coupled devices (CCD), and active-pixel sensors (CMOS) in the present state-of-the-art, prior to the reductionist more general "cell"-like synthesis–we can think of two unexpected, but conceivable and tenable breakthroughs cognate with the technology that are worth to be referred.

One is, definitely, the use of biopolymers as paper for the use of electronic applications, namely paper transistors recurring to metal oxide semiconductor (MOS), complementary (CMOS) circuits, and eventually transparent conductive oxides, i.e., paper-based electronics or papertronics [5, 6]. Most importantly in the case of a simple and universal device architecture in correlation with the novel U-Mentalism CA, it could literally be possible having always and ever a paper copy of every book in the world in the same paper organic substrate, also electronic component (dielectric), and charge storage media, an upturn revival of the inceptive idea of the Great Library of Alexandria and Mouseion since Ptolemy Soter I, center of Hellenistic civilization and epitome of Classical Greece, where the study of natural philosophy (φύσις) found its ἀκμή(acme).

The other conceivable breakthrough is directly correlated with the possible use of transparent oxide electronics as a backbone to U-Mentalism Assembly Language programming. Because in U-Mentalism there will be the need to instruct in symbolic RGB/bytes machine code modules, "frames" and "films" through instances of time in tunnels of "pixels" or, in fact, any other instances of "cell"-like exponential-prone alike basis, having access to novel semi-conductor amorphous oxides or applications with high transparency and electrical conductivity, can open the gates to create an endless array of philosophical and programming short-cuts over typical digital images. In reality, beyond Thin Film Transistors ("active matrix" TFT), Liquid Crystal Display ("passive matrix" LCD), and Organic Light Emitting Diode (OLED), the transparent semiconducting oxides (TSOs) and transparent conducting oxides (TCOs) can help the technology to directly assemble the building block-structure-luminous response mechanism itself, bridging optoelectronics with programming, and possibly breaching into Photo-Voltaic modules (PV or solar panels) or electronics Organic Solar Cells (OSC), including polymer solar cells.

Lastly, the author would like to reiterate the U-Mentalism "O"-to-"C" cybernetic analogy with photosynthesis, the very definition of "synthesis of light", already expanded in a preliminary paper, in all likelihood with improved understanding as of now close to the conclusion:

U-Mentalism is mainly intended to be a programming synthesis of light through typical (digital) images, organized as symbolic-informational truth-equivalent programming language abstracts. Photosynthesis puts together a synthesis of light, carbon dioxide and water into glucose at reaction centre proteins with chlorophyll (digital images), wherein to the fore roots have absorbed water (computability) from the soil, through the stem (programming language abstracts and

paradigms) and through the leaves (programming languages). This is why to the exact chlorophyll complementary light (diagonalization) absorbance centre chloroplast organelle (pixel) there is, at large, a leaf lamina (frame), as a surface area to capture the light, under light's every possible and each necessary time-image. There is, in the overall process of photosynthesis, a light-dependent cycle and a light-independent cycle. In the light-dependent or light cycle (scanner-kinescope), as an effect, short-term stores of energy are produced, enabling their transfer to drive other reactions (computer vision & multiple-view geometry; U-Mentalism Recollection), while in the light-independent cycle (printer-iconoscope; U-Mentalism Collection), the so called Calvin cycle, the atmospheric carbon dioxide is incorporated into organic carbon compounds (U-Mentalism Assembly Language Programming), and dependent on the previous light-dependent reactions (semantic isomorphic correspondence), are then used to form further carbohydrates, such as glucose, the most important source of energy metabolism in bioenergetics (cybernetics) [1].

4. CONCLUSIONS

In the present study, following closely the theoretical and practical keystone of the provisional Patent Application at INPI (Portuguese Institute of Industrial Property) (nº 116408), designated as U-Mentalism, a method to perform computation at or near the speed of light, resorting to "(typical) digital image" RGB-to-binary in singular or multiple nodes/serversin a network, on the Internet, in its entirety a philosophically-meaningful new computer architecture, is displayed its simplest baseline, adjustable for the research and industry communities. Foremost, the proper discrepancy between the imagetic frame of reference of U-Mentalism in relation to the "O(ntological)" and "C(omputational)" approaches is elucidated. No substitute of the latter can prepare ahead the in-depth comprehension of the intrinsic method of typical digital images coincident with non-standard positional numeration base with isomorphic many-bijective modules recursive powers. Ensuing, typical digital images permutations in partial and distributed UTM(s) in a network, on the Internet, is shown to be the proper context for the technology to be undertaken, which suits the passage to a vaguely prosaic, but matter-of-fact indisputable, comparison of the fundamentally futuristic trait of the invention with the current data nº1 TOP500 Linpack supercomputer as of November 2020, the Fugaku supercomputer at the RIKEN Center for Computational Science in Kobe, Japan.

REFERENCES

[1] Homem, Luís, (2019) "What is U-Mentalism?",*Journal of Advances in Computer Networks*, Vol. 7, No. 1, pp.18-24.

[2] Turing, Alan M., (1937) "On Computable Numbers, with an application to the Entscheidungsproblem",*Proceedings of the London Mathematical Society*, 2, 42 (1), pp. 230–65.

[3] Reinsel, David &Gantz, John &Rydning, John, *Data Age 2025,*"The Digitalization of the World, From Edge to Core", *An IDC White Paper - #US44413318, Sponsored by Seagate* pp. 1-24.

[4] Turing, Alan M., (1950) "Computing Machinery and Intelligence", *Mind*, LIX (236) pp. 433-460.

[5] Barquinha, Pedro & Martins, Rodrigo & Pereira, Luis & Fortunato, Elvira, (2012) "Transparent Oxide Electronics, from Materials to Devices" *Wiley, a John Wiley & Sons, Ltd., Publication*

[6] Martins, Rodrigo & Gaspar, Diana & Mendes, Manuel J.& Pereira, Luis & Martins, Jorge & Bahubalindruni, Pydi & Barquinha, Pedro & Fortunato, Elvira (2018), "Papertronics: Multigate paper transístor for multifunction applications", *Applied Materials Today* 12 (2018) pp.402-414

AUTHOR

Luís Homem (Lisbon, 21/12/78) has a degree in philosophy (2008) at the University of Lisbon, having also completed a Ma degree in logic and philosophy of science at the University of Salamanca with the thesis "Topics in Programming Languages, a Philosophical Analysis through the case of Prolog" (2018). Being a doctorate integrated member of the Center for Philosophy of Sciences of the University of Lisbon (CFCUL), the author has developed research mainly in philosophy of logic, science and language.

AN INTERACTIVE APPLICATION TO ASSIST BIOLOGY LEARNING USING AUGMENTED-REALITY

Wenxi Li[1], Marisabel Chang[2], Yu Sun[2]

[1]Northwood High School New City, Irvine, CA 92620
[2]Department of Computer Science, California State Polytechnic University, Pomona, USA

ABSTRACT

As students learn biology in term of molecule, cells, or proteins, cross section of 2D image is a traditional method study the details of them. However, this method cannot bridge the gap between reality and students' imagination based on 2D image. This paper proposes a tool which can assist students to learn biology knowledge more effectively through augmented-reality. The experiments on accuracy and performance of the image pattern recognition indicates that FAST algorithm is the best choice currently. It reaches the highest accuracy of 94.5%.

KEYWORDS

Augment reality, Vuforia, Blender, Unity

1. INTRODUCTION

When coming to a biology molecule-for example, cells and proteins-it can be quite hard for students to visualize. Traditionally, students can see cross sections of the molecule in 2D images. However, there is always a gap between the sides to sides as well as the reality's conception and the audience's imagination. For example, the overlapping sections in tissues would be hard to recognize when given a cross-section. The emergence of 3D digitalization has altered such scenarios profoundly. When applying 3D modeling to biological molecules, instructors may describe and convey the knowledge more vividly and students may understand the designs more fully and more accurately in a more convenient way.

Open Problem: Although students can use diverse methods to learn biological molecule such as 3D digital and 3D physical models, they are no efficiency since they are costly, complex and inaccessible. Some digital 3D modeling techniques and systems that have been proposed to demonstrate biology structures include websites like sketchfab and Turbosquid [12][13]. Sketchfab and Turbosquid are platforms that sell, publish, share, and buy 3D model content. However, these websites are constructed as data bases for 3D models and their goal is to establish a platform for people to post and trade their 3D models. These platforms do not have enough biology models specifically made for school purposes, therefore, a lot of the biology models they have are either too abstract or complex for student understanding. Besides digital 3D modeling, using physical 3D biology molecule models for demonstration is another option. However, physical 3D models are relatively inefficient, considering making a physical model is usually time consuming and limited people can have access to the model [20]. Nevertheless,

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 109-117, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101510

physical models are usually rigid, unlike digital models which students can zoom in and out and view the interior of the models.

Solution: built a 3D biological molecules model using Augmented Reality and Computer Vision. Software's or websites specifically dedicated to demonstration of 3D biological molecules is a relatively innovative and unique idea. When coming to the purpose of teaching and demonstration to facilitate student understanding, a software specifically dedicated to biology models is especially practical, efficient, and engaging, since it can be downloaded on digital devices easily and gives students a thorough understanding of the biology molecule.

For our mobile application, we used Unity3D, Blender and Vuforia. Unity3D is the game engine we used to import the 3D biological molecule models and 3D target model to create the mobile application. We chose Unity3D since it supports cross platform development for both android and IOS. Blender is an open-source 3D modeling software tool that we used to create some of the 3D biological molecule models.[5][7] Vuforia is an Augmented-Reality software for mobile application developed by Qualcomm [14]. It uses computer vision technology to track and recognize image in real time [4]. I used it to assist recognition of the images.

Two experiments have been conducted to verify the following two aspects of the system:
Experiment 1: The accuracy of the image pattern recognition using different algorithms. The core image pattern recognition algorithms rely on the image feature extraction methods. We have tested the different feature extraction algorithms to recognize the image pattern based on the chosen sampled test images, such as FAST, Difference of Gaussians, and Determinant of Hessian. It turns out that FAST has the highest accuracy of 94.5%.

Experiment 2: The speed of image pattern recognition using different algorithms. To guarantee a smooth user experience, the recognition algorithms should be fast enough to provide a real-time experience. We tested the training dataset collected using different feature extraction algorithms to measure the performance. Although most of the speed are similar to each other, the FAST is on average faster than the rest.

In this paper, challenges during the development of this project and sample design structure will be discussed in detail in Section 2. Experiments implemented to examine the validity of this program will be discussed in section 4. Published works similar to or related to this project will be discussed in section 5. Finally, concluding remarks as well as future perspectives will be discussed in section 6.

2. CHALLENGES

Building a mobile application for the first time with a new programming language is not an easy task. Throughout the course of the development, we ran into several challenges that needed overcoming. Here is a brief overview of some of the most difficult challenges that we faced when developing this app.

2.1. Challenge 1

Initially we intended to use existing biology models on the market. However, after research there's not sufficient models that fit my education purpose of demonstration to students. For example, the neuron structures we found on sketchfab are relatively accurate however complex models, that different parts like axons and dendrites are hard to identify for students [18]. To solve this, we decided to build my own 3D models. We decided to build the models in blender,

since blender is a free and open-source graphic software that gives users a lot of freedom and flexibility to model anything they want. While crafting the 3D models, we realized I need to craft a 3D model by adjusting each point, edge, and face. This limits the complexity of my models since we did not have matured 3D modeling skills. Therefore, we re-chose the biological molecules for modeling.

2.2. Challenge 2

There are also issues in constructing 3D models themselves [19]. For example, originally, we intended to use two cylinders and a UV sphere to construct each single phospholipid to create a phospholipid bilayer model for the action potential demonstration, but there is a gap in the connection of the cylinder and the UV sphere. After we deleted one face of the UV sphere, the open-space is a polygon instead of a circle, which the cylinder doesn't fit perfectly into. Therefore, we changed the construction method to that like a tree branch. I extended one face of the UV spear to make it protrude out, and we adjusted the protruded part to make it look like the phospholipid tail.

2.3. Challenge 3

There were lots of problems during the coding process as well. At first, the objects did not turn at the desired speed, so different parameters were tried for x, y, and z to get the preferable turning. When testing the capture function, the 3D model appeared but it did not show its full feature. To solve this issue, camera position and the image to model ratio was adjusted. This allowed the model to appear right at the place of the image and be the appropriate size that can rotate properly and show the full feature of the models. These are just examples of the changes and improvements, and in my mind one most interesting aspect of constructing an application is constantly refining it.

3. SOLUTION

3.1. Augmented Reality Model

We used Vuforia Engine to create model target of biological molecules. In model target, we do not need to scan the real object instead we use computer-aided design (CAD). Using 3D data makes the recognition and tracking of the object more powerful and faster. To create the model target we require Unity 3D, which is a Vuforia Model target generator, physical object, matching 3D CAD model and Vuforia Engine.

There are some characteristics that we need to consider when we create our 3D biological molecule objects to augment and optimize our AR model for better performance. First, the objects must have sufficient geometric details and the object must be rigid. Also, we reduce our 3D model down just to the part that the camera can identify to recognize the object.

To make our augmented reality application, we used Blender, Unity and Vuforia. (See figure 1) First, we created 3D model of Biological molecules by using Blender; In Blender, we used UV wrapping process to imitate the texture and material of biological molecules. Then we created our Vuforia database that contains target images (biological molecules). In this database we upload the image of biological molecules in png and jpg format.[9] After we created the image target and 3D models, we imported the target images and the 3D models to Unity, so the 3D models can be mapped with the target images. [10][11]

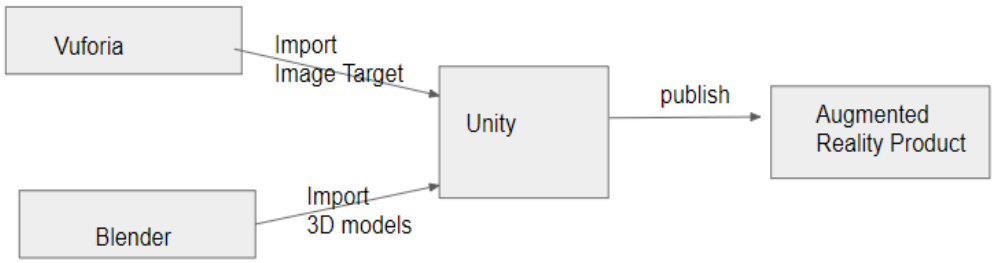


Figure 1: Components of Augmented Reality Model.

3.2. Computer Vision Mobile Phone

We used computer vision library Vuforia to recognize the 2D image. Vuforia provides diverse feature detectors such as Scalar-Invariant Feature Transform (SIFT) and Speeded up Robust Features (SURF) [16]. In order to get the most accurate approach, we analysed 3 different detection algorithms and compared their result. The approaches that we have chosen are Features from Accelerated Segment Test (FAST), Difference of Gaussians, and Determinant of Hessian. FAST is a corner detection algorithm that focused on efficiency [15]. Difference of Gaussians is an algorithm that involves the subtraction between 2 Gaussian kernels with different standard deviations [17].

3.3. Mobile Phone Application

Our solution is a program built based on Unity that can scan and recognize specific pictures and demonstrate a 3D model along with explanation of the corresponding image. This application can be used in different platform such as Android, iOS and Windows. This application contains 3 different levels of depths. (see Figure 2) First there is a menu page, with three buttons “capture”, “collection”, and back. When a user clicks on the capture button, the application enables the camera of the phone, so the application can recognize the specific 2D biological molecule image. After the specific image is recognized, the page is updated with the biological molecule information and its corresponding 3D model of the image. For example, in figure 3, we can see in the right side the 3D Brain image on top of the 2D image of the Brain. The 3D model is pre-built, and images are preloaded into the database. I scanned some images from the Campbell AP Bio textbook chapter 48(Neurons, synapses, and signaling), and built corresponding models.[2]

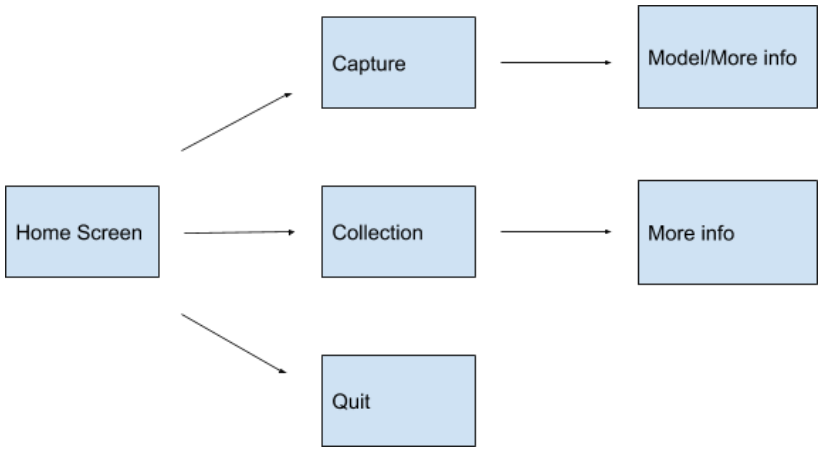


Figure 2: Overview of 3D Biological molecule image User Interface

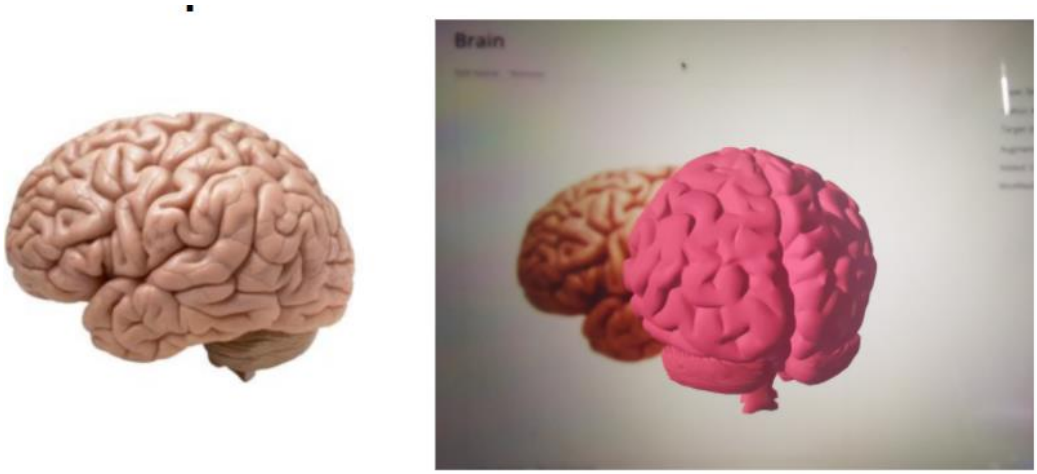


Figure 3: 2D image of a brain image from a book(left). 3D image of a brain using Augmented - reality (right)

Apart from scanning, users can also see the collection when clicked on the “collection” button, where there is a list of buttons users can scroll through to select the model they want to view. (See Figure 4) When clicked on the specific model, the page is directed to the 3D model page. In Figure 5, we can see that after the user clicked in one of the categories, the information of the chosen biological molecule is showed on the screen with its specific image. On the left and right, we can observe protein channel and neuron information with its corresponding image, respectively.

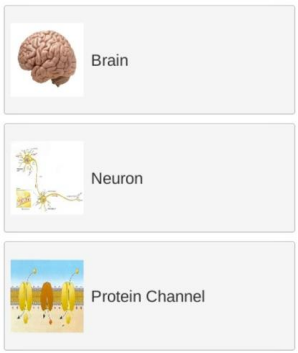


Figure 4: Collection Menu

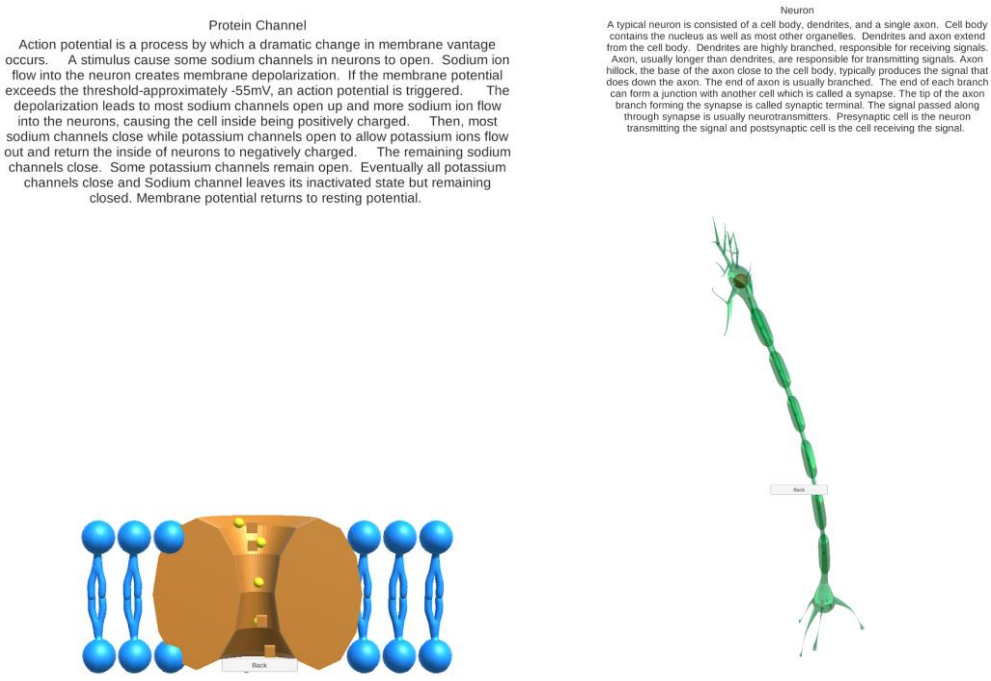


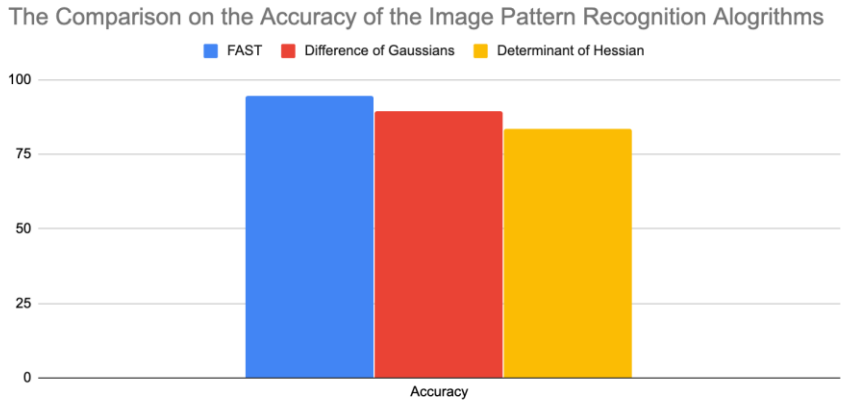
Figure 5: More Information Screen. Protein Channel (Left). Neuron (right).

4. EXPERIMENT

Two experiments have been conducted to measure the performance of the system.

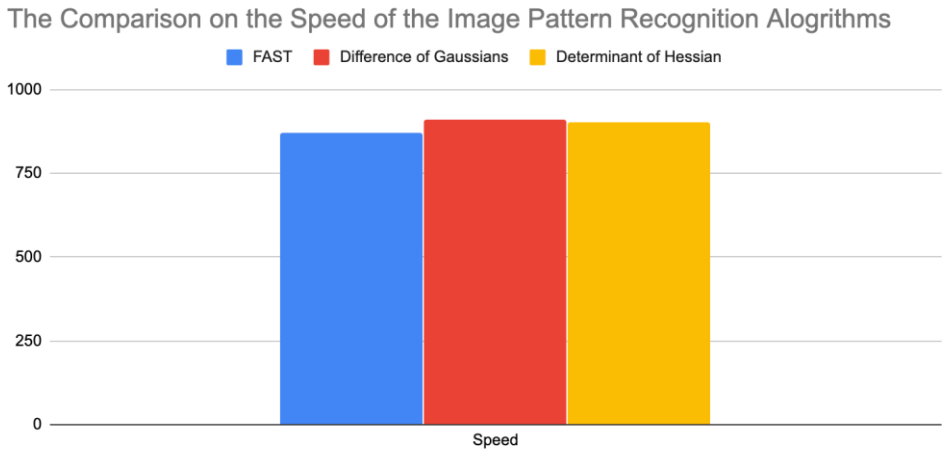
4.1. The accuracy of the image pattern recognition

The core experience and performance of the AR application is based on the accuracy of the image pattern recognition based on the view captured by the camera. We have tested the accuracy of the image pattern recognition using different algorithms. The core image pattern recognition algorithm relies on the image feature extraction methods. Different feature extraction algorithms to recognize the image pattern based on the chosen sampled test images are experimented, such as FAST, Difference of Gaussians, and Determinant of Hessian. It turns out that FAST has the highest accuracy of 94.5% as shown in the chart below.



4.2. The speed of the image pattern recognition

The speed of image pattern recognition using different algorithms. In order to guarantee a smooth user experience, the recognition algorithms should be fast enough to provide a real-time experience. We tested the training dataset collected using different feature extraction algorithms to measure the performance. Although most of the speed are like each other, the FAST is on average faster than the rest as shown in the Figure below.



5. RELATED WORK

Frantz, T. et all addressed the data tracking issue of Microsoft's HoloLens, a tool in neuronavigation, by using Vuforia and Unity[1]. They demonstrated that HoloLens with Vuforia achieves better result and hologram stability than using it without Vuforia. In this work, we used Vuforia to recognize biological molecule images and Unity to create our application.

Amin, D. and Govilkar, S. presented a comparative study of Augmented Reality SDK's such as Metaio SDK, Vuforia SDK, Wikitude SDK, D'Fusion SDK and ARToolKit SDK. [3] They compared them based on license type, platform supported, marker generation, tracking and overlaying capability. Even though Vuforia has some disadvantages such as it does not provide utility function that can load 3D model from diverse formats, we decided to use it since it can support different platforms (Android, iOS and Windows).

Takala T. et all implemented a 3D modeling with Blender.[6] In their study, professional and novice 3D artists used their program to create 3D models. Even though the application was easy and fun to use, it has posing accuracy issues. In our work, we used Blender since it works very well with Vuforia and Unity.

Liu X. et all described how to use Vuforia and Unity to design a game.[8] They explained how to set up Environment Development and Production Process of AR Tower Defense Game. First, the explained step by step of how to establish the environment. Then the step that you must follow to Process of AR Tower Defense Game Generation. Finally, how to generate the game. In their application they can play video, set animation, and interact by using virtual button mode.

6. CONCLUSION AND FUTURE WORK

Learning biological molecules can be very difficult and challenge for learners since they need to try to visualize the biological molecules so they can understand their functionality. They can only use 2D images that are provided in Biology books or Sketchfab and Turbosquid – platforms that buy, sell, share and publish 3D content model- and 3D physical biological molecule models. However, these methods are inefficient and costly. In order to solve this issue, we designed an Augmented-Reality application for Biological molecules that students can use to learn and study biological molecules. Our application has a collection mode in which users can access to a database that contains Biological molecules. In this database, there are information of biological molecules and their corresponding picture. Another option is the capture mode in which learners it to visualize a biological molecule in 3D. This capture mode uses the camera to recognize the 2D image of a biological molecule as input and output the 3D image and information of the target image.

To develop, the AR application, we used Vuforia, Blender and Unity. We decided to use Vuforia, an AR software, since it supports iOS, Android and Window platforms, which were our target platforms. Also, we decided to use Blender, a 3D modeling, and Unity, a game engine, to show the visualization of the biological molecules and they work very well with Vuforia.

We used 3 different algorithms: FAST, Difference of Gaussians, and Determinant of Hessian for the image pattern recognition. Our experiments show that FAST algorithm is more effective and efficient since it is faster than other algorithm and provides highest accuracy of 94.5%.

There are some limitations like the range data in Vuforia database and 3D models are limited. As result, we plan to construct more 3D models for a wider range of biology molecules and integrate the models into the software in the next version. In addition, we plan to add a feature that create animation effect and labels for the 3D models, so that learners can understand and study biological molecules.

REFERENCES

[1] Frantz, Taylor, et al. "Augmenting Microsoft's HoloLens with vuforia tracking for neuronavigation." *Healthcare technology letters* 5.5 (2018): 221-225.
[2] Urry, Lisa A, Michael L. Cain, Steven A. Wasserman, Peter V. Minorsky, and Jane B. Reece. Campbell Biology. New York: Pearson, 2017. Print.
[3] Amin, Dhiraj, and Sharvari Govilkar. "Comparative study of augmented reality SDKs." International Journal on Computational Science & Applications 5.1 (2015): 11-26.
[4] Linowes, Jonathan, and Krystian Babilinski. Augmented Reality for Developers: Build practical augmented reality applications with Unity, ARCore, ARKit, and Vuforia. Packt Publishing Ltd, 2017.

[5] Flavell, Lance. Beginning blender: open source 3d modeling, animation, and game design. Apress, 2011.
[6] Takala, Tuukka M., Meeri Mäkäräinen, and Perttu Hämäläinen. "Immersive 3D modeling with Blender and off-the-shelf hardware." 2013 IEEE Symposium on 3D User Interfaces (3DUI). IEEE, 2013.
[7] Hess, Roland. The essential Blender: guide to 3D creation with the open source suite Blender. No Starch Press, 2007.
[8] Liu11, Xinqi, Young-Ho Sohn, and Dong-Won Park. "Application Development with Augmented Reality Technique using Unity 3D and Vuforia." International Journal of Applied Engineering Research 13.21 (2018): 15068-15071.
[9] Peng, Fuguo, and Jing Zhai. "A mobile augmented reality system for exhibition hall based on Vuforia." 2017 2nd International Conference on Image, Vision and Computing (ICIVC). IEEE, 2017.
[10] Glover, Jesse. Unity 2018 augmented reality projects: build four immersive and fun AR applications using ARKit, ARCore, and Vuforia. Packt Publishing Ltd, 2018.
[11] Borycki, Dawid. Programming for Mixed Reality with Windows 10, Unity, Vuforia, and UrhoSharp. Microsoft Press, 2018.
[12] McCue, T. J. "How to find cool things to 3D print: Sketchfab, MyMiniFactory, Thingiverse." (2019).
[13] Squid, Turbo. "3D Models, Plugins, Textures, and more at Turbo Squid."
[14] Vuforia, Vuforia - enable your apps to see, https://www.vuforia.com/
[15] Gao, Qing Hong, et al. "A stable and accurate marker-less augmented reality registration method." 2017 International Conference on Cyberworlds (CW). IEEE, 2017.
[16] Andreasson, Karl Johan. "Stabilizing Augmented Reality views through object detection." (2017).
[17] Lv, Yaqi, et al. "Difference of Gaussian statistical features based blind image quality assessment: A deep learning approach." 2015 IEEE International Conference on Image Processing (ICIP). IEEE, 2015.
[18] Msanjurj, PatrickZhiaran, Préfontaine, A., Uspalenko, V., Zoilo, Budi, . . . Mumladze, N. (n.d.). Publish & find 3D models online. Retrieved September 09, 2020, from https://sketchfab.com/
[19] Bartonek, Dalibor, and Michal Buday. "Problems of Creation and Usage of 3D Model of Structures and Theirs Possible Solution." Symmetry 12.1 (2020): 181.
[20] Cooper, A. Kat, and M. T. Oliver-Hoyo. "Creating 3D physical models to probe student understanding of macromolecular structure." Biochemistry and Molecular Biology Education 45.6 (2017): 491-500.

A CONTEXT-AWARE AND GEO-BASED MOBILE APPLICATION TO AUTOMATE THE NOTIFICATION OF PUBLIC HEALTH ISSUES

Angela Xiang[1] and Yu Sun[2]

[1]Portola High School, Irvine, California, USA
[2]California State Polytechnic University, Pomona, California, USA

ABSTRACT

Coronavirus disease 2019 (COVID-19) is causing an ongoing pandemic. Social distancing and quarantine are the few effective methods to reduce the spreading risk of the coronavirus among people. As business starts to open up and quarantine policies become looser, the risk of COVID-19 spreading becomes greater [1]. This paper describes the development of a context-aware and geo-based mobile application to automatically track an individual's surroundings and calculate the exposure risk to a public health issue, such as COVID-19. Our application uses other user's data and online databases with information on COVID-19 cases to calculate a percentage revealing the user's possible risk at that location.

KEYWORDS

Flutter, Python Flask, Firebase, iOS, Android.

1. INTRODUCTION

Coronaviruses are a group of RNA viruses that usually cause infections in the respiratory tracts [2]. Most coronaviruses lead to mild illnesses, such as the common cold. However, some coronaviruses can be fatal. The SARS-CoV-2 virus caused an COVID-19 outbreak in China that quickly spread around the world in early 2020. The virus spreads primarily through person-to-person contact, which leads to respiratory tract infections that can affect the sinuses, nose, throat, windpipe, and lungs [3]. Within the span of a few months, the SARS-CoV-2 virus spread to 215 countries, causing many of these countries to enforce strict isolation policies [4]. As these policies loosen up and daily life resumes, the risk of SARS-CoV-2 spreading becomes greater. This study aims to develop a mobile application that will assist enforcing social distancing measures by calculating users' risks of contracting the virus based on their geographical locations and surroundings based on big data analysis. Users will be notified when they are in high-risk areas, and using this application, they can seek areas with lower risk levels. Providing real-time information on the risk level associated with a certain location may help slow the spread of SARS-CoV-2 virus. Measures such as social distancing and quarantine have been adopted in many countries around the world to slow the COVID-19 spread. Since this virus spreads primarily through person-to-person contact, isolating people in their own homes and maintaining a 6-feet distance while in public helped to slow the spread of the SARS-CoV-2 virus [5]. However, as businesses begin to open up and isolation measures begin to loosen, the risk for COVID-19 exposure may increase again.

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 119-130, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101511

Another measure proposed to slow the COVID-19 spread relies on facial masks, protective equipment, and frequent hand-washing. The idea is that these protective measures can prevent droplets from traveling and potentially spreading the virus into new hosts [6]. Facial masks provide effective protection since they shield the nose and mouth from droplets that contain bacteria and viruses. A good habit of thorough hand-washing, for at least 20 seconds, is also helpful to prevent the spread of the SARS-CoV-2 virus from hands to others [7]. Ideally, 100% of the population would wear masks or other protective gear, but in reality, many people do not wear masks due to lack of awareness or other reasons [6]. This increases the exposure risk to more people to the virus. The Institute of Health Metrics and Evaluation suggests that around 33,000 deaths can be avoided by October if 95% of people wear a mask [6]. Hospitals are also keeping medical records of patients infected with SARS-CoV-2. These medical records are used by public health workers to trace infections up the chain of command to learn how the disease spread [5]. This method, known as contact tracing, allows professionals to control the spread of disease. However, these medical records are not available to the general public, and cannot help people to identify when and where they may be at risk.

Our proposed method is a mobile application that provides personalized real-time information on geo-based risks of potential COVID-19 exposure with context-aware features. The application uses geographic location data to gather information about the user's surroundings, such as potential hot spots and large gatherings, as well as user data to find nearby users with COVID-19 diseases. When the user's position changes, the application recalculates a new risk percentage according to the new environment, providing the user with real-time updates calculated specifically for that user. Our application also provides the user with the risk at nearby locations, allowing users to scout for potentially safer locations if necessary. Many existing methods utilize data from medical records or confirmed COVID-19 cases, however some of such data is not available to the general public and is not updated frequently. Our application uses geo-location and self-reported health data to predict and calculate risks for users. This allows our application to rapidly generate a real-time risk that is updated whenever and wherever the user moves to a new location. Our application is also widely available to the general public, and anyone who has a smartphone can use it anywhere at any time. Therefore, we believe that this application could help users to reduce the potential exposure risk to COVID-19 by providing real-time warning to users of the calculated risk at their location.

In the case where geo-location data is limited or unavailable, we demonstrate how the combination of user data and a machine-learning algorithm predicts the user's risk. We experimented with different models, parameters, and data sets, and we found that the machine learning model algorithm predicts the risk with the highest accuracy. We also created a machine learning algorithm to predict the risk based on users most frequently visited locations. For this algorithm, we also experimented with different models, parameters, and data sets to produce the model with the highest accuracy. These experiments produced more accurate machine learning algorithms so that in the case where the application does not have sufficient data, it can predict a potential risk. Since our application primarily uses geo-location and self-reported user health data, these machine learning models fix the issue of not having sufficient data. Especially for locations where there is limited geo-data and when there are limited users, these experiments increase the accuracy of the risk algorithm. Increasing the accuracy of our risk calculation means that users will better understand their surroundings and be more aware of their risks. By providing more accurate warnings and risks, users can take more social distancing precautions or avoid high risk areas by finding and staying in locations with lower risks. This will help users reduce their risks to potential exposure to COVID-19.

The rest of the paper is organized as follows: Section 2 discusses the challenges that people face when exercising social distancing; Section 3 describes our solution to the aforementioned

challenges in Section 2; Section 4 presents the experiments we did; Section 5 compares our application to other SARS-CoV-2 tracking applications; Finally, Section 6 concludes the current work and gives the directions for future work.

2. CHALLENGES

The current measures to reduce the COVID-19 spread rely on social distancing precautions and CDCdata, however these methods do not provide people with real-time analysis of their surroundings. There is a need for a mobile application that offers a personalized geo-based real-time context-aware risk analysis of the potential diseases that the user is exposed to. However, there are several challenges one has to overcome to develop such a mobile application.

2.1. Challenge 1: How to enable a personalized geo-based real-time analysis?

One challenge in developing the mobile application is to enable a personalized geo-based real-time analysis. Real-time analysis requires not only the live data of the user's geographic locations and surroundings, but also the real-time data of location and the self-declared health information of nearby people. When one of these factors change, the risk is re-calculated, providing a real-time analysis that is up-to-date. This requires a large amount of data and the capability of big data analysis.

2.2. Challenge 2: How to integrate context-aware types of risk features?

The second challenge in developing the mobile application is to enable context-aware features to calculate the risks. This context-awareness may help to increase the precision of the risk by taking the user's surroundings into account. Using the user's position, the application searches the surroundings for possible events or hot spots, such as restaurants, tourist attractions, and malls. If there are events or hot spots nearby, then the risk algorithm will factor that into the risk percentage. This allows the risk to calculate a more precise risk analysis by better understanding the surroundings. With a more precise risk analysis, the user can better understand the surroundings and find a relatively safe area to stay in.

2.3. Challenge 3: How to predict and provide the information for the areas that do not have sufficient data available?

The third challenge in developing the mobile application is to enable risk analysis for areas that do not have sufficient data. For some areas, there could be little data on the COVID-19 spread rate, possible hot spots, or health status of nearby users. Under such cases, the mobile application would use a machine learning algorithm to predict the potential exposure risk. The machine learning algorithm would be able to predict the potential exposure risks by using data on other locations, such as the latitude, longitude, number of hot spots nearby, and number of events occurring nearby.

3. SOLUTION

This study develops a context-aware and geo-based mobile application to automatically track an individual's surroundings and calculate the exposure risk to a public health issue, such as COVID-19. It functions with three main components: the frontend UI, backend server, and database. Users first interact with the UI of the application. The frontend of the application consists of the visualization, format, pictures, textboxes, etc. Users first sign in, or if they have

already created an account, they can log in. Once logged in, users can pick locations on Google Map and find the COVID-19 risks at those locations.

The Google Maps on the frontend of the application is connected to a backend server. This component of the application contains algorithms and HTTP APIs to find the user's frequently visited locations, calculate the potential exposure risks, and search for the risks at other locations. The backend server is also connected to a database that stores users' data. Storing latitude, longitude, name, age, email, etc., the database provides the necessary data for the server to calculate a real-time geo-based risk analysis. The database is also connected to the frontend of the application. In storing the login data for each user, it assists in the login and signup process in the UI.

The frontend of the application was developed using Flutter, an open-source UI development kit used to create both the Android and iOS versions of the application. The frontend consists of the visualization, format, pictures, textboxes, etc. When the application is initially opened, the user is directed to a welcome page that gives the option to either "Sign Up" or "Login" (Figure 1). For first-time users, the "Sign-Up" button allows them to create new accounts. Clicking the "Sign-Up" button directs users to another screen that allows them to input their name, age, email, and password, creating a new user in the system (Figure 2). For returning users, the "Login" button allows access to the existing account, where the user's data is logged. Clicking the "Login" button brings users to the "Login" Page (Figure 3), where they can type their emails and passwords to access their accounts.

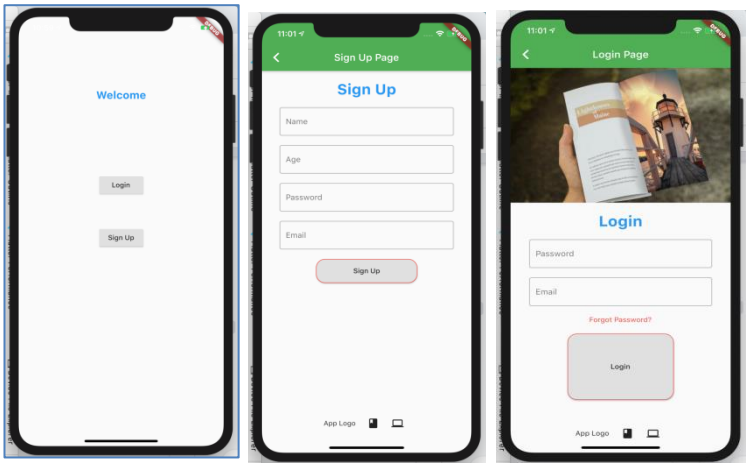


Figure 1. Welcome Page Figure 2. Sign Up Page Figure 3. Login Page

The images of Welcome Page, Sign Up Page, and Login Page of the mobile application.

After signing up or logging into the account, the user can access three pages through the bottom navigation bar. The first page is the "My Locations" Page, where users can see a Google Map with pins that displays the user's commonly visited locations (Figure 4). Clicking on each pin will reveal a textbox that shows the location's name, description, as well as risk factor calculated by the app. The second page is the "Nearby" Page, which shows users a Google Map centered around their current location with pins displaying the surrounding locations with a high potential risk factor, such as restaurants, schools, shopping malls, tourist attractions, etc (Figure 5). Similar to the

"My Locations" Page, clicking on the pins will open a textbox revealing the name, type of hot spot, and risk factor of the location. The third page is the "Profile" Page, where users can see

their profile information, such as the name, age, and diseases they may have (Figure 6). On this page, users also have the option of editing their disease information by turning on a switch and entering the diseases that they have.

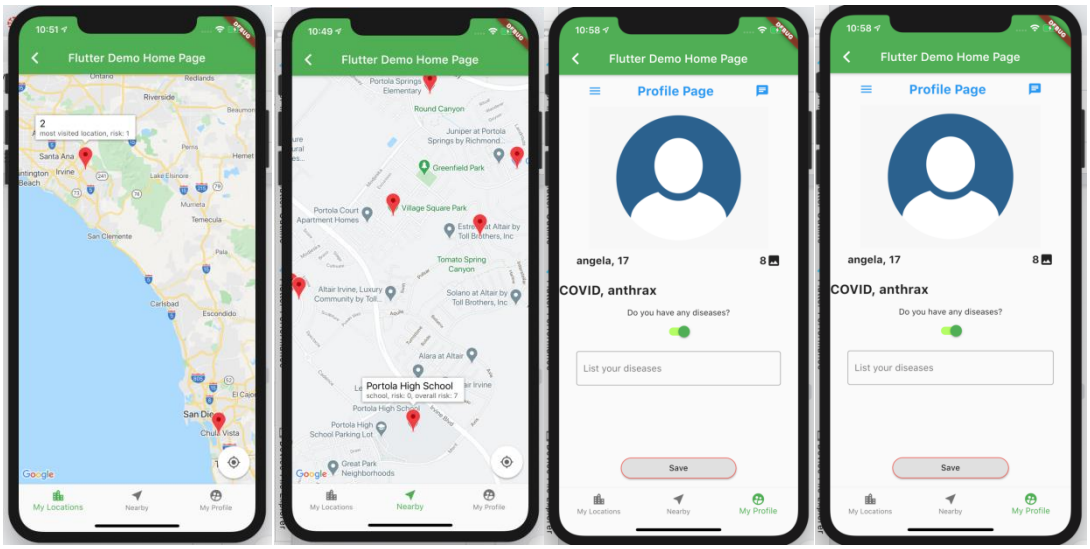


Figure 4. "My Locations" Page

Figure 5. "Nearby" Page

Figure 6. "Profile" Page

Figure 7. "getNearbyUsers" Page

The images of "My Locations", "Nearby", "Profile", and "getNearbyUsers" Pages of the mobile application.

The backend server of the application was created using a Python Flask server to create 4 main HTTP APIs. Python Flask is a micro web framework, which routes an HTTP request to the specified controller. This calls a function in the server, and an HTTP response is then returned. The "My Locations" and "Nearby" Pages use this Python Flask server to find nearby users, find nearby hot spots, retrieve a user's most visited locations, and calculate a risk percentage for the user. Within the Python Flask server, there are four main HTTP APIs that carry out these tasks. The first HTTP API is called getNearbyUsers, and it does the task of getting nearby users (Figure 7). This API uses three parameters, which are the user's current latitude position, longitude position, and a given radius to search for nearby users. The function then loads all users by retrieving the data from the Firebase database. By using the Haversine formula that calculates the distance between two points on a sphere, the function then returns the users that are located within the specified radius of the user.

The second HTTP API is called getNearbyHotPlacesWithAllTypes, and it gets the nearby hot spots (Figure 8). This function uses three parameters: the user's current latitude, current longitude, and a specified radius. The function first specifies the types of hot spots, such as bars, restaurants, tourist attractions, malls, etc. The function then loads the hot spots that are located within the specified radius around the user's location using the Google Maps API. Finally, the function returns the hot spots, as well as the risk at that hot spot.

The third HTTP API is called getMostVisitedLocations, and it gets the user's most frequently visited places (Figure 9). This function loads the user's data from the Firebase database, and it counts the number of times the user visited each location. Counting the number of visits, the function returns the locations with the highest number of visits.

The last API, get_location_risk, calculates the risk for each location (Figure 10). It works by taking the current location from a user, and it checks if another user with a disease is within a specified radius of the user. For every user nearby that has diseases, the risk increases by one. The function then returns the risk for the user's current position.

This application uses a Firebase database to store users' login, profile, and location information. The Firebase Real time Database stores and syncs user data from the application. Firebase stores this user data in JSON format, which consists of attribute-value pairs and array data types. Within the database, there are two main branches in which the data is sorted: users and logs. The users branch is further categorized into smaller branches for each user, which is labeled with a unique uid. Within these individual users' branches, the name, age, email, and diseases are stored. The other main branch, logs, tracks the user's location. Within the logs branch, there are also branches for each user labeled with the user's unique uid. These branches contain the latitude and longitude of the user at different time stamps, creating a log of all locations that the user has visited.

```
@app.route('/getNearbyUsers/<cur_lat>/<cur_lon>/<radius>')
def get_nearby_users(cur_lat, cur_lon, radius):
    cur_lat = float(cur_lat)
    cur_lon = float(cur_lon)
    radius = float(radius)
    print(cur_lat, cur_lon, radius)
    # 1. get all the user data from the firebase database
    # 2. return all the users near by based on a given radius
    result = json.loads(
        requests.get(
            'https://coronavirus-app-1637b.firebaseio.com/users.json').text)
    users_list = []

    for key in result:
        # print(key)            # key
        # print(result[key])    # value
        if 'longitude' in result[key] and 'latitude' in result[key]:
            # print(result[key]['longitude'])
            # print(result[key]['latitude'])

            # filter out the users who's distance is out of the radius range
            if distance(cur_lat, result[key]['latitude'], cur_lon,
                        result[key]['longitude']) <= radius:
                users_list.append(result[key])

    print(users_list)
    return json.dumps(users_list)
```

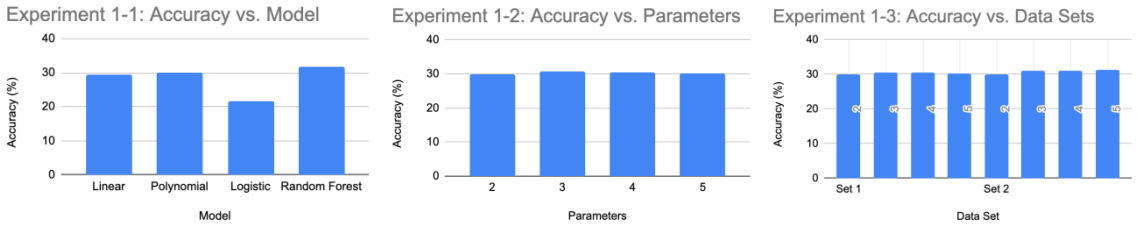
Figure 8. get Near by Hot Places With All Types

```python
@app.route('/getNearbyHotPlacesWithAllTypes/<cur_lat>/<cur_lon>/<radius>')
def getNearbyHotPlacesWithAllTypes(cur_lat, cur_lon, radius):

    types = [
        'bar', 'restaurant', 'school', 'airport', 'university', 'hospital',
        'supermarket', 'store', 'shopping_mall', 'tourist_attraction'
    ]
    nearby_places = []

    # call the server to get all the list
    # and SAVE it here in the variable
    user_list = get_all_users_object()
    overall_risk = 0

    for type in types:
        res = requests.get(
            "https://maps.googleapis.com/maps/api/place/nearbysearch/json?location="
            + cur_lat + "," + cur_lon + "&radius=" + radius + "&type=" + type +
            "&key=AIzaSyCxFM696RNw-aqQmM67jA7-7LogJ9uBUt0")
        print(
            "https://maps.googleapis.com/maps/api/place/nearbysearch/json?location="
            + cur_lat + "," + cur_lon + "&radius=" + radius + "&type=" + type +
            "&key=AIzaSyCxFM696RNw-aqQmM67jA7-7LogJ9uBUt0")
        res_obj = json.loads(res.text)
        # print(res_obj['results'])
        list_places = res_obj['results']

        for place in list_places:
            # print(place['name'])
            # print(place['geometry']['location']['lat'])
            # print(place['geometry']['location']['lng'])
            loc_risk = get_location_risk(
                user_list, place['geometry']['location']['lat'],
                place['geometry']['location']['lng'], 1)
            new_place = {
                'name': place['name'],
                'type': type,
                'risk': loc_risk,
                'lat': place['geometry']['location']['lat'],
                'lng': place['geometry']['location']['lng']
            }
            nearby_places.append(new_place)
            overall_risk += loc_risk

    res = {'overall_risk': overall_risk, 'places': nearby_places}

    return json.dumps(res)
```

Figure 9. get Most Visited Locations

4. EXPERIMENT

To evaluate the accuracy of the risk algorithm, we experimented with the different models, parameters, and data set features to find the most accurate machine learning model. The first experiment was conducted to find a machine learning model that predicts the risks for a new geo-location without available data. This experiment has three parts: experiment 1-1 tests different regression models, experiment 1-2 tests different polynomial parameters, and experiment 1-3 tests different data set features. The machine learning model was created with 5000 data sets for each experiment. The accuracy of each experiment was calculated using the machine learning algorithm and comparing the prediction with the actual risk. For experiment 1-1, different machine learning models were applied to the same data set to find out which model would produce the most accurate algorithm. The models used are linear, polynomial with a power of 2, logistic, and random forest regressions. Experiment 1-2 tested the polynomial model parameters 2, 3, 4, and 5, and tested which model would have the highest accuracy. Experiment 1-3 tested two data sets: one set with four inputs (latitude, longitude, number of hotspots, and number of events) and the other set with two inputs (number of locations and number of events).

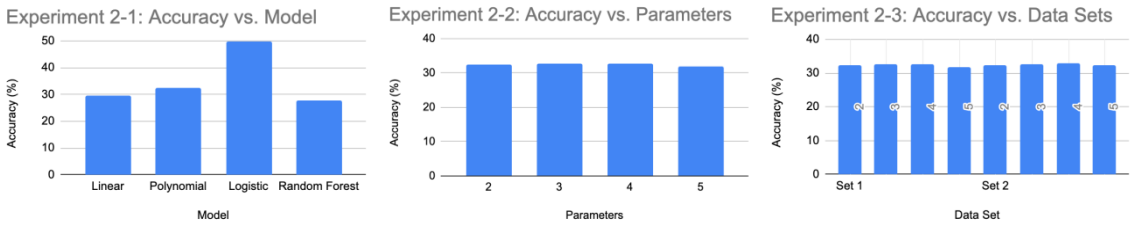


(a) (b) (c)

Figure 10. The results of experiment 1

The results of experiment 1-1 can be seen in Figure 10. The Random Forest model has the highest accuracy at 31.772%. In comparison, the Logistic model has the lowest accuracy of the four models, with an accuracy of 21.6%. This proves that the geo-based risk does not follow a Logistic model. The results of experiment 1-2 reveals that 3 parameters in the polynomial model produces a higher accuracy at 30.568%, but not by much. The models with 2, 4, and 5 parameters have similar accuracies to the model with three parameters, with the least accurate model (2 parameters) being less than 1% away from the most accurate (3 parameters). As mentioned previously, experiment 1-3 compares two data sets with different inputs and different parameters. The most accurate model was produced using Data Set 2 and 5 parameters, and on average, Data Set 2 produced more accurate models than Data Set 1.

The second experiment was conducted to find a machine learning model that predicts the risks for a user based on the most-visited locations. This experiment has three parts: experiment 2-1 tests different regression models, experiment 2-2 tests different polynomial parameters, and experiment 2-3 tests different data set features. The machine learning model was created with 5000 data sets for each experiment. The accuracy of each experiment was calculated using the machine learning algorithm and comparing the prediction with the actual risk. For experiment 2-1, different machine learning models were applied to the same data set to find out which model would produce the most accurate algorithm. The models used are linear, polynomial with a power of 2, logistic, and random forest regressions. Experiment 2-2 tested the polynomial model parameters 2, 3, 4, and 5, and tested which model would have the highest accuracy. Experiment 2-3 tested two data sets: one set with four inputs (type of location, number of nearby hot spots, number of visits, and the duration of each visit) and the other set with three inputs (type of location, number of hotspots, and duration of each visit).



(a) (b) (c)

Figure 11. Results of experiment 2

The results of experiment 2-1 can be seen in Figure 14. The Logistic model has the highest accuracy at 49.8%. In comparison, the Random Forest model has the lowest accuracy of the four models, with an accuracy of 27.721%. This proves that the geo-based risk does not follow a Random Forest model. The results of experiment 2-2 (Figure 11) reveals that 4 parameters in the polynomial model produces a higher accuracy at 32.649%, but not by much. The models with 2,

3, and 5 parameters have similar accuracies to the model with four parameters, with the least accurate model (5 parameters) being less than 1% away from the most accurate (4 parameters). As mentioned previously, experiment 2-3 (Figure 11) compares two data sets with different inputs and different parameters. The most accurate model was produced using Data Set 2 and 4 parameters, and on average, Data Set 2 produced more accurate models than Data Set 1.

Experiment 1 tried to find the most accurate model for predicting risks for new geo-locations without prior data. The most accurate algorithm was one with a Random Forest model, 3 parameters, and using Data Set 2 (a data set with 2 inputs, specifically the number of hot spots and number of locations). Experiment 1-1 proved that using a Random Forest model resulted in a higher accuracy, and a Logistic model had the lowest accuracy, revealing that the geo-location vs. risk does not follow a Logistic pattern. Experiment 1-2 found that a 3-parameter polynomial model had the highest accuracy, but overall, there was not a significant difference between a 2, 3, 4, or 5-parameter polynomial model. Experiment 1-3 found that the model using Data Set 2 and a 5-parameter polynomial model was the most accurate. This is interesting since in experiment 1-2, the 3-parameter model was the most accurate, not the 5-parameter model. This is perhaps due to the fact that in experiment 1-2, the models had very similar accuracies that only deviated by less than 1%. Experiment 1-3 also reveals that on average Data Set 2 had a higher accuracy, suggesting that the input of latitude and longitude in Data Set 1 made the risk algorithm less accurate. Since the risk algorithm did not use the latitude and longitude, it makes sense that Data Set 2 was more accurate. The latitude and longitude position does not directly relate to the risk at the geo-location, and the number of nearby hot spots and events have a greater impact on the risk.

Experiment 2 tried to find the most accurate model for predicting exposure risks based on users' most visited locations. The most accurate algorithm was one with a Logistic model, 4 parameters, and using Data Set 2 (a data set with 3 inputs, specifically the type of location, number of hotspots, and duration of each visit). Experiment 2-1 proved that using a Logistic model resulted in a higher accuracy, and a Random Forest model had the lowest accuracy, revealing that the geo-location vs. risk does not follow a Random Forest pattern. Experiment 2-2 found that a 4-parameter polynomial model had the highest accuracy, but overall, there was not a significant difference between a 2, 3, 4, or 5-parameter polynomial model. Experiment 2-3 found that the model using Data Set 2 and a 4-parameter polynomial model was the most accurate, although there is less than a 1% difference in accuracy between models trained with Data Set 1 and Data Set 2.

5. RELATED WORK

Coveley, M. et al [8] developed a tracking system for infectious diseases. This system consists of transmitter circuits that record the infected people within the confined space. The system works by identifying the people within the location, establishing a record for each person with a time and date, and storing these records [8]. These records can be retrieved to generate an audit record that can help in tracking the disease [8]. Since this tracking system relies on transmitter circuits spaced out over a confined location, it is designed for and best suited to be used in a hospital with medical professionals and potential carriers of the disease inside. Our application also tracks users with diseases; however, it utilizes location data of nearby users to warn someone of potential exposure. This feature allows our application to be used anywhere and anytime as long as the user has a mobile device with them.

Segal, E. et al built an international consortium to track COVID-19 spread [9]. This system relies on participants to self-report COVID-19 symptoms. Participants will also report their geospatial location, time, age, demographic information, and pre-existing medical conditions [9]. This data is used to help track COVID-19 around the world. This is very similar to our application, since

both rely on self-reported data and a large group of users/participants. However, our application takes this one step further by using the self-reported data to calculate a personalized risk analysis for users. Not only does the risk analysis utilize other user's self-reported data, but it also uses geo-location data to find nearby hot spots and large events that could increase the risk of COVID-19.

Boulos, M. et al describes a range of online and mobile GIS mapping systems that track COVID-19 [10]. Some of these systems include Johns Hopkins' CSSE dashboard (Figure 12), World Health Organization's dashboard (Figure 13), and HealthMap (Figure 14). All of these systems track COVID-19 cases on a map, as well as figures, tables, and graphs showing the trends. The CSSE dashboard uses diagnosed cases based on symptom array and chest imaging, and the WHO dashboard uses laboratory-confirmed cases [10]. HealthMap uses machine learning and language processing to sift through news reports to track COVID-19 [10]. HealthMap also offers a feature to find "outbreaks near me" [10]. This is very similar to our application, that uses user and location data to find the risk and nearby users with diseases.

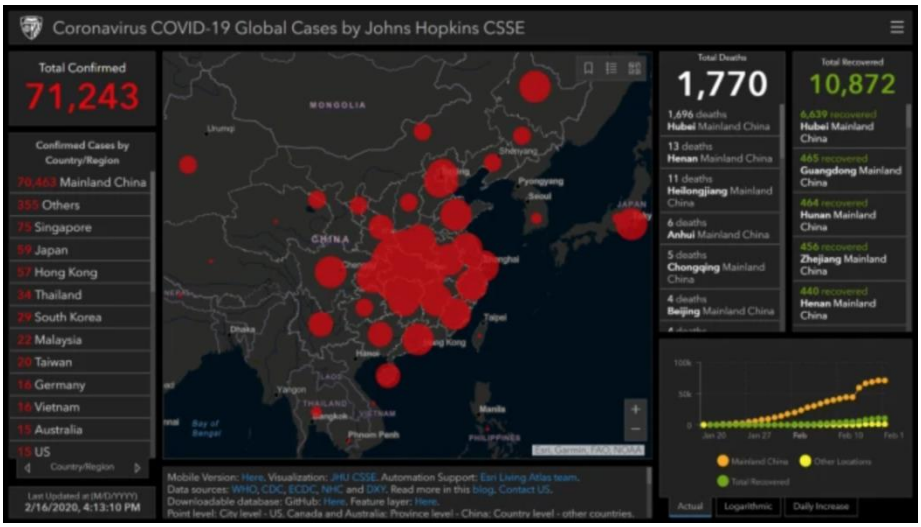


Figure 12. Johns Hopkins CSSE dashboard

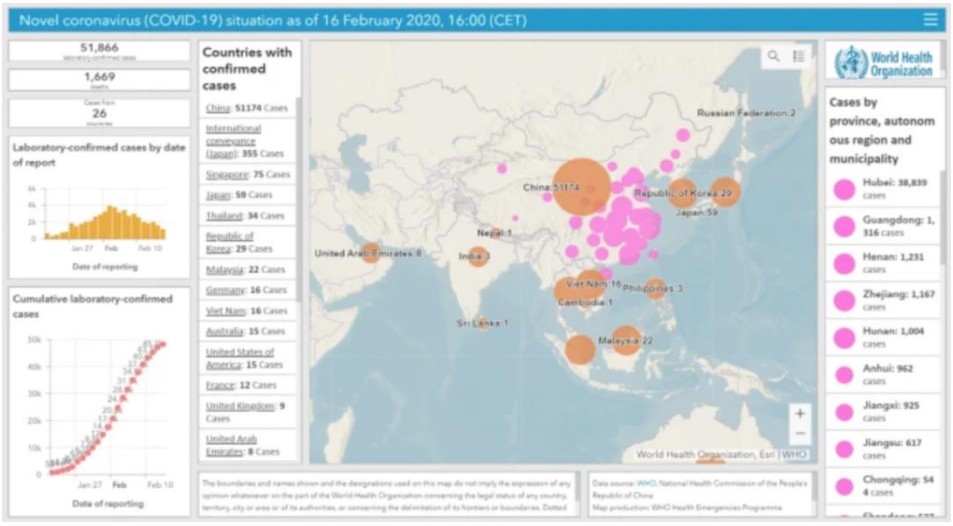


Figure 13. WHO dashboard

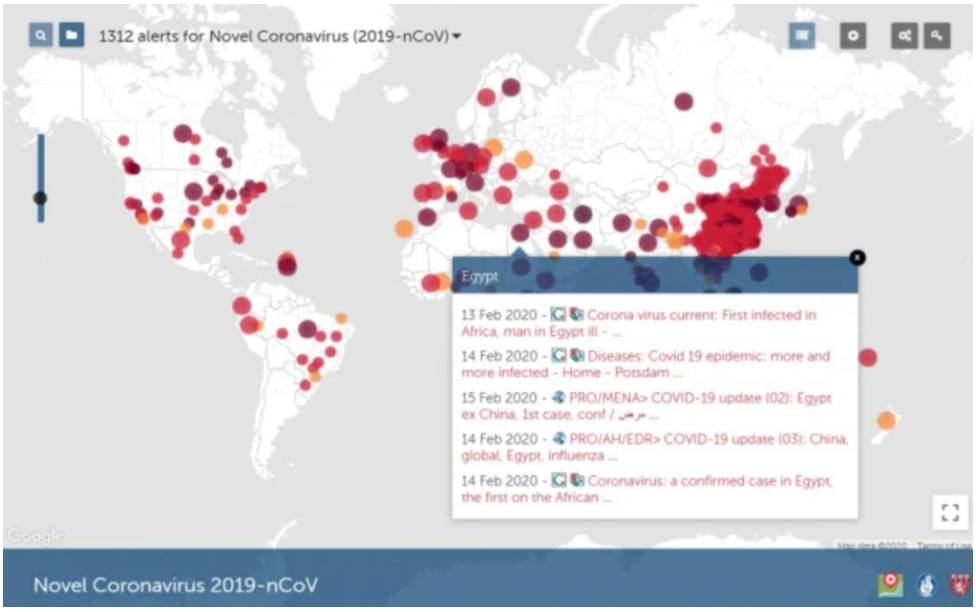


Figure 14. HealthMap

6. CONCLUSION AND FUTURE WORK

In summary, our application tracks COVID-19 among its users and provides a personalized real-time geo-based risk analysis for its users. Using geo-location data and self-reported disease data from users, the application calculates a risk factor with the nearby users, hot spots, and events. By warning users of the calculated exposure risks at their location, this application helps slow the spread of COVID-19. Users are also able to see nearby locations that may have a smaller risk, allowing them to move to a relatively safer location. In cases where the geo-location of a user has no available data, the application would use a machine learning algorithm to predict the risk. Our experiments tried to find the most accurate algorithm by adjusting the regression model, polynomial parameter, and features of the input set. The results show that using a Random Forest model with 3 parameters trained with an input of [number of hotspots, number of events] will result in a more accurate model. The estimation accuracy of these models are between 20 - 30%. With the addition of the machine learning algorithm, this application is able to predict a risk at locations without sufficient data, which greatly expands the coverage of this application.

However, there are some limitations in this application. First, the risk algorithm partially relies on self-reported disease data. If there are not enough users, then the risk might be underestimated, losing the risk algorithm's accuracy. Another limitation is that the machine learning algorithm may not be optimized. The experiment revealed that individually, the best features led to a higher accuracy. But when the best features are combined, the resulting algorithm may not have the highest accuracy. A third limitation is that the risk algorithm could be better optimized using more factors. Currently, the risk algorithm uses self-reported user disease data, nearby hot spots, and nearby large events. This algorithm could be further improved to increase the estimation accuracy on COVID-19 exposure risks.

Further development will address these limitations. The risk algorithm could be further experimented with and developed so that it does not heavily rely on self-reported user data. Using more location data, such as the number of nearby laboratory-identified COVID-19 cases, could

lessen the reliance on many users self-reporting data. Furthermore, conducting experiments that combine the best features could optimize the accuracy of the machine learning algorithm. Testing different combinations of features would lead to the best overall model.

REFERENCES

[1] Johns Hopkins Coronavirus Resource Center (2020) Impact of Opening and losing Decisions by State, https://coronavirus.jhu.edu/data/state-timeline/new-confirmed-cases/california/1

[2] Nat'l. Inst. Allergy & Infectious Diseases (2020) Coronaviruses Overview, https://www.niaid.nih.gov/diseases-conditions/coronaviruses

[3] WebMD (2020) Coronavirus and COVID-19: What You Should Know, https://www.webmd.com/lung/coronavirus

[4] Coronavirus Update (Live) Worldometer Reported Cases and Deaths by Country, Territory, or Conveyance, https://www.worldometers.info/coronavirus/?utm_campaign=homeAdUOA?Si

[5] Centers for Disease Control and Prevention (2020) Monitoring and tracking the disease, https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/about-epidemiology/monitoring-and-tracking.html

[6] Bai, N. (2020) Still Confused About Masks? Here's the Science Behind How Face Masks Prevent Coronavirus,University of California San Francisco, https://www.ucsf.edu/news/2020/06/417906/still-confused-about-masks-heres-science-behind-how-face-masks-prevent

[7] Centers for Disease Control and Prevention (2020) Follow Five Steps to Wash Your Hands the Right Way, https://www.cdc.gov/handwashing/when-how-handwashing.html.

[8] Coveley, M. & Huang, Y. (2010) Method and apparatus for cataloging and poling movement in an environment for purposes of tracking and/or containment of infectious diseases, US Patent No. US 7,817.046 B2. https://patents.google.com/patent/US7817046B2/en

[9] Segal, E., Zhang, F., Lin, X., et al. (2020) "Building an International Consortium for Tracking Coronavirus Health Status", Nat. Med., Vol. 26, pp1161–1165. https://doi.org/10.1038/s41591-020-0929-x.

[10] Kamel Boulos, M.N. & Geraghty, E.M. (2020) "Geographical tracking and mapping of coronavirus disease COVID-19/severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) epidemic and associated events around the world: how 21st century GIS technologies are supporting the global fight against outbreaks and epidemics", Int'l. J. Health Geography, Vol.19, No. 8, https://doi.org/10.1186/s12942-020-00202-8.

[11] Oliver, N., Lepri, B., Sterly, H., Lambiotte, R., et al. (2020) "Mobile phone data for informing public health actions across the COVID-19 pandemic life cycle", Science Advances, Vol. 6, eabc0764, DOI: 10.1126/sciadv.abc0764

[12] Motley, J., Cowls, J., Taddeo, M. & Floridi, L. (2020) "Ethical guidelines for COVID-19 tracing apps", Nature, Vol. 582, pp29 - 31. https://www.nature.com/articles/d41586-020-01578-0

[13] Drew, D., Nguyen, L.H., Steves, C.J., et al. (2020) "Rapid implementation of mobile technology for real-time epidemiology of COVID-19", Science, Vol. 368, pp1362-1367. DOI: 10.1126/science.abc0473.

[14] Couture, V., Dingel, J.I., Green, A.E., Handbury, J., & Williams, K.R. (2020) "Measuring Movement and Social Contact with Smartphone Data: A Real-Time Application to COVID-19", NBER Working Paper No. 27560, https://www.nber.org/papers/w27560.pdf

[15] Ahmed, N., Michelin, R.A, Xue, W., Ruj, S., et al. (2020) "A Survey of COVID-19 Contact Tracing Apps", IEEE Access, Vol. 8, pp134577-134601, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9144194.

SMARTCALLERBOT: A MULTI-LEVEL INCOMING CALL NUMBER DETECTION AND BLOCKING USING CONTEXT-AWARD TECHNIQUE AND ARTIFICIAL INTELLIGENCE

Kaiwen Fu[1], Marisabel Chang[2] and Yu Sun[2]

[1]Fairmont Preparatory Academy, Anaheim, CA 92801, USA
[2]Department of Computer Science, California State Polytechnic University, Pomona, USA

ABSTRACT

Recently, I have received many spam calls every day. My phone number is associated with many essential accounts due to this reason, so I could not change a new phone number. Sometimes the spam call wakes me up at 7 am on the weekend. This situation has sustained from March to June. It bothers my life. I have tried to put them in my phone blacklist, however every time the number call in is different, so my blacklist does not help so much. This paper proposes an application to automatically detect the call content and tell the user what kind of call it is. We applied our application to the call, especially from other states or countries. The results show that the app can detect if the call is a spam call or not.

KEYWORDS

Voice Recognition, Spam detection, Firebase, Artificial intelligence

1. INTRODUCTION

The background of my topic is for people to have a better life, to expect the spam call to bother people's lives [11][12] My argument is very important because sometimes the spam call can worry about people's emotions; for example, when you are having a meeting with someone famous, at this moment, the spam calls in. It will give you an awkward situation. Another example, you try to wake up late at the weekend because you have too much work during the weekdays, you try to give yourself some relaxation; however, a spam call comes in at 7 in the morning. You thought it was your friend or your boss calling you for some emergency; however, when you pick up the phone, it is a spam call, it will drive you crazy and upset. My topic is important because it can solve this kind of problem and give people a better life.

Some of the spam call resistor techniques and systems that have been proposed to remind you this is a spam call, which allows the user to decide to pick up the phone or not [13][14][15] However, these proposals cannot really fix the basic problem of spam call, which is rarely the case in practice. Their implementations are also limited in scale, with samples given for the common app online, they usually just block the call for you but not telling you what's the info about the call. They cannot get the content of the call because their method used cannot be too sophisticated and often results in blocking the wrong call [16] [17] In this paper, we follow the same line of research by detecting the call. Our goal is to pick up the call and detect the content and analyze the call for the user. Our method is inspired by AI voice recognizer. There are some

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 131-138, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101512

good features of using AI to detect spam calls. First it prevents the probability of banning the wrong number. Second, the user can know what the content information was about in the call.

2. CHALLENGES

2.1. Challenge 1

The first challenge we meet is when shall the application pick up the phone or not. This is a challenge because sometimes people do not want the app to automatically pick up any call by themselves. For example, when people are having a meeting, they want the application to pick up any call for them, but when people are waiting for someone to come, they definitely want their phone to ring.

2.2. Challenge 2

The second challenge is how we can make the app record the phone call. There are some challenges that the android system does not allow the application to record the phone call unless the application is verified by the government. However, we need this to be able to work in our application, otherwise the whole system will not work.

2.3. Challenge 3

The third challenge we met is we have to enable the voice recognition system into our application so that when the people are talking on the phone, the application can detect what the person is talking about and get the text format of the call. After we get the voice recognition done, we have to add the keyword search also. We have to have the text version of the call first and then let the application to search some keyword in the call to detect whether the call is spam or not.

3. SOLUTION

3.1. Overview of the Solution

An overview of the system is presented in Figure 1. An incoming phone call comes and the voice call is recorded and sent to database. The system interprets the voice recorded and verifies if the phone call is a spam or not. Finally, the result is sent to the phone application.

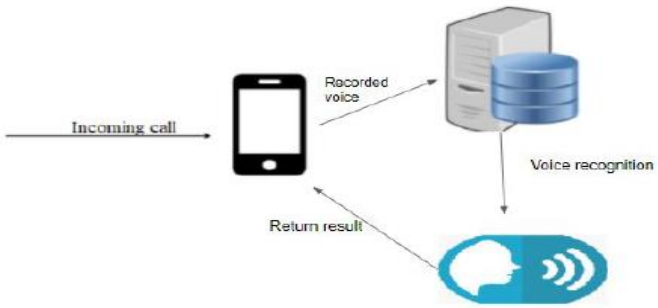


Figure 1. Overview of the Solution

3.2. Data Storage

We utilize Android studio to develop our application and Firebase to store recorded voice. Android Studio is an integrated development environment uses for building application for Android phones, tables, Android TV, and Android Auto [4][5] [7] Firebase is a platform created by Google for developing web and mobile application [1][2]

To store a voice recorder, we created an instance of the Firebase Storage and used this instance to create a storage reference from our application. Then we got the path of the recorded voice and upload the MP4 file to Firebase.

```
static public UploadTask uploadFile(String firebaseDirectory , String path){

    FirebaseStorage storage = FirebaseStorage.getInstance();
    // Create a storage reference from our app
    StorageReference storageRef = storage.getReference();

    Uri file = Uri.fromFile(new File(path));
    // Create a reference to 'images/mountains.jpg'
    StorageReference ref = storageRef.child( firebaseDirectory + file.getLastPathSegment());
    UploadTask uploadTask = ref.putFile(file);

    return uploadTask;

}
```

Figure 2. Upload File to Firebase

3.3. Voice Recognition

We used Python for the system and Speech-to-text from Google Cloud to recognize the voice recorded. Speech-to-text use Google's AI technologies to convert the voice into text [3] [6] In our system, we interpret the voice recorded and verify if the message contains the keywords that demonstrate that the message is a spam or not.

3.4. Android Application

As a shown in Figure 3, the app has a button that enable the blocking spam call. When the blocking spam call is enabled, and the app closes and runs in the background. Also, this screen has the option of change the setting and manage the blacklist.

Figure 3. Turn on Blocking Spam Call

The App has a functionality to add keywords that would be used to verify if the phone call is a spam or not. To set up the spam call App, an user would add keywords that the user would think that a spam phone would contain in its voice message to the call assistant list. (see Figure 4)

Figure 4. List of keywords to be considered as a spam

Because the app is running in the background, the app would notice when an incoming phone call come. Thus, when the incoming phone call ends, the process would process and interprets the voice recorded and return a message that notify if the phone call is a spam or not. If the phone call is an spam a red button appears next to the voice message otherwise a green button appears next to the voice message. (see figure 5)

Figure 5. List of phone call message. Spam phone call (Red button). Real phone call (Green button)

4. EXPERIMENT

At present, the most up to date technologies on voice recognition are Amazon Transcribe, Microsoft Azure Speech and Google Cloud Speech. They allow developers to translate voice into text automatically. We conducted an experiments on the three APIs and made informative comparison and justify our choice.

• Azure Speech to Text

The key feature for Microsoft Azure Speech is that it supports custom speech and acoustic models, which make users to alter original speech recognition under special circumstances.
Also, Azure Speech can deliver speech in real-time. This feature is able to provide timely feedback to users and help them to adjust their speech in some way. Microsoft Azure Speech provides a very popular interface - REST API, which is widely used in application development today.

• Amazon Transcribe

Amazon Transcribe can translate audio file into text, then make useful detection based on the text. Even Amazon cannot translate speech to text in real-time, it can automatically recognize multiple speakers and can show a timestamp.

• Google Cloud Speech

To realize speech to text, google developed its own speech-to-text engine, which process both short audio snippets for voice interfaces and longer audio for transcription. It supports 120 languages in real time or from pre recorded audio files.

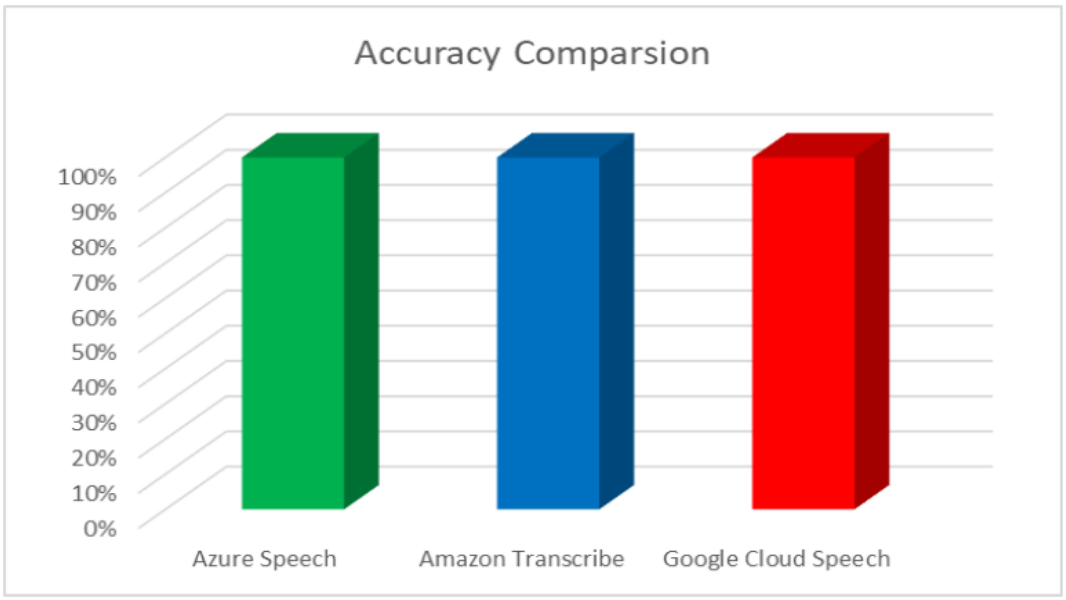


Figure 6. Accuracy Comparison

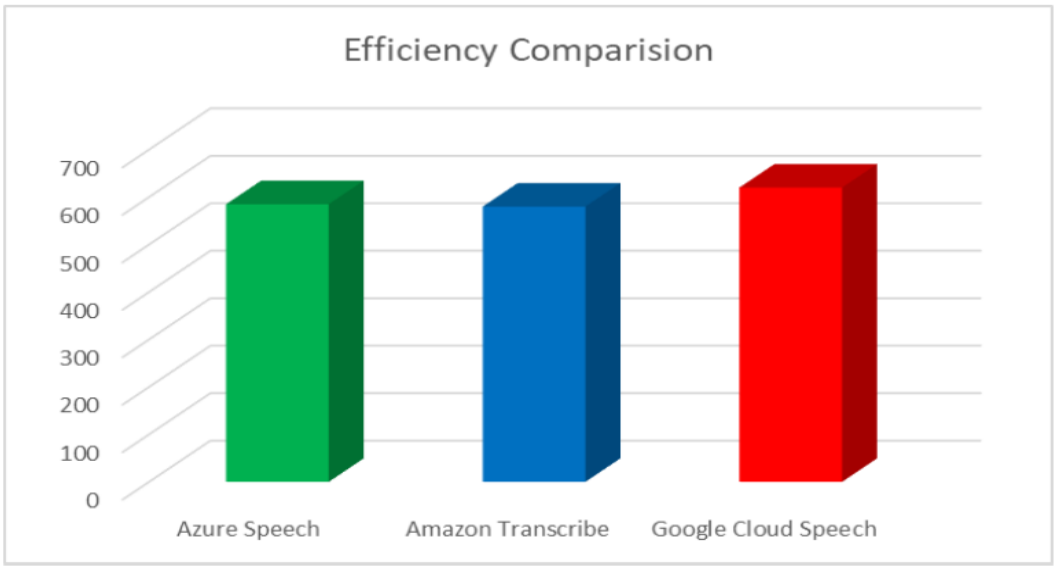


Figure 7. Efficiency Comparison

We did an experiment on the above three APIs and analyze them based on the same input speech. There results show all of them roughly lie in the same level. Google Cloud Speech show a slightly better on efficiency.

5. RELATED WORK

Seneviratne, S., et all presented an Adaptive Boost classifier to identify if an app is a spam or not. [8] They created their training dataset by manually label a sample of removed apps. Then they utilized the metadata of apps to run the classification. In their result, they estimate that at least 2.7 % of the apps at the app market are spam.

Coles, Scott, et al proposed a system to retrieve important information that agents can use to help customer.[9] They developed a voice recognition system that pick up a plurals keywords while the customer and agents are talking on the phone. These keywords are utilized to search information in the database that can be very helpful for the agent and customer. In our project, we used voice recognition to identify if a phone call is a spam or not. As different of this approach, we set up the keywords to interpret a incoming phone call.

Yoav, T. developed a system that displays targeted advertising on end-user device such as a mobile, a static device and/or a computing device.[10] The system utilized voice recognition in speech that is taking from automated systems such as an interactive voice response (IVR) or voice mail system. Then the system picks some keywords from the conversation and identify user's targeting items, such as user past behavior and user's physical location to search possible ads that can be interested to the user. Finally, the system displays the ads on the end-user device.

6. CONCLUSION AND FUTURE WORK

In this project, we proposed an artificial intelligent approach for blocking spam calls. Our mobile app utilizes the recorded voice call and voice recognition system to identify if an incoming phone call is a spam or not. First, the phone call is recorded and sent to the Firebase. Then the speech-to-text gets the voice recorded from the database and interpret the voice recoded. To identify a spam call, we set up keywords that we believe that are associated with spam calls.

As future work, we plan to add phone number of spam calls to our database, so that the system can block a phone call automatically if the phone number is in our database.

One limitation that is related with this app is that it incoming phone call needs to end to verify if it is spam or not. One feature that we plan to add to the app is to have the ability to set a recorded time for the phone call, so that when the incoming call reaches the recorded time, the recorded voice is sent to the Firebase and the system can validate if the incoming call is a spam or not.

In the future, we plan to improve application in efficiency, accuracy, and usability. our application will be able to reach its full potential.

REFERENCES

[1] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.
[2] Moroney, Laurence, Moroney, and Anglin. Definitive Guide to Firebase. Apress, 2017.
[3] Bijl, David, and Henry Hyde-Thomson. "Speech to text conversion." U.S. Patent No. 6,173,259. 9 Jan. 2001.
[4] Zapata, Belén Cruz. Android studio application development. Packt Publ., 2013.

[5] Developer, Android. "Android Developer." línea]. Available: https://developer. android. com (2009).
[6] Ballinger, Brandon M., et al. "Speech to text conversion." U.S. Patent Application No. 12/976,972.
[7] Powar, Swapnil, and B. B. Meshram. "Survey on Android security framework." International Journal of Engineering Research and Applications 3.2 (2013): 907-911.
[8] Seneviratne, Suranga, et al. "Early detection of spam mobile apps." Proceedings of the 24th International Conference on World Wide Web. 2015.
[9] Coles, Scott, et al. "Dynamic information retrieval system utilizing voice recognition." U.S. Patent Application No. 10/191,225.
[10] Tzruya, Yoav M. "Voice-Recognition Based Advertising." U.S. Patent Application No. 12/566,189.
[11] Azad, Muhammad Ajmal, and Ricardo Morla. "Caller-REP: Detecting unwanted calls with caller social strength." Computers & Security 39 (2013): 219-236.
[12] Whitworth, Brian, and Elizabeth Whitworth. "Spam and the social-technical gap." Computer 37.10 (2004): 38-45.
[13] Mcrae, Matthew Blake, Kendra Sue Harrington, and Allen Joseph Huotari. "Method and system device for deterring spam over internet protocol telephony and spam instant messaging." U.S. Patent No. 7,992,205. 2 Aug. 2011.
[14] Sahin, Merve, Marc Relieu, and Aurélien Francillon. "Using chatbots against voice spam: Analyzing Lenny's effectiveness." Thirteenth Symposium on Usable Privacy and Security ({SOUPS} 2017). 2017.
[15] Rao, Anup, et al. "Method and system for deterring SPam over Internet Protocol telephony and SPam Instant Messaging." U.S. Patent Application No. 11/203,449.
[16] Nassar, Mohamed, and Olivier Festor. "Labeled voip data-set for intrusion detection evaluation." Meeting of the European Network of Universities and Companies in Information and Communication Engineering. Springer, Berlin, Heidelberg, 2010.
[17] Narayan, Akshay, and Prateek Saxena. "The curse of 140 characters: evaluating the efficacy of SMS spam detection on android." Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices. 2013.

MINIMISING DELAY AND ENERGY IN ONLINE DYNAMIC FOG SYSTEMS

Faten Alenizi and Omer Rana

School of Computer Science and Informatics, Cardiff University, Cardiff, UK

ABSTRACT

The increasing use of Internet of Things (IoT) devices generates a greater demand for data transfers and puts increased pressure on networks. Additionally, connectivity to cloud services can be costly and inefficient. Fog computing provides resources in proximity to user devices to overcome these drawbacks. However, optimisation of quality of service (QoS) in IoT applications and the management of fog resources are becoming challenging problems. This paper describes a dynamic online offloading scheme in vehicular traffic applications that require execution of delay-sensitive tasks. This paper proposes a combination of two algorithms: dynamic task scheduling (DTS) and dynamic energy control (DEC) that aim to minimise overall delay, enhance throughput of user tasks and minimise energy consumption at the fog layer while maximising the use of resource-constrained fog nodes. Compared to other schemes, our experimental results show that these algorithms can reduce the delay by up to 80.79% and reduce energy consumption by up to 66.39% in fog nodes. Additionally, this approach enhances task execution throughput by 40.88%.

KEYWORDS

Dynamic Fog Computing, iFogSim, Computational Offloading, Energy Consumption, Minimising Delay, Dynamic Resource Management, Internet of Things (IoT).

1. INTRODUCTION

Cloud computing plays an important role in processing tasks generated by IoT devices [1]. However, as the number IoT devices increases, so does the amount of generated data, and processing these data in the cloud may incur significant overhead in multi-hop networks. This is because the cloud server is usually remote and spatially distant from the IoT devices, which leads to high transmission latency, downgraded performance of latency-sensitive applications and network congestion [2]. To address this issue, fog computing (FC) – considered an extension of cloud computing (CC) [3, 4] – has been introduced by CISCO [5] which acts as an intermediary between the cloud and end-user devices. This brings processing, storage, and networking services spatially closer to end devices to reduce latency and network congestion. The main idea behind FC is to deploy computing resources, i.e. fog nodes (FNs), at the network edge [3, 6]. FNs can be routers, gateways, and access points to which end devices offload their computationally intensive tasks. However, FNs are characterised as resource-limited devices [4, 7] because they cannot handle all requests emanating from IoT devices located in their radius of coverage. To overcome this problem, computational offloading within the fog computing paradigm is one of the solutions that helps to improve the utilisation of available resources at FNs and allows for the processing of end-user tasks [8-10].

In fog computing, computational offloading refers to the cooperation between fog nodes in the same layer or with upper layers in which overloaded FNs delegate part of the workload to underloaded FNs within their proximity [9, 11] (we refer to such proximal fog nodes as forming a "neighbourhood"). Offloading, in this sense, refers to the sharing of the workload amongst the

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 139-158, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101513

nodes to minimise the overall latency of end-user tasks and improve QoS for user applications [11-13]. Computational offloading aims to maximise usage of the available fog resources, however at the expense of increasing energy use of fog nodes. An open research challenge considered in this work is understanding when a FN will decide to dynamically offload its workload and to which other FNs in the neighbourhood. This timely decision making is critical but challenging, especially within a dynamic system, where tasks generation cannot be known a priori. Another important question is understanding how to manage energy of fog resources, while still exploiting most of the available resources. The optimization problem of minimising the delay and the energy consumption has been decomposed into two sub-problems, named delay minimization problem and energy saving problem. The main objectives of this work can be summarised as following: (i) maximising the utilisation of resource-constrained fog nodes; (ii) minimising application loop delay; (iii) improving throughput of user tasks; (v) optimising energy consumption of fog nodes. This paper is structured as follows. The related works are introduced in Section 2. We describe the model of the fog computing system and constraints in Section 3. In Section 4, we explained the problem formulation. In Section 5, the proposed algorithms are presented. Simulations are conducted in Section 6 with numerical results, and concluding remarks are drawn in Section 7 with future work.

2. RELATED WORK

This section is divided into two main parts. The first focuses on computational offloading between entities within a specific system model; the second addresses the effect of dynamically controlling servers to improve power efficiency (e.g. either switching servers on and off as required or running them at lower capacity to improve power efficiency).

2.1. Computational Offloading

Offloading computing tasks to a cloud data centre and/or neighbouring fog computing servers has received considerable attention in other publications [4, 8, 9, 12-15]. Due to the differences in system models, existing works can be categorised as outlined below:

2.1.1. Computation Offloading in IoT-Fog-Cloud Computing Systems

IoT devices can be tasks generators and can process their own tasks aided by either fog or cloud resources, or both. The following scenarios illustrate this point:

(i) An end-user device either processes its tasks locally or chooses to offload them to the nearest fog node for processing (this is referred to as the IoT-Fog computing system). In this scenario, Tang et al. [12] aimed to maximise the total number of executed tasks on IoT devices and fog nodes while meeting the deadline requirements under energy constraints. The authors formulated the problem as a decentralised, partially observable offloading optimisation problem in which end users are partially aware of their local system status which includes the current number of remaining tasks, the level of battery power and the availability of the nearest fog node resource (availability is based on the number of tasks in the fog node's queue). These criteria are used to determine whether to process tasks locally or offload them to the nearest fog node. The suggested solution enables the IoT device to make an approximate optimal decision based on its locally observed system while meeting the delay requirements. Chen and Hao [15] investigated the task offloading problem in a dense software-defined network. The authors describe the problem as a mixed-integer nonlinear problem and decomposed it into two sub-problems: (1) deciding whether the end-user device should process its tasks locally or offload them to the edge device and (2) determining how many computational resources should be given to each task. To solve these sub-problems, the authors proposed an efficient software-defined task offloading scheme. The results of their proposed scheme, compared to random and uniform offloading schemes, demonstrate the effectiveness of their solution in decreasing the overall task execution time and the end-user device's energy consumption.

(ii) To determine whether end-user tasks should be processed on the end-user device, on the fog nodes or on the cloud servers (this is referred to as the IoT-Fog-Cloud computing system), Sun et al. [14] Since processing tasks are not limited to fog nodes, most tasks are processed either on the IoT devices (if they have the computational capacity), or on cloud servers as long as the task deadline is not violated. This leads to fewer tasks being processed within the fog environment, thereby reducing the energy consumed in fog nodes. Computational offloading has been studied in fog radio access networks [13] to achieve minimum system cost, which is the weighted sum of total energy consumption and total offloading latency. In their work, task latency only involves computation and transmission latency – no queuing latency is considered. To enhance the offloading decision, along with improving resource allocation for computation and radio resources, Zhao et al. have formulated the problem as a non-linear, non-convex joint optimisation problem. Their proposed solution has proven its effectiveness in comparison both to the mobile cloud computing system (MCC), in which all user tasks are processed on a cloud server, and to the mobile edge computing system (MEC), in which all end-user tasks are executed in the edge computing system. This is attributed to the fact that, in their model, both the fog and the cloud computation resources are available to support the offloading scheme.

(iii) In this scenario, in addition to the previous scenario, fog nodes can make the decision to offload the workload partially or fully to another fog node (this is referred to as the IoT-Fog-Fog-Cloud computing system. In [11], Yousefpour et al. proposed a delay-minimisation policy to reduce the overall service delay. In their work, the decision for a fog node to process its upcoming task(s), or either to offload to one of its neighbours (horizontal cooperation) or to the cloud server, is based on the estimated queue waiting time. If the offloading (queue waiting time) threshold has been reached, then a fog node will select one of the neighbouring fog nodes in its domain to offload its upcoming tasks to. The selection of the best neighbouring fog node is based on minimising total propagation delay plus queuing delay. Three different models have been considered and compared. In the first model, there is no processing of tasks in the fog system (NFP), so an IoT device either processes its own requests or sends them to a cloud data centre. The second model only allows for the processing of light computational tasks in the fog system with heavy computational tasks being processed in the cloud (LFP). In the third model, fog computing can process all types of tasks (AFP). The results show that AFP achieved the minimum average service delay compared to the two previous models.

2.1.2. Computation Offloading in Fog-cloud Computing Systems

In this approach the end-user devices offload all their computational tasks to the associated fog nodes for processing. The associated fog nodes can choose to offload part of their computational workload to another fog node or to the cloud, thereby exploiting fog and cloud resources to process end-user tasks. Gao et al. [9] investigated dynamic offloading and resource allocation, formulating the problem as a stochastic network optimisation to minimise delay and power consumption while ensuring the stability of all queues in the system. They present a predictive offloading and resource allocation approach that focuses on the trade-off between energy consumption and delay. Their approach suggests that increasing the allocation of processing resources in fog nodes causes a reduction in delay but increases energy consumption due to the processing of additional tasks and vice versa. The authors demonstrate the benefit of their approach compared to other schemes.

In [4], Xiao and Krunz developed a workload offloading strategy that maximises the average response time of all end-user tasks that are given a power efficiency constraint. In their experiment, power consumption is measured as the power spent on offloading each unit of received workload, but the power consumed to execute workloads is not considered. The decision for fog nodes to start cooperating and offload the workload is made through an agreement between the parties, and the amount of workload to be offloaded is based on the workload processing capabilities and the workload arrival rates. Based on their results, cooperation between fog nodes

helps to decrease the average response time. They also observed a fundamental trade-off between the average response time and the power efficiency of the fog node. The authors suggested that in order to optimise the power efficiency of the fog computing systems the response time of end-user tasks should be set to its maximum tolerable point, which means that when end-user tasks can stand higher delay there is no need to offload tasks to save energy. In addition, the authors stated that with delay-sensitive applications it is better to equip fog nodes with high-power consumption so that they are able to share more of their workload with other fog nodes, thus minimising response time.

2.1.3. Computation offloading in Fog Computing Systems

In this computing system, only fog resources are available to process end-user tasks, but no processing in the cloud is considered. Considering computation resources, Mukherjee et al. [16] designed a scheduling policy that manages to meet the deadline constraint of end-user tasks. In their scheduling policy, the deadline requirements of a task determine whether a fog node places it in its high priority queue, in its low priority queue or offloads it to one of its neighbouring fog nodes within the same tier. The decision whether to process the task or offload it to its neighbours is based on the availability of a neighbour with a lower transmission delay plus lower queue length of a specified type. Their results show the effectiveness of their proposed policy when compared to (a) an approach where no offloading is involved, and the fog node assigns its upcoming tasks randomly to one of its two queues with no consideration to priority and (b) an approach where workload offloading occurs between fog nodes with random task scheduling to any queue without considering their priority.

2.2. Dynamic Server Energy Management

Related work in this area is classified into two sections based on the environment in which this technique has been applied, which are Cloud Computing systems and Wireless Local Area Networks (WLANs). This technique has not been applied in fog computing system, while it has proved its efficiency in other environments.

2.2.1. Cloud Computing System

To save energy in a cloud environment, it has been proposed that servers should be dynamically shut down [17, 18] or put into sleep mode [19-21]. The authors in [17-21] investigated the problem of Virtual Machine (VM) placement to save energy and still maintaining QoS. In their work, underloaded data centres were detected and shutdown as per [17, 18] or put in a sleep mode as per [19-21]. All VMs in those data centres where then be migrated to other active underloaded data centres. This is done to minimise the energy consumed by cloud computing systems and is called 'VM consolidation'. For overloaded data centres, different VM selection methods have been proposed to decide which VMs should be migrated to other active data centres. Furthermore, a switched-off data centre could be activated to accommodate the migrated VMs to ensure QoS requirements in the system are met. The researchers saved the most energy when putting idle-mode data centres into sleep or shutdown mode.

Mahadevamangalam in [19] stated that in a cloud environment, idle-mode data centres with no workload consume energy equivalent to 70% of the energy consumed by data centres that are fully utilised and in busy mode. Therefore, shutting down idle-mode data centres will save up to 70% of the energy consumed in a cloud environment.

2.2.2. Wireless Local Area Networks (WLANs)

In WLANs, putting access points (APs) into sleep mode or switching them off has improved the energy efficiency of WLANs. In [22], Marsan and Meo found that in a group of APs that partially overlap, having one AP in each group to monitor the system and serve the upcoming users while all others are switched off can reduce energy consumption by up to 40%. In addition, if all APs

are switched off during idle periods, e.g. at night, energy consumption could be reduced by a further 60%. Li et al. [23] proposed a state transitions-aware energy-saving mechanism in which APs are not just switched on and off based on user demand, but there is also an intermediate stage that helps make the switching frequency as low as possible. This is to avoid frequently switching APs on and off as this will shorten their service life and also to avoid latency and energy overheads when APs are switched on.

2.3. Summary

In connection with minimising delay, computational offloading has been proposed in the literature [4, 8, 9, 11-16, 24]. Computational offloading can be deployed offline or online. In offline deployment, computational offloading decisions are made at the system design stage. All the required information about the system is known beforehand and is based on historical or predictive knowledge, such as the computational capacity of fog nodes, the total number of IoT devices and their workload (number of requests). In online deployment, the decision of computational offloading takes place at run-time and considers the current system status and process characteristics, such as the current waiting time. Most research investigating the offloading problem have considered the offline approach [4, 8, 9, 14, 15, 25] , while online approach has limited coverage [11-13, 16]. This shed lights on the importance of investigating the online computational offloading method, our approach primarily makes use of the online approach. In addition to that, none of the state-of-the-art fog computing models explore the impact of varying the offloading threshold on the system performance.

Fog computing is designed to place computational resources near the end users. To minimise delay, the end-users send their requests to the nearest fog node. However, if the fog node is overloaded, existing mechanisms focus on offloading part of this workload to the cloud for processing. However, there might be other nearby underloaded fog nodes that could help to process the workload to further minimise delay. This is called ‘fog cooperation’, but so far it has received limited coverage [4, 9, 11, 16].

In terms of minimising both delay and energy in the fog paradigm, most studies have addressed either minimising energy at IoT devices and ignoring the energy spent at the fog paradigm [12, 13, 15] or investigating the trade-off between these two aspects withing fog systems [4, 9, 14] because executing more tasks at fog nodes will reduce delay and consume more energy while executing fewer tasks at fog nodes and sending the rest to the cloud will increases delay but reduces energy consumption at fog paradigm. Therefore, most research addresses the balance between delay and energy by processing the workload on IoT devices, fog nodes or cloud servers if the QoS is satisfied. This results in fewer tasks being processed by the fog, thus consuming less energy as long as the QoS is met (e.g. deadline of users’ tasks). However, there might be underloaded or idle-mode fog nodes that could be switched off to save energy but still maintain the benefits of fog architecture, i.e. executing more tasks at fog nodes and thus minimising delay. To the best of knowledge of this paper’s authors, minimising both delay and energy at the same time and applying dynamic server energy management by switching on/off fog nodes have not been addressed before in the fog system.

3. SYSTEM MODELLING AND CONSTRAINTS

System model is presented in section 3.1, and Types of Connections and Constraints is described in section 3.2.

3.1. System Model

Network diagram is described in section 3.1.1, and application module description in section 3.1.2.

3.1.1 Network Diagram

An overview of the fog computing architecture is shown in Figure.1 and consists of three layers:

- **The IoT devices layer:** this layer contains of mobile vehicles. The vehicle node has a set of sensors. Each sensor transmits different types of tasks and an actuator and once they are within the coverage radius of a fog node, they will send their tasks. Two types of tasks are emitted by the mobile vehicle. The first type is non-urgent and contains information such as current location, speed, and direction of the vehicle. The second task is an urgent request that requires a quick response. For example, this task may contain a video stream of a moving vehicle's surroundings, which requires quick processing by fog nodes to help avoid collisions. This might be important, especially for autonomous driverless vehicles.
- **Fog computing layer:** this layer consists of a set of fog nodes and a fog controller. Fog nodes reside in roadside units (RSU) that are deployed in different areas of a city. Fog nodes can communicate with each other if they are located within each other's vicinity [26]. Fog nodes can form an ad hoc network between themselves to share and exchange data. All fog nodes are logically connected to the fog controller which monitors the performance of all fog nods and manages the resources. The fog nodes are static and receive two different types of tasks from all vehicles within their radius. These tasks are called priority and non-priority tasks. Regarding priority tasks, fog nodes process requests generated by a user's sensor and send the response back to the user. For non-priority tasks, fog nodes do some processing of the information provided by the vehicles within their range and send the results to the cloud for further analysis and storage for retrieval by traffic management organisations.
- **Cloud computing layer:** this layer contains cloud servers. It manages and controls the traffic at the city-level based on historical data received by fog nodes.

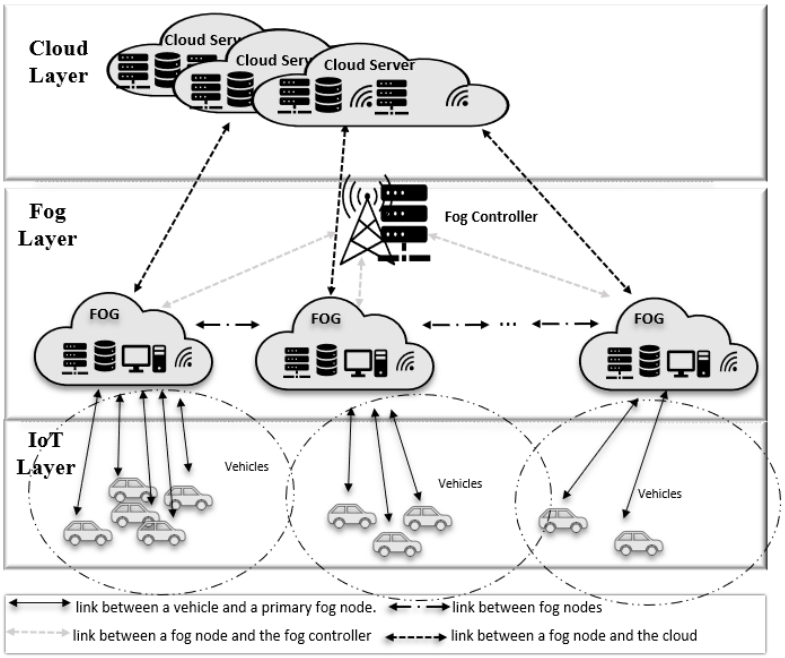


Figure 1: Fog Computing Model

3.1.2 Application module description

The application model of this study consists of three modules named Road Monitor, Global Road Monitor and Process Priority Tasks. The first two modules are responsible for traffic light control systems and the last module is only for processing end-user priority tasks. The function of each of these modules is as follows:

• **Road Monitor**: this module is placed in fog nodes. If a vehicle enters an area within the coverage of a fog node, the sensor automatically sends the current car location, its speed, weather conditions and road conditions to the connected fog node for analysis. Then the module processes these data and the results are sent to the cloud for further analysis.

• **Global Monitor**: this module is placed in the cloud and receives the collected data from fog nodes (after being processed by the Road Monitor module), analyses these data and stores the results.

• **Process Priority task**: this module is placed in fog nodes and is responsible for processing the priority requests from the user. The results are then sent back to the user. The application in iFogSim is represented as a directed acyclic graph (DAG) = (M, E) where M is the set of application modules deployed = {m1, m2, m3, ... , mn}, e.g. Process Priority Task, Road Monitor and Global Road Monitor modules. Between application modules, there is a set of edges belonging to E, which represents the data dependencies between application modules. This is shown in Figure.2.

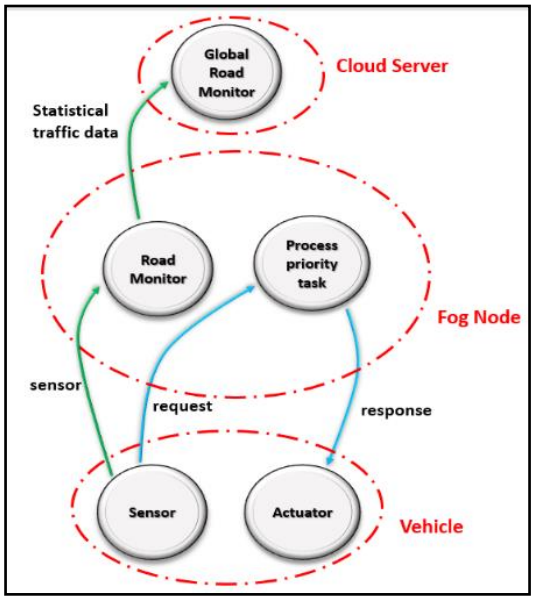


Figure 2: Directed Acyclic Graph (DAG) of the application model.

3.2 Types of Connections and Constraints

This section describes the connections between a vehicle and a fog node, between fog nodes, and between fog nodes and cloud. Also, the set of constraints involved within these connections.

3.2.1 Connection between Vehicles and Fog nodes

The connection between a vehicle and a fog node is made with communication and processing constraints.

- **Communication Constraints**

Vehicles connect to the fog node if and only if it is located within its communication range, as constraint (1)

$$\boldsymbol{D_{v,f} \leq max\ Coverage_f;\ \forall\, v \in V, \forall\, f \in FN} \qquad (1)$$

Where V is all vehicles, v one vehicle, FN is all fog nodes and f is one fog node. $D_{v,f}$ is the distance between a vehicle v and a fog node f, is calculated as

$$D_{v,f} = \sqrt{(X_v - X_f) + (Y_v - Y_f)}; \quad \forall v \in V, \forall f \in FN \tag{2}$$

where (X_V, Y_V) and (X_f, Y_f) are the coordinates of a vehicle and a fog node, respectively. If a vehicle is located within the coverage radius of more than one fog node it will connect to the nearest fog node. This to reduce delay because the expected arrival time of the task at the connected fog node depends on the transmission and the propagation delay, but the propagation delay depends solely on the distance between the two connected objects. Propagation delay (PD) is calculated as

$$PD = \frac{D_{v,f}}{PS} \tag{3}$$

Following [27], we assume that the speed of signal propagation (PS) is equal to the speed of light, $c = 3 \times 10^8$.

- **Processing Constraints**

For fog nodes to process user tasks, application modules in which these tasks are processed should be placed at fog nodes. To ensure the placement of these application modules, application modules require CPU, Ram and Bandwidth capacity so that fog nodes will have enough CPU, Ram and Bandwidth capacity to place these application modules, thus processing end-user tasks at the fog paradigm.

$$\sum_{i=0}^{M} Required_{Capacity}\, m_i \leq \sum Available_{Capacity}\, f;\ \forall m_i \in \mathbf{M}, \forall f \in \mathbf{FN} \tag{4}$$

Where $Required_{Capacity}$ for each application module = {CPU, Ram, Bandwidth} and the fog node capacity = {CPU, Ram, Bandwidth}. Constraint (4) ensures that the total required capacity of all application modules should not exceed the available capacity of the fog node in which they should be placed. In iFogSim, if the capacity required to place application modules exceeds the available capacity of fog nodes, the system will iterate through upper tiers fog computing system until it reaches the cloud and places these application modules. The CPU required for an application module is calculated as following:

$$CPU = NV * (Rate * TaskCPU) \tag{5}$$

Where NV is the total number of connected vehicles to a fog node, and TaskCPU is the task CPU length which is the number of instructions contained in each task in Million Instructions Per Second (MIPS). Rate is calculated as:

$$Rate = \frac{1}{Transmission\ Time\ in\ ms} \tag{6}$$

The placement of application modules in iFogSim is done before running the system and starting the emission of tasks. If the number of vehicles increases, this will impact the required CPU capacity for an application module. In this case the number of connected vehicles for each fog node is limited as constraint (7).

$$\sum_{i=0}^{V} v_i f_j \leq MAX_{vehicle\ number};\ \forall v_i \in V, \forall f_j \in FN \tag{7}$$

3.2.2 Connection between Fog nodes

This section describes the waiting queue for fog nodes in which the offloading decision is determined, how fog nodes communicate and the selection criteria for the best neighbouring fog node.

- **Fog nodes' waiting queue**

Each fog node maintains a waiting queue into which tasks are placed upon their arrival at the fog node. Fog nodes process one task at a time. Once the execution of that task is completed the fog node will check its waiting queue and process the next task according to its scheduling policy, i.e. first come, first served. This process continues until no tasks are in the waiting queue. Following the work [11], the waiting queue time triggers the decision to start computational offloading to neighbouring fog nodes. To start sharing workloads, the queue waiting time (T^{Queue}) should exceed the offloading threshold, e.g. 50ms, 100ms or 200ms.

$$T^{Queue} > Max_{threshold} \tag{8}$$

T^{Queue} is calculated as

$$T^{Queue} = \sum T_i * T_i^{process} + \sum T_z * T_z^{process};\ \forall\ i, z\ \in T \tag{9}$$

Where T_i and T_z are the total number of tasks of the type *i* and *z*, e.g. priority or non-priority. T is all tasks and $T_i^{process}$ is the expected execution time of a specific task and calculated as

$$T^{process} = \frac{TaskCPU}{F\_MIPS * N\ of\ PS} \tag{10}$$

Where F_MIPS is the total mips available in a fog node and N of PS is the total number of processing units allocated in that fog node.

- **Coverage Method**

To achieve area coverage, several fog nodes are required. Fogs can also overlap to achieve maximum coverage as in [28] see Figure. 3.

Figure 3: Overlapping Fog Nodes.

- **Selecting the Best Neighbouring Fog Node**

The process of selecting the best neighbouring fog node follows the work in [11]. It happens when a fog node reaches its offloading threshold, e.g. 50ms, 100ms or 200ms waiting queue time, for each upcoming task that is generated from vehicles in the coverage range of this fog node.

The neighbouring fog nodes of a fog node are the fog nodes that are located within the coverage radius of the fog node itself. This is shown in constraint (11)

$$\boldsymbol{d_{ij} \leq Coverage_{radius};\ \forall\, i,j \in FN} \quad (11)$$

Where d_{ij} is the distance between fog nodes i and j. In Figure. 3, FOG 2 and FOG 3 are the neighbouring fog nodes for FOG 1. Also, FOG 1, FOG 4, FOG 5 are the neighbouring fog nodes for FOG 3. The criteria for selecting the best neighbouring fog node depends on two factors. First, the neighbouring fog node should be within the communication range of the primary fog node. Second, and most importantly, a neighbouring fog node should have the minimum sum of waiting queue time plus propagation delay amongst all available neighbours.

$$\boldsymbol{Min \sum T^{Queue} + PD} \quad (12)$$

PD is calculated as

$$\boldsymbol{PD = \frac{D_{f,f'}}{PS}} \quad (13)$$

($\boldsymbol{D_{f,f'}}$) is the distance between fog nodes f and f'and it is calculated similar the distance between a vehicle and a fog node ($\boldsymbol{D_{v,f}}$) and propagation speed PS is equal to the speed of light, its value 3×10^8, this done similar to the work in [27].

3.2.3 Between Fog Nodes and the Cloud

When fog nodes finish the processing of non-urgent tasks the results are sent to the cloud for further analysis and processing by the application module named Global Road Monitor. In the current work, the cloud is the least to be considered in sharing the workload of fog nodes when they reach the offloading threshold. This is due to the availability of neighbouring fog nodes and in order to get maximum usage of the available resources in the fog system. However, if all neighbours reach their offloading threshold, the primary fog node will determine to send the task to the cloud if its queue waiting time is higher than transmission delay caused by sending the task for processing to the cloud and getting the results back. Due to the powerful computational capabilities at the cloud server compared to fog nodes, queueing delay is neglected so tasks are processed upon their arrival [29-31] .

4. PROBLEM FORMULATION

The optimisation problem of minimising the delay and the energy consumption has been decomposed into two sub-problems: the delay minimisation problem and the energy saving problem.

4.1 Delay Minimization Problem

The response time includes the round-trip time for transmitting the workload between a user and the associated fog node. It includes the transmission delay, propagation delay, queuing delay and processing delay. If the workload is processed by the vehicle's primary fog node then the service latency is calculated as

$$\boldsymbol{T = T^{sTv} + 2\,X\,(T_{vTf}^{Transmision} + PD_{vTf}) + T^{Queue} + T^{procss} + T^{vTa}} \quad (14)$$

Where $\boldsymbol{T^{sTv}}$ and $\boldsymbol{T^{vTa}}$ is the latency time between a vehicle and its sensor, and between the vehicle and its actuator, respectively. $\boldsymbol{T_{vTf}^{Transmision}}$ is transmission delay between the vehicle and its

primary fog node. It is based on the network length of the task and the bandwidth, and it is calculated as

$$T^{Transmision} = \frac{Network\ Length\ of\ Task}{Bandwidth} \quad (15)$$

If the primary fog node decides to offload the workload to one of its neighbours, then the latency is calculated as

$$T = T^{sTv} + 2x(T_{vTf}^{Transmsiion} + PD_{vTf}) + 2x(T_{fTf}^{Transmission} + PD_{fTf}) + T^{Queue} + T^{Process} + T^{vTa} \quad (16)$$

If the primary fog node decides to send the task to the cloud, then the latency is calculated as

$$T = T^{sTv} + 2x(T_{vTf}^{Transmsiion} + PD_{vTf}) + 2x(T_{fTc}^{Transmission}) + T^{Process} + T^{vTa} \quad (17)$$

4.2 Energy Saving Problem

By minimising the power consumption of fog nodes, the overall cost of electricity consumption and environmental impact is reduced. Each fog node has two power modes: idle and busy. The fog node's power is said to be in idle mode when the fog node is not doing any task processing and in busy mode when the fog node is busy processing tasks. The energy consumed is the power spent when a fog node is processing workload and when the fog node is switched ON and not doing any processing. The total energy consumption in iFogSim is calculated as in [32] as

$$E = PR + (TN - LUT) * LUP \quad (18)$$

Where PR is previous total energy consumed in this fog node, TN is the time now which is the time that the updateEnergyConsumption () is called when utilising this fog node, LUT is the last time this fog node has been utilised and finally LUP which is the fog node last utilization power status, which is idle power or busy power, the value of this is based on the predefined parameters when creating a fog node. The problem of minimizing delay and energy is formulated as follows:

$$\text{Min} \sum T \ \& \ \sum E$$

$$\text{s.t.} \ \ (1), (7), (4)$$

$$T^{Queue} \leq Max_{threshold} \quad (19)$$

$$P_F + P_N = 1, P_F \ \& \ P_N = \{0, 1\} \quad (20)$$

Equation (1) ensures the connection between a fog node and a vehicle that is located within its communication range. Equation (7) ensures the number of vehicles connected to one fog node does not exceed the threshold number. Constraint (4) ensures the placement of application modules at fog nodes. Equation (19) ensures the stability of fog nodes' queues so that, to process its upcoming tasks, the waiting queue time should not exceed its threshold. In constraint (20), PF and PN mean that if the task is processed in its primary fog node, then PF = 1 and PN = 0 and vice versa. Therefore, the task is either processed in the primary fog node or one of its neighbours.

5. PROPOSED ALGORITHMS

An approach that combines two algorithms has been proposed to solve the above stated problem. The first algorithm is called dynamic task allocation and the second is called dynamic resource saving. In this paper, both stated algorithms need to work together to achieve the intended outcome.

5.1 Dynamic Task Scheduling (DTS):

The aim of this algorithm is to minimise delay by allowing cooperation between fog nodes in terms of workload sharing, to maximise the resource utilization and maximise throughput. The fog controller is not involved in the selection of the best neighbouring fog node, it is mainly involved in the DEC algorithm. Also, in regards to DTS algorithm, if the best neighbour is switched OFF, the fog controller will send a signal to switch ON the selected best neighbour, this is further explained in section 5.2.

The process of offloading a task based on the queue waiting time of the fog nodes was originally proposed by [11]. In [11], the task can be offloaded multiple times, which means that if the primary fog node decides to offload the upcoming task to its neighbour *i*, by the time this task arrives at fog node *i*, fog node *i* might have reached its offloading threshold. Then fog node *i* will select fog node *j* to offload this task to, resulting in offloading this task multiple times and adding additional transmission and propagation delay. As stated by [11], multiple task offloading will increase the delay compared to only allowing the task to be offloaded one time, and this is applied to the current work. The technique is shown in Figure. 4.

When a fog node receives a task, if this task is the first task in its queue it will immediately process it, if not, it will check its queue waiting time. If its queue did not reach its offloading threshold, e.g. 50ms, 100ms or 200ms, the task will be added to its queue, but if the queue reaches its threshold the fog node will check if the task has been offloaded by another fog node. If it has, then it will add this task to its queue. If it has not been offloaded by another fog node it will select the best neighbour to offload this task to, according to the criteria described in section 3.2.2. If the best neighbour reaches its offloading threshold during the selection process and before offloading the task, then the primary fog node will make the decision whether to offload the task to the cloud for processing or process the task locally. This is determined when comparing the transmission delay caused by sending the task for processing to the cloud and getting the results back with the queuing delay of the fog node itself. if the queueing delay is higher, then the fog node will send the task for processing to the cloud, else, the task will be processed locally at the primary fog node.

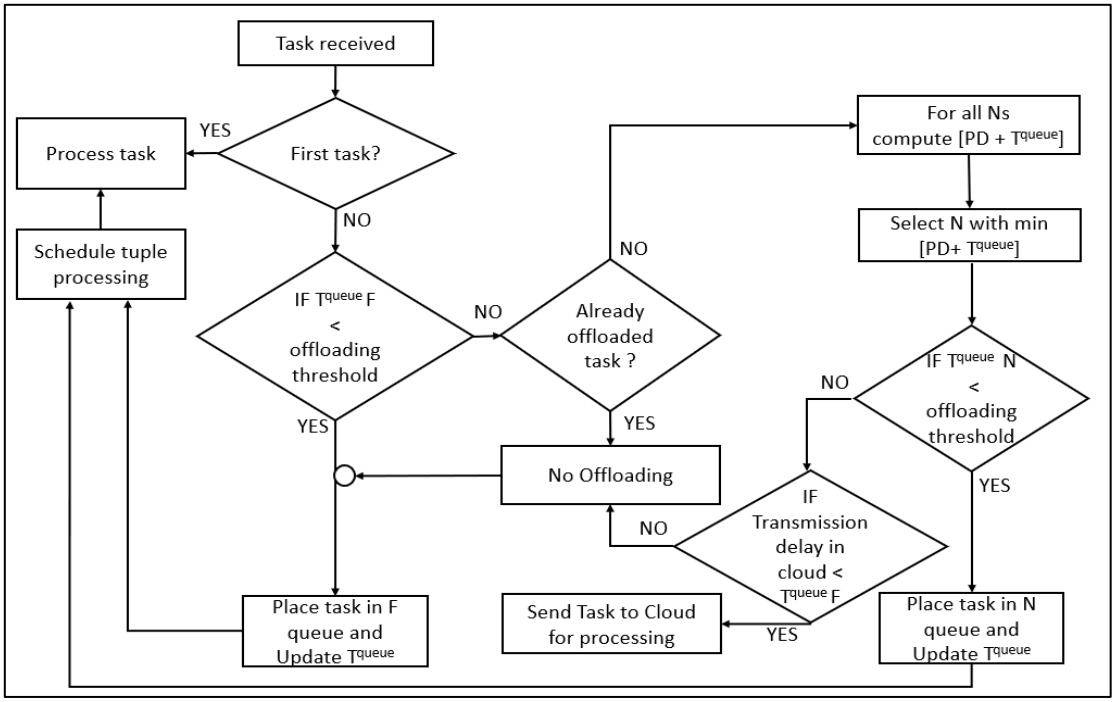


Figure 4: Flowchart of Dynamic Task Scheduling Algorithm.

5.2 Dynamic Energy Control (DEC)

The need for 24/7 availability of fog nodes poses a challenge on energy efficiency and cost since the fog provider needs to maintain available resources that may be used but are not continuously needed. If a fog node is not needed it should be turned off to save energy. Dynamic energy control (DEC) has been proposed in order to optimise resource utilisation by dynamically deciding when to switch off an active fog node(s) and conserve overall system energy. The pseudo code of our proposed algorithm is given in Algorithm 1.

In this system, the ON and OFF switching of fog nodes is carried out by the fog controller which runs algorithm 1 each time it receives information about the system. Fog nodes update the fog controller with their information so that fog controller can make the appropriate decision to save energy. At the beginning of the simulation, all fog nodes are switched OFF.

Algorithm 1 Dynamic Energy Controlling

Input: System Data: 1- current waiting time; 2- current processing states; 3- if awaiting task/s

Output: Sending signals to switch ON/OFF determined FNs

```
Fog Controller receives System data
for all FNs do:
  if (FN. status ==OFF)
    if (FNQueueSize! = 0)
      Send Signal ON
    else
  else
    if (processingStatus =1) //fog node is not processing task/s
      Send Signal OFF
    else
  end if
end for
```

6. PERFORMANCE EVALUATION

In this section, we first provide the details of the simulations, then we investigate the performance of our two comined algorithms.

6.1 Simulation Environment Settings

iFogSim has been used to simulate the environment. It is a toolkit developed by Gupta et. al [33], which is an extension of the CloudSim simulator. It is a toolkit allowing the modelling and simulation of IoT and fog environments and is capable of monitoring various performance parameters, such as energy consumption, latency, response time, cost, etc. For this research, the three-tier fog system was established first as shown by the simulation in Figure. 1. The simulation was run with one cloud server, seven fog nodes, the fog controller, and a total of 50 vehicles. Two fog nodes connected to 25 vehicles, but the other five fog nodes are not connected to any vehicles. This is done to vary the workload amongst fog nodes because if all fog nodes have the same workload then offloading will not be beneficial [11]. Each vehicle transmits two different tasks every 3ms. The parameter values used in the simulation is in Tables 1 - 5.

Table 1: Application Modules Requirements.

Module	CPU (mips/vehicle)	BW (Mbps)	Ram (GB)
Process priority task	333.33	1000	10
Road Monitor	300	1000	10
Global Road Monitor	99.99	1000	10

Table 2: Tasks details.

Task Type	Processed module	CPU length (MIPS)	Network Length (Mbps)
Request (urgent)	Process priority task	1000	1000
Sensor (nonurgent)	Road Monitor	900	500
Statistical traffic data	Global Road Monitor	300	500

Table 3: Entity Configurations in iFogSim.

Characteristics	Vehicle	Fog nodes	Cloud servers
CPU (MIPS)	0.0	15100	448000
RAM (MB)	0	40000	40000
Uplink BW (Mbps)	1000	1000000	1000000
Downlink BW (Mbps)	1000	1000000	1000000
Rate Per MIPS	0.0	0.001	0.01
Level	2	1	0

Table 4: Power Consumption with ON/OFF.

Device	Power Consumption (W) when device is ON		Power Consumption (W) when device is OFF
	Idle	Busy	Power
Fog Node	83.4333	107.339	0.0
Cloud Server	16*103	16*83.25	No

Table 5: Latency values between entities.

Between		Link latency (ms)
Cloud	Fog node	100 ms
Fog node	Neighboring FN	2 ms
Vehicle	Fog node	[1-5] depends on location
Sensor/Actuator	Vehicle	1 ms

6.2 Experiments

The conducted experiments are shown in Table 6. The metrices used to measure the performance are:

- **Service latency** as the average round trip time for all tasks processed in the fog environment
- **Throughput**, which is measured as the total number of processed tasks within a time window.
- **Total Energy Consumption** in fog environment

Table 6: Set of Conducted Experiments Details.

Experiment		Dynamic Task Scheduling		Dynamic Energy Controlling
no	name	Yes/No	When	
1	No offloading	no	-	No
2		no	-	Yes
3	Offloading-50	yes	50 ms	No
4		yes	50 ms	Yes
5	Offloading-100	yes	100 ms	No
6		yes	100 ms	Yes
7	Offloading-200	yes	200 ms	No
8		yes	200 ms	Yes

6.2.1 Average round trip time

There are two control loops in the simulation:

• Sensor → Process Priority Tasks → Actuator. This control loop represents the path of the priority requests, and it is called Control loop A.

• Sensor → Road Monitor → Global Road Monitor. This control loop represents the path pf the non-priority requests, and it is called Control loop B.

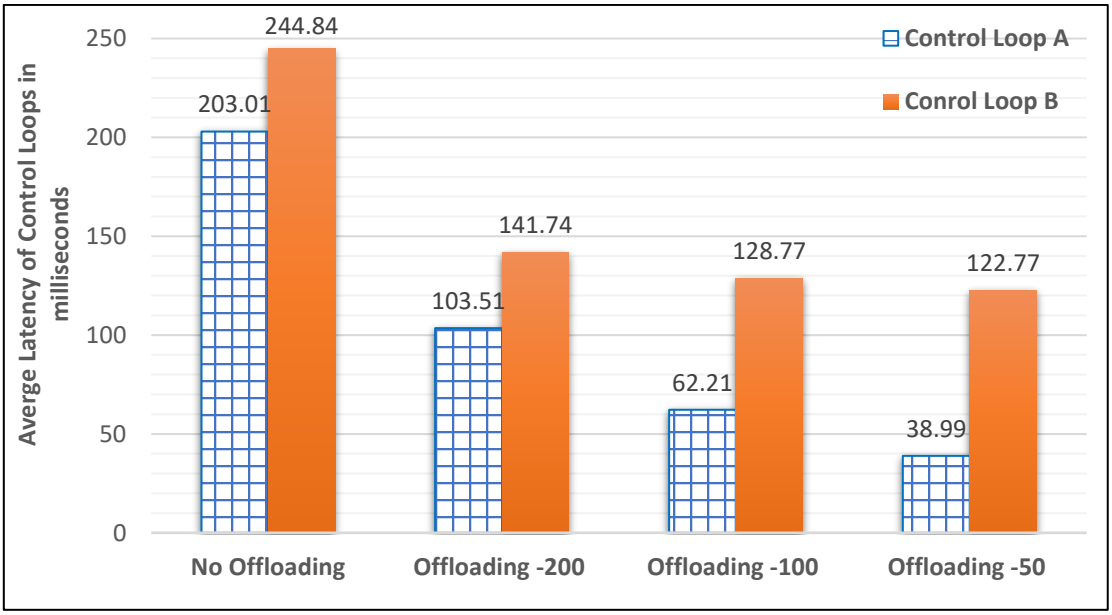


Figure 5: Average Round Trip Time with no-offloading and different Offloading Thresholds

The aim here is to minimise the average round-trip time for control loop A, in which the result is going back to the users, compared to control loop B, in which the user tasks should be processed at fog nodes and the results sent to the cloud for further analysis and storage. The results in Figure. 6 show that when a fog node is not offloading its tasks to the neighbouring fog nodes, the average round trip time for all the processed tasks for control loop A is 203.01ms. This is due to the long queuing delay. However, the average round trip is minimised when the offloading threshold is set to 50ms. This is because more neighbours are involved in the process of executing tasks. With a 50ms threshold, the average latency of the control loop was reduced by 80.79% compared to the no-offloading case.

6.2.2 Throughput Evaluation

According to the results in Figure. 7, when the offloading threshold is set to 50ms the number of executed tasks is increased by almost 40.88% compared to the no-offloading method. As with the no-offloading method, many tasks are waiting to be executed in the queue compared to when the offloading threshold is set to 50ms, the threshold where fog node cooperation is allowed and workloads (tasks) are shared.

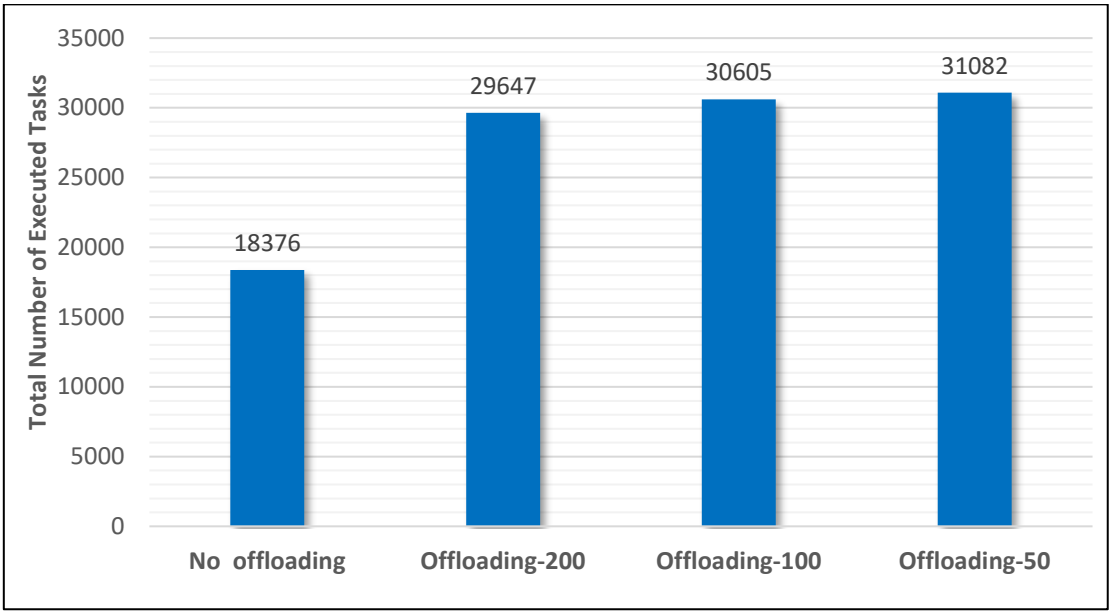


Figure 6:Number of executed Tasks in Fog Nodes with no offloading and different Offloading Thresholds

6.2.3 Total Energy Consumption

In cases where a dynamic energy control algorithm is not applied the highest energy consumption in the fog environment occurs when the offloading threshold is set to 50ms. This is because more fog nodes are involved in the execution process and are in their busy mode power mode. This compares to the no-offloading method where only two fog nodes are busy processing tasks while the rest of the fog nodes are not doing any processing and are in their idle power mode (see Figure 8). In the no-offloading method, DEC saves around 66.39% of power. This power was spent powering on unused fog nodes, which cause a wastage in resources. Applying the DEC algorithm helps to minimise the total energy consumed in the fog environment by 2.59%, 3.84% and 6.37% with the various offloading thresholds of 50ms, 100ms and 200ms, respectively. The reason for a low energy saving with various offloading thresholds compared to a high energy saving with the no-offloading approach is that the workload of the primary fog nodes is high, thus sharing some of their workloads with their neighbours. As a result, neighbours staying ON most of the time helps to process these tasks.

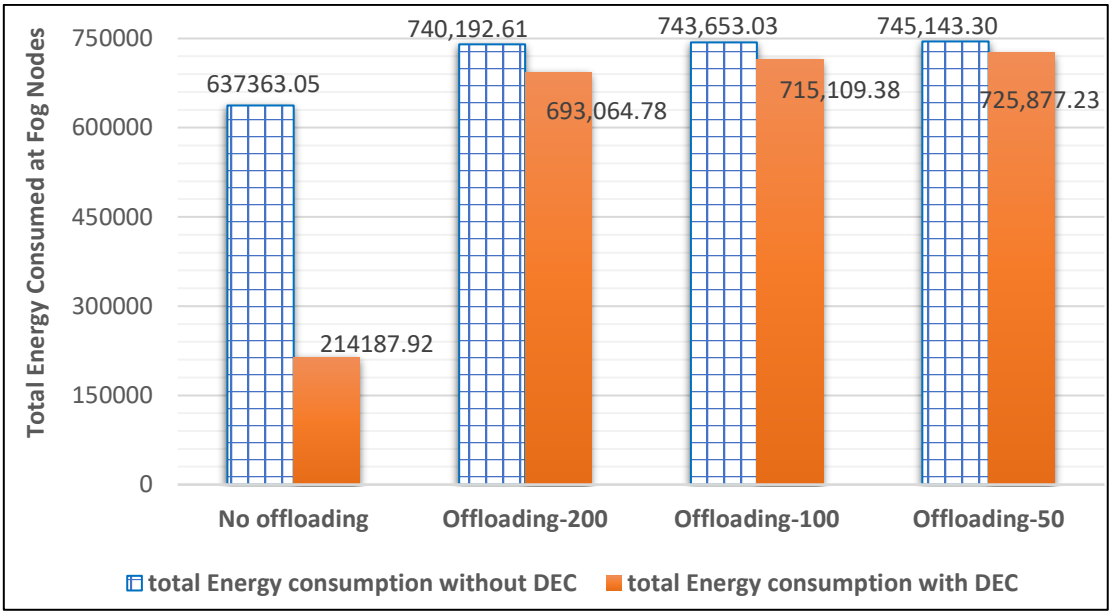


Figure 7: Total Energy Consumed in the fog environment with no offloading and different offloading thresholds with and without Dynamic Energy Controlling (DEC) algorithm

Varying the offloading threshold of the queuing delay, such as 50ms, 100ms and 200ms does improve service latency and throughput. However, this is not the case with energy consumption because more energy is spent by the fog system when the offloading threshold is set to 50ms. This is because more fog nodes are involved in the execution process and therefore require more power to work efficiently. However, after applying the DEC algorithm, energy consumption was reduced. Varying the offloading threshold has not been addressed before in other publications, but it does have a positive impact on overall results. However, this technique depends on the number of neighbouring fog nodes that are willing to help and the availability of these fog nodes. This will be addressed in future work.

7. CONCLUSION

In this paper, we studied the problem of minimising service latency and power consumption in fog computing systems and proposed a combination of two efficient and effective algorithms: dynamic task scheduling (DTS) and dynamic energy control (DEC).

In future work, latency and energy overhead caused by activating switched off fog nodes should be considered, and their impact on the system should be addressed. This is because fog nodes are operational devices that require time and energy to boot up in contrast to previous work that powers on switched off datacentres without considering latency and energy overhead. Also, frequent switching ON and OFF of edge devices might lead to edge device failure in the long term and shorten the life of edge devices. Therefore, the frequency of switching fog nodes on and off should be considered and minimised.

REFERENCES

[1] H. R. Arkian, A. Diyanat, and A. Pourkhalili, "MIST: Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications," *Journal of Network and Computer Applications*, vol. 82, pp. 152-165, 2017.

[2] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of network and computer applications,* vol. 98, pp. 27-42, 2017.

[3] K. Ma, A. Bagula, C. Nyirenda, and O. Ajayi, "An iot-based fog computing model," *Sensors,* vol. 19, no. 12, p. 2783, 2019.

[4] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, 2017: IEEE, pp. 1-9.

[5] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing: Principles, architectures, and applications," in *Internet of things*: Elsevier, 2016, pp. 61-75.

[6] H. F. Atlam, R. J. Walters, and G. B. Wills, "Fog computing and the internet of things: a review," *big data and cognitive computing,* vol. 2, no. 2, p. 10, 2018.

[7] B. Jamil, M. Shojafar, I. Ahmed, A. Ullah, K. Munir, and H. Ijaz, "A job scheduling algorithm for delay and performance optimization in fog computing," *Concurrency and Computation: Practice and Experience,* vol. 32, no. 7, p. e5581, 2020.

[8] Q. Wang and S. Chen, "Latency-minimum offloading decision and resource allocation for fog-enabled Internet of Things networks," *Transactions on Emerging Telecommunications Technologies,* p. e3880, 2020.

[9] X. Gao, X. Huang, S. Bian, Z. Shao, and Y. Yang, "Pora: Predictive offloading and resource allocation in dynamic fog computing systems," *IEEE Internet of Things Journal,* 2019.

[10] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of everything*: Springer, 2018, pp. 103-130.

[11] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the internet of things," in *2017 IEEE international conference on edge computing (EDGE)*, 2017: IEEE, pp. 17-24.

[12] Q. Tang, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Decentralized Computation Offloading in IoT Fog Computing System With Energy Harvesting: A Dec-POMDP Approach," *IEEE Internet of Things Journal,* 2020.

[13] Z. Zhao *et al.*, "On the design of computation offloading in fog radio access networks," *IEEE Transactions on Vehicular Technology,* vol. 68, no. 7, pp. 7136-7149, 2019.

[14] H. Sun, H. Yu, G. Fan, and L. Chen, "Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture," *Peer-to-Peer Networking and Applications,* vol. 13, no. 2, pp. 548-563, 2020.

[15] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications,* vol. 36, no. 3, pp. 587-597, 2018.

[16] M. Mukherjee, M. Guo, J. Lloret, R. Iqbal, and Q. Zhang, "Deadline-aware Fair Scheduling for Offloaded Tasks in Fog Computing with Inter-fog Dependency," *IEEE Communications Letters,* 2019.

[17] M. A. H. Monil, R. Qasim, and R. M. Rahman, "Energy-aware VM Consolidation Approach Using Combination of Heuristics and Migration Control."

[18] A. Mosa and N. W. Paton, "Optimizing virtual machine placement for energy and SLA in clouds using utility functions," *Journal of Cloud Computing,* vol. 5, no. 1, p. 17, 2016.

[19] S. Mahadevamangalam, "Energy-aware adaptation in Cloud datacenters," ed, 2018.

[20] M. A. H. Monil and R. M. Rahman, "Implementation of modified overload detection technique with VM selection strategies based on heuristics and migration control," in *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, 2015: IEEE, pp. 223-227.

[21] M. A. H. Monil and R. M. Rahman, "VM consolidation approach based on heuristics, fuzzy logic, and migration control," *Journal of Cloud Computing,* vol. 5, no. 1, p. 8, 2016.

[22] M. A. Marsan and M. Meo, "Queueing systems to study the energy consumption of a campus WLAN," *Computer networks,* vol. 66, pp. 82-93, 2014.

[23] F. Li, X. Wang, J. Cao, R. Wang, and Y. Bi, "A State Transition-Aware Energy-Saving Mechanism for Dense WLANs in Buildings," *IEEE Access,* vol. 5, pp. 25671-25681, 2017.

[24] Q. Zhu, B. Si, F. Yang, and Y. Ma, "Task offloading decision in fog computing system," *China Communications,* vol. 14, no. 11, pp. 59-68, 2017.

[25] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal,* vol. 5, no. 1, pp. 283-294, 2017.

[26] S. F. Abedin, M. G. R. Alam, N. H. Tran, and C. S. Hong, "A Fog based system model for cooperative IoT node pairing using matching theory," in *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2015: IEEE, pp. 309-314.

[27] S. A. Soleymani *et al.*, "A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing," *IEEE Access,* vol. 5, pp. 15619-15629, 2017.

[28] F. T. Zohora, M. R. R. Khan, M. F. R. Bhuiyan, and A. K. Das, "Enhancing the capabilities of IoT based fog and cloud infrastructures for time sensitive events," in *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, 2017: IEEE, pp. 224-230.

[29] G. Lee, W. Saad, and M. Bennis, "An online secretary framework for fog network formation with minimal latency," in *2017 IEEE International Conference on Communications (ICC)*, 2017: IEEE, pp. 1-6.

[30] S. El Kafhali, K. Salah, and S. B. Alla, "Performance Evaluation of IoT-Fog-Cloud Deployment for Healthcare Services," in *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*, 2018: IEEE, pp. 1-6.

[31] L. Liu, Z. Chang, and X. Guo, "Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices," *IEEE Internet of Things Journal,* vol. 5, no. 3, pp. 1869-1879, 2018.

[32] D. Rahbari and M. Nickray, "Scheduling of fog networks with optimized knapsack by symbiotic organisms search," in *2017 21st Conference of Open Innovations Association (FRUCT)*, 2017: IEEE, pp. 278-283.

[33] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275-1296, 2017.

USING MACHINE LEARNING IMAGE RECOGNITION FOR CODE REVIEWS

Michael Dorin[1, 2], Trang Le[2],
Rajkumar Kolakaluri[2] and Sergio Montenegro[1]

[1]Aerospace Information Technology, Universität Würzburg,
Würzburg, Germany
[2]Engineering, University of St. Thomas, St. Paul, MN, USA

ABSTRACT

It is commonly understood that code reviews are a cost-effective way of finding faults early in the development cycle. However, many modern software developers are too busy to do them. Skipping code reviews means a loss of opportunity to detect expensive faults prior to software release. Software engineers can be pushed in many directions and reviewing code is very often considered an undesirable task, especially when time is wasted reviewing programs that are not ready. In this study, we wish to ascertain the potential for using machine learning and image recognition to detect immature software source code prior to a review. We show that it is possible to use machine learning to detect software problems visually and allow code reviews to focus on application details. The results are promising and are an indication that further research could be valuable.

KEYWORDS

Code Reviews, Machine Learning, Imagine Recognition, Coding Style

1. INTRODUCTION

Code reviews are policy in many software development organizations, and it is commonly believed that code reviews are an economical way to discover faults before a software product is deployed. Indeed, it is even suggested that code that has not been adequately reviewed has twice the faults of reviewed code [1]. However, many software engineers are overwhelmed with work, so proper code reviews are often not done. The reviewability of software is affected by many factors such as documentation, logic, semantics, and syntax. Source code includes aspects that might even be considered aesthetic, and aesthetic aspects might turn tedious and possibly overwhelm the review process [2]. In a paper by Yazdani and Manovich, non-photographic images, such as screenshots and images of text messages, were analysed and found they could be useful in predicting social trends [3]. This paper aims to evaluate the possibility of using "screenshots" of source code with machine learning image recognition as part of the software code review process. Tools to reduce monotonous tasks related to reviews could be very valuable. This paper begins by discussing the readability aspects of code and estimates the impact style has on reviews. We then created images of poorly styled code and properly styled code and used machine learning to train an image recognizer to identify poorly formatted code and present positive results. Creating source code “screenshot images” for analysis could be part of automating code reviews. Using automation as part of the review process could make software engineers more efficient.

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 159-167, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101514

2. RELATED WORK

2.1. Code Reviews

As code reviews are an essential topic, many papers are written each year to address the review process's problems. In the paper, "Confusion in Code Reviews," the authors recognize that code reviews do not always go smoothly and identify items confusion in the review process [4]. Fatima et al. discuss the good and bad consequences of feedback in the review process [5]. A vital feature these papers discuss is related to problems of the code review process, and by automating some of the toils of the review process, we believe it is possible to improve the overall quality of the review.

2.2. Machine Learning and Image Analysis

Machine Learning and Image Recognition has been used with success in many areas. For example, Lin et al. describe the successful use of deep learning for laser positioning [6]. An even more applicable subject is image processing and sentiment analysis. Qian et al. analysed twitter messages attempting to capture human expressiveness with image recognition [7]. Zhang et al. (2015) describe microblogs' sentiment analysis by integrating text and image features [8]. Although these papers were not software related, they positively demonstrate the success of machine learning in the context of image analysis as well as showing the possibility of detecting text sentiment.

2.3. Machine Learning and Source Code

Concerning research related directly to software source code, the paper, "Aesthetics Versus Entropy in Source Code," found that evaluating code beauty could be used for style checking [9]. Other studies have used machine learning and deep learning in code review systems to analyse code errors automatically. Bielki et al. introduced a machine learning-based system where the analyser learned to produce static analysis tools using a decision tree algorithm [10]. The system showed a coverage improvement but mentioned scalability and generalizability could be improved. Gupta and Sundaresan created a system using a 'long short-term memory' network called DeepCodeReviewer, which learned to review from human reviews. The authors explain in their paper they plan to improve the DeepCodeReviewer tool to learn continuously and personalize itself to a team or a repository [11]. These papers demonstrate the applicability of machine learning to the code review process, but do not address reviews using image processing.

3. BACKGROUND FOR METHODS

3.1. Data from Previous Work

This paper uses some data originally gathered in preparation for the 2019 IEEE Aerospace Conference in Big Sky, Montana (Aeroconf). At Aeroconf, we wanted to study what stylistic issues were most problematic for code reviews [7]. For this study, we created 'code snippets' and asked programmers to determine the proper outcome should the code be executed. This survey demonstrated that problematic code not only takes longer to review, but it is more often reviewed incorrectly. The survey showed that improperly formatted code had a review success of less than 90% on average, and on average it took about 22.5 seconds longer to review than properly formatted code. Some feedback received from this presentation indicated that many issues could be avoided simply by following coding standard rules. We agree, as in general, the issues

identified were stylistic, not logic-based. With this in mind we suggest these issues may be spotted visually, analogous to a tumour in a medical CT scan.

3.2. Scope of the Problem

Even though modern code editors can enforce properly formatted code, we were still surprised to see how much existing code violates style rules. It seems that even though modern tool kits are helpful, some issues of poorly formatted code linger. To demonstrate the ramifications of this problem, we downloaded several hundred projects from GitHub and scanned them for the common issues. As shown in Table 1, we discovered that most projects had at least some software issues and two projects had more than 15% of their lines associated with a issues. Figure 1 shows this table graphically. We used static analysis tools 'nsiqcppstyle'[12] and 'lizard'[13] to identify issues.

Table 1.

Percent	Value
0 to 1%	180
1.1% to 2%	181
2.1% to 3%	121
3.1% to 4%	51
4.1% to 5%	36
5.1% to 6%	23
6.1% to 9%	22
9.1% to 12%	15
12.1% to 15%	3
More than 15%	2
Increased Time	21%

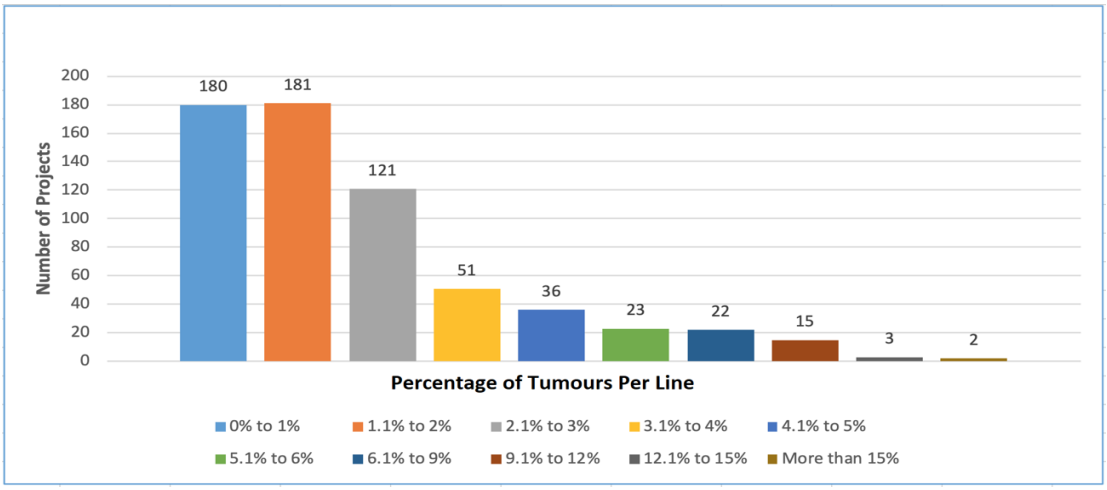


Figure 1. Percentage of Lines Associated with Tumours

3.3. Impact of Tumours

We will examine a hypothetical project to illustrate the possible consequences of tumours in source code. To arrive at a reasonable size for our hypothetical project, we analysed projects in our collection with very few tumours, specifically the projects where 0% to 2% of their lines were associated with a tumour, as shown in Figure 1. Static analysis of this group showed the median number of lines was 61,649, and the median number of tumours was 499, with the

probability of a line being part of a tumour being 0.008. Our hypothetical project was given characteristics based off these numbers and is shown in Table 2 with the results. We used the results of the Aeroconf [7] study to estimate how long code with and without tumours takes to review, and in our hypothetical project the presence of tumours increased the review time by 21%. It is also important to note that not only does the presence of tumours increase the review time, but it also reduces the accuracy of the review to lower than 90% [7].

Table 2. Hypothetical Project Specifications

Description	**Value**
Project Lines	61,649
Project Tumours	499
Code Segments Lines	56
Segment Count	1,100
No TumourSegment Review Time	37.5 Seconds
Segment Review Time with Tumour	59.5 Seconds
No Tumours Review Time	**11.5 Hours**
Tumour Review Time	**14.5 Hours**
Increased Time	**21%**

3.4. Deep Learning Review

In recent years, deep learning and convolutional neural networks (CNN) have been shown to be efficient in resolving complex non-linear problems and have become a prevalent approach for a broad range of tasks [14]. By automating the process of feature learning, CNN takes advantage of the concept of local information and effectively detects different deep features in multiple successive stacked layers [15]. This has resulted in CNN becoming one of the most popular methods for image recognition and classification. In this paper, we leveraged the powerful capacities of CNN to develop our recognition system.

VGG-19 and ResNet50 are state-of-the-art convolutional neural networks that are trained on the ImageNet dataset for solving image classification tasks in computer vision [15][16]. VGG-19 is a classic convolutional neural network that has 19 layers with trainable weights, in which exists 16 Convolutional layers and 3 fully connected layers [17]. ResNet50 (Residual Networks) has 50 layers [16]. They are both applied to solve tasks related to transfer learning. These networks are used for this study as they possess a rich feature representation and can likely yield good results.

4. METHODS AND ALGORITHMS

4.1. Identification of the Tumours

As previously mentioned, hundreds of GitHub projects were selected in order to conduct the analysis [18]. Problematic code was identified through static analysis and image files were created. Good code snippets were likewise identified and separated, and the resulting image files were also produced. The process for image creation is shown in Figure 2. In total, a set of about 44,000 images were prepared and collected for model training and testing, with about 38,000 images labelled as non-tumour and the remaining labelled as tumour (about 6,000). However, due

to the imbalanced distribution ofdata, only about 6,000 of the non-tumour images were randomly selected, along with the 6,000 images of tumour data. Doing this, the model potentially avoids the problem of overfitting, which is a major factor leading to the poor performance of a model and boosts its overall performance significantly.

Figure 3. Example Source Code Image

The created images are monochromatic and 228 pixels by 280 pixels in size. An example is shown in Figure 3.

4.2. Implementation

In training the models, a Keras framework [19][20] is utilized with a Tesla T4 GPU [21]. All the models are rigorously trained over 100 epochs. Data is broken down into batches of 64 images. Different optimizers were tried to accelerate the training process, and stochastic gradient descent was selected with a learning rate of 0.001. Also, ImageDataGenerator [22] is employed to progressively manipulate and load data in batches and help monitor the training process.

As shown in Figure 4, ImageDataGenerator implements multiple stages in a single platform, ranging from the preprocessing of data, the code-snippets in this case, to the training and evaluating the resulting models. Using ImageDataGenerator makes the training and testing straightforward, and it also helps effectively monitor the training. We choose this class as an intermediary is because of the automation and convenience it offers. ImageDataGenerator is a very beneficial tool and is heavily used for big datasets demonstrating training automation and computational efficiency.

In this project, two advanced architectures for image classification, VGG-19 [23] and ResNet50 [24], as well as a customized CNN, were adopted. We felt that traditional machine learning methods would be ineffectual in capturing spatial and local features in images. Accordingly, we adopted the advancements of deep neural networks as the principal proposition of this study. In particular, VGG19 and ResNet50 were trained on top of two fully connected layers interwoven with a dropout layer and a pooling layer in between.

Transfer learning is regularly used with pre-trained models as a starting point for further development on the task at hand. Consequently, we exploited transfer learning to utilize the well-tuned pre-defined architectures to accelerate the result and speed up the training. With minor modifications on this predictive task, pre-trained models can harness the proven feature learning capacity and yield better outcomes.

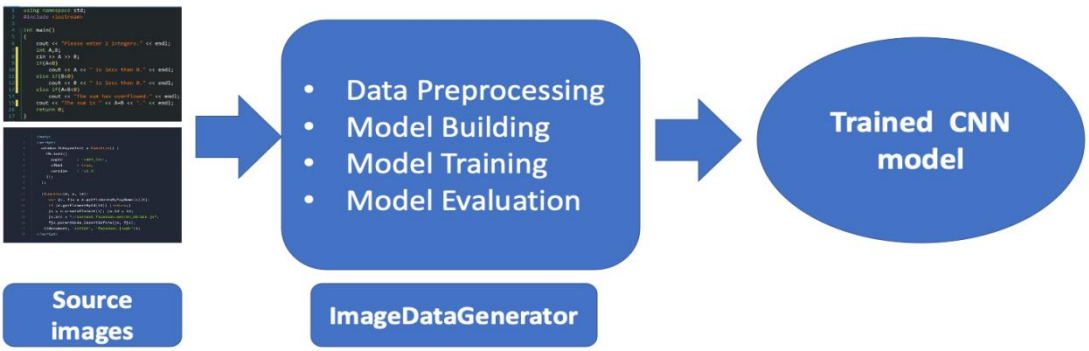


Figure 4. Pre-processing Data and Training Models

The customized CNN consists of three convolutional blocks which are followed by two fully connected layers at the end for classifying whether a given snippet of source code contains a tumour or not. By training the network, the model will be learning the weights and adjusting according to the training process that the model went through.

All the models are trained on the same data for the sake of accuracy rate comparison. The model selection is executed manually after the training and the model which achieved the best result in model evaluation phase is selected as a final model for predictions (Figure 5).

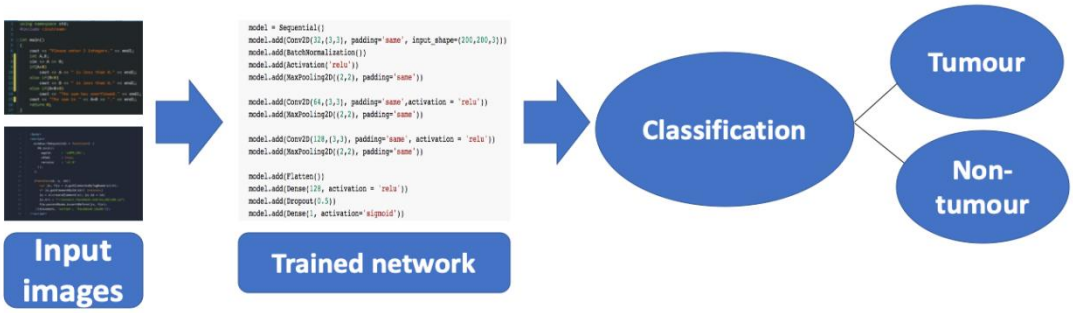


Figure 5. Model Evaluation

5. RESULTS AND DISCUSSION

To evaluate performance of the architectures, two metrics were used: accuracy and F1-score. Accuracy is adopted to measure how accurate the models are and F1-score is also elected so that the way the models make predictions can be observed and accessed more closely. We tried to balance the distribution of tumour and no-tumour training images, yet the probability of predicted images still might not be well distributed, so steps were taken to help mitigate noises and over fitting. The customized CNN architecture was found to be the best model with 80% accuracy and a 0.79 F1-score. VGG-19 and ResNet50 did not seem to perform as well on this task with results of only 59% and 55% respectively in terms of accuracy. Relevant comparisons are shown in Table 3.The problem might lie in the inherent nature and the amount of data presented. The

application of transfer learning makes a smooth transition to solve the problem, but the models do not align well with the data leading to the performances being weak. In addition, the amount of training data is not sufficient to bear the number of layers that constructs the notable distinction of the two state-of-the-art architectures. Our customized CNN has fewer layers and a less complicated architecture then ResNet50 and VGG-19. Also, since this is a binary classification problem, it seems to fit the data better than other pretrained powerful models such ResNet50 or VGG-19.

Table 3. Model Accuracy Rate Comparison

Description	**Accuracy**	**F1-Score**
VGG19	59%	0.62
ResNet50	55%	0.61
Our CNN	80%	0.79

As we performed a reduction in training data earlier, class distribution over the data is balanced, and hence the problem of overfitting was avoided. This is exhibited via the results in the confusion matrix shown in Table 4. The number of correctly predicted tumours is appropriate for the number of tumours exists in the dataset. In other words, the precision and recall rates of the target classes are relevant and consistent, which demonstrate the efficiency of the model.

The deep convolutional network designed for this project achieved an impressive outcome as compared to VGG-19 and Resnet50. Using a convolutional model, 80% accuracy was achieved in detecting software tumours in the code snippets, which shows it is possible for a CNN to identify software segments with code tumours.

Even though we obtained a satisfying outcome, there are still gaps that could be filled in order to achieve a better effect. On the one hand, data is essential for any CNN model to operate well on a given task. Therefore, with improved tools to extract and process data, we could expect the model to perform better, and accordingly gain more effect. On the other hand, a larger and more sophisticated network can be employed to learn more complex features with the introduction of both spatial and temporal information into the network. These topics should be addressed in future research.

Table 4. Classification Report of the Final Model

Description	**Precision**	**Recall**	**F1-Score**
No tumour	0.77	0.85	0.81
Tumour	0.83	0.74	0.78
Macro average	0.81	0.80	0.80
Weighted Average	0.81	0.80	0.80

6. CONCLUSION

Even for humans, visually recognizing a code tumour in software is a difficult task. Applying deep learning image classification brings a huge advancement and a positive outcome. This paper shows it is possible to use a convolutional neural network with up to 80% accuracy to identify patterns in code. Though our work identifies patterns that might be spotted by static analysis tools, it is possible that other tumour styles can be identified and discovered by our method. It seems it is possible to identify any bad looking code using machine learning and image recognition. Other methods should certainly be explored and researched since this work shows promising results, and tools can be created to enhance code review practices and perhaps other aspects of software development. Further research is required to make this system practical and useful.

ACKNOWLEDGMENTS

Special thanks to the University of St. Thomas for supporting this work.

REFERENCES

[1] Bavota, G. & Russo, B. (2015). "Four eyes are better than two: On the impact of code reviews onsoftware quality," 2015 IEEE 31st International Conference on Software Maintenance and Evolution, ICSME 2015 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 81–90. doi:10.1109/ICSM.2015.7332454

[2] Kozbelt, A. ; Dexter, S.; Dolese, M.; Seidel, A. (2012). "The Aesthetics of Software Code: A quantitative exploration," Psychology of Aesthetics, Creativity, and the Arts, Vol. 6, No. 1, 57–65. doi:10.1037/a0025426

[3] Yazdani, M.& Manovich, L. (2015). "Predicting social trends from non-photographic images on Twitter," Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015, Institute of Electrical and Electronics Engineers Inc., 1653–1660. doi:10.1109/BigData.2015.7363935

[4] Ebert, F., Castor, F., Novielli, N., &Serebrenik, A. (2019). "Confusion in code reviews: Reasons, impacts, and coping strategies," In 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 49-60). IEEE.

[5] Fatima, N.; Nazir, S.; Chuprat, S. (2020). "Understanding the Impact of Feedback on Knowledge Sharing in Modern Code Review," Institute of Electrical and Electronics Engineers (IEEE), 1–5. doi:10.1109/icetas48360.2019.9117268

[6] Lin, C.-S.; Huang, Y.-C.; Chen, S.-H.; Hsu, Y.-L.; Lin, Y.-C. (2018). "The Application of Deep Learning and Image Processing Technology in Laser Positioning," Applied Sciences, Vol. 8, No. 9, 1542. doi:10.3390/app8091542

[7] Dorin, M. & Montenegro, S. (2019). "Eliminating Software Caused Mission Failures," IEEE Aerospace Conference Proceedings,IEEE Computer Society. doi:10.1109/AERO.2019.8741837

[8] Zhang, Y.; Shang, L.; Jia, X. (2015). "Sentiment analysis on microblogging by integrating text and image features," Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 9078), Springer Verlag, 52–63. doi:10.1007/978-3-319-18032-8_5

[9] Coleman, R. & Boldt, B. (2017). "Aesthetics versus entropy in source code," In Proceedings of the International Conference on Software Engineering Research and Practice (SERP) (pp. 113-119). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[10] Bielik, P.; Raychev, V.; Vechev, M. (2016). "Learning a Static Analyzer from Data," Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10426 LNCS, 233–253

[11] Gupta, A. &Sundaresan, N. (2018). "Intelligent code reviews using deep learning," doi:10.1145/nnnnnnn.nnnnnnn

[12] Yoon, J & Kunal, T. (2014). "Cpp style checker in python," From https://github.com/kunaltyagi/nsiqcppstyle, accessed 21-7-2020

[13] Yin,T. (2012). "A simple code complexity analyser without caring about the C/C++ header files or Java imports, supports most of the popular languages," fromhttps://github.com/terryyin/lizard, accessed 21-7-2020

[14] Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; Chen, T. (2018). "Recent advances in convolutional neural networks, Pattern Recognition, Vol. 77, 354–377. doi:10.1016/j.patcog.2017.10.013

[15] Shin, H. C.; Roth, H. R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R. M. (2016). "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," IEEE Transactions on Medical Imaging, Vol. 35, No. 5, 1285–1298. doi:10.1109/TMI.2016.2528162

[16] Akiba, T., Suzuki, S., & Fukuda, K. (2017). "Extremely large minibatch sgd: Training resnet-50 on imagenet in 15 minutes," arXiv preprint arXiv:1711.04325.

[17] Mateen, M.; Wen, J.; Nasrullah; Song, S.; Huang, Z. (2018). "Fundus Image Classification Using VGG-19 Architecture with PCA and SVD," Symmetry, Vol. 11, No. 1, 1. doi:10.3390/sym11010001

[18] GitHub (2020). "The world's leading software development platform," from https://github.com/, accessed 21-7-2020
[19] Keras (2020). "Keras: the Python deep learning API," from https://keras.io/, accessed 21-7-2020
[20] Manaswi, N. K. &Manaswi, N. K. (2018). "Understanding and Working with Keras, Deep Learning with Applications Using Python," Apress, 31–43. doi:10.1007/978-1-4842-3516-4_2
[21] NVIDIA (2020). "NVIDIA T4 Tensor Core GPU for AI Inference | NVIDIA Data Center," from https://www.nvidia.com/en-us/data-center/tesla-t4/, accessed 21-7-2020
[22] Karis (2020). "Image data preprocessing," from https://keras.io/api/preprocessing/image/, accessed 21-7-2020
[23] He, K.; Zhang, X.; Ren, S.; Sun, J. (2016). "Deep residual learning for image recognition," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Vol. 2016-December), IEEE Computer Society, 770–778. doi:10.1109/CVPR.2016.90
[24] Simonyan, K. & Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556.

AN INTELLIGENT AND DATA-DRIVEN MOBILE PLATFORM FOR YOUTH VOLUNTEER MANAGEMENT USING MACHINE LEARNING AND PREDICTIVE ANALYTICS

Alyssa Huang[1] and Yu Sun[2]

[1]Arnold O. Beckman High School, Irvine, CA 92602, USA
[2]California State Polytechnic University, Pomona, CA, 91768, USA

ABSTRACT

Volunteering is very important to high school students because it not only allows the teens to apply the knowledge and skills they have acquired to real-life scenarios, but it also enables them to make an association between helping others and their own joy of fulfillment. Choosing the right volunteering opportunities to work on can influence how the teens interact with that cause and how well they can serve the community through their volunteering services. However, high school students who look for volunteer opportunities often do not have enough information about the opportunities around them, so they tend to take whatever opportunity that comes across. On the other hand, as organizations who look for volunteers usually lack effective ways to evaluate and select the volunteers that best fit the jobs, they will just take volunteers on a first-come, first-serve basis. Therefore, there is a need to build a platform that serves as a bridge to connect the volunteers and the organizations that offer volunteer opportunities. In this paper, we focus on creating an intelligent platform that can effectively evaluate volunteer performance and predict best-fit volunteer opportunities by using machine learning algorithms to study 1) the correlation between volunteer profiles (e.g. demographics, preferred jobs, talents, previous volunteering events, etc.) and predictive volunteer performance in specific events and 2) the correlation between volunteer profiles and future volunteer opportunities. Two highest-scoring machine learning algorithms are proposed to make predictions on volunteer performance and event recommendations. We demonstrate that the two highest-scoring algorithms are able to make the best prediction for each query. Alongside the practice with the algorithms, a mobile application, which can run on both iPhone and Android platforms is also created to provide a very convenient and effective way for the volunteers and event supervisors to plan and manage their volunteer activities. As a result of this research, volunteers and organizations that look for volunteers can both benefit from this data-driven platform for a more positive overall experience.

KEYWORDS

Machine learning, Predictive Analytics, Flutter, Volunteer Management, Scikit-learn

1. INTRODUCTION

Community service and volunteering events are prevalent, not only helping serve the underserved but also providing people, particularly younger students, with the opportunity to better their community and influence those around them [3]. Since many universities seek students who demonstrate the passion and ability to impact those around them, community service and volunteering often play a significant role in college applications. Aside from general community

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 169-183, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101515

service events such as packing food boxes for the poor and donating to toy drives, there are also specialized events that focus on a particular topic or subject. Participating in volunteering projects relating to a student's interest is both gratifying and beneficial to the student's resume [1][2].

Identifying the volunteering events that fit volunteers has never been easy for high school students, as they might need to take their interests and abilities, logistics of locations, time commitment, skills required for the jobs, and even the attitude of the organization staff into consideration. In reality, however, most high school students will just take any volunteering opportunity available to them because they don't have enough information they need in order to make a better-informed judgment about whether the event fits their pursuit, passion, and ability. On the other hand, it is also very challenging for the volunteer event supervisors to evaluate the volunteers and select those who are qualified for the jobs. Instead, they may just follow a first-come, first-serve basis, along with some minimum basic requirements. Such practices may result in unfavorable volunteering outcomes: volunteers may not really be passionate about or capable of doing the jobs that were assigned to them, and organizations do not get the quality services they expect from the volunteers. In most cases, there is also very little that organizations can do with the quality of the volunteer service since most volunteer jobs are free. In the meantime, there are many qualified high school teenagers out there who are looking for volunteer opportunities — they just don't have a centralized platform to help them identify opportunities that fit them.

This research paper focuses on building an intelligent, data-driven platform that allows volunteers to plan and manage their volunteer activities and enables organizations to identify the volunteers who are most likely to fit the job by finding connections between volunteer profiles, their expected event or job performance, and best-fit volunteer opportunities.

Open Questions. In this paper, we aim to answer the following inquiries:

1. *Given the user's previous performance on events, interests, and age, among other profile attributes, as well as the event's type, time commitment requirement, etc., how well will the user perform in volunteering for this event?*

2. *From the user's interests and previously participated events, what event should be recommended to the user?*

Many studies have found other methods in predicting volunteer activity and outcomes. For example, one study found that grouping volunteers into different motivational groups will predict volunteer outcomes better than knowing only the motives of the volunteers. On the other hand, another study proves that the motives of volunteers have a weaker than expected effect on the length of volunteerism and service. Rather, a volunteer's identity and expectations are what best predicts the period of volunteerism. A different study links subjective experiences before and during volunteering to the ultimate satisfaction and length of volunteer service. This study, however, aims to predict the effects of volunteerism and volunteer interest on volunteer outcomes.

Solution. In order to make reasonable predictions to answer the previously mentioned queries, a three-part system was created to gather data, store data, and make predictions using the data. To gather data, we implemented a multi-use mobile application that volunteers and supervisors may use to create and join events, log and approve hours, and log volunteer interests, etc. The app is easily navigable and contains many functions that volunteers and supervisors need. Any data gathered from this app is stored inside an organized cloud database. The database has functions that also allows for secure user authentication, so users can log in safely. At any time, the app may query any data from the database about the user to display on the app. A machine learning

framework that gathers data from volunteers and makes predictions was implemented as well. We generated dummy data, mimicking data of real volunteers, in order to train the machine learning algorithms. Through multiple trials, we were able to find the best-performing algorithms to use in predicting each question. In the end, the chosen algorithms are run on a server that the app may access at any time to show prediction results to users. The use of dummy data allowed the machine learning algorithms [4][5] to train and produce the best possible results. Together, the three systems work together to accurately make the best predictions possible.

Various experiments were devised in order to find the top-performing algorithm for making predictions for each question. For each question, multiple experiments were run, differing each time in areas such as algorithm parameters, volunteer feature sets, and training data sizes. Changing these areas allows us to determine if a certain prediction model depends heavily on one of these areas. Training the machine learning algorithms to make the best possible prediction required tens of thousands of generated dummy data. To easily compare the prediction results with a separate dataset to prove its accuracy, the data is randomly split into two sets: a training set and a test set. Through running multiple trials, the mean accuracy of the prediction was generated for classification algorithms, and the coefficient of determination, the R2 value, of the prediction for regression algorithms. Each experiment generated multiple mean accuracies and R2 values that were then graphed in order to find the top-performing algorithm for each question. Analysis of the graphs shows that each prediction model yields a consistent accuracy of prediction through multiple trials, however, the accuracy of prediction differs greatly between multiple models. Any further interesting finds were noted as well. It was found that generally, one prediction model trumped the other prediction models, regardless of changes in algorithm parameters, volunteer feature sets, or training data sizes.

The rest of this paper is structured as follows: Section 2 will detail the multiple challenges faced in this study and how they were overcome; Section 3 will describe the methodology and solution in greater detail; Section 4 details the experiments that were performed in this study, as well as a thorough analysis of the results; Section 5 will list any related works that have also been done regarding volunteering; and lastly, Section 6 will conclude the study and state any future work that may be done.

2. CHALLENGES

In order to build the data-driven application system, a few challenges have been identified as follows.

2.1. Challenge 1: The Heterogeneous Data Communication Between Multiple Roles

Traditional management of volunteers by event supervisors include email, texting, or online chat services. However, these methods of communication may be slow and require extra planning. This application aims to provide instant management of events and volunteers. Appropriated for multiple types of users from volunteers to supervisors, this application must facilitate data communication between the supervisors and the volunteers. Volunteers must be able to see event data and join events the supervisors have created, and supervisors must be able to view volunteer profile information, create events, and manage volunteers. When a supervisor creates an event, there must be some way for volunteers to view the event and join the event, and when a volunteer joins an event, the supervisor must be able to view the volunteer and his/her profile. Real-time data updates must be available for users, so no communication errors occur. To solve this challenge, a cloud database will be used, so the application can read and write data at any moment. Any information that users log, including hours, events, or profile changes, will be

written to the cloud database. This will allow other instances of the application to read the changes that have been made.

2.2. Challenge 2: The Complexity of the User Roles and the Associated Data Structures

In order to ensure efficient and effective communication between roles, any data logged by users must be structured to allow for easy retrieval and writing of data. Since the application is targeted towards both supervisors and volunteers, there are many types of data subsets that need to be organized and collected. These may include variables such as volunteer hours log, volunteer joined events, supervisor created events, or profile changes. Without the correct organization, data may be redundant or lost, causing problems in readability and query time. The data being collected is organized into multiple collections and subcollections. For this application, there are three main entities: profile information, organizations, and events. Profile information includes any data regarding users, such as name, email, age, gender, joined/created events are logged under this entity. The data in the Organizations collection includes organization name, join code, administrators, members, etc. The Events collection includes the organization it belongs to, date, time, supervisors, volunteers, etc. Together, all three entities are connected, and often, data from one entity may be used to query data from another entity. For example, accessing the profile information will allow us to view the list of events that the volunteer is currently signed up for, and using this list of events, further information regarding each event may be queried from the Events collection.

3. SOLUTION

This application is an intelligent, data-driven volunteer management platform that allows volunteers to track hours and join events and allows supervisors to create and manage activities. The application has three components that ensure the highest quality performance for the user: the frontend application, the backend cloud database, and the backend machine learning server. The application, written in Dart in Flutter [7], can run on both iOS and Android devices. It hosts many features that allow both volunteers and supervisors to keep track of their workflow. The app is also able to communicate with the cloud Firebase. The cloud database uses Google Firebase [8] and stores data as JSON objects [9]. Google Firebase contains a multitude of features, such as analytics, authentication, and storage. The application directly writes data into the Firebase and is later able to read the data. Lastly, the machine learning code is written in Python and hosted on a server. The Python server will access any data gathered by the application and use the data to make predictions about the user. There are multiple machine learning algorithms, to ensure each query uses the best prediction model to its needs. These three components constantly exchange information. The main technical challenge is keeping the three systems in check and communicating correctly, with data produced on two different occasions: the frontend application and the backend Python code. It is crucial to keep all data organized and stored in the correct places to avoid any errors in communication between the systems. The following section will describe each component in further detail.

The frontend of this application was built in the IntelliJ integrated development environment (IDE) using Google's open-source UI software development kit Flutter. Flutter uses the object-oriented programming language, Dart [6].

The application, targeted towards both the volunteers and supervisors of events, will provide each user access to one of two platforms: the volunteer only platform and the supervisor only platform. Each platform has its respective pages, depending on the user. Upon entering the app,

all users will be prompted with the same login screen, with the option to either login or sign up for an account. After logging into an account, a home screen will appear with a bottom navigation bar containing different pages depending on whether the user is a volunteer or a supervisor. In the case of the volunteer only, the pages Home, Events, Hours, and Profile are shown. Figure 1, shown below, displays the volunteer only platform. In each page, users will be able to view their enrolled events, view any upcoming events, log their volunteer hours, and edit their profile, among other features. Any data, including hours, registered events, and profile changes, is stored in a cloud database that the application will be able to access.

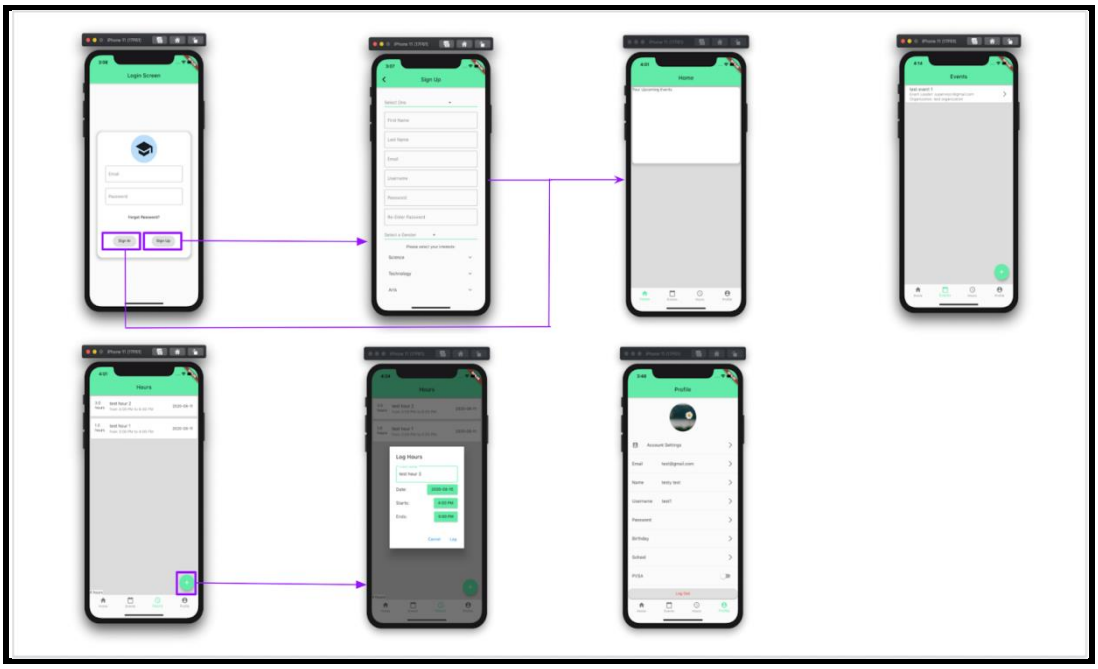


Figure 1. The Volunteer Only platform storyboard

The supervisor only platform follows a similar flow, shown in Figure 2. After logging into a supervisor only account, the home page appears, with a bottom navigation bar that contains the pages Home, Events, Organizations, and Profile. In these pages, the user will be able to see their upcoming events, approve volunteer hours, create and manage events, join and create volunteer organizations, and edit their profile, among other features. Any data shown in these pages are stored in the cloud database as well.

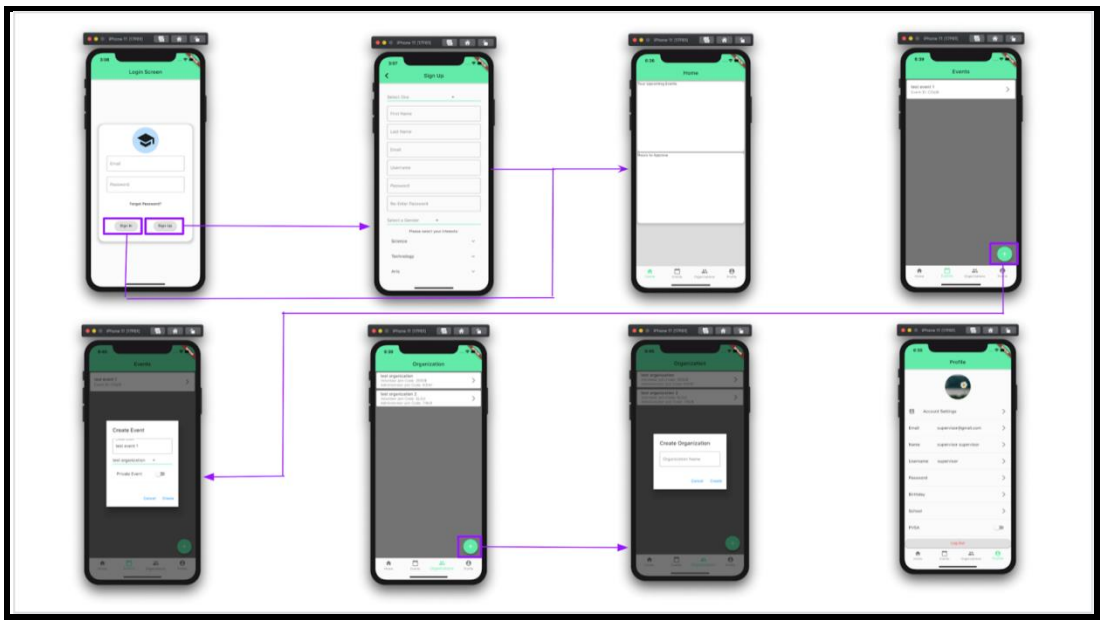


Figure 2. The Supervisor Only platform storyboard

The application includes complex code to load data from the firebase. This function is attained by employing elements such as asynchronous functions and future data types. The use of the two components mentioned before allows the application to run multiple tasks at once and store and use data that will exist in the future. Shown in Figure 3 is an example of the code. In the snippet, a query is sent to the database using an asynchronous function. The use of an asynchronous function allows the application to make a query to the Firebase while other tasks are running or loading. Usually, when using a variable, there must be a declaration of what the variable is representing. However, this can be problematic if the data does not exist yet, such as while data is being loaded from a database. The use of a future data type solves this problem, as instead of throwing an error, it can wait for the data to be a non-null type before showing the data.

```
Future<Map> retrieveEventData(String organization, String eventCode) async {
  var eventInfo;
  await firestore
      .collection('organizations')
      .document(organization)
      .collection('events')
      .where('eventID', isEqualTo: eventCode)
      .getDocuments()
      .then((value) {
    var documents = value.documents[0];
    eventInfo = {
      'organization': documents['name'],
      'eventLeader': documents['eventLeader'],
      'eventID': documents['eventID'],
    };
  });
  return eventInfo;
}
```

Figure 3. Snippet 1 of the application code

In order to collect the data, the application implements a page that asks users for their interests. Figure 4 shows the code implementation of the application. In the snippet of code, multiple Dart widgets are utilized to create elements on the screen, including the ExpansionTile, CheckboxListTile, and the Text widgets. These widgets allow users to check their interests and input any preferences they may have.

```
Container(
  child: Column(
    children: [
      Text('Please select your interests:'),
      ExpansionTile(
        title: Text('Science'),
        children: [
          CheckboxListTile(
            title: Text('Biology'),
            value: _biologyChecked,
            controlAffinity: ListTileControlAffinity.leading,
            onChanged: (bool value) {
              setState(() {
                _biologyChecked = value;
              });
            },
          ), // CheckboxListTile
          CheckboxListTile(
            title: Text('Chemistry'),
            value: _chemistryChecked,
            controlAffinity: ListTileControlAffinity.leading,
            onChanged: (bool value) {
              setState(() {
                _chemistryChecked = value;
              });
            },
          ), // CheckboxListTile
          CheckboxListTile(
            title: Text('Physics'),
            value: _physicsChecked,
            controlAffinity: ListTileControlAffinity.leading,
            onChanged: (bool value) {
              setState(() {
                _physicsChecked = value;
              });
            },
          ), // CheckboxListTile
        ],
```

Figure 4. Snippet 2 of the application code

The backend of this application includes two portions: the machine learning server and the cloud database. To make predictions regarding the users, the machine learning model and cloud database communicate consistently. Since user data is stored in the cloud database, the machine learning model must be able to read data from the database and produce a result and subsequently store this result into the cloud for the front end to query.

Google Firebase is used to store and read data efficiently. The Firebase contains features that allow user authentication by email, phone number, or other accounts without any additional code. Using this feature, users may create accounts and safely log in through the app. In the Firebase, there are three main types of entities stored as documents in collections of the same name: users, organizations, and events (Figure 5). Each entity has its own properties, and some have their own subcollections. The Firebase is organized sensibly as such so that the data can be queried efficiently by the application.

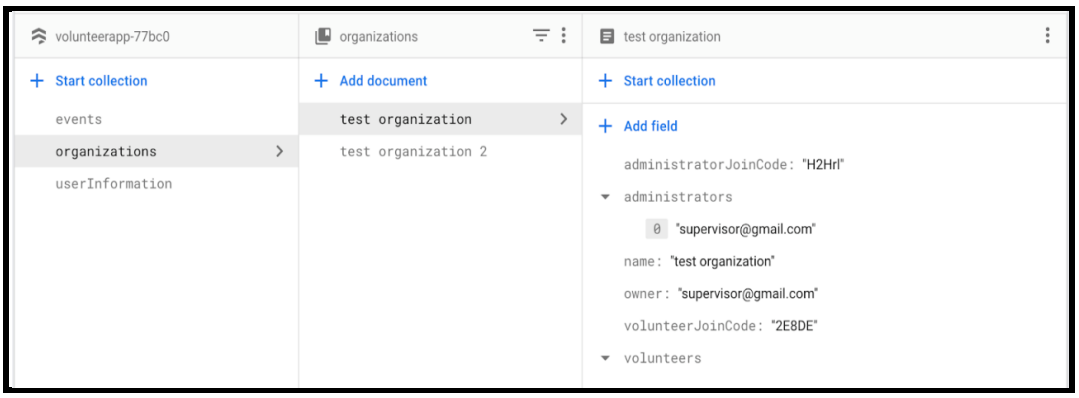


Figure 5. The "organizations" collection within the Firebase

The machine learning framework in the backend is written in Python and employs Scikit-learn, a software machine learning library for Python. Scikit-learn features a myriad of regression, classification, and clustering algorithms. Scikit-learn is used to make predictions on the two inquiries mentioned above.

Multiple algorithms are tested to find the top-performing algorithms for each prediction. To train the prediction models, tens of thousands of accounts of dummy data is generated, with data including age, gender, interests, etc. The variables within the data are linked, for example with volunteer performance increasing with age. Next, to train the algorithms, a class in the Sci-kit Model Selection library named train test split is used (Figure 6). This class divides the dummy data into two groups; one to train the model and the other to test the model. This allows the predictions of the training set data to be compared accurately with the actual results from the test data. After running multiple training tests, the algorithm that consistently performed the highest was confirmed.

```
X, y = load_iris(return_X_y=True)

X_train, X_test, y_train, y_test = train_test_split
(input_data, output_data, test_size=0.2, random_state=0)

model = linear_model.LinearRegression()
model.fit(X_train, y_train)
print(f"{model.score(X_test, y_test)}")
```

Figure 6. A snippet of the Python code

4. EXPERIMENT

When signing up to volunteer for an event, a volunteer may not know how well they will perform in the event due to not completely understanding the requirements and expectations of volunteering for the event. In this experiment, we aim to find the best machine-learning algorithm to predict a volunteer's performance on a scale of one to ten based on preceding experience, user information, and event information.

To train the algorithm, data from previous users and their ratings after each event is used. When a prediction is required, it will take into account the current user's profile information and the user's past ratings to make a prediction on how well their performance on the specified event will be. Using the Scikit-learn machine learning library in Python, multiple algorithms, including linear regression, polynomial regression, random forest classifier, k-nearest neighbor classifier, Gaussian naive Bayes, and random forest bagging classifier, are tested for accuracy in prediction [13][14][15]. Each prediction model is trained with dummy data variables that include volunteer age, event type (science, engineering, math, writing, public speaking, etc.), interests, and a rating on the event. Although all volunteer profile information and the event type is randomly generated, the performance is tied to these generated numbers. For example, if the event type is science, and if the user is interested in science, then the minimum performance will be increased. Similarly, the minimum performance of the volunteer will increase with the age of the volunteer.

Multiple experiments are run, each with multiple trials. Each experiment alters one part: algorithm parameters, volunteer feature sets, or training data sizes. Through these experiments, we will be able to determine if an algorithm depends on certain factors to produce the best results. Each algorithm is then tested and scored to find the most accurate prediction model.

Figure 7 illustrates the results of experiment 1. Over twelve trials, the random forest bagging classifier and the Gaussian naïve Bayes consistently outperformed the rest, followed closely by the random forest classifier. By contrast, the linear regression prediction model consistently scored the lowest.

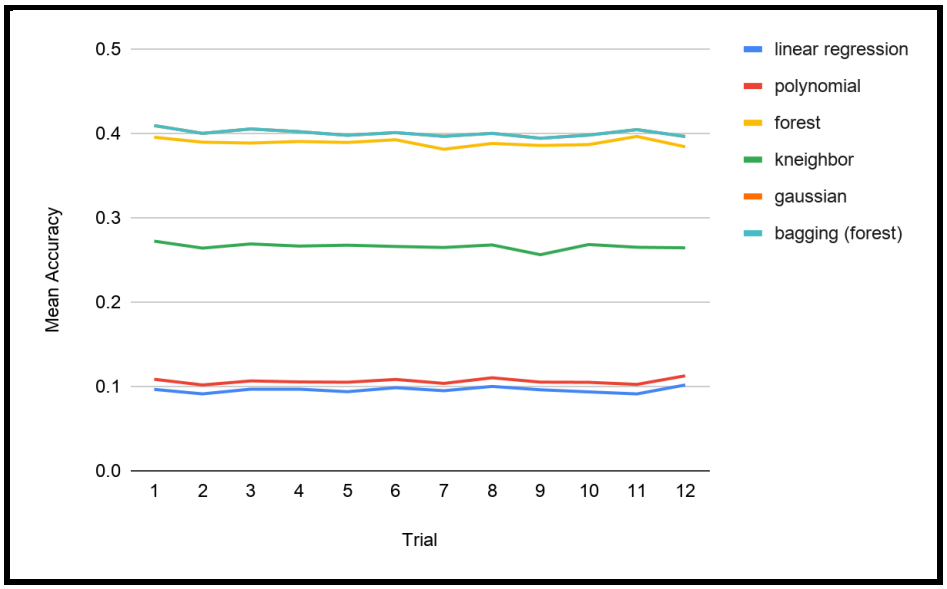


Figure 7. Experiment 1.1 Results Graph

An interesting find was that the random forest bagging classifier and the Gaussian naive Bayes algorithms scored exactly the same numbers, as can be seen in Table 1. After further examination, it was concluded that this was an error in the Python code.

Table 1. Experiment 1.1 Results

	r^2 value		mean accuracy on the given test data and labels	mean accuracy on the given test data and labels	mean accuracy on the given test data and labels	mean accuracy on the given test data and labels	mean accuracy on the given test data and labels
trial	linear regression	polynomial	svm	forest	kneighbor	gaussian	bagging (forest)
1	0.09659642072	0.1085847523		0.3956	0.2724	0.40945	0.40945
2	0.0912802714	0.1018627497		0.3898	0.26415	0.40025	0.40025
3	0.09686422492	0.1066349464		0.38875	0.2691	0.40545	0.40545
4	0.0968569656	0.1054718431		0.3906	0.26655	0.40225	0.40225
5	0.09385395767	0.1050779491		0.38945	0.2676	0.39795	0.39795
6	0.09858931497	0.108387892		0.39265	0.2661	0.40115	0.40115
7	0.09501043918	0.1037082948		0.3814	0.2649	0.39675	0.39675
8	0.1001078768	0.110352902		0.38825	0.2679	0.40025	0.40025
9	0.09611727669	0.1052227573		0.38585	0.25635	0.3944	0.39445
10	0.09364955503	0.104988994		0.3869	0.2684	0.3982	0.3982
11	0.09123142702	0.102510737		0.3966	0.26515	0.40465	0.40465
12	0.1018312775	0.112689413		0.38435	0.2645	0.3965	0.3965

To ensure that the random forest bagging classifier would be the best predictor of volunteer performance, another experiment (Experiment 2) was conducted, changing the feature set of the volunteer dummy data. In addition to the previous data set, which included volunteer age, event type (science, engineering, math, writing, public speaking, etc.), interests, and a rating on the event, this experiment added extra variables: the number of hours required for the volunteer to volunteer in the event and the volunteer's previous average ratings in previous events. These variables were once again linked to the randomly generated performance number: the higher the volunteer's previous rating, the higher the minimum performance of the volunteer will be, and the more hours are required for the event, the lower the minimum performance of the volunteer will be.

Figure 8, the graph for experiment 2, shows that the polynomial regression, random forest classifier, and the Gaussian naive Bayes algorithms scored very similarly. In contrast with Experiment 1, where polynomial regression was one of the worst-performing algorithms, polynomial regression scored exceptionally.

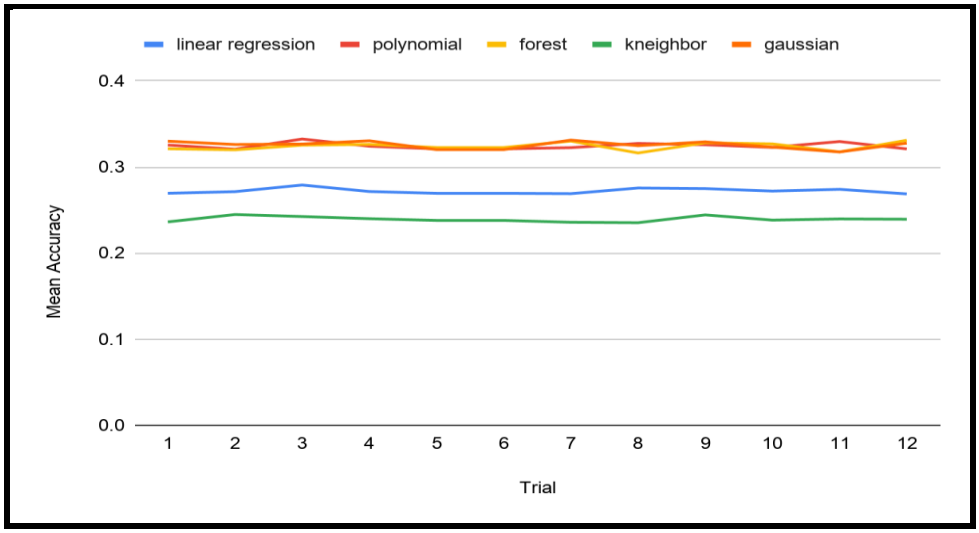


Figure 8. Experiment 1.2 Results Graph

In experiment 3, we altered the model parameters of one prediction model. Since the random forest bagging classifier did not currently include any parameters, we altered the parameters of

the second-best predictor thus far, the random forest classifier. The random forest classifier requires two parameters: n-estimators and max depth.

In the results, shown in Figure 9, it is clear that most changes in parameters yielded similar results, except when n-estimators was a small number and max depth was a large number.

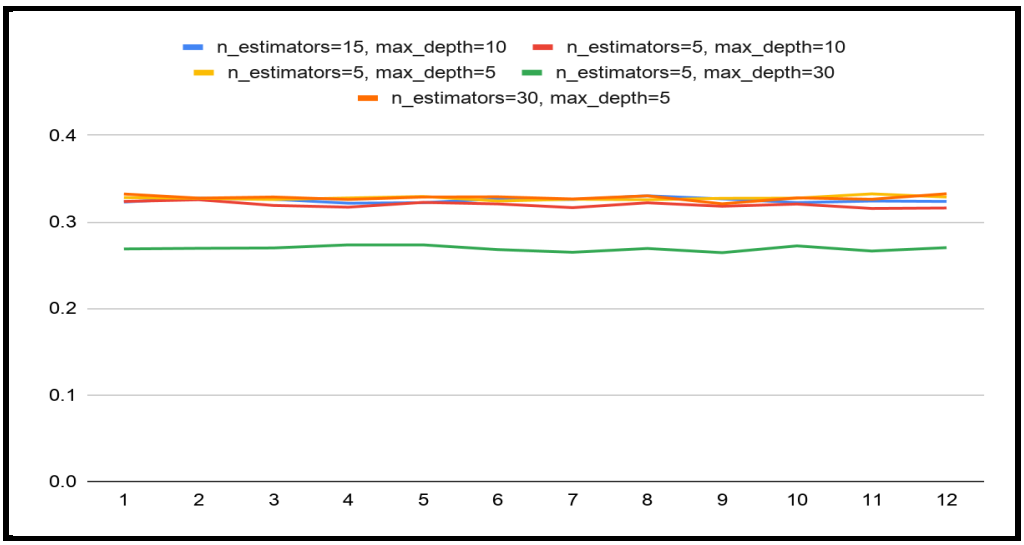


Figure 9. Experiment 1.3 Results Graph

Experiment 4 explored the impact of sample size on the accuracy of the prediction. In Figure 10, it is apparent that in the trials with 1000 volunteers, the prediction model scores less consistently than the trial with 10000 volunteers does. It was found that rather than increasing the accuracy of the prediction models, training the prediction models with more sets of data would increase the consistency of the predictions.

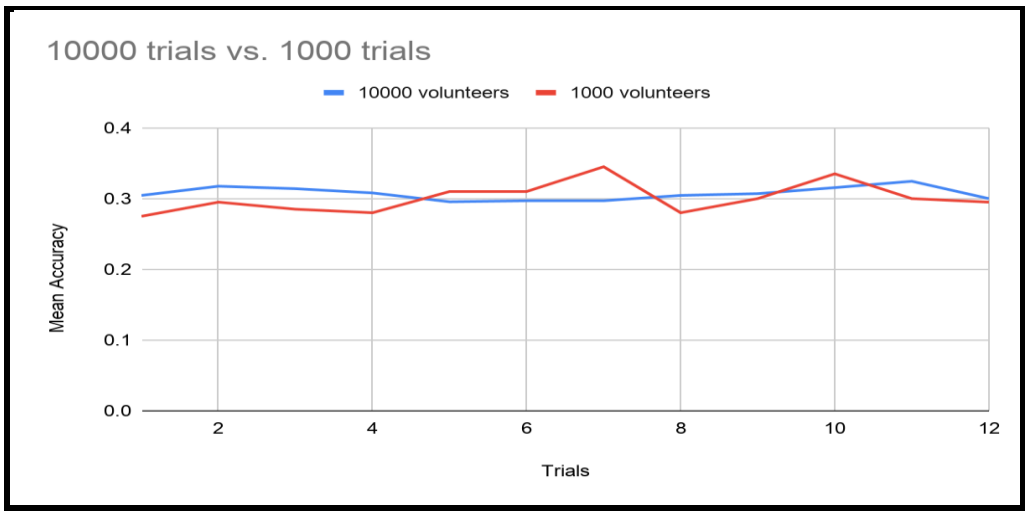


Figure 10. Experiment 1.4 results graph

Overall, it was found that the random forest bagging classifier was the highest performing prediction model when trying to predict the volunteer's performance. Furthermore, avoiding the

use of random forest algorithm parameters of a low n-estimators value and a high max depth value and training the datasets with more trials will yield better results.

The application will employ event recommendations for users to volunteer in. This will improve volunteer and supervisor experience in multiple ways: 1) Volunteer exposure to suitable events will increase, allowing volunteers to serve in more events, and 2) The supervisors' events will be advertised to volunteers who are more interested and experienced, increasing the quality of volunteer service. In this last experiment, the goal was to accurately recommend a category of the event to a volunteer based on the events they had previously volunteered in, profile information, and event information.

As performed in the previous experiment, the machine learning algorithms are trained using previous user data. Both previous users' event data and the user's current data will be used in order to make the prediction. The Scikit-learn machine learning library in Python allows usage of linear regression, polynomial regression, support vector machine, random forest classifier (n-estimators = 30 and max_depth = 30), k-nearest neighbor classifier (n-neighbors = 3), and Gaussian naive Bayes to make predictions [13][14][15]. Each prediction model is trained with dummy data variables including the number of each specific type of event (science-oriented, computer-science-oriented, writing oriented, public-speaking-oriented, math-oriented, and engineering-oriented) that the volunteer has served in. The generated recommendation is related to the number of events of each type that the volunteer has attended.

Within this experiment, multiple experiments with multiple trials were run. Each experiment alters one of each: algorithm parameters, volunteer feature sets, and training data sizes. These experiments will allow us to determine which machine learning algorithms will predict recommendations for users the best.

In experiment 1, the only data taken into account was the amount of each type of event the volunteer participated in. The type of event that the volunteer served most in was the recommended event. Figure 11 illustrates the results of experiment 1. The two highest-scoring prediction models were the support vector machine and the random forest classifier, and the random forest classifier scored slightly higher than the support vector machine on certain accounts. Linear regression was the lowest scoring prediction model

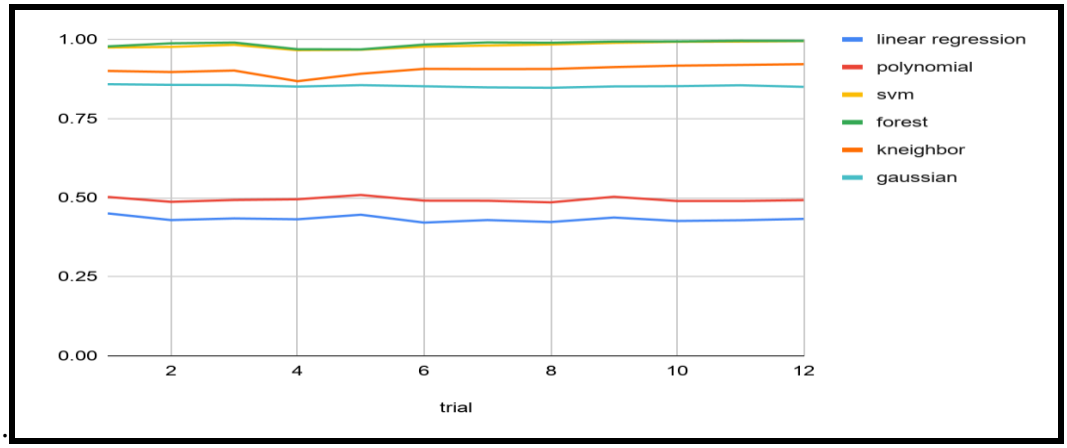


Figure 11. Experiment 2.1 results graph

During experiment 2, the two model parameters, n-estimators and max-depth, of the highest-scoring model, the random forest classifier, were altered. Figure 12 shows the results of the

second experiment. Overall, the different groups scored similarly, with the exception of n-estimators = 5 and max depth = 5, and n-estimators = 30 and max depth = 5. The highest scoring was n-estimators = 30 and max depth = 30, which is the same model parameters used in experiment 1.

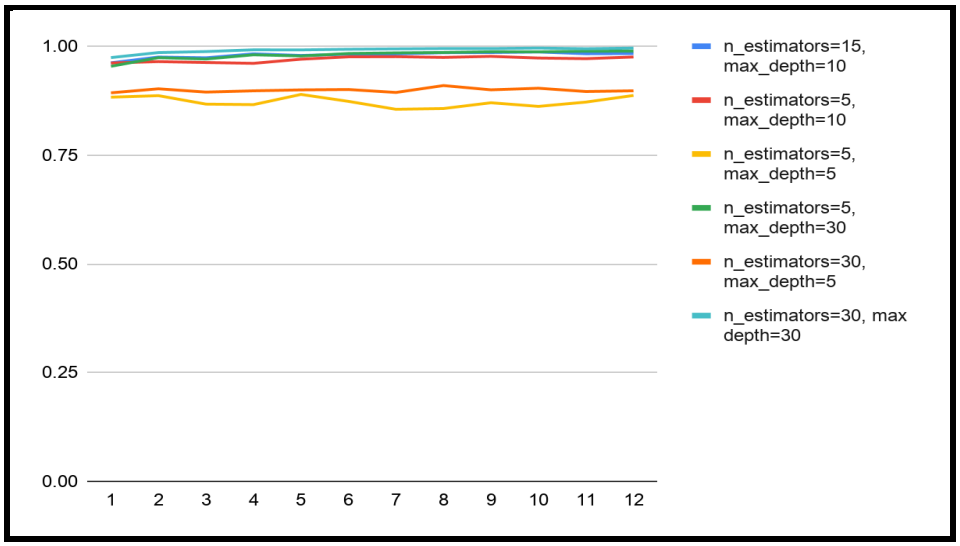


Figure 12. Experiment 2.2 results graph

In experiment 3, the impact of sample size on the result was once again tested. The highest scoring prediction model, the random forest classifier, was tested in this experiment. Figure 13 shows the unanticipated results of the experiment. With the introduction of more trials, the machine learning algorithm performed worse than with fewer trials.

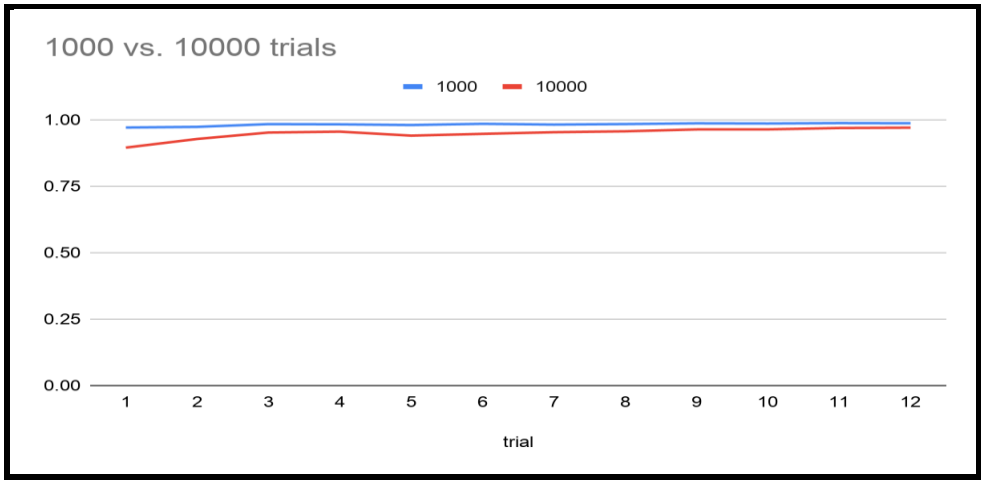


Figure 13. Experiment 2.3 Results Graph

5. RELATED WORK

Stukas, A. et al [10] propose an index that calculates the volunteer's matches to one or more of six motivational categories identified in previous research. The authors of the paper prove that the use of this index will predict volunteer outcomes better than volunteer motives and affordances alone can predict. Furthermore, the authors find the less structured organizational contexts are,

the greater the total matches' effect may be. In contrast, this paper uses machine learning algorithms to predict multiple outcomes, one of which is volunteer performance based on traits like age, gender, interests, etc. Rather than categorizing volunteers, this study connects the volunteer and his/her information directly to volunteer performance. Connections between volunteer performance and organizational structures are not covered in this paper.

Finkelstein, M. et al [11] predicted how volunteerism is affected by multiple factors, such as initial motives, personality, and role identity. The paper found that a volunteer's identity and expectations were the greatest predictors of the length of volunteering. Instead of searching for the basis of volunteerism, this paper aims to find patterns between a volunteer's disposition and the types of events volunteers may want or should volunteer in. These patterns will be used to predict events to recommend to the volunteer or the effectiveness of volunteering in an event to the volunteer's final goal.

Davis, M et al [12] explores how volunteer behavior is influenced by subjective experiences during volunteering and preceding experiences before volunteering. Two-hundred-thirty-eight real volunteers over nine organizations were studied over the course of a year. Findings indicate that, consistent with their model, volunteer involvement is influenced by the satisfaction felt from volunteering, but the continuance of volunteerism is not. This study uses dummy data and aims to predict how both objective and subjective factors influence the performance of the volunteers. These factors may include age, gender, and interests.

6. CONCLUSION AND FUTURE WORK

There are many organizations seeking volunteer services, and teenagers are valuable resources of energy, talent, and creativity to fulfill those volunteer jobs. While there is a fair amount of emphasis on how volunteer services can enhance a teen's college and scholarship applications, we should all realize that the meaning and value of volunteer services can go far beyond that. Being able to participate in the right service opportunities can contribute to developing a sense of self-confidence for the teens and can help shape their life-time opinions on the volunteer services to serve the needs of the less fortunate around them. Teens who have a positive experience in volunteer service are usually more responsible and have higher self-esteem and resilience when they become adults. Therefore, it is meaningful and valuable to build an intelligent, data-driven, and easy-to-use volunteer platform that provides teenage volunteers the information they need to choose the right service opportunities that accommodate their interests, talents, and time commitment. This will allow users to get involved with something they are truly passionate about and are capable of doing.

In this research paper, we first built a platform to collect the volunteer profiles and event activity information and turned them into structured data. Then we use advanced machine learning methods and predictive analytics to turn data into valuable and actionable intelligence for the volunteers and organizations. By identifying the best machine learning algorithms to predict the two queries introduced previously, volunteers will now be able to identify the right volunteer opportunities for them, while organizations will benefit from improved services provided by the best-fit volunteers. The results of this research can be used by all the volunteers and organizations providing volunteering opportunities for a brand-new, much more effective, and enriching volunteer experience in the future.

A major limitation of this research is that the dummy data that we used was generated by code; therefore, there could be some inconsistencies between the dummy data and real volunteer data. In the real world, there are much more demographic variables and behavior attributes that could be included in the model. Future research will examine further connections between real

volunteer profiles, predicted volunteer performance, and event recommendations based on the real data we collect from the platform using the mobile application.

In addition, we also plan to structure a teenage volunteer cycle model to trace a volunteer's profile and behavior data from the day they sign up, to the day they go to the college or exit the system. When a volunteer leaves the system, we will collect data on which universities they get accepted and what majors they pursue in college. Based on this, we could further improve our prediction models by factoring in the correlation between volunteering events and volunteers' future colleges and majors. This would be extremely valuable to the high school teens and their parents as they will then have free access to custom, data-driven volunteer opportunity recommendations to help them plan for their volunteer services during their high school years. We are very excited to introduce this intelligent platform with advanced built-in predictive analytics to the volunteer teens, their parents, and the organization volunteer supervisors.

REFERENCES

[1] Van Willigen, Marieke. "Differential benefits of volunteering across the life course." The Journals of Gerontology Series B: Psychological Sciences and Social Sciences 55.5 (2000): S308-S318.

[2] Morrow-Howell, Nancy, Song-Iee Hong, and Fengyan Tang. "Who benefits from volunteering? Variations in perceived benefits." The Gerontologist 49.1 (2009): 91-102.

[3] Wilson, John. "Volunteering." Annual review of sociology 26.1 (2000): 215-240.

[4] Michie, Donald, David J. Spiegelhalter, and C. C. Taylor. "Machine learning." Neural and Statistical Classification 13.1994 (1994): 1-298.

[5] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of Machine Learning Research 12 (2011): 2825-2830.

[6] Bracha, Gilad. The Dart programming language. Addison-Wesley Professional, 2015.

[7] Wu, Wenhao. "React Native vs Flutter, Cross-platforms mobile application frameworks." (2018).

[8] Moroney, Laurence, Moroney, and Anglin. Definitive Guide to Firebase. Apress, 2017.

[9] Peng, Dunlu, Lidong Cao, and Wenjie Xu. "Using JSON for data exchanging in web service applications." Journal of Computational Information Systems 7.16 (2011): 5883-5890.

[10] Stukas, A. A., Worth, K. A., Clary, E. G., & Snyder, M. (2009). The Matching of Motivations to Affordances in the Volunteer Environment: An Index for Assessing the Impact of Multiple Matches on Volunteer Outcomes. Nonprofit and Voluntary Sector Quarterly, 38(1), 5–28. https://doi.org/10.1177/0899764008314810

[11] Finkelstein, Marcia A., Louis A. Penner, and Michael T. Brannick. "Motive, role identity, and prosocial personality as predictors of volunteer activity." Social Behavior and Personality: an international journal 33.4 (2005): 403-418.

[12] Davis, M. H., Hall, J. A., & Meyer, M. (2003). The First Year: Influences on the Satisfaction, Involvement, and Persistence of New Community Volunteers. Personality and Social Psychology Bulletin, 29(2), 248–260. https://doi.org/10.1177/0146167202239050

[13] Segal, Mark R. "Machine learning benchmarks and random forest regression." (2004).

[14] Bunea, Florentina, Alexandre B. Tsybakov, and Marten H. Wegkamp. "Aggregation for Gaussian regression." The Annals of Statistics 35.4 (2007): 1674-1697.

[15] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." Emerging artificial intelligence applications in computer engineering 160.1 (2007): 3-24.

EXPLAINABLE AI FOR INTERPRETABLE CREDIT SCORING

Lara Marie Demajo, Vince Vella and Alexiei Dingli

Department of Artificial Intelligence, University of Malta, Msida, Malta

ABSTRACT

With the ever-growing achievements in Artificial Intelligence (AI) and the recent boosted enthusiasm in Financial Technology (FinTech), applications such as credit scoring have gained substantial academic interest. Credit scoring helps financial experts make better decisions regarding whether or not to accept a loan application, such that loans with a high probability of default are not accepted. Apart from the noisy and highly imbalanced data challenges faced by such credit scoring models, recent regulations such as the `right to explanation' introduced by the General Data Protection Regulation (GDPR) and the Equal Credit Opportunity Act (ECOA) have added the need for model interpretability to ensure that algorithmic decisions are understandable and coherent. An interesting concept that has been recently introduced is eXplainable AI (XAI), which focuses on making black-box models more interpretable. In this work, we present a credit scoring model that is both accurate and interpretable. For classification, state-of-the-art performance on the Home Equity Line of Credit (HELOC) and Lending Club (LC) Datasets is achieved using the Extreme Gradient Boosting (XGBoost) model. The model is then further enhanced with a 360-degree explanation framework, which provides different explanations (i.e. global, local feature-based and local instance-based) that are required by different people in different situations. Evaluation through the use of functionally-grounded, application-grounded and human-grounded analysis show that the explanations provided are simple, consistent as well as satisfy the six predetermined hypotheses testing for correctness, effectiveness, easy understanding, detail sufficiency and trustworthiness.

KEYWORDS

Credit Scoring, Explainable AI, BRCG, XGBoost, GIRP, SHAP, Anchors, ProtoDash, HELOC, Lending Club

1. INTRODUCTION

1.1. Problem Definition

Credit scoring models are decision models that help lenders decide whether or not to accept a loan application based on the model's expectation of the applicant being capable or not of repaying the financial obligations [1]. Such models are beneficial since they reduce the time needed for the loan approval process, allow loan officers to concentrate on only a percentage of the applications, lead to cost savings, reduce human subjectivity and decrease default risk [2]. There has been a lot of research on this problem, with various Machine Learning (ML) and Artificial Intelligence (AI) techniques proposed. Such techniques might be exceptional in predictive power but are also known as black-box methods since they provide no explanations behind their decisions, making humans unable to interpret them [3]. Therefore, it is highly unlikely that any financial expert is ready to trust the predictions of a model without any sort of justification [4]. Model explainability has recently regained attention with the emerging area of

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 185-203, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101516

eXplainable AI (XAI), a concept which focuses on opening black-box models in order to improve the understanding of the logic behind the predictions [5, 6]. With regards to credit scoring, lenders will need to understand the model's predictions to ensure that decisions are made for the correct reasons. Furthermore, in adherence to existing regulations such as the GDPR 'right to explanation' and the ECOA, applicants have the right to know why they have been denied the loan. Therefore, credit scoring models must be both exceptional classifiers and interpretable, to be adopted by financial institutions [7, 8]. Formally, in this work we refer to model interpretability as the model's ability to explain or to present in understandable terms to a human [9]. The terms *explainability*, *interpretability*, *understandability* and *comprehensability* are used interchangeably in this work.

There are a number of challenges posed when working with XAI, including questions like "who are the explanations for (experts or users)?", "what is the best form of representation for the explanations?" and "how can we evaluate the results?" [10]. The literature still lacks precise answers to these questions since different people require different types of explanations. This lead to ambiguity in regulations and solutions needed [11]. The literature includes very few entrances of such interpretable credit scoring models, most of which provide only a single dimension of explainability. Therefore, in this work, we shall be addressing this gap by proposing a credit scoring model with state-of-the-art classification performance on two popular credit datasets (HELOC and Lending Club Datasets) and enhanced by a 360-degree explanation framework for model interpretability by bringing together different types of explanations.

1.2. Aims and Objectives

Our goal of an interpretable credit scoring model can be decomposed into the following two main objectives:

1. Model interpretability of the implemented credit scoring model by providing human-understandable explanations through different XAI techniques (Section 3.3)
2. A comprehensive approach for evaluation of model interpretability through both human subjective analysis and non-subjective scientific metrics (Section 4)

Details about how these objectives have been met are found in the rest of this paper, which is organized as follows. Chapter 2 includes a review of existing methods in the XAI domain. A detailed workflow of the system is discussed in Chapter 3. Chapter 4 includes all the experiments carried out to evaluate the interpretability performance, whilst any limitations, improvements and conclusions are finally discussed in Chapter 5.

2. RELATED WORK

Back in 1981, [12] state that the ability to explain decisions is the most highly desirable feature of a decision-assisting system. Recently, XAI has gained high popularity. It aims to improve the model understandability and increase humans' trust. There have been various efforts in making AI/ML models more explainable in many applications, with the most popular domain being image classification [13, 14, 15].

The authors in [16] state that dimensionality reduction like feature selection and Principle Component Analysis (PCA) can be an efficient approach to model interpretation since the outcome can be intuitively explained in terms of the extracted features. Štrumbelj and Kononenko in [17] propose a sensitivity analysis based model, which analyses how much each feature contributes to the model's predictions by finding the difference between the prediction

and expected prediction when the feature is ignored. Such explanations are given in the form of feature contributions.

Trinkle and Baldwin in [18] investigate whether Artificial Neural Networks (ANNs) can provide explanations for their decisions by interpreting the connection weights of the network. They conclude that performance was restricted due to the use of just one hidden layer and state that such interpretation techniques are not robust enough to handle more hidden layers. Baesens et al. in [19] contributed to making ANNs more interpretable by making use of NN rule extraction techniques to investigate whether meaningful rule sets can be generated. They implemented three NN rule extraction techniques being Neurorule, TREPAN and Nefclass, and the extracted rules were then presented in a decision tree structure since graphical representations are more interpretable by humans.

The authors in [20] propose Layer-wise Relevance Propagation (LRP), a post-hoc interpretability model for interpreting the individual predictions of a Deep Neural Network (DNN) rather than the model itself. It propagates back through the layers of the network until reaching the input and pinpoints the regions in the input image that contributed the most to the prediction. In [21], Yang et al. propose Global Interpretation via Recursive Partitioning (GIRP), a compact binary tree that interprets ML models globally by representing the most important decision rules implicitly contained in the model using a contribution matrix of input variables. Ribeiro et al. in [22] propose Local Interpretable Model-agnostic Explanations (LIME), a novel technique that explains any classifier's predictions by approximating them locally with a secondary interpretable model. While these are local explanations, the global view of the model can be presented by selecting a set of representative and non-redundant explanations. In [23], Ribeiro et al. introduce another novel model-agnostic system to explain the behaviour of complex models. They propose Anchors, which are intuitive high precision IF-THEN rules that highlight the part of the input, which is used by the classifier to make the prediction. It is shown that Anchors yield better coverage and require less effort to understand than LIME. The authors in [24] propose SHapley Additive eXplanations (SHAP), a unified framework for interpreting predictions. SHAP are Shapley values representing the feature importance measure for a particular prediction and are computed by combining insights from 6 local feature attribution methods. Results show that SHAP are consistent with human intuition.

In 2018, Fair Isaac Corporation (FICO) issued the Explainable Machine Learning Challenge in aim of generating new research in the domain of algorithmic explainability. They challenged participants to create ML models that are both accurate and explainable, in aim of solving the credit scoring problem using the HELOC financial dataset. The final models were qualitatively evaluated by data scientists at FICO. The winners, Dash et al. [25], propose Boolean Rules via Column Generation (BRCG), a novel global interpretable model for classification where Boolean rules in disjunctive normal form (DNF) or conjunctive normal form (CNF) are learned. Column generation is used to efficiently search through the number of candidate clauses without heuristic rule mining. BRCG dominates the accuracy-simplicity trade-off in half of the datasets tested, but even though it achieves good classification performance and explainability, methods like the RIPPER decision tree still obtain a better classification accuracy in many of the datasets, including HELOC. The authors state that limitations include performance variability as well as the reduced solution quality when implemented on large datasets. In [26], Gomez et al. propose another solution to this challenge. They make use of a Support Vector Machine (SVM) model with a linear kernel for classification, where features were first discretized into ten bins to get rid of outliers and ensure scalability and manageable time complexity. For explainability, they implement an updated version of Anchors [23], which finds key features by systematically perturbing the columns and holding others fixed. They combine instance-level explanations and global-level model interpretations to create an interactive application visualising the logic behind

the model's decisions, identifying the most contributing features in a decision. Using a greedy approach, they also suggest the minimal set of changes required to switch the model's output. Chen et al. are also mentioned for their great work in [27], who propose a globally interpretable model known as the 'two-layer additive risk model', achieving accuracy similar to other neural networks. The model is decomposable into subscales, where smaller models are created from subgroups of features and eventually combined to produce the final default probability. The decomposable nature of the model allows it to produce meaningful components that identify the list of factors that contribute most to the model's predictions, providing rule-based summary explanations for global interpretability and local case-based explanations for local interpretation.

All these XAI techniques present their ability in making ML/AI models more interpretable. The last three mentioned techniques are credit scoring models that provide good classification performance as well as local and/or global explainability. In fact, they present an interesting evolution of explainability within credit scoring, motivating the main goal of this project. The winner of FICO's Explainable Machine Learning Challenge in 2018, BRCG [25], is considered as a state-of-the-art of XAI in credit scoring and is therefore selected as the benchmark paper for this work.

3. METHODOLOGY

For this work, an interpretable credit scoring model is proposed. As depicted in Figure 1 the data is first preprocessed and then a classification function is adopted to classify the data instances. Subsequently, the classifier is extended by three XAI methods to provide a 360-degree explanation framework. Therefore, the pipeline of the system comprises of three main sequential phases:

1. **Data handling & preprocessing**: transforming and preparing data for classification
2. **Classification**: classifying data into predetermined labels
3. **eXplainable AI**: appending interpretable explanations to the classification predictions

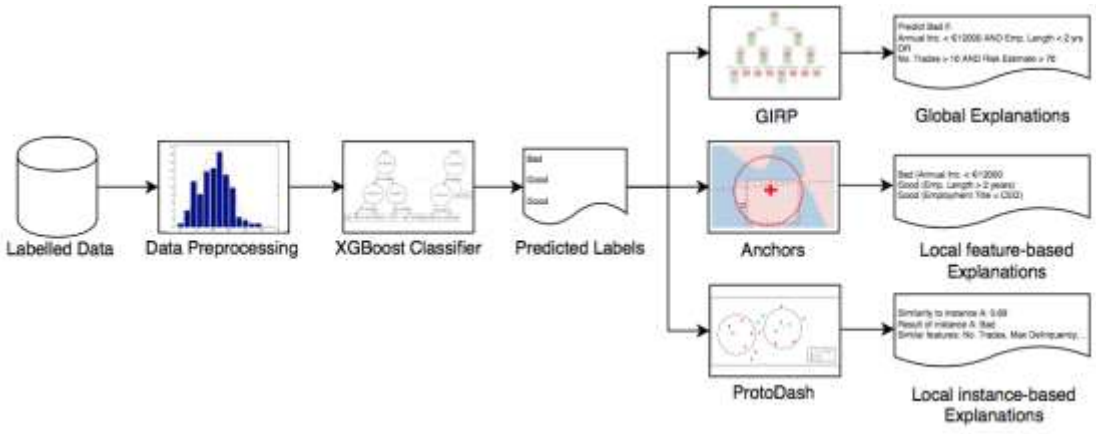


Figure 1. Pipeline of proposed Interpretable Credit Scoring model.

3.1. Data Handling and Preprocessing

Considering that our classifier requires supervised learning, the first concern is the preparation and handling of the data. Two datasets are used in this project:

- The Home Equity Line of Credit (HELOC) Dataset which contains around 10,000 instances with 24 different features (21 numerical and 3 categorical).
- The Lending Club (LC) Dataset which includes around 2.3 million loan applications with 145 features of different types.

The HELOC Dataset is used by the benchmark paper [25] and therefore used for fair comparison, whilst the LC Dataset is quite popular in the credit scoring literature [28, 28, 30, 31] and is used to further evaluate the implemented model. Figure 2 depicts the different stages undertaken during the preprocessing phase until the data is ready to be used by the model.

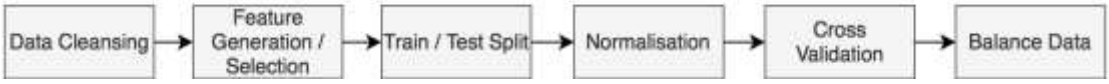


Figure 2. Pipeline of the different stages during the data preprocessing phase.

Firstly, the data is cleaned by handling any special values in the HELOC Dataset, converting the target variables into binary labels, removing outliers, imputing empty values in the numerical features with the mean and imputing empty values in categorical features with unused category values, converting categorical features using one-hot encoding, and eliminating noisy data. Next, some additional variables are computed for the Lending Club Dataset such as *Loan Amount to Annual Income* since ratios are better for deep learning classifiers, and feature selection is performed on the LC Dataset through analysis of the correlation matrix and change in classification performance. Furthermore, both datasets are split with 75:25 ratio using stratification and the training set is further split using stratified 10-fold cross validation. The data for each fold is then normalised using both the min-max normalisation and standard scaling techniques and best method is identified through evaluation. Finally, the training data of the LC Dataset is balanced using four different resampling techniques and best method is also identified through evaluation.

3.2. Classification

Initially, a DNN model was chosen as the classification function, however after abundant experiments and evaluation it was noted that the performance was not as satisfactory as expected. As observed by the authors in [32] and [33], DNNs often suffer from reduced performance on certain credit scoring datasets. After replicating the work in [29], who use an ANN and present a high accuracy, results show that their model was actually overfitting and classifying all instances into the same class (i.e. giving a high accuracy due to data being highly unbalanced). From the literature [32, 34, 35] it was observed that other methods like Random Forest (RF) and Extreme Gradient Boosting (XGBoost) perform strongly in credit scoring. Hence, for this work, the most commonly used classification techniques are implemented and compared in order to find the best performing algorithm in credit scoring on both the HELOC and LC Datasets. The ML models implemented include Logistic Regression, Decision Tree, Random Forest, ANN and XGBoost. A "light" version of the BRCG algorithm by Dash et al. [25], which is a more efficient variant of the method, is also implemented in this work to enable better comparison with more metrics between the classification techniques.

Results show that the XGBoost model does not require any data normalisation and that the oversampling technique gave the best performance on the Lending Club Dataset. Using the optimal parameter values found by the grid search on the HELOC Dataset portrayed classification performance improvement on both datasets. It is concluded that XGBoost yields less Type-I and Type-II errors than LBRCG on the HELOC Dataset whereas LBRCG yields less

Type-I errors (5,112 loans) but much more Type-II errors (27,767 loans) than XGBoost on the LC Dataset. This signifies that the XGBoost model is better at maintaining a good balance between Type-I and Type-II errors and improves performance over the LBRCG model by an F1-Score of more than 3% on the HELOC Dataset and an F1-Score of over 7% on the LC Dataset.

3.3. eXplainable AI

Given their opaque nature, deep learning techniques lack interpretability and are unable to explain their decisions. Therefore, the main goal of this project is to enhance the implemented credit scoring model by augmenting it with a 360-degree explanation framework such that it can provide different types of explanations for its predictions. Many new XAI methods have been recently published in the literature, some of which have not yet been explicitly applied to the credit scoring domain.

It is important to note that different people in diverse situations require different explanations [8]. There are three different personas that require explainability in credit scoring, being (i) loan officers that are said to prefer local sample-based explanations, (ii) rejected loan applicants that are said to prefer local feature-based explanations, and (iii) regulators or data scientists that are said to prefer global model explanations. Hence, a single XAI method does not suffice to provide all the explanations required [11]. In this project, we aim to propose an explainable credit scoring model that provides 360-degree interpretability by producing explanations for each of the three different personas mentioned. Table 1 represents the three different state-of-the-art XAI methods used to yield the different explainability dimensions required. The following subsections describe the implementation details of the XAI method implemented for each of the three different explanation types.

Table 1. The XAI methods used to generate each explanation type and the format of the explanation provided by each method.

Explanation Type	XAI Method	Explanation Form	Reference
Global	SHAP+GIRP	Decision Tree / IF-THEN rules	[24], [21]
Local feature-based	Anchors	DNF rule	[23]
Local instance-based	ProtoDash	Prototypical instances	[36]

3.3.1. Global Explanations

Global explanations are explanations that provide a global understanding of how the classification model works overall. They interpret the reasoning behind the general logic used by the model when making its predictions. Such explanations are usually preferred by regulators and managers since they are mostly concerned with the global understanding of the credit scoring model rather than the individual explanations of each instance. This is because regulators and data scientists are responsible of ensuring that the model is being correct, fair and compliant in its predictions.

As discussed previously, the benchmark BRCG method [25] is a directly interpretable supervised learning method that provides Boolean rules to globally explain its logic. Therefore, in this project, we aim to implement an XAI technique that provides similar or better global explanations to BRCG. For this part of the XAI objective, a SHAP+GIRP method is implemented. GIRP [21] is a post-hoc method that is capable of interpreting any black-box model by extracting the most important rules used by the model in its predictions. It is a very recent model that depicts state-of-the-art capabilities in model-agnostic interpretability. It is important to note that, to the best of our knowledge, GIRP has not yet been explicitly applied to the credit risk

problem and no formal academic results have been presented for this domain. The explanation provided by GIRP for the global understandability of the model is given in the form of a decision tree, however the IF-THEN rules that make up the tree are also extracted and provided for variety.

GIRP makes use of a contribution matrix to generate a decision tree consisting of the most discriminative rules contained in the trained model, which is then pruned for better generalisation to form the Interpretation Tree. The contribution matrix contains the contributions of each input variable to the prediction of each instance. To generate this contribution matrix, Lundberg and Lee's SHAP [24] method is used and is implemented using the SHAP Python library. Using the SHAP package, an explainer is created over the XGBoost model to generate SHAP values for each feature for each prediction, constructing the contribution matrix for GIRP. For the implementation of GIRP, source code was adapted from a Github repository (https://github.com/west-gates/GIRP [Accessed: 10/08/2020]) containing an implementation of GIRP on text classification. The code was updated such that the methods handle tabular data rather than words from text extracts. The rules extracted from GIRP include a number of conditions in their IF statement and a default rate in their THEN section. The larger the default rate, the higher the risk.

3.3.2. Local Feature-based Explanations

Local explanations are explanations that provide a local understanding on how and why a specific prediction was made. Such explanations are said to be preferred by loan customers since they are mostly interested in why they have been denied the loan and what is the reasoning behind the model's prediction for their particular loan application. This type of explanation can be provided in the form of feature relevance scores or rules. It is important to note that local explanations are not provided by the benchmark BRCG model. However, in this work, we aim to go above and beyond global explainability. As discussed, Lu et al. in [8] state that different people in different scenarios require different explanations and therefore, we aim to provide further model interpretability through local explanations.

In this work, the local feature-based explanations are generated using the post-hoc Anchors method from [23]. The explanations are given in the form of rules containing conditions on the most important features for the model prediction. The original Python implementation of Anchors from [23] is used. Anchors generates an anchor rule that is iteratively increased in size until a predetermined probability threshold is reached. The outputted rule is the shortest rule with the largest coverage and closest estimated precision to the threshold, that explains the model prediction. The anchor rule contains the features and feature values that contributed to the model prediction. An anchor rule is a sufficient condition, which means that other data points that satisfy it should have, with an x% probability, the same prediction as the original data point. The probability x is set to 90% in this work. It was noted that for the HELOC Dataset, the resulting anchor rule only holds for the data point it was built for (from the entire test data), even when reducing the probability threshold x to 50%. Furthermore, the outputted anchor rules contain an average of 35 conditions, which might make the rule hard to read and consequently uninterpretable. Therefore, we implemented a further extension that iterates over the partial anchors in the main anchor to find the shortest partial anchor that still holds for the data point. This obtains rules that contain an average of 4 conditions. Finally, the derived rule of each data point is used as the local feature-based explanation for its prediction.

3.3.3. Local Instance-based Explanations

Similar to local feature-based explanations, these explanations provide a local understanding on individual predictions rather than the model as a whole. Such explanations are said to be preferred by loan officers since they are interested in validating whether the prediction given by the model for a loan application is justified. Therefore, it is said that a loan officer would gain more confidence in the model's prediction by looking at other similar loan applications with the same outcome, and hence understanding why a loan application has been denied compared to other loan applications that were previously accepted and then ended up defaulting [11]. This type of explanation is usually provided in the form of prototypes (i.e. similar data points from the dataset). Again, it is important to note that this type of local explanations is not provided by the benchmark BRCG model but in this work, we aim to provide further model interpretability by providing explanations for the three personas that require explainability in credit scoring.

In this work, local instance-based explanations are generated using the post-hoc ProtoDash method by Gurumoorthy et al. [36]. The explanations are given in the form of two prototypical data points that have similar features. The implementation of ProtoDash by AIX360 [11] is used. ProtoDash employs the fast gradient-based algorithm to find prototypes of the data point in question as well as the non-negative importance weight of each prototype. The algorithm aims to minimize the maximum mean discrepancy metric and terminates when the number of prototypes m is reached. For this work, $m=6$ is used and the two prototypes with the largest weight from the outputted six are selected as the final exemplar-based explanation.

4. EVALUATION & RESULTS

The aim of this study is to enhance the credit scoring system with interpretability such that its predictions are also justified by reasons. However, these reasons need to make sense and need to be simple enough for easy understanding by both domain experts and layman. Therefore, the analysis of the explanations is important since it moves us away from vague claims about interpretability and towards evaluating methods by a common set of terms [4]. There are three evaluation approaches for XAI being (i) functionally-grounded, where some formal definition of interpretability is used as a proxy for explanation quality analysis, (ii) application-grounded, where human experts evaluate the quality of the explanations in the context of the end-task, and (iii) human-grounded, where lay human-subject experiments are carried out to test the explanation quality regardless of its correctness [4, 27].

The majority of the papers that do perform evaluation adopt one of the last two evaluation approaches, making use of human subjects as their evaluators. However, as noted by [46], evaluating interpretable systems using only human evaluations can imply a strong bias towards simpler descriptions that might not completely represent the underlying reasoning of the method. In this project, we address this gap by adopting a comprehensive evaluation approach, where apart from the usual human subjective analysis, the interpretability efficiency is also analysed through functionally-grounded techniques so as to provide results in non-subjective and scientific metrics. Since this type of evaluation approach is rarely used throughout the XAI literature, it is difficult to compare to existing XAI techniques and interpretable systems in terms of such metrics [10]. Table 2 lists the hypotheses (A-F) that are tested during each of the three analysis approaches.

Table 2. The hypotheses tested by each XAI evaluation approach.

	Hypothesis Description	Functionally-grounded	Application-grounded	Human-grounded
A	The explanations provided are complete and correct	✓	✓	
B	The explanations provided are effective and useful		✓	✓
C	The explanations provided are easily understood		✓	✓
D	The explanations provided boost trust in ML models		✓	✓
E	The explanations provided are sufficiently detailed		✓	✓
F	Different explanations are required by different people		✓	

The below subsections include the evaluation results of each of the three types of analysis approaches on the explanations provided.

4.1. Functionally-Grounded Analysis

There are two types of functionally-grounded measures being complexity-based that analyses the complexity of the rule base, and semantics-based that analyses whether the semantics of the rules associate with the membership function. Motivated by Martens et al. [38] and the taxonomy proposed in [39], the below functionally-grounded measures are used in this study:

- **Number of unique rules**
- **Average number of rule conditions**
- **Consistency of rules** by checking if any contradicting rules exist and if rules are similar for instances of the same class
- **Completeness/Fidelity** by computing the percentage of instances where the model and the rules agree on the label

One might suggest that a lower number of rules are preferred since it makes it easier to follow through and keep track of things. However, more information in an explanation can also help users build a better mental model [40]. The consistency of the rules determines whether the method is trustworthy and reliable in its logic, whereas the completeness of the rules identifies how well they explain the decision function of the model, testing Hypothesis A. In the below subsections, analysis of each explanation type is carried out using the defined measures.

4.1.1. Global Explanations

As discussed in Section 3.3.1, the global explanation is provided in the form of a decision tree, but for better comparison with the BRCG global explanation, the IF-THEN rules were also extracted from the tree. For BRCG, satisfying the DNF rule signifies that the loan application is likely to default, whilst for the implemented SHAP+GIRP method, satisfying either one of the IF-THEN rules results in a specific default rate.

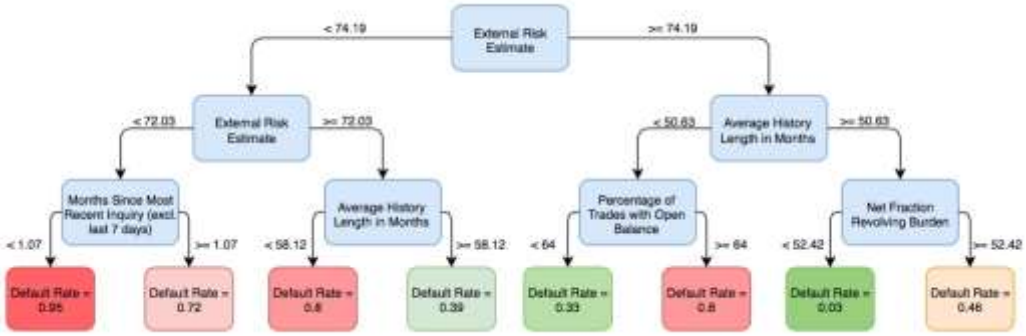


Figure 3. Global explanation for credit scoring model via SHAP + GIRP on HELOC Dataset.

Figure 3 illustrates the global explanation for the implemented XGBoost model on the HELOC Dataset, given in the form of a decision tree. As shown, when the *External Risk Estimate* feature has a smaller value, the loan application has a higher default risk. Moreover, a smaller *Percent Trades W Balance* leads to a lower default rate. These observations are inline with the monotonicity constraints of the HELOC Dataset. Table 3 depicts the evaluation results for both the BRCG and SHAP+GIRP global explanations on both datasets. For BRCG, each clause in the DNF rule (i.e. the rule of ANDs) is considered as a separate rule since either one of these clauses must be satisfied to satisfy the whole DNF rule. As shown, the BRCG model outputs fewer rules than the implemented SHAP+GIRP model. Impressively, BRCG manages to explain all the model predictions with just two rules of two conditions each for the HELOC Dataset and just one rule of two conditions for the LC Dataset. This could be possible due to BRCG's greedy approach in generating the rules. For the implemented SHAP+GIRP approach, the number and complexity of the rules could be easily adjusted by the maximum tree depth. Using a maximum tree depth of 2 resulted in 4 rules of 2 conditions each. As later shown during the application-grounded analysis (Section 4.2) having more rules and features could lead to more justifiable explanations that lead to more trust. In fact, most of the domain experts interviewed in this work suggested that having more features would have improved the explanations.

With regards to consistency, it is demonstrated in Table 3 that the global explanation provided by both models contains no conflicting and/or contradicting rules. This shows that the global explanation provided for XGBoost by the implemented SHAP+GIRP method is reliable and logical. When considering the completeness metric (i.e. the fidelity of the explanation to the predictions), the global explanation provided by the BRCG method is almost 100% complete for both datasets. BRCG achieves this high completeness rate because it is intrinsically explainable. On the other hand, the global explanation of the implemented XGBoost model achieves a completeness rate of around 90% on the HELOC Dataset and around 97% for the Lending Club Dataset. The loss in fidelity is most probably due to the extrinsic nature of the XAI method since the explanation was extracted from the XGBoost model through two levels of external processes, firstly SHAP to extract the feature contributions and then GIRP to form the global interpretation tree using these contributions. Therefore, it seems reasonable for such an explanation to have a lower completeness rate than an explanation extracted from a natively interpretable method. It is interesting to note that when increasing the maximum tree depth of the GIRP method to 100 on the HELOC Dataset, as done for other applications in [21], the completeness rate increased slightly by around 1%. However, the interpretability of the model was then greatly diminished since the decision tree became very hard to follow and the number of rules and conditions increased immensely. Therefore, completeness must be slightly sacrificed for considerably better interpretability. All in all, the completeness rate achieved by the SHAP+GIRP method is still quite high for both datasets, confirming Hypothesis A.

Table 3. Functionally-grounded metrics on global explanations.

	Metric	BRCG	SHAP + GIRP
HELOC	Number of unique rules	2	8
	Average number of rule conditions	2	3
	Consistency of rules	Yes	Yes
	Completeness rate	99.96%	89.95%
LC	Number of unique rules	1	8
	Average number of rule conditions	2	3
	Consistency of rules	Yes	Yes
	Completeness rate	99.66%	96.88%

4.1.2. Local Feature-Based Explanations

As discussed in Section 3.3.2, the local feature-based explanations provided by Anchors are given in the form of DNF rules that contain conditions on the most important features. Since BRCG does not provide local feature-based explanations, these explanations cannot be compared to the benchmark.

	Example 1	Example 2
installment	689.89	762.08
int_rate	14.65	8.9
loan_amnt	20000	24000
credit_age	5021	6180
delinq_2yrs	0	0
inq_last_6mths	2	1
revol_util	77.4	21.9
open_acc	14	12
annual_install_to_inc	0.15	0.13
revol_bal_to_mnth_inc	3.88	1.63
term	36 months	36 months
sub_grade	F5	A2
home_ownership	MORTGAGE	OWN

Explanation for Example 1:

Predicted as **Bad** because:

home_ownership = MORTGAGE
AND sub_grade = F5
AND annual_install_to_inc > 0.12
AND revol_bal_to_mnth_inc > 3.20

Explanation for Example 2:

Predicted as **Good** because:

home_ownership = OWN
AND annual_install_to_inc > 0.04
AND sub_grade = A2

Figure 4. Two examples of the local feature-based explanations via Anchors on LC Dataset

Figure 4 illustrates two examples of the provided local feature-based explanations from the Lending Club Dataset. As opposed to the global explanation, a local explanation is data-point specific. Therefore, the number of rules is equal to the size of the test set. As shown in Table 4 the average number of rule conditions is 4 for both datasets. This is quite a reasonable number of conditions since it provides sufficient details without complicating the rule too much. Moreover, the local feature-based explanations provided are consistent and 100% complete for each dataset since each rule is faithful to the data point and prediction it is explaining, confirming Hypothesis A.

Table 4. Functionally-grounded metrics on local feature-based explanations.

	Metric	Anchors
H	Average number of rule conditions	4
	Consistency of rules	Yes
	Completeness rate	100%
LC	Average number of rule conditions	4
	Consistency of rules	Yes
	Completeness rate	100%

4.1.3. Local Instance-Based Explanations

As discussed in Section 3.3.3, the local instance-based explanations provided by ProtoDash are given in the form of prototypical instances from the data. This type of explanation assumes that loan applications that exhibit similar behaviours might end up in the same situation. It is important to note that BRCG does not provide local instance-based explanations.

Figure 5 illustrates an example local instance-based explanation provided for the LC Dataset. The *Loan application* column lists the feature values for the application at hand, whereas the two other columns list the feature values along with the target class and prototype weight of the two prototypical samples extracted as local instance-based explanations. Identical feature values are highlighted in green.

	Loan application	Prototype A	Prototype B
Target Class		Good	Good
Prototype Weight		52%	48%
installment	445.1	186.61	587.71
int_rate	7.07	7.49	7.12
loan_amnt	14400	6000	19000
credit_age	4687	25933	6178
delinq_2yrs	0	0	0
inq_last_6mths	0	0	0
revol_util	33	88.2	44.6
open_acc	12	13	14
annual_install_to_inc	0.08	0.03	0.13
revol_bal_to_mnth_inc	3.66	14.88	5.16
term	36 months	36 months	36 months
sub_grade	C1	D1	A5
home_ownership	MORTGAGE	MORTGAGE	MORTGAGE

Figure 5. Sample local instance-based explanation via ProtoDash on LC Dataset.

Similar to the local feature-based explanations, such explanations are data-point specific, such that the prediction of each instance in the test set is explained through the use of two prototypical instances from the training set. It is noted that functionally-grounded analysis on local instance-based explanations in terms of number of rules, rule conditions and consistency is meaningless since these explanations are not given in the form of rules. As proven in this work and as stated in [37], local explanations are more faithful than global explanations.

4.2. Application-Grounded Analysis

As stated in [37], there is a lack of formalism on how this type of XAI evaluation, or any type of XAI evaluation for that matter, must be performed. Application-grounded analysis requires human domain experts to quantify the correctness and quality of the explanations provided by performing real tasks. In credit scoring, loan officers are considered experts the area since they have comprehensive knowledge of loan requirements and banking regulations.

In this project, interviews were carried out with seven different loan and/or risk officers employed at different banking and financial institutions around Malta (Bank of Valletta, HSBC, APS, BNF, Lombard). Each interview was around an hour long. The authors in [41] state that 5-10 respondents are needed to get reasonably stable psychometric estimates for evaluating the communality of answers. Application-grounded evaluation helps to identify the actual impact of the proposed model in a real-world application, since it directly tests the objective that the system was built for, giving strong indication on the actual success. To keep the duration of the

interviews as short as possible, the evaluation was performed for just the HELOC Dataset. During the interviews, the domain experts were presented with a total of 3 tasks; one for each explanation type (global, local feature-based, and local instance-based). It is important to note that throughout the interview, it was observed that the interviewees' limited knowledge on the dataset contributed to an undesirable decrease in understandability. Therefore, it is worthy to mention that if the questions could make use of a dataset that the experts are used to and confident with, their understandability would have been certainly improved.

4.2.1. Global Explanations

Table 5 depicts the evaluation results achieved for each question from this section. As a general note, the *Result* column represents the literal result of the question, the *Percentage* column represents the result in the form of a global percentage, whilst the *Hypothesis* column lists the hypotheses confirmed by each question.

Table 5. Evaluation results acquired from interviews on the global explanation.

Question/Task Description	Result	Percentage	Hypothesis
Forward prediction task	7/7 experts	100%	A & C
Accept/reject loan task	7/7 experts	100%	A & C
Preference of tree or rules?	6/7 experts	86%	-
How well explanation clarifies prediction?	31/35	89%	B
Is explanation sufficiently detailed?	6/7 experts	86%	E
How much explanation increased trust in ML models?	26/35	74%	D

Firstly, the domain experts were presented with a forward prediction task where they were requested to interpret the model's prediction given the global explanation. 100% of the domain experts managed to correctly complete this task and reach the same conclusion as the model, confirming the understandability and correctness of the explanation. For the second task, the experts were requested to use the model's prediction and explanation to indicate whether they would accept or reject the loan application. Despite their limited dataset knowledge, all seven experts agreed with the model's prediction. Both these tasks confirm Hypotheses A and C.

Interestingly, six out of seven experts prefer the decision tree representation of the explanation. This implies that most humans find graphical representations more interpretable and easier to follow. Confirming this observation, the authors in [37] state that visualisations are the most human-centred technique for interpretability. This suggests that the global explanation provided as a decision tree might be preferred to BRCG's global explanation provided as a DNF rule, even though the BRCG explanation is simpler when comparing the rule-form of both methods.

Confirming Hypothesis B, the domain experts indicate that they believe that the explanation adequately clarifies the prediction, marking the Likert scale with an average score of 4.5. Moreover, six out of seven experts indicate that they are satisfied with the level of detail provided by the global explanation, confirming Hypothesis E. Finally, despite being quite a controversial question since people outside the AI community might find it hard to trust ML models, the domain experts indicate that the global explanation increased their trust by an average of 74%, confirming Hypothesis D.

4.2.2. Local Feature-Based Explanations

Table 6 depicts the results achieved for each question from this section. All the seven domain experts agreed with the model's prediction when asked to accept/reject a loan application, determining that the explanation provided is correct and easy to understand, thus confirming Hypothesis A and C. Regarding the explanation's ability to clarify the prediction, a total score of 29/35 (i.e. 83%) and an average score of 4.1 was achieved, which is slightly lower than the score achieved for this same question on the global explanation, but still confirms Hypothesis B. Six out of seven experts indicate that the local feature-based explanation provided is sufficiently detailed, confirming Hypothesis E. Finally, it is shown that on average, the local feature-based explanation increased the trust of the domain experts in ML models by 77%, which is slightly better than the score acquired for the global explanations. This confirms Hypothesis D. From the general impression given by the experts and as later described in Section 4.2.4, it was observed that most of the experts preferred the local feature-based explanation over the global explanation. An interesting point that was highlighted by one of the experts is that complete trust in the model is not necessarily required since the model should be there to help rather than make the decisions itself. Therefore, the ability to understand the reasoning behind the model's decision should provide enough trust to use the model.

Table 6. Evaluation results acquired from interviews on the local feature-based explanation.

Question/Task Description	Result	Percentage	Hypothesis
Accept/reject loan task	7/7 experts	100%	A & C
How well explanation clarifies prediction?	29/35	83%	B
Is explanation sufficiently detailed?	6/7 experts	86%	E
How much explanation increased trust in ML models?	27/35	77%	D

4.2.3. Local Instance-Based Explanations

Table 7 depicts the results achieved for the local instance-based explanations. Similar to the other two types of explanations, all seven domain experts agreed with the model's prediction when asked to accept/reject a loan application, confirming Hypotheses A and C. With regards to the explanation's ability to clarify the prediction, an average score of 3.3 is achieved, suggesting that this type of explanation was not as favoured and possibly implies that local instance-based explanations are not as effective and useful as the other two types. Most of the experts also specified that having three or four prototypes as part of the explanation, instead of two, would have been more useful. This could be easily resolved by adjusting the number of prototypes outputted from ProtoDash. Finally, it is shown that the local instance-based explanation increased the trust of the domain experts in ML models by an average of 74%. This is the same score achieved for the global explanation, which is slightly lower than that achieved for the local feature-based explanation. Having said this, the results confirm Hypothesis D. In general, whilst a few of the experts liked the idea behind the prototypical explanations, two experts expressed their disagreement with comparing loan applications to each other. They state that every case is different even if they show similar traits. Moreover, unpredictable changes can cause loan applications with very good traits to default and hence comparing with such application can cause unreliable results.

Table 7. Evaluation results acquired from interviews on the local instance-based explanation.

Question/Task Description	Result	Percentage	Hypothesis
Accept/reject loan task	7/7 experts	100%	A & C
How well explanation clarifies prediction?	23/35	66%	B
Is explanation sufficiently detailed?	6/7 experts	86%	E
Two prototypes are enough?	2/7 experts	30%	-
How much explanation increased trust in ML models?	26/35	74%	D

4.2.4. Final Thoughts

Finally, each domain expert was asked to choose their preferred type(s) of explanation(s). Some of the domain experts specified that all the explanation types are meaningful in different setups and recommended that all the three types should be available since a user might require a second form of explanation to confirm what is understood from the first explanation. The most preferred type of explanation is the local feature-based rule explanation, whilst the local instance-based explanation is the least preferred. However, the fact that each type of explanation was selected as a preferred explanation by one or more experts confirms that not everybody prefers the same thing. Therefore, this confirms Hypothesis F and thus affirms the efforts in this work with regards to providing three types of explanations for better subjective interpretability.

4.3. Human-Grounded Analysis

Human-grounded analysis includes conducting simpler human-subject experiments that still maintain the importance of the target application. Such evaluation can be carried out by lay humans, allowing for a bigger subject pool. This analysis focuses mainly on evaluating the quality and interpretability of the explanations rather than the correctness to ensure that the provided explanations are interpretable not just by domain experts but even lay humans.

In this project, human-grounded analysis is performed through questionnaires, which were sent over to a number of subjects of different age, gender, occupation, education level and marital status. Authors in [42] and [43] suggest that 10 to 30 participants are an adequate sample size. For the analysis, a Google Form was posted on a number of Facebook groups with members having different backgrounds, and 100 participants have completed the questionnaire. To keep the questionnaire as simple as possible, the evaluation was performed for just the HELOC Dataset. The participants are given a case scenario, where they are asked to imagine themselves as a loan applicant that has been denied a loan and has been provided with the model explanation for their denial. It is important to note that some of the features for the loan application were removed, whilst the rest were given in easy terms to keep the task simple and easy to complete. Since it is said that loan applicants prefer explanations that are related to their own case, only a feature-based explanation is used for this analysis as the participants are representing the loan applicants (which are lay human) in real life. The participants are asked to fill out a total of 5 questions using Likert scales, yes/no selection and textual answers.

Through the human-grounded analysis, some interesting observations were made. Firstly, 87% of the participants (i.e. participants that marked Question 1 with a score of 3, 4, or 5) were satisfied with the local feature-based explanation provided and, on average, the participants were 74% satisfied with the explanations. Moreover, 89% of the participants (i.e. participants that marked Question 2 with a score of 3, 4, or 5) found the explanation to be profitably understandable with the explanation achieving an average understandability of 78% amongst all the participants. The

explanation provided also helped 17% of the participants to be have 100% more trust in ML models, whilst 38% of the participants were convinced to have more trust in ML models with 80% confidence. On average, with the help of the local feature-based explanation, the participants were 70% convinced to have more trust in AI models. This question is rather controversial since lay humans may have less knowledge on ML and AI and might therefore find it harder to trust such models. It is assumed that trustworthiness in such AI and ML models will increase with time as their use continues to expand and the models continue to improve in terms of interpretability [44, 45]. Furthermore, 72% of the participants found the explanation to be sufficiently detailed, whilst others suggested that more features, an overall risk rating or visualisation charts should be added. All in all, these results further confirm that the local feature-based explanations satisfy Hypotheses B-E.

5. CONCLUSIONS

In this work, a credit scoring model with state-of-the-art classification performance on the HELOC and Lending Club Datasets and comparable explainability to the benchmark BRCG model by Dash et al. [25] is proposed. The implemented credit scoring model incorporates the XGBoost algorithm, which demonstrates its capability of keeping a good balance between Type-I and Type-II errors. Furthermore, in aim of boosting the explainability of the black-box XGBoost model, a 360-degree explanation framework is developed by augmenting three separate post-hoc XAI techniques to provide three different types of explanations. A SHAP+GIRP method provides global explanations, Anchors provides local feature-based explanations and ProtoDash provides local instance-based explanations. Changing the classification function requires no changes in the interpretability component of the proposed model since the implemented XAI methodologies are model-agnostic and can be extracted from the current system pipeline and appended to a new classifier. It is shown, through the functionally-grounded analysis, that all the types of explanations provided are simple, consistent and complete. With regards to global explanations, it is shown that the provided explanation is comparatively as simple as the explanation produced by the benchmark BRCG model (in terms of number of rules and rule conditions). The application-grounded analysis deduced that six out of seven domain experts preferred the visual representation of the provided global explanation, which further suggests that the provided global explanation (in the form of a decision tree) might be preferred over the DNF rule of the BRCG model. The two other types of explanations are implemented over and above the global explanation and enable the implemented credit scoring model to be explained in alternative forms. In fact, the results of the application-grounded analysis present that the most preferred type of explanations are the local feature-based explanations, which are not provided by the benchmark BRCG model. It was also concluded that most of the financial experts interviewed found the explanations provided to be useful and have potential to be implemented in the system adopted by their bank. Through the rest of the evaluation, it was shown that the three types of explanations provided are complete and correct, effective and useful, easily understood, sufficiently detailed and trustworthy.

Future work can be focused on implementing a more user-specific solution with capabilities that allow the user to manage parameters such as the decision tree depth of the global explanations, the number of features in the local feature-based explanations, and the number of instances in the local instance-based explanations. Moreover, a possible improvement to the explanations, which was a popular suggestion amongst the domain experts interviewed, is to combine the global and local explanations in order to generate a decision tree that provides both global and local reasoning by highlighting the path and leaf node that is satisfied by the loan application in question. Future works suggested by lay humans through the human-grounded analysis include adding an overall risk rating or a percentage of eligibility to the explanation.

ACKNOWLEDGEMENTS

The authors would like to gratefully thank Andrew Borg and BRSAnalytics for their financial support. Moreover, many thanks go to Susan Vella, Maria Grech, Rosalie Galea, Anthony Bezzina, Ian Cunningham, Jeremy Aguis and Mariella Vella for their help and time with the interviews, as well as the participants that took the time to fill out the questionnaire.

REFERENCES

[1] Bekhet, H. A., & Eletter, S. F. K. (2014). Credit risk assessment model for Jordanian commercial banks: Neural scoring approach. *Review of Development Finance*, *4*(1), 20-28.

[2] Mester, L. J. (1997). What's the point of credit scoring?. *Business review*, *3*(Sep/Oct), 3-16.

[3] Bonacina, M. P. (2017, November). Automated Reasoning for Explainable Artificial Intelligence. In *ARCADE@ CADE* (pp. 24-28).

[4] Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

[5] Fernandez, A., Herrera, F., Cordon, O., del Jesus, M. J., & Marcelloni, F. (2019). Evolutionary fuzzy systems for explainable artificial intelligence: why, when, what for, and where to?. *IEEE Computational Intelligence Magazine*, *14*(1), 69-81.

[6] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., & Zhu, J. (2019, October). Explainable AI: A brief survey on history, research areas, approaches and challenges. In *CCF International Conference on Natural Language Processing and Chinese Computing* (pp. 563-574). Springer, Cham.

[7] Gilpin, L. H., Testart, C., Fruchter, N., & Adebayo, J. (2019). Explaining explanations to society. *arXiv preprint arXiv:1901.06560*.

[8] Lu, J., Lee, D., Kim, T. W., & Danks, D. (2020, February). Good Explanation for Algorithmic Transparency. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (pp. 93-93).

[9] Došilović, F. K., Brčić, M., & Hlupić, N. (2018, May). Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 0210-0215). IEEE.

[10] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Chatila, R. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82-115.

[11] Arya, V., Bellamy, R. K., Chen, P. Y., Dhurandhar, A., Hind, M., Hoffman, S. C., ... & Mourad, S. (2019). One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*.

[12] Teach, R. L., & Shortliffe, E. H. (1981). An analysis of physician attitudes regarding computer-based clinical consultation systems. *Computers and Biomedical Research*, *14*(6), 542-558.

[13] Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., & Darrell, T. (2016, October). Generating visual explanations. In *European Conference on Computer Vision*(pp. 3-19). Springer, Cham.

[14] Core, M. G., Lane, H. C., Van Lent, M., Gomboc, D., Solomon, S., & Rosenberg, M. (2006, July). Building explainable artificial intelligence systems. In *AAAI* (pp. 1766-1773).

[15] Van Lent, M., Fisher, W., & Mancuso, M. (2004, July). An explainable artificial intelligence system for small-unit tactical behavior. In *Proceedings of the national conference on artificial intelligence* (pp. 900-907). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[16] Vellido, A., Martín-Guerrero, J. D., & Lisboa, P. J. (2012, April). Making machine learning models interpretable. In *ESANN*(Vol. 12, pp. 163-172).

[17] Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, *41*(3), 647-665.

[18] Trinkle, B. S., & Baldwin, A. A. (2007). Interpretable credit model development via artificial neural networks. *Intelligent Systems in Accounting, Finance & Management: International Journal*, *15*(3-4), 123-147.

[19] Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management science*, *49*(3), 312-329.

[20] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, *10*(7), e0130140.

[21] Yang, C., Rangarajan, A., & Ranka, S. (2018, June). Global model interpretation via recursive partitioning. In *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (pp. 1563-1570). IEEE.

[22] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144).

[23] Ribeiro, M. T., Singh, S., & Guestrin, C. (2018, April). Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[24] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765-4774).

[25] Dash, S., Gunluk, O., & Wei, D. (2018). Boolean decision rules via column generation. In *Advances in Neural Information Processing Systems* (pp. 4655-4665).

[26] Gomez, O., Holter, S., Yuan, J., & Bertini, E. (2020, March). ViCE: visual counterfactual explanations for machine learning models. In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (pp. 531-535).

[27] Chen, C., Lin, K., Rudin, C., Shaposhnik, Y., Wang, S., & Wang, T. (2018). An interpretable model with globally consistent explanations for credit risk. *arXiv preprint arXiv:1811.12615*.

[28] Serrano-Cinca, C., Gutiérrez-Nieto, B., & López-Palacios, L. (2015). Determinants of default in P2P lending. *PloS one*, *10*(10), e0139427.

[29] Gupta, D. K., & Goyal, S. (2018). Credit risk prediction using artificial neural network algorithm. *International Journal of Modern Education and Computer Science*, *11*(5), 9.

[30] Malekipirbazari, M., & Aksakalli, V. (2015). Risk assessment in social lending via random forests. *Expert Systems with Applications*, *42*(10), 4621-4631.

[31] Mancisidor, R. A., Kampffmeyer, M., Aas, K., & Jenssen, R. (2020). Deep generative models for reject inference in credit scoring. *Knowledge-Based Systems*, 105758.

[32] Marceau, L., Qiu, L., Vandewiele, N., & Charton, E. (2019). A comparison of Deep Learning performances with others machine learning algorithms on credit scoring unbalanced data. *arXiv preprint arXiv:1907.12363*.

[33] Tsai, C. F., & Wu, J. W. (2008). Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert systems with applications*, *34*(4), 2639-2649.

[34] Aithal, V., & Jathanna, R. D. (2019). Credit risk assessment using machine learning techniques. *International Journal of Innovative Technology and Exploring Engineering*, *9*(1), 3482-3486.

[35] Salvaire, P. A. J. M. (2019). *Explaining the predictions of a boosted tree algorithm: application to credit scoring* (Doctoral dissertation).

[36] Gurumoorthy, K. S., Dhurandhar, A., Cecchi, G., & Aggarwal, C. (2019, November). Efficient Data Representation by Selecting Prototypes with Importance Weights. In *2019 IEEE International Conference on Data Mining (ICDM)* (pp. 260-269). IEEE.

[37] Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, *6*, 52138-52160.

[38] Martens, D., Baesens, B., Van Gestel, T., & Vanthienen, J. (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European journal of operational research*, *183*(3), 1466-1476.

[39] Gacto, M. J., Alcalá, R., & Herrera, F. (2011). Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, *181*(20), 4340-4360.

[40] Kulesza, T., Stumpf, S., Burnett, M., Yang, S., Kwan, I., & Wong, W. K. (2013, September). Too much, too little, or just right? Ways explanations impact end users' mental models. In *2013 IEEE Symposium on Visual Languages and Human Centric Computing* (pp. 3-10). IEEE.

[41] Hoffman, R. R., Mueller, S. T., Klein, G., & Litman, J. (2018). Metrics for explainable AI: Challenges and prospects. *arXiv preprint arXiv:1812.04608*.

[42] Isaac, S., & Michael, W. B. (1995). *Handbook in research and evaluation: A collection of principles, methods, and strategies useful in the planning, design, and evaluation of studies in education and the behavioral sciences*. Edits publishers.

[43] Hill, R. (1998). What sample size is "enough" in internet survey research. *Interpersonal Computing and Technology: An electronic journal for the 21st century*, *6*(3-4), 1-12.

[43] Jain, S., Luthra, M., Sharma, S., & Fatima, M. (2020, March). Trustworthiness of Artificial Intelligence. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 907-912). IEEE.

[45] Ashoori, M., & Weisz, J. D. (2019). In AI we trust? Factors that influence trustworthiness of AI-infused decision-making processes. *arXiv preprint arXiv:1912.02675*.

[46] Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018, October). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)* (pp. 80-89). IEEE.

AUTHORS

Lara Marie Demajo has degrees in M.Sc. in Artificial Intelligence and B.Sc. in Information Technology (Hons.) in AI, both from University of Malta. Her work has won various prizes including IEEE Best ICT Project and first place in FICTeX Final Year Project awarded by the Dean of Faculty of ICT. She has over 4 years experience in software development.

Dr Vince Vella brings over 25 years of senior technical leadership and management experience. Currently, he holds the position of CTO at Computime Software, BRSAnalytics and CTLabs. He holds a PhD from the Centre for Computational Finance and Economic Agents (CCFEA), University of Essex. Vince is also a lecturer within the Department of AI at University of Malta, mainly responsible for the MSc AI – Fintech stream. His main interests overlap Artificial Intelligence, Machine Learning and Computational Finance, particularly in the areas of AI Managed Funds, Algorithmic Trading, decentralized AI and AI for Anti Money Laundering.

Prof Alexiei Dingli is a Professor of Artificial Intelligence (AI) at the Department of AI within the University of Malta. He has been conducting research and working in the field of AI for the past two decades. His work was rated World Class by international experts and won various prizes including; the Semantic Web Challenge, the first prize by the European Space Agency, the e-Excellence Gold Seal award, the First Prize in the Malta Innovation Awards, the World Intellectual Property Organization (WIPO) award for Creativity and the first prize of the Energy Globe award by the UN, amongst others. He has published several peer-reviewed papers and books in the field. He also formed part of the Malta.AI task-force aimed at making Malta one of the top-AI countries in the world where he chaired the working-group on AI in work & education. Prof Dingli also assists various local and international organizations during their transformation towards becoming AI companies.

A NEW FRAMEWORK OF FEATURE ENGINEERING FOR MACHINE LEARNING IN FINANCIAL FRAUD DETECTION

Chie Ikeda, Karim Ouazzane and Qicheng Yu

School of Computing and Digital Media,
London Metropolitan University, London, UK

ABSTRACT

Financial fraud activities have soared despite the advancement of fraud detection models empowered by machine learning (ML). To address this issue, we propose a new framework of feature engineering for ML models. The framework consists of feature creation that combines feature aggregation and feature transformation, and feature selection that accommodates a variety of ML algorithms. To illustrate the effectiveness of the framework, we conduct an experiment using an actual financial transaction dataset and show that the framework significantly improves the performance of ML fraud detection models. Specifically, all the ML models complemented by a feature set generated from our framework surpass the same models without such a feature set by nearly 40% on the F1-measure and 20% on the Area Under the Curve (AUC) value.

KEYWORDS

Financial Fraud Detection, Feature Engineering, Feature Creation, Feature Selection, Machine Learning

1. INTRODUCTION

Online banking services has expanded rapidly, and in tandem, fraudulent activities via the internet and credit cards have increased substantially. According to Financial Fraud Action UK in 2020, the financial fraud losses registered a record high of £824.8 million in 2019 [1]. Payment card and remote banking account for 60% of the whole fraud losses. Evidently, the fraud detection system (FDS), used by many financial institutions, has not caught up with the advancement in fraud schemes. To address constant changes in fraud schemes, the FDS has incorporated machine learning (ML), but it is still challenging to reveal new fraudulent patterns by applying ML to raw data only.

The recent studies in financial fraud detection have further adopted feature engineering, which is an essential work in data preparation for ML. Feature engineering involves two main progresses: feature creation in which feature candidates are created from original data, and feature selection in which features are selected among the candidates as an input for ML.

Broadly, feature creation is classified into two types: feature transformation and feature aggregation. Feature transformation creates features by transforming original data using some functions, which typically adopt mathematical or statistical functions. The recent example in the field of financial fraud detection includes Bahnsenet al [2] who use the statistical function of the von Mises distribution to transform interval time between the last transaction and the latest

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 205-220, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101517

transaction by each individual customer. Feature transformation is also useful to convert values in categorical features into numerical values because ML algorithms unable to directly deal with categorical features. For instance, Dummy variables can represent a single class from a categorical feature by a set of binaries with the exact same information.

Feature aggregation creates features by aggregating some patterns observed from original data. Feature aggregation combines various features from multiple tables into a new summary form, e. g., average amount of transaction by each individual customer, and number of accesses to an online banking account per month. For example, Yesilkanat et al. [3] and Y.Xie et al [20] use feature aggregation to express a sequential pattern of transactions and create new features by combining original data such as the place (such as an ATM location), the amount, and the time of transaction.

Feature selection – another progress in feature engineering - selects relevant features from the candidates created in feature creation for ML algorithms. By doing so, it addresses two issues: effectiveness and compatibility. It selects effective features that improve ML model predictions. It also makes features readily useable for a different type of ML algorithms.

In financial fraud detection, a variety of ML algorithms have been used. They include support vector machine (SVM), random forests (RF), logistic regression (LR), K-means, local outlier factor (LOF), neural networks (NN). These ML algorithms are broadly classified into two types: supervised learning and unsupervised learning. Supervised learning uses historical transaction records including a fraud flag and learns the different patterns between fraud and non-fraud data, while unsupervised learning deals with big data and observes latent patterns without learning fraud flags from past data. Unsupervised learning has more potential to reveal underlying fraud patterns than supervised learning by multiplying data without training. Lee et al. [4] use a feature selection process for unsupervised learning for credit card fraud detection and show that a detection accuracy of the unsupervised learning model with selected features is better than that of the same model but without feature selection. Varmedja et al. [5] use a feature selection process for supervised learning models such as Naïve Bayes (NB) and LR, and show the effectiveness with selected features.

Despite these progresses in the field of financial fraud detection, in the process of feature creation, most studies use either feature aggregation or feature selection separately.

Even if one type of feature creation is used, few studies use feature selection before putting features into ML models. Conversely, even if feature selectin is used, few studies use feature creation before selection features; most of the studies select variables from original data.

Against the background, in this paper, we propose a new framework of feature engineering for ML in financial fraud detection. Specifically, our framework consists of feature creation process and feature selection process jointly. In feature creation process, both techniques of feature aggregation and feature transformation are used to create feature candidates, which could improve an accuracy of ML models. Subsequently, feature selection process evaluates the candidate features in terms of classification report and the Area Under the Curve (AUC). Features are then selected based on the evaluation and are used as an input for appropriate ML algorithms.

The salient aspect of this framework is three-fold. First and most importantly, the combination of creation process and selection processes: use of feature aggregation and feature transformation jointly to create important feature candidates, and selection from the feature candidates based on evaluation by specific ML models. Second, in feature selection process, we consider compatibility between features and individual ML algorithm and built the framework that can

accommodate any ML fraud detection models, which does not rely on a certain specific ML model. Third, few studies of feature engineering in financial fraud detection for unsupervised learning exist yet. We believe that performance of unsupervised learning models can be improved when using the selected important features based on our framework;

The rest of this paper is organised as follows. In Section 2, we review the techniques and recent development of feature engineering in general study and for financial fraud detection. In section 3, we describe about a real-life dataset from a European bank. Then, in Section 4, we present our development of new framework to create and evaluate effective features for fraud detection model. Afterwards, the experimental composition and the results is shown in Section 5. Finally, conclusion and discussion of the paper are given in Section 6.

2. RELATED WORKS

This paper is closely related to the recent literature on a fraud detection framework that incorporates feature engineering methods. One frequently used feature engineering approach combines two or more features from original data into new ones to represent customer's behaviour on transaction. J.M.Kanter et al [26] developed a cross domain framework that generalises three parts of features, which are Label, Segment, Featurise (L-S-F), to customise the process of feature creations. This feature engineering framework is a general concept to improve an accuracy of machine learning models. Y.Lucas et al. [19] built a conceptual framework of generating history base features using Hidden Markov Models (HMM). The framework calibrates the similarity between an observed sequence and the sequences of past fraud transactions inspected for the cardholders. These examples of feature engineering framework in the financial field are for supervised learning algorithms such as Decision Tree (DT), Random Forests (RF) and Logistic Regression (LR), while Nargesian et al. [8] and Heaton [9] introduce the frameworks for improving an accuracy of unsupervised learning algorithms: Deep Learning (DL), Recursive Neural Network (RNN) and Convolutional Neural Network (CNN) as credit card fraud detection models. The framework for unsupervised learning algorithms applies mathematical functions on a single feature in original data to create new features for improving an accuracy of fraud detection models. Xinwei et al. [6] developed a fraud detection system that uses a progressive feature engineering process based on "Homogeneity-oriented behaviour analysis (HOBA) using a deep learning model. HOBA uses four categories: Recency, Frequency, Monetary value, and Location, to categorise into some small groups based on the similar characteristic on transactions. These papers demonstrate the effectiveness of using feature creations for prediction models.

Feature creation methods in financial fraud detection are roughly divided into two categories: feature aggregation and feature transformation. The aggregated features are used for observing user's behaviour in transactions. Y.Xie et al. [20] developed a rule-based feature engineering method for credit card fraud detection that considers both individual behaviour and group behaviour, and creates group features that classify regular and fraudulent transactions. C.Whitrow et al. [21] introduced the new feature aggregation technique for credit card fraud detection that calculates over transactions observed by a fixed time window and between maximum and minimum amounts. Bahnsen et al. [2] created aggregated features by applying the statistical function of the von Mises distribution on interval time between the last transaction and the latest transaction by each individual customer.

Feature transformation transforms the original features into new ones to describe the original data. The methods of feature transformation applying mathematical functions such as log, square, normalization, addition, subtraction, multiplication, division, mean and standard deviation on

each attribute in a dataset are utilised in our framework and these methods are shown the effectiveness of improving an accuracy of machine learning models in general feature engineering studies [8, 9, 25, 27]. For example, J.M.Kanter et al [27] developed the Deep Feature Synthesis algorithm to create features for relational datasets. The algorithm observes relationships in the data and then sequentially applies mathematical functions among the data. Other feature transformation methods in the field of financial fraud detection are for unsupervised learning algorithms including deep learning [6, 22, 23, 24], and they show a high level of effects for unsupervised learning models.

Another feature engineering approach is to select significant features for specific ML algorithms. Lee et al. [4] use a feature selection method for unsupervised learning in credit card fraud detection to select relevant features to a target and they use feature selection methods such as filter, wrapper and embedded. Brodley et al. [10] employ the Expectation-Maximization clustering method that disperse separability and maximum likelihood. Xinwei et al. [6] select relevant features using Chi2 technique in feature selection for classification of e-commerce websites. D. Varmedja et al. [5] concluded that feature selection and balancing unbalanced label dataset should be carried out to enhance a credit card fraud detection for machine learning algorithms. Through the whole results of experiments using the selected features presented that feature selection is remarkably significant in achieving meaningful results.

These studies show the importance of feature selection by a comparison of the performances between ML models built with selected features and other ones built with only original features. Though many studies of feature engineering have proven the effectiveness of feature creation and feature selection individually, they seldom implement both methods together in one framework. In this paper, we use feature engineering methods of feature creation process and feature selection process jointly for ML in financial fraud detection.

3. ONLINE BANKING DATA ON TRANSACTIONS

An online payment dataset is provided by a European bank to verify the effect of the framework and it contains approximately 29,000 transactions across about 2,692 account holders in 3 days. The ratio of fraud labels is about 7% of all transactions. This dataset is partially extracted from over 100,000 transactions for a tentative experiment. In future work, we will examine with the full of transactions after verifying the effect of the framework in this paper.

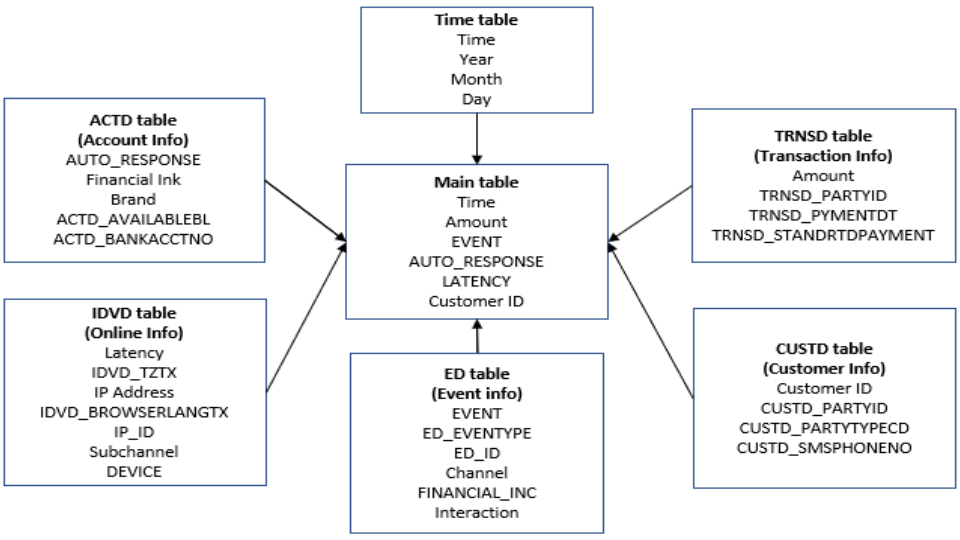


Figure 1. data modelling

The dataset, which is integrated from different tables such as time, account, online, customer's info, transaction, events, is as shown. Descriptions of each feature in the dataset are described in Table 1.

Table 1: Description of Original Features

Attributes	Description	Attributes	Description
ACTD_BANKACCTNO	Account's bank account number	CUSTD_ SMSPHONENO	SMS phone number
ACTD_AVAILABLEBL	Available balance	LATENCY	Latency
TRNSD_FASTER STANDARDPAYMENTIND	Faster or Standard payment indicator	IP Address	Access IP Address
ED_EVENTTYPETX	Type of payments	Interaction	Internet banking, branch, mobile, Tel
Customer ID	Customer Party ID	Time	Access date time / Timestamps
EVENT	Event of transaction	Financial INC	Transfer bank name
IDVD_INTESSIONID	Internet Section ID	Brand	Financial Institute name
IDVD_TZTX	Time zone of transaction	Sub channel	Sub-channel name
IDVD_USERAGE0TTX	Online user agent	DEVICE	Access devices
AUTO_RESPONSE	Auto response	IP_ID	Online banking ID

4. FEATURE ENGINEERING FRAMEWORK FOR FINANCIAL FRAUD DETECTION

The main contribution of our framework is to join two processes of feature creation and feature selection illustrated in Figure 2.

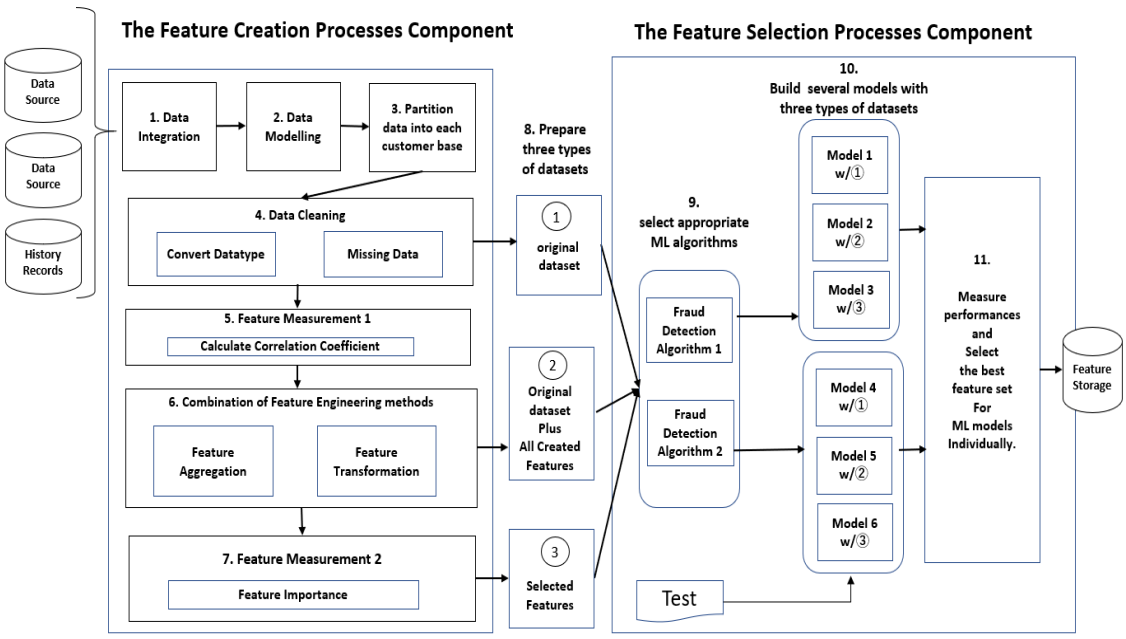


Figure 2. Feature Engineering Framework for Fraud Detection Models.

4.1. Feature Creation Processes

In the feature creation component, there are seven steps to create feature candidates and measure important features. The raw data collected from various sources is a mess and needs to be cleaned by dealing with data formats and missing values before implementation of feature engineering. The processes from step 1 to step 5 are relevant to pre-processing feature engineering, specifically in step 5, similar attributes are removed from original data to avoid over fitting by using correction coefficient as an evaluation method.

(a) Feature Aggregation based on Customer Behaviour

Feature aggregation represents customer's behaviour on online transaction. The original data is grouped by each customer ID to build an individual customer's profile. Aggregation makes more detailed features that express the individual customer's regular patterns by combining two or more attributes from various tables as shown in Figure 3 below.

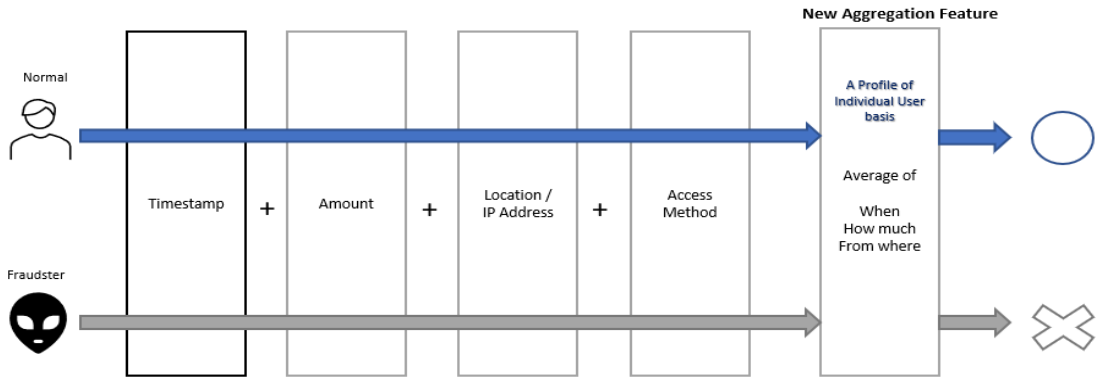


Figure 3. Image of Combining Multiple Features

In Table 2 describes some examples of feature aggregation that enable ML algorithms to learn various patterns of customer' behaviour and to classify a fraud pattern more easily.

Table 2. Feature Aggregation

Attributes	Combinations
Time	Days since the last transactions Hours since the last transactions Minutes since last transactions Days since the last access by same device Hours since the last access by same device Minutes since the last access by same IP address Hours since the last access by same IP address Days since the last event type occurred Hours since the last event type occurred Days since the last transaction occurred from specific location/ATM Hours since the last transaction occurred from specific location/ATM

IP Address	IP address of access device since last transaction IP address of access device since last transaction
Amount	Amount of the last transaction Amount of the last transaction from specific location/ATM Amount of the last transaction via IP address
Channel	Channel type when each event is occurred
Event Type	Event type accessed via IP address Event type accessed by a specific device

(b) Feature Transformation based on mathematical functions

We selected several mathematical functions to transform a single feature to different aspects. Some examples of mathematical functions used for transformation features are shown in Table 3.

Table 3. Feature Transformation

Examples of Mathematical Functions	Formula/Equations
Confidence Interval	a statistic estimation formula that uses the normal distribution for observing a point estimate by calculating ***maximum***, ***minimum***, ***median***, and ***mean***
Standard Deviation	a method of scaling the values based on z-score which calculates the following equation. $Z=(x-\mu)/\sigma$ where x:value to be transformed, μ: mean value of the data, σ: standard deveation
Binning	a way to group figures of continuous numbers into bins
Clustering (K-Means)	a way to group a set of spots into clusters based on a distance measure. Customer's info can be classified with the distances from an actual and some groups based on similar data patterns by using k-means
Linear	The equation: Let A1, ……, An be n matrices having dimension K x L. $B = \alpha_1 A_1 + \ldots + \alpha_n A_n$
Logarithm	Log transformation is one of the popular transformation. $X'_1 = log(xi)$

Now, we created approximately 42 feature candidates in the real-life dataset using feature aggregation and feature transformation methods as described in Table 4.

Table 4. New created features based on aggregation and transformation

Feature Engineering Time Series	Description
Year	Transaction year
Month	Transaction month
Day	Transaction day
Hour	Transaction hour

Minute	Transaction minute
Second	Transaction second
Weekday	Transaction weekday

Day of year	Days of year from transaction
Feature Engineering **Clustering**	**Description**
Class	Clustering group by k-means based on customer characters
Aggregations based on customer behaviour	**Description**
Customer ID conf Rate	Attributed rate scale by confidence on customer ID
ED_EVENT conf Rate	Attributed rate scale by confidence on Event Type
Action Type conf Rate	Attributed rate scale by confidence on Action Type
DEVICE conf Rate	Attributed rate scale by confidence on Device frequency
Amount conf Rate	Attributed rate scale by confidence on Amount
Customer ID EVENT par Day	Group by customer ID and Event frequency per day
Customer ID IP Address par Day	Group by customer ID and IP address frequency per day
Customer ID DEVICE par Hour	Group by customer ID and device frequency per hour
Customer ID USER count Minute	Group by customer ID and user agent counts per minute
Customer ID Channel count Minute	Group by customer ID and channel counts per minute
Customer ID counts	Count each customer ID
New feature	Time to next transaction for each customer
Transformations based on mathematical method	**Description**
Latency diff	Difference Latency
Amount diff	Difference Amount
Day diff	Difference Day
Hour diff	Difference Hour
Minute diff	Difference Minute
Access min	Minimum access time
Access max	Maximum access time
Access std	Standardization of Access time
LATENCY std	Standardization of Latency
Amount std	Standardization of Amount
Amount log	Log Transform of Amount
Min log	Log Transform of Minute
Sec log	Log Transform of Second
Day bin	Binning of Day
Min bin	Binning of Minute
Channel Event	Linear combinations (Channel and Event Type)
Action IP	Linear combinations (Action type and IP address)
Event Latency	Linear combinations (Event and Latency)
Event Sub Device	Linear combinations (Event Type and subchannel and device)
Event INC Code	Linear combinations (Event Type and Auth code and FC type)

Day of year	Days of year from transaction
Feature Engineering Clustering	**Description**

Class	Clustering group by k-means based on customer characters
Aggregations based on customer behaviour	**Description**
Customer ID conf Rate	Attributed rate scale by confidence on customer ID
ED_EVENT conf Rate	Attributed rate scale by confidence on Event Type
Action Type conf Rate	Attributed rate scale by confidence on Action Type
DEVICE conf Rate	Attributed rate scale by confidence on Device frequency
Amount conf Rate	Attributed rate scale by confidence on Amount
Customer ID EVENT par Day	Group by customer ID and Event frequency per day
Customer ID IP Address par Day	Group by customer ID and IP address frequency per day
Customer ID DEVICE par Hour	Group by customer ID and device frequency per hour
Customer ID USER count Minute	Group by customer ID and user agent counts per minute
Customer ID Channel count Minute	Group by customer ID and channel counts per minute
Customer ID counts	Count each customer ID
New feature	Time to next transaction for each customer
Transformations based on mathematical method	**Description**
Latency diff	Difference Latency
Amount diff	Difference Amount
Day diff	Difference Day
Hour diff	Difference Hour
Minute diff	Difference Minute
Access min	Minimum access time
Access max	Maximum access time
Access std	Standardization of Access time
LATENCY std	Standardization of Latency
Amount std	Standardization of Amount
Amount log	Log Transform of Amount
Min log	Log Transform of Minute
Sec log	Log Transform of Second
Day bin	Binning of Day
Min bin	Binning of Minute
Channel Event	Linear combinations (Channel and Event Type)
Action IP	Linear combinations (Action type and IP address)
Event Latency	Linear combinations (Event and Latency)
Event Sub Device	Linear combinations (Event Type and subchannel and device)
Event INC Code	Linear combinations (Event Type and Auth code and FC type)

4.2. Feature Selection Processes

Three types of datasets are set up after the processes in the feature creation component. The first dataset is original features, the second one is a set of original features and created features in the feature aggregation and transformation processes. The last dataset is only selected features from the second one based on feature importance. In the feature selection component, any ML algorithms for fraud detection can be chosen according to user's needs. In the framework, weselected two ML algorithms of support vector machine (SVM) and isolation forest (IF). SVM is a supervised learning algorithm and popularly used for fraud detection in many studies

[3,11,12,13]. In their studies, performance of SVM is steady and fine. IF is an unsupervised learning algorithm and works well for anomaly detection [14,15,16]. These ML algorithms use the three datasets individually to build each model and evaluate their results based on classification report and AUC. Eventually, the best feature sets can be selected for each ML model.

(a) Feature Importance

As an evaluation method of relevant features, we select feature importance from RF model to measure the relative importance of each input feature. Scores of feature importance are calculated by the training data used to the model. In the RF model, every node indicates a status of how to split values in an individual feature. The status is based on impurity, which is Gini impurity or information gain (entropy) in case of classification. While training the RF model, feature importance of each feature is computed how much a single feature contributes to reducing the weighted impurity. The figure 4 describes feature importance of each feature in the second dataset. It indicates that many importance features with high scores are the created features by feature engineering methods. Following this evaluation result, we selected 46 features out of 66 features in the second dataset.

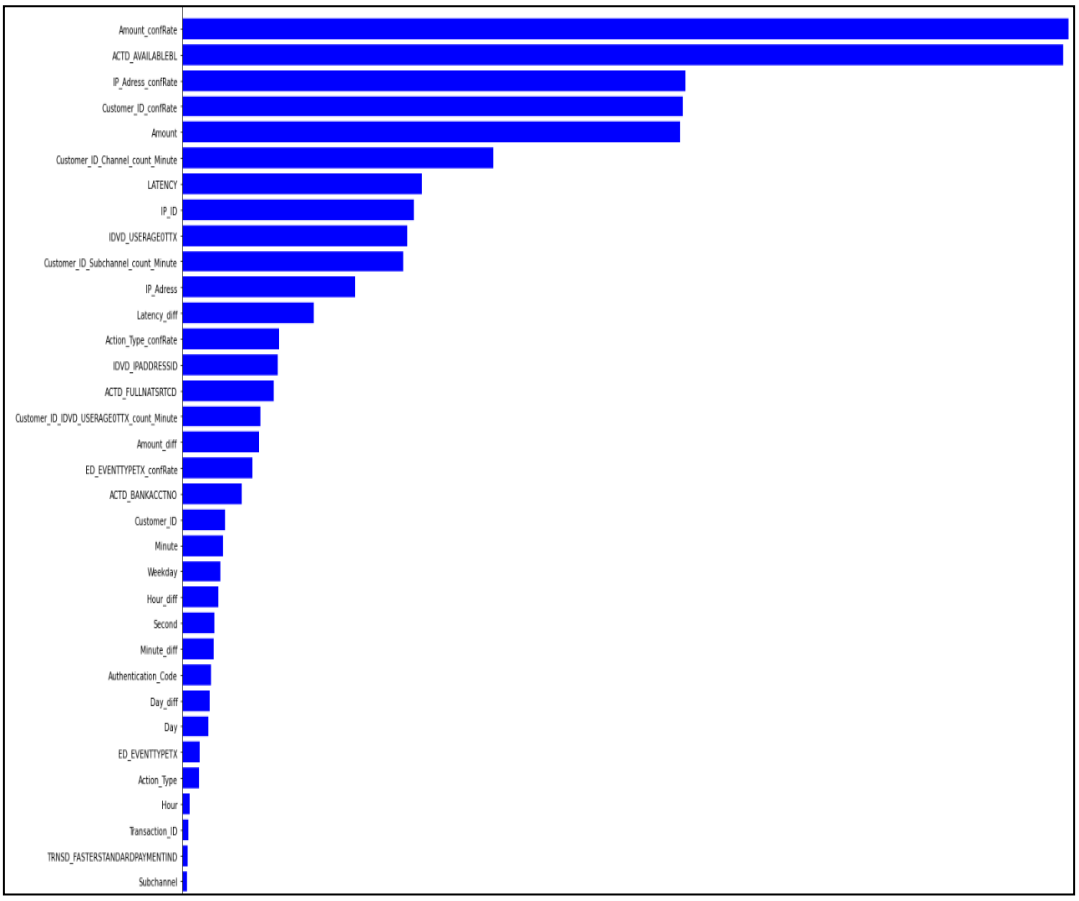


Figure 4. Feature Importance

(B) Fraud Detection Algorithms

- **Support Vector Machine**

Support vector machine (SVM) is a supervised learning algorithm and a popular classification method in financial fraud detection [3,11,12,13] to group values in dataset by applying a boundary line, called a hyper plane, which segregates a fraud pattern from normal patterns[18]. The best boundary will be determined by finding a hyper plane where splits the two classes of data locations by calculating maximum distance between the two classes shown in figure 5. A hyper plane is defined by the following function [18],

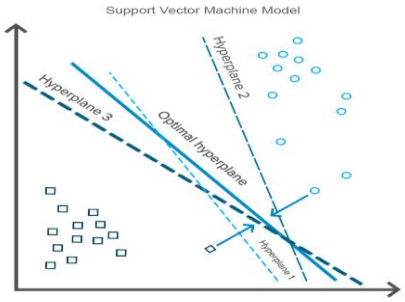


Figure 5. support vector machine approach

Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right) \quad (1)$$

Constraints:

$$y_i - wx_i - b \le \varepsilon + \xi_i \quad (2)$$
$$wx_i + b - y_i \le \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \ge 0$$

Linear SVM: (3)

$$y = \sum_{i=1}^{N}\left(a_i - a_i^*\right)\cdot\langle x_i, x\rangle + b$$

(4)

- **Isolation Forest Algorithm**

Isolation forest (IF) is an unsupervised learning algorithm for anomaly detection [14,15,16] and consists of multiple isolation trees which are created by repeating swiftly and randomly selecting attributes between the maximum and minimum values. Attributes values of anomalous instances are commonly different from the regular instances. The median depth of the instance in the forest which is consisted of multiple isolation trees is calculated to give a measure of the normality and anomalous scores of the instance. Equation of the algorithm is described as following:

$$Anomaly\ Score\ (S) = 2^{\frac{-E(h(k,m,N)}{c(n)}}$$

$$,where\ c(n) = 2(\ln(n-1) + 0.5772156649) - 2\left(\frac{n-1}{n}\right)$$

$$,where\ n\ is\ a\ number\ of\ data\ points\ in\ a\ chosen\ sample$$

$$,where\ E(h(k,m,N)) = \frac{\sum_{i=1}^{N}\begin{cases} if\ k == 1, \sum_{j=1}^{M} 1 \\ else,\ \sum_{j=1}^{M} 1 + c(k) \end{cases}}{N}$$

$$,where\ N\ is\ a\ total\ number\ of\ trees$$

$$,where\ M\ is\ a\ total\ number\ of\ binary\ splits$$

$$,where\ k\ is\ a\ total\ number\ of\ data\ points\ in\ the\ final\ node\ (exit\ node)$$

Equation 1. Calculation in isolation forest

Anomaly scores are calculated by the average cross multiple trees in the forest. In figure 6 and figure 7 show each sub dataset that was split randomly and the isolated data point of a non-anomalous point and an anomalous point [17].

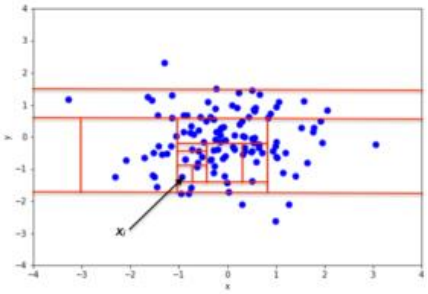


Figure 6. Isolated data point of a non-anomalous point [17]

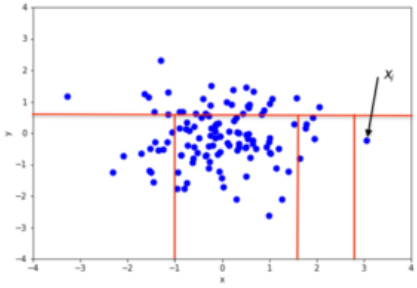


Figure 7. Isolated data point of an anomalous point [17]

5. MODELLING AND RESULTS

In the experiment of the feature engineering framework, six different models based on SVM and IF techniques are developed with three different types of feature sets, which are only original features, all created features and original features, selected features based on feature importance shown in Table 5. And subsequently, their performance is analysed and compared. Under Jupiter Notebook, python with sklearn library is used to create and evaluate features, and build SVM and IF models. As the performance evaluation methods, we use AUC and a classification report including precision, recall and F1-score. Each measurement is proceeded depends on how many target variable of fraud flag (“1”) is correctly detected by each model.

Table 5. Selected Features from All features in the dataset

Selected Features	Description
ACTD_AVAILABLE	Available balance
ACTD FULLNATSRTCD	Available transfer code
ACTD BANKACCTNO	Available bank account
Amount conf Rate	Attributed rate scale by confidence on amount
Latency diff	Difference Latency
Latency	Latency
Event Latency	Event latency
Event Act	Event action
Event INC Code	Event Inc code
IDVD USERAGETTX	Online user agent
Sub Channel PERSONAL	Sub channel type
Action IP	Action IP
Action Type conf Rate	Attributed rate scale by confidence on Action type
Amount	Transaction amount
Minute	Transaction minute
Hour	Transaction hour
Day	Transaction day
weekday	Transaction weekday
Customer ID IDVD USERAGE count Minute	Group by customer ID and online user agent frequency per minute
Customer ID Channel count Minute	Group by customer ID and channel frequency per minute
Customer ID counts	Group by customer ID counts per day
Amount diff	Difference Amount
Device DIGITAL	Access device and access type
ED EVENT TYPETX conf Rate	Attributed rate scale by confidence on event type
Minute diff	Difference Minute
Hour diff	Difference Hour
Day diff	Difference Day
Transaction ID	Transaction ID

Table 6: Performance of each model using three types of feature sets

Classifiers	F1-Measure	Precision	Recall	AUC
SVM with original data (1)	0.73	1.0	0.57	0.79
IF with original data (1)	0.25	0.24	0.26	0.59
SVM with all features (2)	0.97	1.0	0.94	0.97
IF with all features (2)	0.40	0.39	0.42	0.68
SVM with selected features (3)	0.95	1.0	0.91	0.95
IF with selected features (3)	0.60	0.57	0.62	0.79

* () ...dataset type

The measurement results of ML models using different feature sets are shown in Table 6. Recall shows the proportion of the actual fraud actions that were accurately detected, while precision donates the proportion of the accurately detected fraud actions to the detected fraud actions. Specifically, the aspect of F1-measure and AUC estimate the overall performance of ML models.

By comparing performances of the ML models using engineered features created by our framework with the ML models using only original features, all ML models using engineered features improve the accuracy in every measurements by nearly 40% on the F1-measure and 20%

on the AUC value. The SVM model using all features achieves the highest F1-measure of 0.97 and the highest AUC of 0.97, while the SVM model using only original data records the F1-measure of 0.73 and the AUC of 0.79. The IF models using created features through our framework have much better F1-measure scores of 0.60 and AUC of 0.79 than the IF model using original data that has the scores of 0.25 on F1-measure and 0.59 on AUC.

We compare the effectiveness of the feature set using all created features with using selected features based on feature importance to evaluate the compatibility between the effective feature set and a specific ML algorithm. The performance of SVM model using all features is better than SVM model using the selected features, whereas the performance of IF model using selected features is better than IF model using all features. The AUC value of SVM model using all features becomes 0.97, whereas the AUC value of SVM model using the selected features is 0.95. The AUC value of IF model using all features becomes 0.68, whereas the AUC value of IF model using the selected features is 0.79. We conclude that the important feature set is not effective for any ML algorithms in common. Finally, by comparing the performance of unsupervised learning models with supervised learning models, the AUC values and F1-measure scores of supervised learning models are higher than unsupervised learning models in every measurements. Overall, the results above demonstrate the effectiveness of the proposed feature engineering framework.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a new framework of feature engineering for ML models in financial fraud detection. What distinguishes our framework from others is that it involves both feature creation and feature selection. In addition, our feature creation process puts together two types of feature creation: feature aggregation and feature transformation. Moreover, our feature selection process is compatible with a variety of ML algorithms. Hence, our framework is general and applicable to many types of ML algorithms used in financial fraud detection and could enhance the existing financial fraud detection models. Using an actual financial transaction dataset from a private bank in Europe, we have shown that our framework improves the accuracy of ML model prediction significantly 40% on the F1-measure and 20% on the AUC value comparing with baseline models. We would like to conclude the paper with two caveats. First, although our experiment using an actual dataset shows an improvement in ML model prediction, the experiment uses standard ML algorithms such as SVM and IF, our framework will be applicable to richer algorithms such as a deep learning algorithm, which has recently attracted attention in financial fraud detection. Using such an algorithm in our framework is listed on our future work. Second, in our experiment, the data are limited to a small subset of large amounts of transactions. It would enhance fraud detection further if more contextual data about customer behaviour and transactions via various devices or online websites are used in our framework. Despite these caveats, we hope that our proposed framework will be useful for financial institutions to fight against financial fraudulent activities

REFERENCES

[1] Financial Fraud Action UK. January to June 2020 fraud update: Payment cards. Remote banking and cheque s.1.: Financial Fraud Action UK, 2020.

[2] A.C.Bahnsen, D.Aouada, A.Stojanovic & B.Ottersten (2016) “Feature engineering strategies for credit card fraud detection”, Expert Systems With Applications, Vol. 51,P134-142.

[3] A.Yesilkanat, B. Bayram, B.A.Koroglu & S.Arslan (2020) “An Adaptive Approach on Credit Card Fraud Detection Using Transaction Aggregation and Word Embeddings”, Semantic Scholar, corpus ID:218980594

[4] H.Lee, D.Choi, H.YIM, E.Choi, W.Go, T.Lee, I.Kim & K.Lee (2018) "Feature Selection Practice For Unsupervised Learning of Credit Card Fraud Detection", Journal of Theoretical and Applied Information Technology, Vol.96, No2, P408-417

[5] D.Varmedja, M.Karanovic, S.Sladojevic, M.Arsenovic & A.Andrela (2019) "Credit Card Fraud Detection - Machine Learning methods", 18th International Symposium INFOTEH-JAHORINA, 20-22 March 2019

[6] Z.Xinwei, H.Yaoci & W.Qili (2019) "HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture", Elsevier Information Sciences,online press

[7] S.Wang, J.Tang & H.Liu (2016) *Feature Selection*, Encyclopaedia of Machine Learning and Data Mining, P1-9, Springer Link

[8] F.Nargesian, H.Samulowits, U.Khurana, E.B.Khalil & D.Turaga(2017) "Learning Feature Engineering for Classification", IJCAI, P2529-2535

[9] J.Heaton (2017) "An Empirical Analysis of Feature Engineering for Predictive Modeling", Cornell University arXiv org, 1701.07852v1

[10] J.G.Dy & C.E.Brodley (2004) "Feature Selection for Unsupervised Learning", Journal of Machine Learning Research Vol. 5, P845-889

[11] C.Wang & D.Han (2018) "Credit card fraud forecasting model based on clustering analysis and integrated support vector machine", 406(1), P13861-13866, Springer Link

[12] Y.Jain, N.Tiwari, S.Dubey & S.Jain (2019) "A Comparative Analysis of Various Credit Card Fraud Detection Techniques", International Journal of Recent Technology and Engineering, Vol7, P2277-3878

[13] M.Khedmati, M.Drfani & M.GhasemiGol (2020) "Applying support vector data description for fraud detection", Cornell University arXiv org, 2006.00618v1

[14] S.P.Maniraj, A.Saini, S.D.Sarkar & S.Ahmed (2019) "Credit Card Fraud Detection using Machine Learning and Data Science", International Journal of Engineering Research & Technology, Vol. 8, P2278-0181

[15] F. Carcillo, Y.L.Borgne, O.Caelen, Y.Kessaci, F.Oble & G.Bontempi (2020) "Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection", ResearchGate, DOI:10.1016

[16] F.Liu, K.M.Ting & Z.H. Zhou (2008) "Isolation Forest", 8th IEEE International Conference on Data Mining: P413-422

[17] Z.Ding & M.Fei (2013) "An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window", Vol. 46, P12-17

[18] S.Patel & S.Gond (2014) "Supervised Machine Learning for Credit Card Fraud Detection", International Journal of Engineering Trends and Technology (IJETT), Vol. 8

[19] Y.Lucas, P.E.Portier, L.Laporte, L.H.Guelton, O.Caelen, M.Granitzer & S.Calabretto (2020) "Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs", ScienceDirect, ELSEVIER,1909.01185v1

[20] Y.Xie, G.Liu, R.Cao, Z.Li, C.Yan & C.Jiang (2019), "A Feature Extraction Method for Credit Card Fraud Detection", 2nd International Conference on Intelligent Autonomous Systems (ICoIAS), 2019.00019, DOI 10.1109

[21] C.Whitrow, D.J.Hand, P.Juszczak, D.Weston & N.M.Adams (2008), "Transaction Aggregation as a Strategy for Credit Card Fraud Detection", Data Mining and Knowledge Discovery, Vol. 18, P30-55

[22] K.Fu, D.Cheng, Y.Tu & L.Zhang (2016), "Credit Card Fraud Detection Using Convolutional Neural Networks", International Conference on Neural Information Processing, P483-490

[23] S.Misra, S.Thakur, M.Ghosh & S.K.Saha (2019), "An Autoencoder Based Model for Detecting Fraudulent Credit Card Transaction", ScienceDirect, Vol. 167, P254-262

[24] J.Jurgovsky, M.Granitzer, K.Ziegler, S.Calabretto, P.Portier, L.Heguelton & O.Caelen (2018), " Sequence Classification for Credit-Card fraud detection", Expert Systems with Applications

[25] F.Nargesian, H.Samulowitz, U.Khurana, E.B.Khalil & D.Turaga (2017), "Learning Feature Engineering for Classification", 26th International Joint Conference on Artificial Intelligene, (IJCAI-17)

[26] J.M.Kanter & K.Veeramachaneni (2016), "Label, Segment, Featurize: a cross domain framework for prediction engineering", IEEE International Conference on Data Science and Advanced Analytics (DSAA), P430-439

[27] J.M.Kanter & K.Veeramachaneni (2015), "Deep Feature Synthesis: Towards Automating Data Science Endeavors", IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2015.734485

AUTHORS

Chie Ikeda holds a master's degree in data Analytics and a PhD student at London Metropolitan University, UK. She is also an assistant manager and data scientist at Aflac Life Insurance Company in Japan. One of her responsibilities at the company is to lead her project team to successfully build an AI model for detecting a fraud.

Karim Ouazzane is a currently a full professor of computing and knowledge exchange at London Metropolitan University. He is Director of research and enterprise in the school of Computing and Digital Media and the University Knowledge Transfer Partnership Director. He is currently the Chair of the European Cyber Security Council at Brussels. He worked and acted as a consultant for a number of companies such as Endress-Hauser, ICI, Power Gen, Schlumberger, Barclays, Lifeline IT and Lloyds Banking Group in the UK in the area of machine learning and cyber security.

Qicheng Yu is a senior lecturer at London Metropolitan University. He leads modules in data mining, programming for data analytics, e-business and e-commerce, database and web applications development and supervises PhD research students.

3-D OFFLINE SIGNATURE VERIFICATION WITH CONVOLUTIONAL NEURAL NETWORK

Na Tyrer[1], Fan Yang[1], Gary C. Barber[1], Guangzhi Qu[1],
Bo Pang[1] and Bingxu Wang[1, 2]

[1]Automotive Tribology Center, Department of Mechanical Engineering, School of Engineering and Computer Science, Oakland University, Rochester, Michigan, 48309, USA
[2]Faculty of Mechanical Engineering and Automation, Zhejiang Sci-Tech University, Hangzhou, Zhejiang, 310018, P.R.China

ABSTRACT

Signature verification is essential to prevent the forgery of documents in financial, commercial, and legal settings. There are many researchers have focused on this topic, however, utilizing the 3-D information presented by a signature using a 3D optical profilometer is a relatively new idea, and the convolutional neural network is a powerful tool for image recognition. The present research focused on using the 3 dimensions of offline signatures in combination with a convolutional neural network to verify signatures. It was found that the accuracy of the data for offline signature verification was over 90%, which shows promise for this method as a novel method in signature verification.

KEYWORDS

Signature Verification, 3D Optical Profilometer, Convolutional Neural Network

1. INTRODUCTION

With the security requirements related to signing contracts, signing checks, etc, handwriting has become a more and more important factor in current society. An individual's signature is easily influenced by emotions and may vary day-to-day or can be completely different over time. Signature fraud is difficult for a human's eye to identify; thus it is important to find an automatic signature verification method. To prevent signature fraud, the handwriting verification system becomes crucial. Handwriting verification is widely used in many fields, it is essential to prevent the forgery of documents in financial, commercial, and legal environments.

Signature verification can be classified into two types: online and offline. For the online type, a special pen and an electronic surface are required. The stored data provides information such as the pen's position, velocity, acceleration, pressure, and angle as a function of time. However, using an electronic pen can be very different from using a pen. For the offline type, only the signed documents are available to be analyzed. While online signature verification provides more information, it needs a specific system that is not always available. What's more, most of the important documents require a handwritten signature. Therefore, exploring improved methods to verify offline signatures is essential.

Several studies have been conducted on this topic. In "Automatic Signature Verification", the author presented the state of the art in automatic signature verification. It contained a

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 221-228, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101518

comprehensive bibliography of more than 300 selected references as an aid for researchers working in the field including both online and offline signature analysis [1]. The researchers summarized two type of features – function features and parameter features. Function features are position, velocity, acceleration, pressure, force, direction, etc. Parameter features are global parameters such as, number of penups or pendowns, time duration of positive, etc, and local parameters such as component priented and pixel oriented. It was concluded that based on current methods, the accuracy is very promising. It also pointed out that the research should be more focused on device interoperability and the need for specific investigations. For the online signature verification, the speed of the pen is used as a dynamic feature of the signature [2]. The pen-input on-line signature verification incorporating pen-position, pen-pressure, and pen-inclinations trajectories was developed in "On-line Pen Input Signature Verifier PPI (pen-Position/ pen-Pressure/pen-Inclination)" [3].

For the offline signature verification, an offline signature verification system using a convolutional neural network was developed [4]. The resulting model was based on pre-trained VGG16 architecture using ICDAR 2011 SigComp dataset to train with transfer learning. J. Coetzer et al. [5] developed an off-line signature verification system that uses features that are based on the calculation of the Radon transform (RT) of a signature image. Each writer's signature is subsequently represented by a hidden Markov model (HMM). In "A smoothness index-based approach for offline signature verification" a method was developed based on a smoothness criterion [6]. Although skilled forgery signatures are very similar to genuine, they are generally less smooth and natural on a detailed scale than the genuine ones. R. Sabourin et al. [7] investigated the use of shape matrices as a mixed shape factor for off-line Signature Verification. Originally, shape matrices have been used for planar shapes, however, the reseachers used shape matrices to compare signatures. They concluded that shape factor is relatively well suited for the global interpretation of signature images. Luiz G. Hafemann et al [8] performed extensive experiments on four deep convolutional neural networks databases which are GPDS, MCYT, CEDAR and Brazilian PUC-PR datasets. The results show large improvement on GPDS-160 database. As shown above, the challenge of offline signature is not only extracting the signature but also to choose a reliable database.

Several research papers considered online signature pen pressure analysis, which requies an electronic pen. However, there is a difference between writing with a digital pen versus writing with an actual pen. With a 3D optical profilometer, offline signature pen pressure analysis can be determined precisely. Previous research has focused on 2D information for the offline signatures, which for highly skilled forgery may appear highly similar to the actual signature, but the pen pressure is harder to mimic. The purpose of this research is to determine the 3D information of offline signatures and utilize a convolutional neural network to allow a computer to extract features of the signature.

2. EXPERIMENTAL PROCEDURE

The present research consisted of two tasks. One is planning to explore deep learning algorithms, i.e. convolutional neural network which is used to train and test the images for 3-D information. Experiments were conducted to test the network by using the signatures during training. The second task was to utilize a Bruker Contour GT-K 3D Optical Profilometer to extract the offline signature. Figure 1 shows a 3-dimensional image letter 'a' with various depths that occur due to indentations in the paper when the letter was written. For example, the bright blue color represents areas where pen pressure was high producing relatively deeper indentations, the green color represents relatively low pen pressure and the red color represents the original surface of the paper.

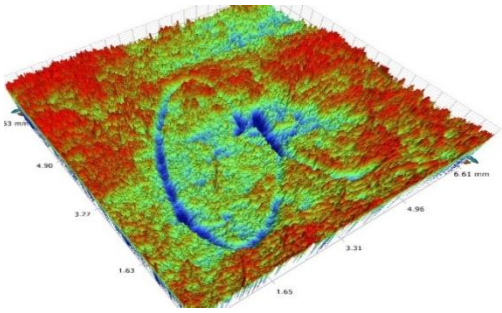


Figure 1. The scanned 3-D image of the handwritten letter ‘a’

Figure 2 shows the typical convolutional neural network (CNN) architecture. Regular neural networks take the input data and convert it into a one-dimensional vector of neurons, while CNN's work in 3 dimensions: width, height, depth. For example, the input is the image. The width and the height are dimensions of the image, and the depth stands for 3 channels (Red, Green, and Blue). Figure 3 shows the neuron in 3 dimensions as visualized in one of the layers. In addition, the neurons in a layer will only be connected to a small region of the layer before it, rather than all of the neurons in a fully connected position. At the end of the CNN architecture, the full image was reduced into a single vector of class scores, arranged along the depth dimension.

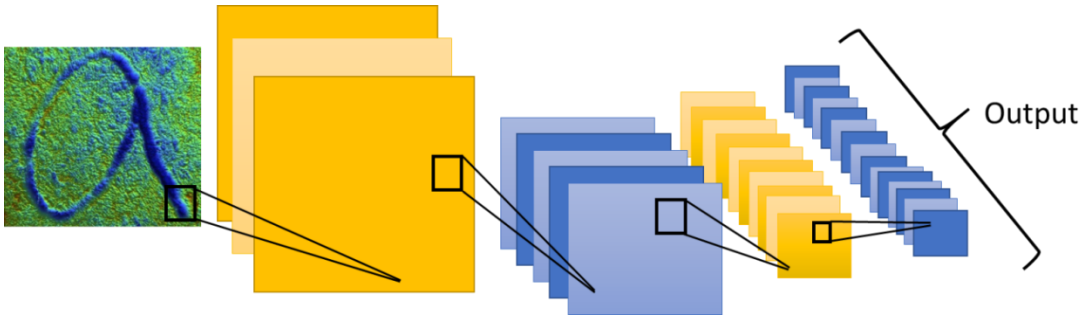


Figure 2. Typical CNN architecture

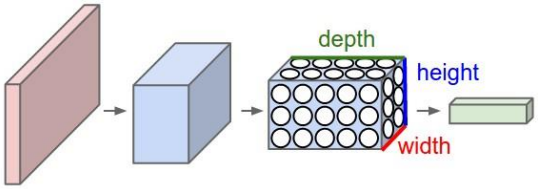


Figure 3. The neuron in 3D as visualized in one layer [9]

The advantage of this technique is to analyze the pen contact pressure on surfaces such as paper, instead of an electronic pen. As shown in Figure 4(a)-(d), there are two participants who wrote “Tom” and “Jim” on a piece of paper. They may not show significant differences by the naked eye. However, using pen pressure as shown in Figure 4(e)-(h) a significant difference can be observed. Using this technique combined with the current signature recognition technique, the accuracy of signature recognition is expected to be enhanced.

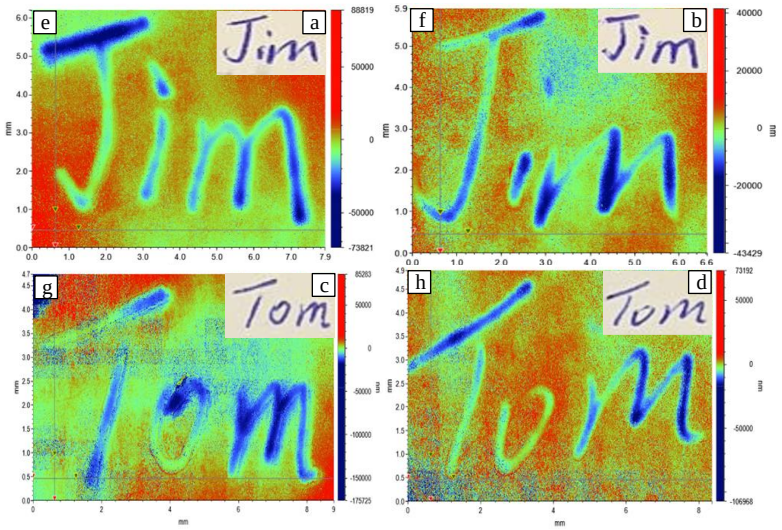


Figure 4. (a) (c) (e) (g) “Jim” “Tom” wrote by Participant One (b) (d) (f) (h) “Jim” “Tom” wrote by Participant Two

To demonstrate the concept of this research, a simply database was created. The database is composed of 80 offline samples written by two participants. Because of the time constraints of the 3-D machine scanning process, the letter ‘a’ is used to demonstrate the concept. In the experiment, every participant wrote 40 letters (35 for training and 5 for testing). Considering the instability which occurs in the process of collecting data, the “signature “is recorded on the same sheet of paper using the same pen to eliminate potential variables.

The optical profilometer has different types of filters to help eliminate noise, remove the tilt of the surface, etc. As shown in Figure 5, it can measure the various depths of the surface in 2D and 3D mode.

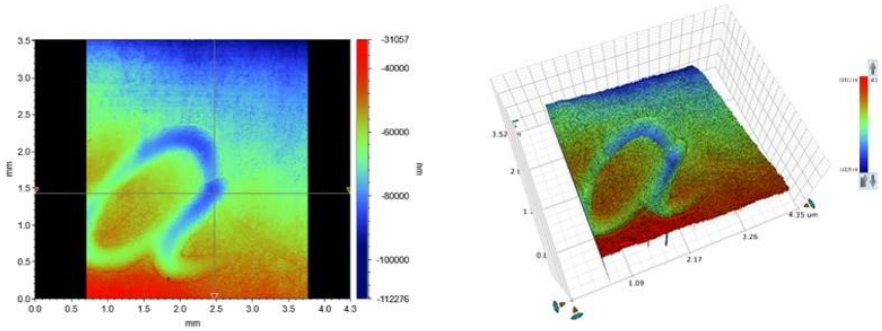


Figure 5. Same handwriting with different mode (left – 2D image, right – 3D image)

3. RESULTS

In the experiments, the database (35 letters of each participant) was divided into a training set (Figure 6) and a testing set in a ratio of 3:7 to train the model, and 5 unseen data were used to test the performance. Five of the thirty-five training letters after the scanning process are shown below:

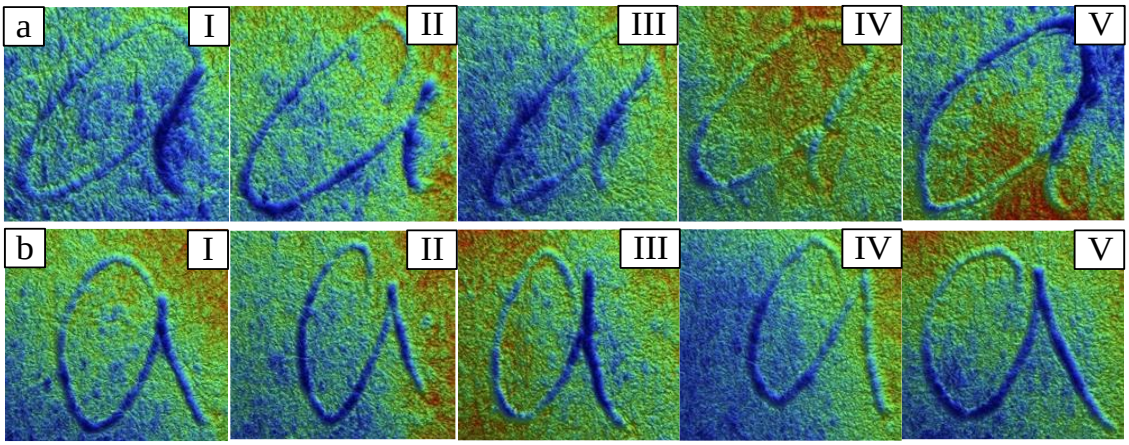


Figure 6. (a) Five training data from Participant One (b) Five training data from Participant Two

For the two participants, five handwritten letters were used to test the accuracy of the model. The data are shown in Figure 7.

For participant one, the 5 data were used to test the model ten times. Then an average accuracy was calculated which was over 90%. Three of the ten results are shown below:

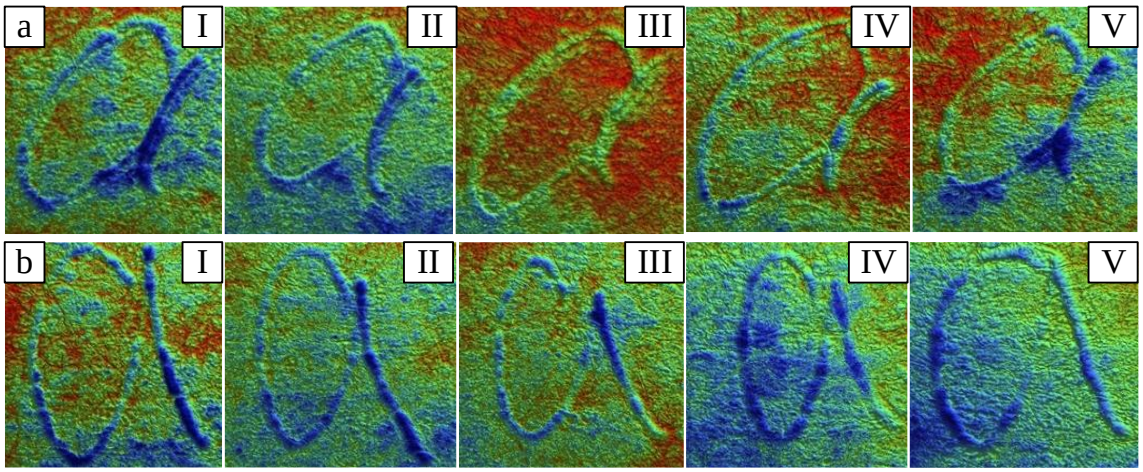


Figure 7. Five handwritten letters for the accuracy tests of the model (a) Participant One (a) Participant Two

Table 1. Summary results of Participant One

Participant One	1st	ConfMat = [1.0 0; 0.1250 0.8750] Accuracy = ***0.9375***
	2nd	ConfMat = [0.9583 0.0417; 0 1.000] Accuracy = ***0.9792***
	3rd	ConfMat = [1.0 0; 0.0417 0.9583] Accuracy = ***0.9792***

For participant two, the same method applied to participant one, three of the ten test results are shown below.

Table 2. Summary results of Participant Two

Participant Two	1st	ConfMat = [1.0 0; 0.0417 0.9583] Accuracy = ***0.9792***
	2nd	ConfMat = [0.9167 0.0833; 0 1.000] Accuracy = ***0.9583***
	3rd	ConfMat = [0.9583 0.0417; 0 1.000] Accuracy = ***0.9792***

4. CONCLUSION

As the results have shown, with the combination of the optical profilometer and the CNN, the accuracy of the data for offline signature verification is very promising. For the 3D signature verification method, a new option is provided to verify signatures and achieve a high accuracy in the analysis of offline handwriting on top of other researchers have achieved In future research, the training database needs to be developed with much larger data inputs as training sets by collecting more offline signatures from various people. The more input data collected, the more accurate the results will be. In addition, a more complicated algorithm to combine all relevant factors, such as handwriting similarity, pen-pressure, and pen-inclination trajectories, etc should be developed. Using the optical profilometer is time-consuming but introducing one more factor is a significant contribution for offline analysis. Even with different types of paper, the relative pen pressure has value for fraud detection.

REFERENCES

[1] D, Impedovo & G, Pirlo, (2006) "Automatic Signature Verification: The State of the Art", IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, Vol. 38, No. 5, pp609.

[2] A, Al-Shoshan, (2008) "Handwritten Signature Verification Using Image Invariant and Dynamic Features", Proceedings of the International Conference on Computer Graphics, Imaging and Visualization (CGIV 2006), Jul 2006, Sydney (Australia). pp173–176.

[3] Y, Komiya & T, Matsumoto, (1999) "On-line Pen Input Signature Verifier PPI (pen-Position/ pen-Pressure/pen-Inclination)", IEEE Smc'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man & Cybernetics, Oct 1999, Tokyo (Japan).

[4] G. Alvarez, B. Sheffer, M. Bryant, Offline Signature Verification with Convolutional Neural Networks

[5] J, Coetzer, B, Herbst & J. Preez, (2006) "Off-Line Signature Verification: A Comparison between Human and Machine Performance", Tenth International Workshop on Frontiers in Handwriting Recognition, Oct 2006, La Baule (France).

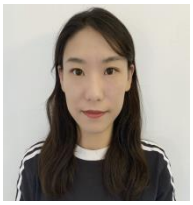
[6] B, Fang, Y, Wang, C, Leung, P, Kwok, K, Tse, Y, Tang, & Y, Wong, (1999) "A Smoothness Index-Based Approach for Offline Signature Verification", Proceedings of the Fifth International Conference on Document Analysis and Recognition ICDAR'99, Sep 1999, Banalore (India). pp785–791.

[7] R, Sabourin, J, Drouhard & E, Wah, (1997) "Shape Matrices as A Mixed Shape Factor for Offline Signature Verification", Proceedings of the Fourth International Conference on Document Analysis and Recognition ICDAR'97, Aug 1997, Ulm (Germany). Vol. 2, pp661–665.

[8] Luiz G.Hafemann, Robert Sabourin, Luiz S.Oliveira, (2017) "Learning Feautres For Offline Handwritten Signaature Verification Using Deep Convolutional Neural Networks", Pattern Recognition 70 (2017) pp 163-176

[9] CS231n Convolutional Neural Networks for Visual Recognition, http://cs231n.github.io/convolutional-networks/

AUTHORS



NA TYRER is currently a Ph.D. student at Automotive Tribology Center at Oakland University. She is closely working with Automotive OEMs as a research assistant to investigate the tribological performance of automotive parts. Her main research field is the tribological behavior of electrical coatings.

FAN YANG received her B.S. degree from the Beijing Information Science and Technology University and M.S. degree from the Oakland University in 2020.

GARY C. BARBER received his PhD from the University of Michigan in 1987. He is currently a Professor and Director of the Automotive Tribology Center at Oakland University in Rochester, Michigan. Dr. Barber is a fellow of the Society of Tribologists and Lubrication Engineers (STLE) and has published more than 125 papers in journals and conference proceedings. His research areas include the effects of various heat treatments on the tribological behavior of steels and cast irons and the use of nanofluids as lubricants.

GUANGZHI QU received the B.E. and M.E. degrees from the Department of Computer Science and Engineering, Beihang University, and the Ph.D. degree from the University of Arizona in 2005. He is currently a professor at Oakland University in Rochester, Michigan. His research areas include the data mining, machine learning, operating systems, and program analysis.

BINGXU WANG received the Ph.D degree from Oakland University, Michigan, USA, in 2018. He is currently an assistant professor of the Faculty of Mechanical Engineering and Automation at Zhejiang Sci-Tech University, China. Dr. Wang is a member of the Society of Tribologists and Lubrication Engineers (STLE) and the Society of Automotive Engineers (SAE), and has published about 28 papers in various peer reviewed journals and conference proceedings. His research areas include 3D surface measurements, metallurgical evaluation, austempered ductile iron, heat treatment design, tribological properties and nanofluids applications.

NEGATIVE SAMPLING IN KNOWLEDGE REPRESENTATION LEARNING: A MINI-REVIEW

Jing Qian[1, 2], Gangmin Li[1], Katie Atkinson[2] and Yong Yue[1]

[1]Department of Intelligent Science, School of Advanced Technology,
Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu Province, China
[2]Department of Computer Science, University of Liverpool,
Liverpool, United Kingdom

ABSTRACT

Knowledge representation learning (KRL) aims at encoding components of a knowledge graph (KG) into a low-dimensional continuous space, which has brought considerable successes in applying deep learning to graph embedding. Most famous KGs contain only positive instances for space efficiency. Typical KRL techniques, especially translational distance-based models, are trained through discriminating positive and negative samples. Thus, negative sampling is unquestionably a non-trivial step in KG embedding. The quality of generated negative samples can directly influence the performance of final knowledge representations in downstream tasks, such as link prediction and triple classification. This review summarizes current negative sampling methods in KRL and we categorize them into three sorts, fixed distribution-based, generative adversarial net (GAN)-based and cluster sampling. Based on this categorization we discuss the most prevalent existing approaches and their characteristics.

KEYWORDS

Knowledge Representation Learning, Negative Sampling, Generative Adversarial Nets.

1. INTRODUCTION

A knowledge graph (KG) is essentially a structural approach to tell facts. It refers to a network whose nodes are real entities or abstract concepts and edges are their in-between relations. Many KGs have gained steady development, such as NELL [1], Freebase [2] and YAGO [3]. They store and express ground-truth facts in the form of a triple (head entity, relation, tail entity) or (subject, predicate, object). Inspired by word embedding [4], people turned to distributed representation of entities and relations instead of one-hot representation that benefits the storage of triples but fails in capturing latent semantics.

Knowledge representation learning (KRL) is also known as knowledge graph embedding (KGE), it attempts to embed entities and relations in the KG into low-dimensional vector space. In recent years, a variety of KRL models have been successively proposed and deployed. Looking at conventional translational distance-based TransE [5], semantic matching-based RESCAL [6] or the state-of-the-art attention-based KBAT [7] and GAATs [8], they aim to learn better knowledge representations to serve knowledge graph completion tasks. KRL models define their own scoring functions on account of different embedding modes, which returns a score to measure the plausibility of the given triple. Mikolov et al. [4] simplifies noise contrastive estimation (NCE)

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 229-238, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101519

[9] to negative sampling with the aim of reducing computational complexity. KRL extends this strategy that ranks observed ("positive") instances higher than unobserved ("negative") ones [10]. As seen in translational distance-based models [5, 11-14], they are optimized through partitioning scores of positives and negatives with an adaptive margin. A large number of negative samples are required in training KRL models. However, most KGs only store ground-truth triples, for the sake of space efficiency. Negative sampling thus plays a pivotal role in the training process. The widely-used negative sampling method is uniform sampling[5, 12], which replaces the head or tail entity of the positive triple with an entity that is uniformly sampled from the entity set of the KG. Nevertheless, such generated negative triples are too obviously incorrect and contribute less as the training goes on, in most cases. Bernoulli sampling[11] applies different probabilities in head and tail replacement to alleviate the problem of false-negative triples. KBGAN [15] and IGAN [16] adversarially train the generator to provide better-quality negatives by applying a pre-trained KRL model as the discriminator. TransE-SNS [17] and NSCaching [18] carry out negative sampling in a more concentrated way. Furthermore, enlightened by CKRL [19], NKRL [20] puts forward a confidence-aware negative sampling method. Yang et al. [21] recently derives the general form of an effective negative sampling distribution, which is of pioneering significance. They are the first to deduce the correlation between positive and negative sampling distribution. Trouillon et al. [22] further studies the number of negatives generated for each positive triple, and elicits that fifty negative samples per positive is a good choice for balancing accuracy and training time.

In this review, we summarize current negative sampling methods and divide them into three categories, sampling from fixed distribution, sampling from GAN-based framework and sampling from custom cluster. Most KRL research focuses on proposing new embedding methods or their applications in downstream tasks, such as knowledge graph completion [23], question-answering [24] and recommendation [25]. Little attention is paid to negative sampling, although it is an influential and crucial step in KRL model training. In KRL surveys [26, 27], negative sampling is mentioned but only in a short space. To the best of our knowledge, this review is the first work to systematically and exhaustively overview existing negative sampling methods in KRL.

Around twelve negative sampling techniques applied in KRL are summarized in our work. Definitions and notations, as well as two necessary assumptions before modelling, are briefly covered in Section 2. Developments in KRL are presented in Section 3, in which we sort KRL models from four perspectives as is routine. Negative sampling is elaborated in Section 4 and this presents our main contribution. Finally, this review finishing with a conclusion and future research directions.

2. DEFINITIONS, NOTATIONS AND ASSUMPTIONS

In a standard KG, $\mathbb{E}$ represents the set of entities, $\mathbb{R}$ represents the set of relations. $\mathbb{D}^+$ and $\mathbb{D}^-$ are sets of the positive triples $\tau^+ = (h, r, t)$ and the counterpart negative triples respectively. The following formula sets out the components of the set $\mathbb{D}^-$. In general cases, one KRL model can be explained by its own-defined scoring function $f_r(h, t)$ where h and t belong to $\mathbb{E}$ and r belongs to $\mathbb{R}$. The relation r maps the head entity h to its tail entity t. The plausibility of each possible triple is measured by the scoring function. The higher the plausibility is, the more probability for the triple being a piece of truth.

$$\tau^- \in \mathbb{D}^-$$

$$\mathbb{D}^- = \left\{(h',r,t)\middle|h' \in \mathbb{E} \bigwedge h' \neq h \bigwedge (h,r,t) \in \mathbb{D}^+\right\} \cup \left\{(h,r,t')\middle|t' \in \mathbb{E} \bigwedge t' \neq t \bigwedge (h,r,t) \in \mathbb{D}^+\right\} \cup \left\{(h,r',t)\middle|r' \in \mathbb{D} \bigwedge r' \neq r \bigwedge (h,r,t) \in \mathbb{D}^+\right\}$$

KRL models are trained under the open world assumption (OWA) [28] or the closed world assumption (CWA) [29]. The CWA states that facts that are not observed in $\mathbb{D}^+$ are false, while the OWA is relaxed to assume that unobserved facts can be either missing or false. Most models prefer the OWA due to the incompleteness nature of KGs. The CWA has two main drawbacks, worse performance in downstream tasks and scalability issues caused by tremendous negative samples [26].

3. KRL MODELS

The goal of KRL is to embed triples (h,r,t) into a low-dimensional continuous vector space. A scoring function $f_r(h,t)$is manually defined to calculate the credibility score for the given triple. Different models embed semantic information into the vector representation of the KG in different ways [26]. By convention, there are mainly two types of KRL models, the Translational Distance-based and the Semantic Matching-based. In recent years, neural networks and additional information (entity type, path, text, etc.) have also been considered.

Translational distance-based models. The main idea of the translation-based models is to measure the distance between the head entity and the tail entity after triples in the KG are vectorized. Inspired by translation invariance in word vectors, TransE [5] considers the relation vector as a transition from the head to the tail, i.e. $h + r \approx t$. A short distance between $h + r$ and t reflects high credibility of the given triple. TransH [11] improves TransE to make it more applicable for modeling complex relations, like one-to-many and many-to-many. Some other variants extend TransE by projecting the embedding vectors of entities into various spaces, such as TransR [12], TransD [13] and TransG [14].

Semantic matching-based models. Compared to translational distance-based models, semantic matching-based models pay more attention to the latent semantics embodied in vectorized entities and relations. They are also called matrix decomposition models. RESCAL [6] is one of the earlier works that defines the scoring function based on semantic matching, in which, the relation vectors compose the mapping matrix $\mathrm{M_r}$ between the head and the tail, and the matrix product $h\mathrm{M_r}t$ is used to measure the plausibility of triples. DistMult [30] simplifies RESCAL by limiting $\mathrm{M_r}$ to be a diagonal matrix, while ComplEx [22] extends DistMult to the complex field to build an antisymmetric-relation model.

Neural network-based models. Applying neural networks in KRL has also seen steady progresses. MLP [31] feeds entities and relations into a fully-connected layer to encode semantic matching. ConvE [32] attempts to fit the scoring function using 2D convolution. By distinguishing relations and entities, RSN [33] introduces a recurrent skip mechanism. KG-BERT [34] is based on Transformer (BERT) to integrate KRL and language model pre-training. Referring to graph neural networks (GNNs), R-GCN [35] is the pioneer to encode relational data with the graph convolutional network framework.

Auxiliary-dependent models. Some work suggests incorporating additional information for improvement. Guo et al. [36] considers the entity type to be an extra piece of information and assumes that entities of the same type ought to be closer in vector representation. PTransE [37]

attempts to describe multi-hop relations between entities through addition, multiplication and RNN rules so that the relation paths between entities can be represented by the vector calculations of relations. In addition, Wang et al. [38] introduces a joint model adding the text information in the embedding process, and Guo et al. [39] comes up with a rule-based KRL model combining some rule information.

All the above models require negative samples during training. Before explaining the necessity of negative sampling in KRL, its roles in word embedding ought to be mentioned. Both word embedding and KGE belong to the scope of unsupervised learning. The softmax function has conventionally served as the training objective that approximately maximizes its log probability by normalizing with respect to all words in the dictionary, which is highly inefficient and computationally expensive. Negative sampling is proposed to simplify the computation. Instead of estimating the probability distribution based on the whole dictionary, the final representations can be obtained through distinguishing the positive sample from a few negative samples that are generated by perturbing the positive one. In view of random walk over graphs, graph structured data is similar to natural language, where nodes are as words and links as context. KRL adopts negative sampling that is trained by discriminating from a preset number of negative samples, rather than modelling conditional on all nodes.

It is noticed that poor negative samples can be easily discriminated and helpless for training. However, most KRL studies center on embedding modes and simply apply uniform sampling to generate negative training triples [26]. At present, only a few works have been devoted to improving the quality of negatives. We outline these methods with the aim of gaining more attention to this field. Besides, conventional and the state-of-the-art KRL models, their applications and future trends, can be found in the representative surveys [26, 27, 40].

4. NEGATIVE SAMPLING

Negative sampling was first proposed in neural probabilistic language models and labelled as importance sampling [41]. Mikolov et al. [4] emphasizes it as a simplified version of NCE [9] to benefit the training of word2vec. Extended by word embedding, KGE also takes negative sampling as the prerequisite for the model training. Poor or too obviously false negatives are hard to capture for latent semantics in the KG and easily cause the zero loss problem as well. Conversely, generating better-quality negative triples will facilitate both the smooth running of the training and the learned embeddings getting desired performance in assessment tasks. Recognising the importance, benefit and standard of negative sampling, many methods have been proposed and many approaches have been tested. We survey the existing strategies and present them in the following schema.

4.1. Fixed distribution-based sampling

Negative sampling methods of this category are broadly used due to their simplicity and efficiency. However, the ignorance of changes over the negative sample distribution can easily result in the vanishing gradient problem and impede the model training.

4.1.1. Uniform sampling

Uniform sampling [5]is the earliest, easiest and most widely-used negative sampling method in KRL. It refers to constructing negative triples by replacing either the head h or the tail t of a positive triple with the entity randomly sampled from the entity set $\mathbb{E}$ according to uniform distribution. However, in most cases, the uniformly sampled entity is unrelated with the corrupted

positive triple, then the formed negative triple is too wrong to facilitate the training. Taking the triple (*London, locatedIn, UnitedKingdom*) as an example, its tail entity *UnitedKingdom* needs to be replaced to produce counterpart negative triples. Under the uniform sampling schema, the generated negatives could be (*London, locatedIn, apple*) or (*London, locatedIn, football*). These low-quality triples will be easily discriminated by the KRL model merely in terms of different entity types, which can slow down the convergence [42]. Similarly, IGAN emphasizes the zero loss problem in the random sampling mode, and explains the little contribution made by the low-quality negatives. Translation-based KRL models prefer adopting a marginal loss function with a fixed margin to distinguish positive triples from negative ones. Unreliable negatives tend to be out of the margin, which easily results in zero loss. Another severe drawback of uniform sampling lies in false-negative samples. After replacing the head in (*DonaldTrump, Gender, Male*) with *JoeBiden*, *(JoeBiden, Gender, Male)* is still a true fact (false negative).

4.1.2. Bernoulli sampling

To alleviate the false negatives problem, Bernoulli negative sampling [11] suggests replacing head or tail entities with different probabilities according to the mapping property of relations. That is, to give more chance of replacing the head in one-to-many relations and the tail in many-to-one relations. *Gender* is a typical many-to-one relation. Replacing the tail in (*DonaldTrump, Gender, Male*) with high probability unlikely cause false negative triples. If setting constraints on entity type, it may generate high-quality negatives. Zhang et al. [43] extends Bernoulli sampling by considering relation replacement following the probability $\alpha = r/(r + e)$, here r is the number of relations and e is the number of entities. The rest $1 - \alpha$ is divided by head entity replacement and tail entity replacement according Bernoulli distribution. Such changes enhance the ability of KRL models in relation link prediction.

4.1.3. Probabilistic sampling

Kanojia et al. [44]proposes probabilistic negative sampling to address the issue of skewed data that commonly exists in knowledge bases. For relations with less data, Uniform or Bernoulli random sampling fails to predict the missing part of golden triplets among semantically possible options even after hundreds of epochs of training. Probabilistic negative sampling speeds up the process of generating corrupted triplets by bringing in a tuning parameter β known as train bias that determines the probability by which the generated negative examples are complemented with early-listed possible instances. Kanojia et al. evaluates probabilistic negative sampling (PNS) over TransR in link prediction, and elicits that TransR-PNS achieves 190 and 47 position gains in Mean Rank on benchmark datasets WN18 and FB15K [5] respectively compared to TransR using Bernoulli sampling.

4.2. GAN-based sampling

GAN is short for Generative Adversarial Network [45]. In the GAN-based framework, the generator is responsible for providing negative samples and the discriminator is the target KRL model. Adversarial training is going on between the generator and the discriminator to optimize final knowledge representations. Reinforcement learning is required for training GAN [18]. The framework can be performed on various KRL models as it is independent of the specific form of the discriminator [16]. GAN is capable of modelling dynamic distribution, its generator has advantages in providing negative samples with better quality consistently. However, potential risks (training instability and model collapse) embodied in reinforcement learning should not be neglected.

4.2.1. KBGAN

KBGAN [15] is the first work to adapt GAN to negative sampling in KRL. It considers selecting one of two translational distance-based KRL models (DistMult [30], ComplEx [22]) as the negative sample generator and one of two semantic matching-based KRL models (TransE [5], TransD [13]) as the discriminator for adversarial training. The generator produces a probability distribution over a candidate set of negatives and selects the one with highest probability to feed into the discriminator. The discriminator minimizes the marginal loss between positive and negative samples to learn the final embedding vectors. KBGAN combines four Generator-Discriminator pairs that show better performance than baselines, which reflects the strength of the adversarial learning framework.

4.2.2. IGAN

Different from that of KBGAN [15]which considers probability-based, log-loss KRL models as the generator, IGAN [16] applied a two-layer fully-connected neural network as its generator to supply better quality negative samples. The discriminator is still the desired KRL model. The embedding vectors of the corrupted positive triple are fed into the neural network and followed by non-linear activation function ReLU. The softmax function is added after to calculate the probability distribution over the whole entity set $\mathbb{E}$ instead of a small candidate set in KBGAN. The quality of the formed negative is measured by the scoring function of the discriminator. IGAN can mine negative samples with relatively high quality during adversarial training but suffers from high computational complexity.

Comparison between GAN-based and self-adversarial sampling. Adversarial Contrastive Estimation (ACE) [46] introduces a general adversarial negative sampling framework for NCE that is commonly used in NLP. RotatE [47] thinks that such adversarial framework is difficult to optimize since it needs to train the discrete negative sample generator and the embedding model simultaneously, which costs a lot in computation. GAN-based sampling has no advantage in efficiency. In order to reduce the risk of training instability caused by reinforcement learning, both KBGAN and IGAN requires to be pre-trained, which gives rise to extra costs. Therefore, RotatE proposes a self-adversarial sampling method based on self-scoring function and avoids the requirement of reinforcement learning. Meanwhile, it outperforms KBGAN in link prediction.

4.3. Custom cluster-based sampling

Sampling from custom clusters means that the desired negative sample is selected from a handful of candidates rather than sampled from the whole entity set. For example, domain sampling [48] suggests to sample from the same domain, and affinity dependent sampling relies on the closeness of entities that are measured by cosine similarity. Two more sampling methods, TransE-SNS and NSCaching, are elaborated in this section. Reducing the sampling scope makes the target of negative sampling more clear, which gains efficiency. Because KGs grow rapidly and update frequently, the constant renewal of custom clusters is essential and skilled.

4.3.1. TransE-SNS

Qin et al. [17] puts forward entity similarity-based negative sampling (SNS) to mine valid negatives. Inspired by the observation that smaller distance between two entity vectors imply their higher similarity in the embedding space, the K-Means clustering algorithm [49] is used to divide all entities into a number of groups. An entity is uniformly sampled from the same cluster of the replaced head entity to complete the corrupted positive triple and when necessary, the tail entity is replaced in the same manner. The negatives generated in a such way should be highly

similar to the given positive triple. Adapting SNS to TransE (TransE-SNS) and then evaluating in link prediction and triple classification, demonstrates that SNS enhances the ability of TransE.

4.3.2. NSCaching

High-quality negative samples tend to get high plausibility measured by scoring functions. Motivated by the skewed score distribution of negative samples, Zhang et al. [18]attempts to only track helpful and rare negatives of high plausibility with cache. NSCaching can be considered to be in the same group of GAN-based methods since they all parametrize the dynamic distribution of negative samples. To be precise, NSCaching is a distilled version of GAN-based methods, because it has fewer parameters, it does not need to be trained through reinforcement learning, and it also avoids the model collapse problem brought by GAN. After storing the high-quality negative triples in cache, NSCaching uniformly samples from the cache and applies importance sampling to update it. With more concentrated sampling and more concise training, NSCaching performs better than GAN-based methods in terms of efficiency and effectiveness.

4.4. Other novel approaches

We find that there are some novel negative sampling methods that cannot be simply classified into the above three categories, such as confidence-aware negative sampling [20] and Markov chain Monte Carlo negative sampling [21].

4.4.1. NKRL

Since human knowledge is innumerable and changeable, bypassing crowdsourcing and manual efforts in building KGs is the mainstream. Noise and conflicts are inevitably involved due to the auto-construction, explosive growth and frequent updates of typical KGs. Xie et al. [19] initially proposes a novel confidence-aware KRL framework (CKRL), and Shan et al. [20]extends this idea to negative sampling in noisy KRL (NKRL). CKRL detects noises but applies uniform negative sampling that easily causes zero loss problems and false detection issues. NKRL proposes a confidence-aware negative sampling method to address these problems, and the concept of negative triple confidence it introduces is conducive to generate plausible negatives by measuring their quality. NKRL also modifies the triple quality function defined in CKRL with the aim of alleviating the false detection problems and improving noise detection ability. Both CKRL and NKRL are performed on translation-based KRL models, and NKRL outperforms CKRL in link prediction task.

4.4.2. MCNS

Yang et al. [21] creatively derives that a nice negative sampling distribution that should be positively but sub-linearly correlated to the positive sampling distribution, and raises Markov chain Monte Carlo negative sampling (MCNS). In the proposed SampledNCE framework, the depth first search (DFS) algorithm is applied to traverse the graph to obtain the Markov chain of the last node, from which negative samples are generated. MCNS uses the self-contrast approximation to estimate positive sampling distribution, and the Metropolis-Hastings algorithm [50] to speed up negative sampling. Embedding vectors are updated by minimizing the hinge loss after inputting the positive sample and the generated negative sample into the encoder of the framework. The importance of negative sampling is proved in the formula derivation. Experiments exhibit that MCNS performs better than all baselines in downstream tasks and wins in terms of efficiency. The proposal of MCNS is based on graph structure models without limitation to KRL, which is a generic solution.

5. CONCLUSIONS

In this short paper we have reviewed negative sampling in KRL. We sketched out existing well known negative sampling methods in three categories. We aimed to provide a basis for selecting the proper negative sampling method to train a KRL model to its best. The majority of KRL studies focus on defining new scoring functions to model multi-relational data in KGs, thus simply selecting the random mode for negative sampling. Nevertheless, as another momentous perspective of KRL, negative sampling is of the same significance with positive sampling. We hope that this review can be of some help to those who are interested in negative sampling. Subsequent work lies in comparing the methods mentioned here by performing link prediction on benchmark datasets. Besides, proposing a new strategy for negative sampling is a challenging attempt but is also under consideration.

ACKNOWLEDGEMENTS

This work is supported by the AI University Research Centre (AI-URC) of Xi'an Jiaotong-Liverpool University through XJTLU Key Programme Special Fund KSF-P-02 and KSF-A-17. We appreciate their support.

REFERENCES

[1] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, "Toward an Architecture for Never-Ending Language Learning," (in English), *Proceedings of the Twenty-Fourth Aaai Conference on Artificial Intelligence (Aaai-10),* pp. 1306-1313, 2010.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *SIGMOD Conference*, 2008.

[3] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *WWW '07*, 2007.

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," presented at the Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, Lake Tahoe, Nevada, 2013.

[5] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," presented at the Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, Lake Tahoe, Nevada, 2013.

[6] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," presented at the Proceedings of the 28th International Conference on International Conference on Machine Learning, Bellevue, Washington, USA, 2011.

[7] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 4710-4723: Association for Computational Linguistics.

[8] R. Wang, B. Li, S. Hu, W. Du, and M. Zhang, "Knowledge Graph Embedding via Graph Attenuated Attention Networks," *IEEE Access,* vol. 8, pp. 5212-5224, 2020.

[9] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics," *J. Mach. Learn. Res.,* vol. 13, no. 1, pp. 307–361, 2012.

[10] B. Kotnis and V. Nastase, "Analysis of the Impact of Negative Sampling on Link Prediction in Knowledge Graphs," 08/22 2017.

[11] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," presented at the Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, Québec, Canada, 2014.

[12] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning Entity and Relation Embeddings for Knowledge Graph Completion," in *AAAI*, 2015.

[13] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge Graph Embedding via Dynamic Mapping Matrix," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*

and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 2015, pp. 687-696: Association for Computational Linguistics.

[14] H. Xiao, M. Huang, and X. Zhu, "TransG : A Generative Model for Knowledge Graph Embedding," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, 2016, pp. 2316-2325: Association for Computational Linguistics.

[15] L. Cai and W. Y. Wang, "KBGAN: Adversarial Learning for Knowledge Graph Embeddings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, 2018, pp. 1470-1480: Association for Computational Linguistics.

[16] P. Wang, S. Li, and R. Pan, "Incorporating GAN for Negative Sampling in Knowledge Representation Learning," in *AAAI*, 2018.

[17] S. Qin, G. Rao, C. Bin, L. Chang, T. Gu, and W. Xuan, "Knowledge Graph Embedding Based on Adaptive Negative Sampling," Singapore, 2019, pp. 551-563: Springer Singapore.

[18] Y. Zhang, Q. Yao, Y. Shao, and L. Chen, "NSCaching: Simple and Efficient Negative Sampling for Knowledge Graph Embedding," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 2019, pp. 614-625.

[19] R. Xie, Z. Liu, and M. Sun, "Does William Shakespeare REALLY Write Hamlet? Knowledge Representation Learning with Confidence," *ArXiv,* vol. abs/1705.03202, 2018.

[20] Y. Shan, C. Bu, X. Liu, S. Ji, and L. Li, "Confidence-Aware Negative Sampling Method for Noisy Knowledge Graph Embedding," *2018 IEEE International Conference on Big Knowledge (ICBK),* pp. 33-40, 2018.

[21] Z. Yang, M. Ding, C. Zhou, H. Yang, J. Zhou, and J. Tang, "Understanding Negative Sampling in Graph Representation Learning," *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* 2020.

[22] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," presented at the Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, New York, NY, USA, 2016.

[23] S. Kazemi and D. Poole, "SimplE Embedding for Link Prediction in Knowledge Graphs," in *NeurIPS*, 2018.

[24] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge Graph Embedding Based Question Answering," *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining,* 2019.

[25] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge Graph Attention Network for Recommendation," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* 2019.

[26] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge Graph Embedding: A Survey of Approaches and Applications," *IEEE Transactions on Knowledge and Data Engineering,* vol. 29, no. 12, pp. 2724-2743, 2017.

[27] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition and Applications," *ArXiv,* vol. abs/2002.00388, 2020.

[28] L. Drumond, S. Rendle, and L. Schmidt-Thieme, "Predicting RDF triples in incomplete knowledge bases with tensor factorization," 03/26 2012.

[29] R. Reiter, "Deductive Question-Answering on Relational Data Bases," in *Logic and Data Bases*, H. Gallaire and J. Minker, Eds. Boston, MA: Springer US, 1978, pp. 149-177.

[30] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding Entities and Relations for Learning and Inference in Knowledge Bases," *CoRR,* vol. abs/1412.6575, 2015.

[31] X. Dong *et al.*, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 08/24 2014.

[32] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D Knowledge Graph Embeddings," *ArXiv,* vol. abs/1707.01476, 2018.

[33] L. Guo, Z. Sun, and W. Hu, "Learning to Exploit Long-term Relational Dependencies in Knowledge Graphs," in *ICML*, 2019.

[34] L. Yao, C. Mao, and Y. Luo, "KG-BERT: BERT for Knowledge Graph Completion," *ArXiv,* vol. abs/1909.03193, 2019.

[35] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling Relational Data with Graph Convolutional Networks," Cham, 2018, pp. 593-607: Springer International Publishing.

[36] S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo, "Semantically Smooth Knowledge Graph Embedding," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China, 2015, pp. 84-94: Association for Computational Linguistics.

[37] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling Relation Paths for Representation Learning of Knowledge Bases," *ArXiv,* vol. abs/1506.00379, 2015.

[38] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge Graph and Text Jointly Embedding," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1591-1601: Association for Computational Linguistics.

[39] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Knowledge Graph Embedding with Iterative Guidance from Soft Rules," *ArXiv,* vol. abs/1711.11231, 2018.

[40] Y. Lin, X. Han, R. Xie, Z. Liu, and M. Sun, "Knowledge Representation Learning: A Quantitative Review," *ArXiv,* vol. abs/1812.10901, 2018.

[41] Y. Bengio and J. Senecal, "Adaptive Importance Sampling to Accelerate Training of a Neural Probabilistic Language Model," *IEEE Transactions on Neural Networks,* vol. 19, no. 4, pp. 713-722, 2008.

[42] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815-823.

[43] Y. Zhang, W. Cao, and J. Liu, "A Novel Negative Sample Generating Method for Knowledge Graph Embedding," presented at the Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks, Beijing, China, 2019.

[44] V. Kanojia, H. Maeda, R. Togashi, and S. Fujita, "Enhancing Knowledge Graph Embedding with Probabilistic Negative Sampling," *Proceedings of the 26th International Conference on World Wide Web Companion,* 2017.

[45] I. J. Goodfellow *et al.*, "Generative adversarial nets," presented at the Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, Montreal, Canada, 2014.

[46] A. Bose, H. Ling, and Y. Cao, "Adversarial Contrastive Estimation," *ArXiv,* vol. abs/1805.03642, 2018.

[47] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space," *ArXiv,* vol. abs/1902.10197, 2019.

[48] Q. Xie, X. Ma, Z. Dai, and E. Hovy, "An Interpretable Knowledge Transfer Model for Knowledge Base Completion," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, 2017, pp. 950-962: Association for Computational Linguistics.

[49] J. Hartigan and M. C. Wong, "Statistical algorithms: algorithm AS 136: a K-means clustering algorithm," 1979.

[50] N. Metropolis, A. W. Rosenbluth, M. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics,* vol. 21, pp. 1087-1092, 1953.

EVALUATING DUTCH NAMED ENTITY RECOGNITION AND DE-IDENTIFICATION METHODS IN THE HUMAN RESOURCE DOMAIN

Chaïm van Toledo, Friso van Dijk and Marco Spruit

Utrecht University, Utrecht, the Netherlands

ABSTRACT

The human resource (HR) domain contains various types of privacy-sensitive textual data, such as e-mail correspondence and performance appraisal. Doing research on these documents brings several challenges, one of them anonymisation. In this paper, we evaluate the current Dutch text de-identification methods for the HR domain in three steps. First, by updating one of these methods with the latest named entity recognition (NER) models. The result is that the NER model based on the CoNLL 2002 corpus in combination with the BERTje transformer give the best combination for suppressing persons (recall 0.94) and locations (recall 0.82). For suppressing gender, DEDUCE is performing best (recall 0.53). Second NER evaluation is based on both strict de-identification of entities (a person must be suppressed as a person) and third evaluation on a loose sense of de-identification (no matter what how a person is suppressed, as long it is suppressed)

.

KEYWORDS

Named Entity Recognition, Dutch, NER, BERT, evaluation, de-identification.

1. INTRODUCTION

De-identification of texts has become a common task within the medical domain, for researching health care records and many other medical documents. The HR field also contains a lot of textual data, but in contrast to de medical domain we did not find any HR text de-identification tools. De-identification of texts provides benefits for organisations. First, the General Data Protection Regulation (GDPR) asks for limiting personal identifiers (PIDs) in processing data. Second, data scientists can work safer with pseudonymised texts, because the PIDs are suppressed. De-identifying methods will reduce the impact of data breaches. And third, organisations can temper employee privacy concerns by removing individual characteristics. Therefore, to realise these benefits, we investigate text de-identification in Dutch governmental HR e-mail correspondence.

This paper transfer existing Dutch medical text de-identification methods to the HR domain. The case organisation is a Dutch governmental HR organisation. The organisation provides employee payments and HR management. More than 100,000 civil servants rely on their systems. Civil servants could contact the organisation's contact centre by phone, e-mail, and chat. The contact centre gets questions like: "how to get tax reduction by buying a bike for commuting?"

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 239-249, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101520

The correspondence system saved more than 300,000 e-mails. Over 2,000 e-mails are manually annotated on eleven characteristics. Next, we trained NER models to recognise names, locations, and organisations. Then, different NER methods and de-identification methods are combined and compared with each other.

The research question of this paper is: To which extent can current de-identification methods anonymise Dutch HR related texts? Our main contributions are bringing de-identification to the HR domain, benchmarking current de-identification methods, updating NER models, and benchmarking NER methods.

The paper is structured as follows: The Background section elaborates on NER, de-identification and policy and PIDs. From this background, the expectation is that updating NER models with the so-called transformers should enhance de-identification results. About PIDs, none of the de-identification methods are specialised in supressing gender, job titles and regular titles. Because of the absence of recognising the classes, we expect that performance will be modest. In the Methods section, this paper elaborates how we update the NER methods and how we combine this with de-identification methods.

The Results section shows the performances of the NER and de-identification methods. The results are divided in three evaluations. The first one evaluates the performance of the current and the updated NER models. The second evaluation shows how the state-of-the-art de-identification methods suppress PIDs correctly. Third evaluation brings the result of to which extend the PIDs are suppressed, no matter how it is labelled by the methods.

After, we discuss the shortcomings of the current de-identification methods and what features future de-identification methods should have. We conclude that current Dutch NER methods can be improved through transformers and that with the TKS method and a state-of-the-art NER method the best results can be achieved in de-identifying HR texts.

2. BACKGROUND

This section overviews three elements of de-identification. The background starts with NER. NER recognise privacy sensitive elements in texts. Second part elaborates about the de-identification methods and the usage of NER systems in de-identification methods. Third and last part explain about the policies of de-identification.

2.1. NER for Dutch

A definition of NER is: “the task of automatically identifying names in text and classifying them into a pre-defined set of categories” [1, p. 1]. Automated text de-identification and NER share the same goal: recognise entities in texts [2]. Three main approaches can be distinguished [3]. The first approach is rule-based, this method contains name lists, regular expressions and other predefined handcrafted rules. Second is an unsupervised learning approach, with clustering, word groups can construct based on context similarity. Third approach is a supervised learning, with predefined labels on words in a corpus. The results are good in this approach, but the construction of corpora for NER is a time-consuming process.

For supervised learning NER, the conditional random fields (CRF) algorithm plays a major role. CRF takes neighbouring tokens into account and so context plays a role in labelling names in texts. Long short-term memory (LSTM) also impacts a major role in NER. LSTM is “capable of remembering information over long time periods during the processing of a sequence” [4, p. 1].

Recent developments show an upcoming rise of transformers, like BERT (Bidirectional Encoding Representations for Transformers), introduced by Google [5]. Test results show improvements in NER because of BERT.

Two known hand-annotated corpora are publicly available to create Dutch NER systems, namely CoNLL-2002 (Conll) and SoNaR-1 (Sonar). Conll has four labels for entity recognition: persons, locations, organisations and miscellaneous. The training corpus contains 218,737 lines of words [6]. The Sonar corpus got two extra labels: products and events. Sonar contains 1 million words. Where the Conll corpus uses news data, Sonar uses news items, manuals, autocues, fiction and reports and 'new' media like blogs, forums, chat and SMS [1].

For Dutch, there is a well-known NER system, namely FROG [7]. FROG detects persons, organisations, locations, products, events, and miscellaneous entities in texts. Another Dutch NER system (with English, Spanish and German) is created by Lample et al. [8], but unfortunately, the Dutch model isn't available online. Another attempt is Polyglot [9], based on Wikipedia and freebase data and brings a service with 40 language models. Performances for NER are mostly tested with the Conll corpora. Table 1 lists the NER systems with support for Dutch.

Table 1: NER systems with support for Dutch

Approach	**Test set**	**Language**	**Rates**
Classifying Wikipedia data [10]	CoNLL-2003	English	F_1-score 85.2
	CoNLL-2002	Dutch	F_1-score 78.6
	CoNLL-2003	German	F_1-score 66.5
	CoNLL-2002	Spanish	F_1-score 79.6
Single CRF classifier [1], [7]	SoNaR	Dutch	F_1-score 84.91
Discriminative Learning, word embeddings [9]	CoNLL-2003	English	F_1-score 71.3
	CoNLL-2002	Dutch	F_1-score 59.6
	CoNLL-2002	Spanish	F_1-score 63.0
LSTM-based [11]	CoNLL-2003	English	F_1-score 84.57
	CoNLL-2002	Dutch	F_1-score 78.08
	CoNLL-2003	German	F_1-score 72.08
	CoNLL-2002	Spanish	F_1-score 81.83
LSTM-CRF [8]	CoNLL-2003	English	F_1-score 90.94
	CoNLL-2002	Dutch	F_1-score 81.74
	CoNLL-2003	German	F_1-score 78.76
	CoNLL-2002	Spanish	F_1-score 85.75

2.2. Text De-Identification Methods

Because of the Health Insurance Portability and Accountability Act (HIPAA) regulations, text de-identification methods originate from the medical domain. These methods use a combination of rule-based and/or statistical approaches. Both approaches are used in the method of Tjong Kim Sang (TKS) et al. [12]. In this method, the user can add names in a list and can also find names with FROG NER. There are also non-statistical approaches like DEDUCE [13]. This rule-based method employs user-added lists of names and other sensitive entities. Table 2 gives an overview of the two different approaches in Dutch.

DEDUCE is created for de-identifying psychiatric nursing notes. The method requires an organisational context for defining important entities. The method brings the most popular names in the Netherlands, but it is recommendable to expanse this list with person names from the organisational systems. The same routine implies for institutions or organisations and locations.

TKS uses NER systems, like Frog, for tokenising the text document. The output is directly annotated in long lists of tokens with labels (entity or O). Like the case of DEDUCE, extra organisational context extends the method. The tokens are already annotated by the NER system or will be looked up in the context lists. At last, the tokens will be analysed with regular expressions for dates, (phone) numbers and e-mails.

Table 2: Overview of previous Dutch text anonymisation research.

Name	Approach	Corpus	Identifiers	Rates
DEDUCE [13]	Rule-based	Psychiatric nursing notes (2,000)	Person, url, institution or organisation, location, phone number, age, date	F_1-score 0.826
TKS [12]	Rule-based, CRF	Data from therapeutic sessions	*Frog:* Persons, locations, events, misc., products, organisations *Method:* numbers, months, days, numbers, month, days, mails, phone numbers	F_1-score0.84 unlabelled score, low micro average 0.55

2.3. Policy and PIDs

De-identified data contains no identifiable elements of individuals. When an element is identifiable to an individual differs from situation to situation, and is also referred to as an anonymity versus utility dilemma [14]. The medical world introduced many de-identifiers. It is helpful to see what criteria there are, because the GDPR does not have an exact specification about what PIDs are. The law describes that personal data: "any information relating to an identified or identifiable natural person" [15, p. 2]. The HIPAA provides eighteen possible identifiers [16].

De-identified data asks for policy. It would be dangerous to give an anonymised dataset to the public. For example, Narayanan and Shmatikov [17] demonstrate a de-anonymisation attack on the Netflix user dataset. Netflix de-identified user data for a recommender system competition, but the public release of this dataset, backfired to the corporation. Hence, three important policies should be considered when working with de-identified data [18]. First, control spread of the data. Second, prevent identification attempts. Third, provide security measurements.

3. METHODS

The methods explain how the evaluation is done. The first two sections explain how the NER is constructed and how the NER and the de-identification methods are evaluated. The third part elaborates about the metrics. The fourth part explains how the data is annotated and how the agreement is scored between the annotators.

3.1. NER Construction and Evaluation

As explained in Background, two Dutch NER applications are publicly available, Polyglot and FROG. We construct four NER models based on the Conll and Sonar corpora. The first two models on both the corpora are trained on the BERT Multilingual Cased (ML), where Dutch is included. The third and fourth models are trained on the two corpora with BERTje [19]. BERTje has specifically been created for the Dutch language. From both Conll and Sonar, we only test persons, locations and organisations. Both contain miscellaneous, but this entity is not annotated in the test dataset. Sonar also contains products and events, but these are also disregarded from

the test dataset. Software for constructing NER transformers based on BERT is derived from Raj's github [20].

3.2. De-Identification Evaluation

Based on the literature, this paper evaluates two de-identification methods, DEDUCE and the method of TKS. The method of TKS uses Frog, but other NERs can also be attached to this method. Therefore, the created NER models based on BERT, are also attached to the method of TKS.

Due to a lack of comparability between DEDUCE and TKS, this research chose not to evaluate DEDUCE's url, phone number and age retrieval performance. DEDUCE can detect somebody's age, or phone number, but there are also other forms of numbers. DEDUCE classifies e-mail and websites as URL, therefore it is difficult to distinguish them from each other. With the method of TKS, this research did not evaluate phone numbers, mails, events, miscellaneous and products, for the same reasons as with DEDUCE. In table 3, our annotation classification labels connects to the DEDUCE and TKS label.

Table 3: Comparison of entity classification identifiers.

Our classification labels	**DEDUCE**	**TKS**
Person	Person	Person
Organisations	Institution	Organisation
Location	Location	Location
Num	...	Num
Date	Date	Month, date, days numbers

3.3. Evaluation metrics

The test performance evaluation metrics are precision, recall and f1-score. Precision is measured by (Σ true positive) / (Σ true positive + Σ false positive). Recall is measured by (Σ true positive) / (Σ true positive + Σ false negative). F1-score is then measured by 2*((precision*recall)/(precision+recall)). This paper appoints recall as the most important metric, because the recall is measured with the false negatives, so the harm is higher when there occurs a false negative than when there occurs a false positive. F1-score is the second-most important metric, because of the combination of false negatives and false positives in the measurements.

Table 4: Generated examples of e-mail questions. Signatures are parsed out of the context

Goodmorning, Please handle the following. [SIGNATURE] Dear colleague, from the 9th of October 2016, I am seconded to the Apeldoorn office of the Tax and Customs Administration, but in the P-Portal my old Ministry of Finance e-mail (j.doe@minfin.nl) is still connected to my account. I would like my new e-mail (john.doe@belastingdienst.nl) to be connected to P-Direct. My employee number is 98706540. Thanks in advantage. Greetings John. [SIGNATURE]
Hello, with permission from the board, I want to change my commuting fee. During long road construction, my commuting distance to the PI Amsterdam increased from 35 km to 41 km. This situation is happening since the 5th of November past year, so I'd like, retrospectively, to get a higher commuting fee since the 5th of November. Sincerely, [SIGNATURE]

3.4. Annotated Data

E-mail data is used from a Dutch government organisation. This e-mail data concerns all kinds of HR related issues. Most of these correspondences relate to HR administration or to modification to personnel files. The dataset contains 2,017 e-mails, with a mean of 133 tokens per e-mail. Table 4 shows two examples of messages. Annotations were made in these e-mails, in total 13,496. Software for making annotations is AnnotatorJS and a SQL database to store these annotations. There is a mean of 6.69 annotations per e-mail, with a standard deviation of 11.21.

Table 5 contains the different PIDs. *Date* is everything what points to a date, i.e. month year, or day. *Weekdays*, like Sunday and Monday, are excluded; this turned out too generic. *Numbers* are defined as a sequence of two or more digits, meaning that 1 or 2 are excluded, but 10 and 222 are not excluded. One digit turned out to be too generic. *Persons* (per) include names, surnames, and initials. This research also includes *Gender*, because gender related words can be seen as a binary. With these words and other identifiers an attacker can easily deduce people. Gender words are like sir, madam (or in Dutch: de heer, mevrouw), but also he, she, son, his. We define *Organisation* (org) as groups of people larger than one person. The *E-mail* (mail) section was difficult, because some e-mails contain a whitespace, this is strictly impossible, probably written by mistake. *Location* (loc) is everything what refers to a physical place, such as a street, postal code, municipal, country, area, region and so forth. *Job title* refers to somebody's job, like policy officer or contact center agent. *Title* includes degrees such as MSc, Dr or Duke. *Code* is anything what refers to an account, like IBAN numbers, usernames and passwords. *Websites* can also reveal the organisation a person is working at, so this research annotates this information as well.

Table 5: Counted PIDs in the dataset

PID	Date	Num	Per	Gender	Org	Mail	Loc	Job title	Code	Title	Website
Num	3628	3338	3086	1567	1011	279	225	120	116	96	28

The first annotator labelled the entire sample of 2,017 e-mails. A second annotator labelled 283 e-mails, 14% of the sample. For measuring the correctness of the labelled representations, an interrater reliability is used with a kappa statistic. According to [21], a kappa score of at least 0.80 is sufficient, a kappa below 0.60 indicates a non-agreement between annotators. A kappa statistic is measured as follow:

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

Pr(a) is the observed agreement between the annotators and Pr(e) is the expected change agreement. The observed agreement is 0.99 and the expected agreement is 0.82. The kappa score is 0.92 and we conclude there is a high agreement between the annotators.

4. RESULTS

The results are distinguished in three evaluations. First, how well do the NER models perform? Second, how do the de-identification methods perform in a "strict" sense? A strict sense means that the suppression both identifies and classifies the entity correctly. The third and last evaluation shows how the de-identification methods perform in a "loose" sense. A loose sense only takes the identification of the suppression into account and not the classification.

All three evaluations are important, because a good NER will recognise entities without organisational knowledge. A high strict de-identification ensures that the utility will not drop too much, because only the sensitive entities are suppressed, and the false positives are low as possible. A loose sense de-identification evaluation is important because suppression is the key to anonymisation.

4.1. NER Evaluation

Table 6 shows the results of two existing NER models, namely Polyglot and FROG, against trained BERT-based models. A remarkable outcome is the fact that the Conll corpus gives highest results with respect to recall. As the creators of Sonar explained, Sonar has a more diverse corpus than Conll. Because of this diversity, the expectation was that Sonar could better recognise entities than Conll. The precision of Sonar BERT ML gives in almost all cases the best results.

Table 6: NER results (underline is highest result in row)

	Metrics	**Polyglot**	**Frog**	**Conll BERT ML**	**Sonar BERT ML**	**Sonar BERTje**	**ConllBERTje**
Per	Precision	0.69	0.68	0.69	0.81	0.77	0.72
	Recall	0.35	0.80	0.86	0.86	0.83	0.88
	F1	0.46	0.73	0.77	0.83	0.80	0.80
Org	Precision	0.66	0.38	0.47	0.71	0.63	0.59
	Recall	0.16	0.46	0.66	0.65	0.54	0.69
	F1	0.26	0.42	0.55	0.68	0.58	0.64
Loc	Precision	0.10	0.09	0.26	0.40	0.34	0.41
	Recall	0.36	0.49	0.64	0.60	0.57	0.65
	F1	0.16	0.15	0.37	0.48	0.42	0.50

Table 7: Strict de-identification results (underline is highest result in row)

	Metrics	**DEDUCE**	**TKS + Frog**	**TKS + Sonar BERT ML**	**TKS + Conll BERT ML**	**TKS + Sonar BERTje**	**TKS + ConllBERTje**
Per	Precision	0.42	0.66	0.64	0.57	0.61	0.59
	Recall	0.77	0.86	0.91	0.87	0.88	0.92
	F1	0.55	0.74	0.75	0.69	0.72	0.72
Org	Precision	0.06	0.28	0.56	0.25	0.51	0.47
	Recall	0.00	0.45	0.68	0.49	0.59	0.68
	F1	0.00	0.35	0.61	0.33	0.55	0.56
Loc	Precision	0.70	0.16	0.23	0.20	0.21	0.26
	Recall	0.26	0.38	0.32	0.47	0.31	0.40
	F1	0.38	0.22	0.26	0.28	0.25	0.31
Date	Precision	0.71	0.97	0.98	0.98	0.97	0.97
	Recall	0.65	0.90	0.94	0.90	0.94	0.94
	F1	0.68	0.96	0.96	0.94	0.96	0.96
Num	Precision	...	0.62	0.62	0.62	0.62	0.62
	Recall	...	0.92	0.93	0.91	0.93	0.93
	F1	...	0.74	0.74	0.74	0.74	0.74

4.2. Strict De-Identification

Table 7 shows the results for strict de-identification. This means that for example a person is identified as an organisation, the person identifier gets a false negative. The BERT-based transformers perform better in almost all NER related fields. It is in the line of expectations that the method of TKS would perform the best regarding the recall in combination with the ConllBERTje model. The NER results section shows that this model also performs best.

The precision can be lower compared to the NER results. The reason for this decrease is the added organisational data. This added data has a positive influence on the recall, because more entities are recognised, but the downside is an overfitting and that influences the precision. Or the problem in other words: sometimes names can be regular words.

4.3. Loose Sense De-Identification

The focus in this section is only whether something is suppressed or not. Loose sense de-identification is when an identifier is, suppressed, a true positive is seen and when an identifier is not suppressed, a false negative is seen. False positives are not considered, because not all methods recognise gender, codes, titles and job titles. A false positive can also be a true positive in another identifier class.

The highest results of table 8 are TKS with the Conll corpus and BERTje for training. The combination preforms best on persons and locations. For persons the BERT based methods are preforming slightly better than Frog. The same is happening with locations and organisations.
Interesting is DEDUCE, as it accounts for gender and title more often. This happens because DEDUCE looks for so-called prefixes at names. These prefixes are like sir, prof, dr and so forth. DEDUCE is not build for recognising words as her/his or son/daughter, so the results remain at a recall of 0.49. However, for recognising titles, it does a good job, with a recall of 0.83. The prefix list was incomplete and should be enlarged with more titles (like nobiliary and accountancy titles).

E-mails are not well recognised; this is due to the tokenisation of the NER methods. The example sentence: “My e-mail is johndoe@example.com” will be tokenised as follows: “My”, “e-mail”, “is”, “johndoe”, “@”, “uu.nl”. The TKS method wants a full e-mail as token and simply selects the e-mail based on its @-sign. The DEDUCE method also contains errors for particularly e-mails, for example when somebody writes their e-mail in capitals or partly in capitals.

Table 8: De-identification results (underline is highest result in row)

Entities	**Metrics**	**DEDUCE**	**TKS + Frog**	**TKS + Sonar BERT ML**	**TKS + Conll BERT ML**	**TKS + Sonar BERTje**	**TKS + ConllBERTje**
Person	Recall	0.79	0.9	0.93	0.89	0.93	0.94
Org	Recall	0.00	0.75	0.81	0.76	0.77	0.78
Location	Recall	0.38	0.76	0.75	0.77	0.76	0.82
Date	Recall	0.61	0.92	0.96	0.92	0.97	0.97
Number	Recall	0.30	0.88	0.89	0.87	0.88	0.89
Gender	Recall	0.49	0.11	0.10	0.16	0.10	0.16
E-mail	Recall	0.53	0.44	0.40	0.40	0.49	0.48
Code	Recall	0.19	0.48	0.40	0.48	0.49	0.60

Title	Recall	0.83	0.13	0.14	0.13	0.15	0.18
Job title	Recall	0.11	0.52	0.49	0.45	0.49	0.51
Website	Recall	0.08	0.08	0.08	0.08	0.00	0.00

5. DISCUSSION

De-identification is a difficult task, as which entities are revealing an individual, and which are not, differs per situation. This research aimed to be as strict as possible in annotating identifiers. Unlike other studies, we also took job titles, titles and gender into account. Of course, it is arguable that not every identifier is as important as the other. Revealing a person's name holds more privacy risk than revealing a person's gender. But a combination of generic identifiers can lead to a high enough specificity that allows for the identification of individuals

None of the methods had a hundred percent score compared to the annotated set and this means that using these methods will not de-identify everybody. Hence, when using a selected method, a researcher should always check data for false negatives. Besides the de-identification techniques, it is arguable to embrace text de-identification workflows in a research organisation. With handles to identify which PIDs are important to suppress.

The BERT transformers had in almost all cases a positive influence on enhancing the identification of the person, organisation, and location entities. A particularly interesting potential future improvement in NER method performance could be to focus more on recognising gender in texts. NER models must in the future handle lower cased names because texts are and will always be noisy and we cannot rely that all writing persons will write correctly. Although we underline that updating and expanding NER corpora is a time-consuming process.

We gathered organisational data, like name and organisation lists for feeding DEDUCE and TKS, however, in both methods this turned out insufficient to reach near-perfect performance. For organisational entities the score of DEDUCE was too low. It is arguable that we didn't gather enough organisational data to fill the organisation list. On the other hand, organisational data is never enough. To illustrate, a person can mention his/her partner's name in an e-mail, but a partner's name doesn't have to be saved in the organisation system. So, when requesting all the existing names from the organisation for the names list, a partner's name in an e-mail doesn't have to be matched and thus there can occur a false negative. Hence, in our result, the NER methods could de-identify at least 75 percent, so applying NER always seems at least a good starting point for de-identifying texts.

The major feature of TKS is that it is modular concerning NER applications. We could easily connect BERT-based NER methods to the de-identification method. We recommend that future text de-identification methods should follow this path and consider easy implementations of other entity recognition methods in a pipeline overview. Thereby, it is recommendable to add the possibility to let a user add de-identification methods to the overall de-identification method. The text de-identification task differs per situation.

6. CONCLUSIONS

There are no strict rules for what should be left out of a text and what should not. Every word in a text could lead to revealing a natural person. We tried to be strict as possible with our annotations and annotate everything what could lead to identifying a person. However, none of the methods aim to include every PID we identified, like gender or job titles.

The loose sense de-identification section shows that the rule-based approaches can detect a person's gender-based salutation but has more trouble to identify a personal name when it is not in the list. The statistical approaches with NER are way better in the situation when a person's name is not in the provided list.

This paper evaluated Dutch NER models and de-identification methods. None of the methods achieve a near-perfect performance. Updating NER models with transformers had a positive influence in all de-identification approaches. This paper showed that the Dutch Conll corpus gives the best results regarding recall performance in combination with the Dutch pretrained transformer BERTje for de-identifying Dutch HR text data.

REFERENCES

[1] B. Desmet and V. Hoste, 'Fine-grained Dutch named entity recognition', Lang. Resour. Eval., vol. 48, no. 2, pp. 307–343, Jun. 2014, doi: 10.1007/s10579-013-9255-y.

[2] B. Wellner et al., 'Rapidly Retargetable Approaches to De-identification in Medical Records', J. Am. Med. Inform. Assoc., vol. 14, no. 5, pp. 564–573, Sep. 2007, doi: 10.1197/jamia.M2435.

[3] J. Li, A. Sun, J. Han, and C. Li, 'A Survey on Deep Learning for Named Entity Recognition', IEEE Trans. Knowl. Data Eng., Mar. 2020, Accessed: Oct. 19, 2020. [Online]. Available: http://arxiv.org/abs/1812.09449.

[4] J. Hammerton, 'Named entity recognition with long short-term memory', in Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, Edmonton, Canada, 2003, pp. 172–175, doi: 10.3115/1119176.1119202.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, Jun. 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[6] E. F. Tjong Kim Sang, 'Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition', 2002, [Online]. Available: https://www.aclweb.org/anthology/W02-2024.

[7] A. van den Bosch, G. J. Busser, W. Daelemans, and S. Canisius, 'An efficient memory-based morphosyntactic tagger and parser for Dutch', Sel. Pap. 17th Comput. Linguist. Neth. Meet. Leuven Belg., pp. 99–114, 2007.

[8] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, 'Neural Architectures for Named Entity Recognition', in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, Jun. 2016, pp. 260–270, doi: 10.18653/v1/N16-1030.

[9] R. Al-Rfou, V. Kulkarni, B. Perozzi, and S. Skiena, 'Polyglot-NER: Massive multilingual named entity recognition', in Proceedings of the 2015 SIAM International Conference on Data Mining, 2015, pp. 586–594, Accessed: Apr. 09, 2019. [Online].

[10] J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran, 'Learning multilingual named entity recognition from Wikipedia', Artif. Intell., vol. 194, pp. 151–175, Jan. 2013, doi: 10.1016/j.artint.2012.03.006.

[11] D. Gillick, C. Brunk, O. Vinyals, and A. Subramanya, 'Multilingual Language Processing From Bytes', in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, Jun. 2016, pp. 1296–1306, doi: 10.18653/v1/N16-1155.

[12] E. F. Tjong Kim Sang, B. de Vries, W. Smink, B. Veldkamp, G. Westerhof, and A. Sools, 'De-identification of Dutch Medical Text', in 2nd Healthcare Text Analytics Conference, Cardiff, Wales, UK, 2019, p. 4.

[13] V. Menger, F. Scheepers, L. M. van Wijk, and M. Spruit, 'DEDUCE: A pattern matching method for automatic de-identification of Dutch medical text', Telemat. Inform., vol. 35, no. 4, pp. 727–736, 2018, doi: 10.1016/j.tele.2017.08.002.

[14] C. van Toledo and M. Spruit, 'Adopting Privacy Regulations in a Data Warehouse A Case of the Anonimity versus Utility Dilemma', in Conference: 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016), Porto, Portugal, Nov. 2016, pp. 67–72.
[15] M. Hintze and K. El Emam, 'Comparing the benefits of pseudonymisation and anonymisation under the GDPR', J. Data Prot. Priv., vol. 2, no. 2, pp. 145–158, 2018.
[16] UC Berkeley, 'UC Berkeley Committee for Protection of Human Subjects', HIPAA PHI: List of 18 Identifiers and Definition of PHI, 2018. https://cphs.berkeley.edu/hipaa/hipaa18.html (accessed Oct. 09, 2018).
[17] A. Narayanan and V. Shmatikov, 'Robust De-anonymization of Large Sparse Datasets', in 2008 IEEE Symposium on Security and Privacy (sp 2008), Oakland, CA, USA, May 2008, pp. 111–125, doi: 10.1109/SP.2008.33.
[18] M. Hintze, 'Viewing the GDPR through a De-Identification Lens: A Tool for Clarification and Compliance', Int. Data Prot. Law, vol. 8, no. 1, pp. 86–101, 2018, doi: 10.2139/ssrn.2909121.
[19] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim, 'BERTje: A Dutch BERT Model', ArXiv191209582 Cs, Dec. 2019, Accessed: Mar. 16, 2020. [Online]. Available: http://arxiv.org/abs/1912.09582.
[20] K. Raj, kamalkraj/BERT-NER. 2019.
[21] M. L. McHugh, 'Interrater reliability: the kappa statistic', Biochem. Medica, vol. 22, no. 3, pp. 276–282, 2012, doi: 10.11613/BM.2012.031.

AUTHORS

Chaïm van Toledo is a PhD candidate in the applied question and answering systems of the Department of Information and Computing Sciences at Utrecht University. As a member of the Applied Data Science Lab, he focuses on transforming organisation data to practical datasets for data science and intelligent systems. His main research objective is to enhance and broaden question and answering systems in the Dutch language.

Friso van Dijk is a PhD researcher in Privacy Governance at the Department of Information and Computing Sciences at Utrecht University. As a member of the Applied Data Science Lab he works with advanced data science techniques to enhance traditional organizational research methods. His primary research interests are in the development of practical tools for navigating privacy decision-making in the organizational context.

Dr. Marco Spruit is an Associate Professor in the Natural Language Processing research group of the Department of Information and Computing Sciences at Utrecht University. As principle investigator in the department's Applied Data Science Lab, his research team primarily works on Self-Service Data Science. Marco's research objective for the coming years is to establish and lead an authoritative national infrastructure for Dutch natural language processing and machine learning to facilitate and popularise self-service data science.

Parallel Data Extraction Using Word Embeddings

Pintu Lohar and Andy Way

ADAPT Centre, Dublin City University, Ireland

Abstract. Building a robust MT system requires a sufficiently large parallel corpus to be available as training data. In this paper, we propose to automatically extract parallel sentences from comparable corpora without using any MT system or even any parallel corpus at all. Instead, we use crosslingual information retrieval (CLIR), average word embeddings, text similarity and a bilingual dictionary, thus saving a significant amount of time and effort as no MT system is involved in this process. We conduct experiments on two different kinds of data: (i) formal texts from news domain, and (ii) user-generated content (UGC) from hotel reviews. The automatically extracted sentence pairs are then added to the already available parallel training data and the extended translation models are built from the concatenated data sets. Finally, we compare the performance of our new extended models against the baseline models built from the available data. The experimental evaluation reveals that our proposed approach is capable of improving the translation outputs for both the formal texts and UGC.

Keywords: Machine Translation, parallel data, user-generated content, word embeddings, text similarity, comparable corpora

1 Introduction

A parallel corpus is the main ingredient for building an MT system. Usually, there are two ways of parallel corpus acquisition, namely: (i) manual development, and (ii) automatic extraction. Although manual development is ideal and is produced in most cases by human translators, this process requires a huge amount of time and effort which is considered to be less practical than automatic extraction of parallel data for MT. One of the easiest ways to accomplish this task is to employ an MT system that translates all the source-language texts into the target language and then performs text similarity in the target language. However, using an MT system is not always the best solution mainly due to the following reasons: (i) it requires a significant amount of time to build the MT system itself, especially if this is an NMT system, (ii) it also takes a long time to translate all the source-language documents into the target language especially for large corpora, and (iii) MT systems for all domains and language pairs are not available. These problems demonstrate that finding a suitable alternative to using an MT system for parallel data extraction is an important aim. In this work, we propose to combine the CLIR, text similarity and word embedding-based approach for extracting parallel sentences from the comparable corpora for both formal texts and UGC, without the help of any MT system or any parallel corpus, thereby saving a significant amount of time

David C. Wyld et al. (Eds): ACITY, DPPR, VLSI, WeST, DSA, CNDC, IoTE, AIAA, NLPTA - 2020
pp. 251-267, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101521

and effort. We use the We use the *Euronews* corpus and hotel reviews (discussed in detail in Section 3) as the comparable corpora for parallel data extraction. We conduct our experiments on English and French texts from these corpora. We consider French as the source- and English as the target language in our experiments. As the CLIR-based searching works at document level, we represent each sentence as a document. Initially, we use the CLIR- component of *FaDA* [17] to index all the source and target language documents and then find a set of suitable candidate target-language documents for each source-language document. Afterwards, we translate[1] all the content words (i.e, after removing stopwords) of the French documents using a French-to-English dictionary.[2] Each of the extracted candidate English documents is then compared with the French document using the average word embeddings of the content words of each English document and that of the English translations of the words in the French document. The word embedding-based similarity is also accompanied by text similarity. The English document with the highest similarity score is selected as the parallel counterpart of the French document.

The remainder of this paper is organised as follows. In Section 2, we discuss some of the existing relevant works in this field. The description of the data sets we use in this work is provided in Section 3. In Section 5, we describe the experimental setup which is followed by the results obtained in Section 6. We perform output analysis in Section 7. Finally, we conclude our work and point out some future possibilities in Section 8.

2 Related work

The extraction of parallel sentences/segments plays an important role in improving MT quality [22, 13]. In general, the issue of parallel data extraction is addressed in different ways. For example, [16] propose a crowdsourcing approach for extracting parallel data from tweets. They attempt to find the translations in tweets instead of translating the texts. [8] extract both parallel sentences and fragments from comparable corpora of Chinese–Japanese Wikipedia to improve statistical MT. [12] apply a domain-biased parallel data collection and a structured methodology to obtain English–Hindi parallel data. Deep learning has gained popularity in this task [5, 11] recently. Many work exploits MT for parallel data extraction [7, 19]. As the alternative resources to parallel data, the comparable corpora are considered as valuable resources for MT. For example, [1] use a multimodal comparable corpus

[1] Note that this is merely a word-to-word translation, not a generic MT

[2] The dictionary is available at: `www.seas.upenn.edu/~nlp/resources/TACL-data-release/dictionaries.tar.gz`

of audio and texts built from 'Euronews'[3] and 'TED'[4] web sites for parallel data extraction. [14] propose a bidirectional method to extract parallel sentences from English and Persian document-aligned Wikipedia. They use two MT systems to translate from Persian to English and the reverse after which an IR system is used for measuring the similarity of the translated sentences. Although many parallel data extraction systems employ MT, it is not always a good idea and so we simply discard the requirement of any MT system and any parallel data at all.

3 Data set

We use two different types of data sets in our experiments: (i) formal text corpora from news domain, and (ii) UGC corpora of reviews.

3.1 Formal text corpora from news domain

The formal text corpora consist of the *Euronews* and the *News commentary* corpus.

- **Euronews corpus:** The *Euronews* corpus [2] is a multimodal corpus of comparable documents and their images. In our experiments, we consider only the documents and not the images as this is beyond the scope of this work. Each document in *Euronews* corpus consists of at least one line of text and many of them contain multiple-line texts with multiple sentences.
- **News commentary corpus:** This data set is comprised of the English–French parallel sentence pairs from the 'News-Commentary' corpus.[5] We refer to this data set as *NewsComm* in short.

Data set	Language	# Documents	# Sentences
Euronews	English	40,421	644,226
	French	37,293	614,928
NewsComm	English	/	246,946
	French	/	246,946

Table 1: Data statistics

Table 1 shows the statistics of the *Euronews* and the *NewsComm* data. We already mentioned earlier in Section 1 that we split each document into multiple sentences in this work. We can see in the above table that in the *Euronews* data, $644K$ English and $614K$ French sentences are obtained from $40K$ English and $37K$ French

[3] https://www.euronews.com/
[4] https://www.ted.com/
[5] http://www.casmacat.eu/corpus/news-commentary.html

documents, respectively. Note that the *NewsComm* data set is simply a parallel corpus at sentence level, not any document level, and so the third column entries are replaced by the '/' character which means 'not applicable' in this case.

3.2 UGC corpora of reviews

- **FourSquare parallel corpus:** This data set contains over $11K$ reviews (or $18K$ sentences) from the French–English parallel corpus of Foursquare restaurant reviews[6] [3]. The reviews were originally written in French, which were then translated into English by the professional translators. The authors also provide the official training, development and test splits for this data set.
- **Hotel review corpus:** The *Hotel_Review* corpus[7] consists of $878K$ reviews from $4,333$ hotels crawled from *TripAdvisor*. Although most of the reviews are in English, some of them are also written in French. Table 2 shows randomly selected three example reviews (two English and one French) from this data set. We highlight the special characters such as newlines, unicodes in red.

Examples	Reviews
1	I stayed at the Hudson Hotel in June and it was awful!!\nStandard Rooms (rate USD 299) are extremely small and the superior ones (USD 359) are tiny as well. \nStaff is not friendly, room wasn\u00b4t ready till 3 p.m.\nEv. in this hotel is very dark (black passages and floor) - you don\u00b4t even have to be claustrophobic to feel you are living your most awful nightmare.
2	Excellent coffee for customers, friendly staff, very good beds and clean rooms! Poor windows because all possible city- and traffic noise from the street hammered your ears. I would use this hotel again though. Sohotel is renovated with style and taste - respecting the history of the building.\nI.S. J\u00e4ms\u00e4nkoski, Finland
3	Cet h\u00f4tel est tr\u00e8s bien situ\u00e9, juste \u00e0 cot\u00e9 de la plage, il est bien entretenu et la literie est de qualit\u00e9. Il propose un petit d\u00e9jeuner relativement copieux, ce qui est pas le cas de tous les h\u00f4tels de LA. Le parking est s\u00e9curis\u00e9. \nPar contre, il est assez mal insonoris\u00e9, et nous avons entendu de bruit de la rue tr\u00e8s tot le matin.

Table 2: Review examples

Note that the newline characters are not always explicitly present even if a new sentence starts. For instance, in example 2, there are no newline characters before the sentences such as '*Poor windows....*' and '*I would use this....*'. In addition, a plenty of unicode characters are present in the hexcode format such as '00b4', '00e9', '00e8' etc. most of which are present in the French review in example 3. Considering these observations, we preprocess the data using the following steps.

[6] https://europe.naverlabs.com/research/natural-language-processing/machine-translation-of-restaurant-reviews/

[7] https://www.cs.cmu.edu/~jiweil/html/hotel-review.html

(i) **Language detection:** We perform language detection[8] in order to detect and extract the English and French reviews from this data set.

(ii) **Sentence splitting:** As our parallel data extraction system is implemented at sentence level, we split the multi-sentence reviews into different parts (sentence) and consider each part as a single document.

(iii) **Unicode conversion:** We convert[9] the characters given in unicode format into the Latin characters. For example, the character '00f4' is converted into 'ô'.

Table 3 shows an original French review (example 3 of Table 2) and its preprocessed version. We highlight all the unicodes in the original review in red and the converted characters in the preprocessed review in blue. Note that 4 sentences are generated from this single review after preprocessing.

Original review	Preprocessed review
Cet h\u00f4tel est tr\u00e8s bien situ\u00e9, juste \u00e0 cot\u00e9 de la plage, il est bien entretenu et la literie est de qualit\u00e9. Il propose un petit d\u00e9jeuner relativement copieux, ce qui est pas le cas de tous les h\u00f4tels de LA. Le parking est s\u00e9curis\u00e9.\nPar contre, il est assez mal insonoris\u00e9, et nous avons entendu de bruit de la rue tr\u00e8s tot le matin.	**Sentence 1:** Cet hôtel est très bien situé, juste à coté de la plage, il est bien entretenu et la literie est de qualité. **Sentence 2:** Il propose un petit déjeuner relativement copieux, ce qui est pas le cas de tous les hôtels de LA. **Sentence 3:** Le parking est sécurisé. **Sentence 4:** Par contre, il est assez mal insonorisé, et nous avons entendu de bruit de la rue très tot le matin.

Table 3: An example review before and after preprocessing

The statistics of the *FourSquare* and *Hotel_Review* data sets is shown in Table 4.

Data set	# Reviews	# Total sentences	# training	# Dev	# Test
FourSquare	11,551	17,945	14,864	1,243	1,838
Hotel_Review	878,561	/	/	/	/

Table 4: Statistics of the FourSquare parallel and the Hotel review data sets

[8] `https://pypi.org/project/langdetect/`

[9] Unicode representation of these characters can be found at: `http://www.fileformat.info/info/unicode/char/search.htm`

4 System description

Our proposed system is composed of the following components: (i) CLIR-based system, (ii) sentence length-based pruning, (iii) average word embeddings, (iv) text similarity, and (v) score combination.

4.1 CLIR-based system

The CLIR component used in this experiment is a part of the open source bilingual document alignment tool *FaDA* [17]. It works in the following steps:

(i) firstly, the source-language and the target-language documents are indexed,

(ii) each source-language document is used to construct a pseudo-query[10] which is considered as the suitably representative of the document,

(iii) all pseudo-query terms are translated into the target-language by a bilingual dictionary and the translated query terms are then searched in the target-language index, and finally

(iv) the *top-n*[11] target-language documents are retrieved.

4.2 Sentence length-based pruning

Prior to performing the word embedding- and the text-based similarities between the source- and the target-language sentences, we exclude some of the comparisons depending upon the sentence-length ratio. This ratio is calculated in terms of the total number of words in the word translations of the source-language document (sentence) and the total number of words in the target-language document (sentence). We set the threshold for this ratio to 0.5, which means that the shorter of the document pair must be at least the half of the longer document in terms of the total number of words they contain. For example, if a French document contains 5 words and an English document contains 20 words, the ratio is 0.25 which is less than the threshold of 0.5. This document pair, therefore, according to our criteria is less likely to be parallel and so is not considered for comparison. The French document must contain at least 10 words to pass this threshold in order to be considered for further similarity measurements. However, 0.5 is not an empirically determined threshold; we choose this value so that very unlikely candidates can be removed from the comparison, albeit some of the invalid pairs still pass the threshold.

In general, the average length ratio of English texts over the French translations is near 1.0 [6] but there are many examples that violate this. For example, consider the English sentence '*I like to propose a toast.*' that contains 6 words and

[10] A *pseudo-query* is the modified form a user's original query in order to improve the ranking of retrieval results compared to the original query.

[11] We use the default value of n used in *FaDA*, where $n = 10$, which means the top 10 candidate target-language documents are retrieved.

its equivalent French translation '*J'aime proposer un toast*' that contains 4 words. The sentence-length ratio in this case is below 0.7 which is far less than 1.0. Therefore, setting a high threshold very close to 1.0 can result in discarding many valid sentence pairs like this one.

4.3 Text similarity

We calculate the text similarity using the following steps:

(i) firstly, we remove all the stopwords from both the French and English documents.

(ii) secondly, we translate the remaining content words of the French document into English using a French–English bilingual dictionary.

(iii) some of the word translations contain stopwords such as *to*, *of* etc. We remove these stopwords.

(iv) finally, we calculate the total number of word matches between the words in the English document and the word-level English translation of the French document.

4.4 Average word vector similarity

Consider the Figure 1 that shows a collection of words represented in a two dimensional space. We can observe that the semantically equivalent words are placed in close proximity. For example, the words *electrical*, *electricity*, *electric* etc. are closely grouped together in the same region. However, this figure shows the simplest representation of how the related words are treated. In reality, the words are represented as a vector of real values in much higher dimensions. In pre-trained word embeddings, the semantically related words usually contain similar vector values.

We now discuss how the average word vectors are actually calculated. Let us consider a sentence S with a sequence of n words: $w_1, w_2, w_3.....w_n$. Let the vector embeddings of the words be $u_{w_1}, u_{w_2}, u_{w_3}.....u_{w_n}$. The average word embedding of S is calculated using the Equation (1) as follows:

$$U_s = \frac{1}{n} \sum_{i=1}^{n} u_{w_n} \tag{1}$$

In our experiments, we use the '*fasttext*' pre-trained Wiki word vectors for English which is made available by [4]. In order to obtain the word embeddings for our experiments, we apply the following steps:

(i) All the stopwords in both the word translations of the French document and the English document are removed,

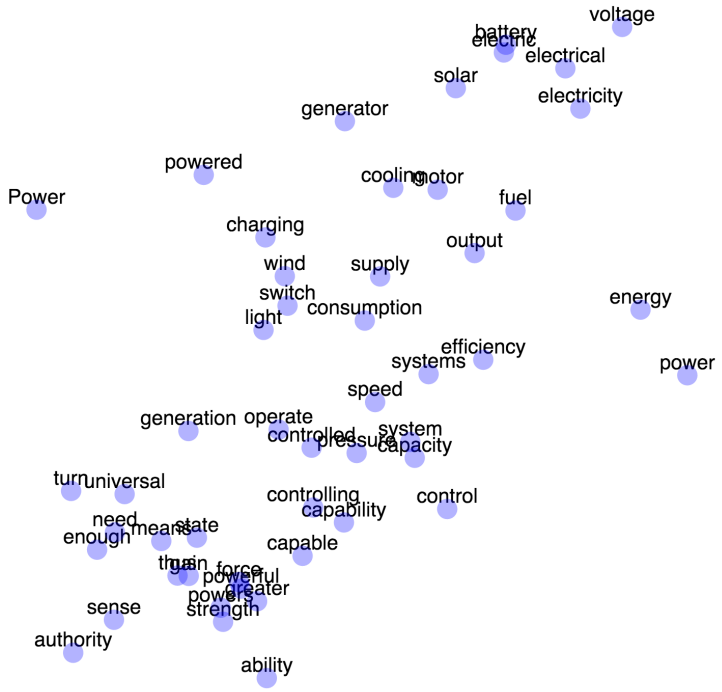


Fig. 1: Example of semantically related words in two dimensional space

(ii) the real word vector values of all the remaining words in the word translations of the French document are retrieved and then the average of all these vector values is calculated,

(iii) the average word vector values for the English document is calculated in a similar manner, and

(iv) the two averages are compared in order to calculate the average word vector similarity.

4.5 Score combination

Once we calculate the text and the average word vector similarities, these scores are then combined to obtain the overall similarity score. The overall similarity score S_{sim} is calculated using the Equation (2) as follows:

$$S_{sim} = w_1 WV_{sim} + w_2 Text_{sim} \tag{2}$$

In the above equation, WV_{sim} and $Text_{sim}$ are the average word vector and the text similarity scores with w_1 and w_2 weight values, respectively.

5 Experiments

5.1 MT configuration

The MT models are built using the freely available open source NMT toolkit 'OpenNMT'[12] [15]. We consider French as the source and English as the target language. In our experiments, we use all the default parameter settings: *RNN* as the default type of encoder and decoder, $word\_vec\_size = 500$, $rnn\_size = 500$, $rnn\_type = LSTM$, $global\_attention\_function = softmax$, $save\_checkpoint\_steps = 5000$, $training\_steps = 100,000$ etc. We evaluate the translation quality using BLEU [18].

5.2 Sentence-level document alignment

We store each sentence of the *Euronews* corpus in a single document which results in creating more than $600K$ documents per language. These documents are then fed as input to the CLIR component of *FaDA*. Once the top n English documents are obtained for a French document, we aim to find its closest semantically equivalent English document. It is, therefore, expected that the total number of extracted sentence pairs is over $600K$. However, it is impractical to consider all these sentence pairs as parallel data because many of them are not semantically equivalent. We, therefore, extract only those pairs that have the similarity score greater than a threshold (discussed in detail in Section 5.4). Table 5 shows the data size of the existing parallel corpora and the extracted sentence pairs from the *Euronews* and *Hotel_Review* data sets.

Text type	**Data set**	**# Sentence**
Formal text	NewsComm parallel	246,946
	Extracted sentence pairs (Euronews)	31,860
UGC text	FourSquare parallel	14,864
	Extracted sentence pairs (Hotel_Review)	6,188

Table 5: Existing parallel corpora vs extracted sentence pairs

5.3 Translation models

Note that we show data combinations for two different types of data sets: (i) formal text, and (ii) UGC text. We built a baseline translation model and an extended translation model for each of the above types of texts.

[12] `https://github.com/OpenNMT/OpenNMT-py`

Models for formal text corpora The extracted parallel sentences from the *Euronews* corpus are used as the additional data set for MT training for formal texts. We build two translation models: one is the baseline model and another is the extended model. The baseline model is built using only the *NewsComm* data whereas the extended model is built using the concatenated data. We held out 1,000 sentence pairs for development and another 1,000 sentence pairs for tuning purposes from the *NewsComm* data. We refer to this baseline model as $Base_{FT}$ and the extended model as Ext_{FT}, where 'FT' stands for 'formal text'. Table 6 shows the data distribution. Each translation model is tuned and tested on the same development and test data sets, respectively.

Model	**Data set**	**# training**	**# Dev**	**# Test**
$Base_{FT}$	News	226,946	1,000	1,000
Ext_{FT}	News + Euronews	253,592	1,000	1,000

Table 6: Data distribution for two different MT models for formal text corpora

Models for UGC corpora Once the sentence pairs are extracted from the *Hotel_Review* data set, we consider them as the additional parallel resource and concatenate with the parallel training sentences of the *FourSquare* corpus. We build following MT models: (i) a baseline model, which is built from the 14,864 parallel training sentences of the *FourSquare* corpus, and (ii) an extended model, which is built from the concatenation of the *FourSquare* data and the sentence pairs extracted from the *Hotel_Review* data set. The baseline model is referred to as '$Base_{UGC}$' and the extended model is referred to as 'Ext_{UGC}'. Table 7 shows the data distribution. Both translation models are tuned and tested on the same development and test data sets, respectively.

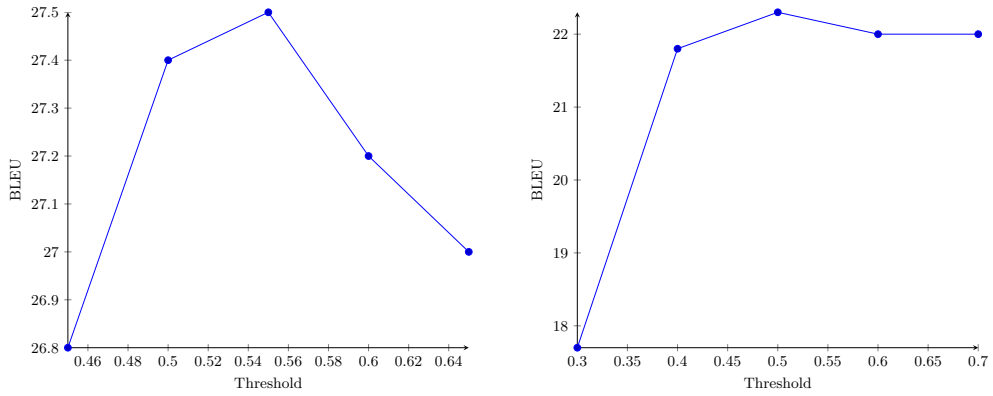
Model	**Data set**	**# training**	**# Dev**	**# Test**
$Base_{UGC}$	FourSquare	14,864	1,243	1,838
Ext_{UGC}	FourSquare + Hotel review	21,052	1,243	1,838

Table 7: Data distribution for two different MT models for UGC text corpora

5.4 System tuning

As we discussed earlier in Section 4, we calculate the overall similarity score of a sentence pair using Equation (2). However, it is required to obtain a threshold

for the the similarity score above which all the sentence pair can be considered as parallel sentences. We explored different threshold values for both the *Euronews* and the *Hotel_Review* data sets.



(a) Formal text: *Euronews* + *NewsComm* (b) UGC text: *Hotel_Review* + *FourSquare*

Fig. 2: Tuning threshold value with BLEU scores

We extract a set of parallel sentence pairs for each threshold value. Aftwerwards, each set is added to the existing parallel resources to build extended translation models. For example, if we set the threshold for similarity score to 0.5, all the sentence pairs whose similarity score are higher than 0.5 can be considered as parallel sentences and would be added to the existing training data. Using this method, different sets of such concatenated data are obtained using different threshold values. We then build different translation models using each data set separately. The model for which we obtain the highest BLEU score is considered as the extended model and the corresponding threshold is considered as the optimal threshold. Figure 2a and 2b show the BLEU score comparison with different threshold values used for extracting sentence pairs from the *Euronews* and *Hotel_Review* data sets, respectively and adding them to the parallel sentences from *NewsComm* and *FourSquare* data sets, respectively. Note that the former combination belongs to formal text and the later one belongs to UGC text. It is obvious from Figure 2a that the BLEU score decreases as the threshold is reduced or increased from 0.55

for the formal text corpus. The highest BLEU score of 27.5 is obtained at this threshold. In contrast, we can observe from Figure 2b that the highest BLEU score of 22.3 is obtained for the UGC text corpus using the similarity threshold of 0.5. We, therefore, set the optimal similarity thresholds for the formal and the UGC text corpora to 0.55 and 0.5, respectively.

6 Results

We show the BLEU scores obtained by the Baselines and the *Extended* models for both the formal text and UGC corpora in Table 8. Note that the formal text corpus is comprised of the parallel sentences from *NewsComm* and extracted sentence pairs from *Euronews* data, whereas the UGC text corpus consists of the parallel sentences from *FourSquare* and extracted sentence pairs from *Hotel_Review* corpus.

Corpus type	**Translation model**	**BLEU score**
Formal text	$Base_{FT}$	27.1
	Ext_{FT}	**27.5**
UGC text	$Base_{UGC}$	22.1
	Ext_{UGC}	**22.3**

Table 8: BLEU score comparison

For formal text corpus, we can observe in Table 8 that the addition of parallel sentences extracted from the *Euronews* corpus using our proposed system improves the BLEU score, i.e. the Baseline ($Base_{FT}$) is outperformed by the extended model (Ext_{FT}) by 0.4 BLEU points. On the comparison, we notice a slight improvement in BLEU score for UGC text corpus, i.e. the Baseline ($Base_{UGC}$) is outperformed by the extended model (Ext_{UGC}) by 0.2 BLEU points. We also perform the statistical significance test of the outputs using MultEval [9]. However, we found that these improvements in BLEU score are not statistically significant as $p > 0.1$.

We notice that the improvement in BLEU score for UGC text corpus (i.e. 0.2 points) is less than that for the formal text corpus (i.e. 0.4 points). One probable reason for this degradation is that the *Euronews* data actually contains some parallel texts. Therefore, extracting and adding them to the existing *NewsComm* parallel training data helps improve the BLEU score to some extent. In contrast, the *Hotel_Review* data set does not contain parallel texts. In fact, the reviews are generated randomly by different users without any translation usage in mind. However, there exists some texts that are very close in meaning even though they are not parallel. Due to this reason, such partially semantically similar texts help improve the BLEU score very slightly.

7 Output Analysis

The improvement in BLEU score shows that our automatic parallel data extraction system helps improve MT quality by supplying additional training data. However, this is the beginning phase of our experiment and further plans are made to extend this work. As of now, we illustrate some example outputs where the *Baseline* models are outperformed by our *Extended* models.

Example	Reference	Baseline model	Extended model
1	But, equally important, workers organized themselves to defend their interests.	But, as important, workers have been organized to defend their interests.	But, equally important, workers have been organized to defend their interests.
2	Overall, however, the inequality gaps are large and, in many cases, growing.	Overall, however, the inequality gap remains acute and in some cases even expansion.	Overall, however, the inequality gap remains deep, and in some cases it expands.
3	Countries that import currently subsidized food will be worse off.	Countries that imports currently will suffer.	Countries that import products currently subsidized will suffer.
4	He ate chocolate and watched NBA games.	He ate chocolate and watched from the NBA games.	He ate chocolate and watched the NBA games.

Table 9: Example outputs: Baseline vs Extended model (formal text corpora)

Let us first show some example outputs produced by the translation models built from the formal text corpora (*NewsComm* and *Euronews*) in Table 9 and explain how the *Baseline* model is outperformed by the *Extended* model. In example 1, the word '*equally*' is missing in the *Baseline* output. The second example shows that the ending phrase '*in some cases even expansion*' of the output produced by the *Baseline* model is grammatically incorrect whereas the *Extended* model produces the phrase '*in some cases it expands*' which is grammatically correct and semantically equivalent to the phrase '*in many cases, growing*' in the reference translation.

In example 3, both translation outputs are erroneous but the output produced by the *Extended* model is better as it includes the word '*products*' which although is not equivalent to the word '*food*' in the reference translation but at least conveys a little bit of similar meaning. Finally, example 4 shows the case where both translation outputs are mostly correct except the presence of extra prepositions. The phrase '*watched the NBA games*' that is produced by the *Extended* model is better than the phrase '*watched from the NBA games*' produced by the *Baseline* when compared with the reference translation.

Table 10 illustrates some example outputs produced by the translation models

built from the UGC text corpora (*FourSquare* and *Hotel_Review* data sets) and shows how the *Baseline* model is outperformed by the *Extended* model.

Example	Reference	Baseline model	Extended model
1	Cozy little teahouse, amazing sweets and teas.	Disgusting room, very good cakes and teas.	Small tea room, very good cakes and teas.
2	A nice atmosphere to hang out with friends.	Friendly atmosphere for a relaxed dinner.	Friendly atmosphere for a walk with friends.
3	The sales assistants are super friendly.	The sales assistants are really welcoming.	The sales assistants are super welcoming.
4	Their famous hot chocolate, one of the best in the world, is worth the wait!	Its suggestion hot chocolate, one of the best in the world is worth the wait!	Its legendary chocolate, one of the best in the world is worth the wait!
5	They do really good burgers.	They serve very good burgers.	They do very good burgers.

Table 10: Example outputs: Baseline vs Extended model (UGC text corpora)

We can notice in the table that although the phrase '*Small tea room*' (in example 1) produced by the *Extended* model is not a proper translation, it is still much better than the completely wrong translation output '*Disgusting room*' produced by the *Baseline* model. In example 2, the phrase '*hang out with friends*' in the reference translation is semantically closer to the phrase '*walk with friends*' (produced by the *Extended* model) than to the phrase '*relaxed dinner*' (produced by the *Baseline* model). Moreover, '*super friendly*' is more synonymous to '*super welcoming*' than '*really welcoming*' in example 3. Furthermore, the word '*suggestion*' (see example 4) is completely meaningless when used before '*hot chocolate*' that is produced by the *Baseline* model. In contrast, although '*legendary chocolate*' is not a proper translation (produced by the *Extended* model), it is partially similar to '*famous hot chocolate*' in the reference. Finally, both of the translation outputs in example 5 are sensible but the output produced by the *Extended* model is closer to the reference.

8 Conclusions and Future work

In this paper, we proposed a parallel data extraction technique from comparable corpora of both formal texts and UGC in order to generate additional parallel training data for MT. Many research works employ MT itself to ease this task. However, it is not always a practical solution because in addition to building the MT system in the first place, it also requires a huge amount of time to translate all the source-language documents of the comparable corpus into the target-language in order to be able to perform the text similarity in the target language. To overcome this situation, we implemented a parallel data extraction system without any help from MT or even any parallel corpus. We initially used the CLIR component of *FaDA* tool

to extract the candidate target-language sentences for a source-language sentence. We then used the average word-embeddings and text similarity with the help of a bilingual dictionary in order to obtain parallel sentences from the *Euronews* and *FourSquare* corpus. These extracted sentence pairs were then concatenated with the existing parallel training data to build the extended translation models which outperformed the baseline systems that are built from only the existing parallel training data.

We noticed that extracting parallel texts from the *Euronews* corpus obtains a slightly higher BLEU score improvement than for the *Hotel_Review* data set. One probable reason is that the *Euronews* data actually contains some parallel texts and so extracting and adding them to the existing *NewsComm* parallel training data helps improve the BLEU score to some extent. In contrast, the hotel reviews are extremely unlikely to contain parallel texts as they are randomly generated by different users without translation usage foreseen. Although not being strictly parallel, some of them are semantically equivalent, and adding them as extra training data improves the BLEU score very slightly over the *Baseline* model. It is, therefore, expected that the BLEU score can be improved further if there exists a considerable amount of parallel texts in a comparable corpus of UGC. As we did not use any MT system or any parallel corpus for this task, our proposed system is very simple and can be easily applied to a large comparable corpus. Our findings in this research are encouraging as our system relies on only the text similarity, word embeddings and a bilingual dictionary, for which the required resources are easily available online. We believe that our proposed model has the potential to benefit further research in this field.

One of the drawbacks in our approach is that we have not compared our system with some of the most popular existing sentence alignment systems. Some examples of well-known works in this field are [20], [10] and [21]. In future, we would like to explore these approaches and apply their sentence alignment systems on the data sets we used in this work. Our main objective is to combine the best performing system with our system. Another possibility is to apply all of them separately and select the sentence alignments that are common outputs generated by all or most of these alignment systems. In addition, we also plan to apply our system to other types of UGC such as tweets, customer feedback, movie reviews etc.

Acknowledgements

This research has been supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106).

Bibliography

[1] Haithem Afli, Loïc Barrault, and Holger Schwenk. Building and using multimodal comparable corpora for machine translation. *Natural Language Engineering*, 22, 11 2015.

[2] Haithem Afli, Pintu Lohar, and Andy Way. MultiNews: A web collection of an aligned multimodal and multilingual corpus. In *Proceedings of the First Workshop on Curation and Applications of Parallel and Comparable Corpora*, pages 11–15, Taipei, Taiwan, November 2017.

[3] Alexandre Berard, Ioan Calapodescu, Marc Dymetman, Claude Roux, Jean-Luc Meunier, and Vassilina Nikoulina. Machine translation of restaurant reviews: New corpus for domain adaptation and robustness. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 168–176, Hong Kong, 2019.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[5] Houda Bouamor and Hassan Sajjad. H2@bucc18: Parallel sentence extraction from comparable corpora using multilingual sentence embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, Miyazaki, Japan, 2018.

[6] Aitao Chen. Cross-language retrieval experiments at clef 2002. *Advances in Cross-Language Information Retrieval*, pages 28–48, 2003.

[7] Chenhui Chu. *Integrated Parallel Data Extraction from Comparable Corpora for Statistical Machine Translation*. PhD dissertation, Kyoto University, 2015.

[8] Chenhui Chu, Toshiaki Nakazawa, and Sadao Kurohashi. Integrated parallel sentence and fragment extraction from comparable corpora: A case study on chinese–japanese wikipedia. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 15(2), 2015.

[9] Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies:*, pages 176–181, Portland, Oregon, USA, 2011.

[10] Luís Gomes and Gabriel Pereira Lopes. First steps towards coverage-based sentence alignment. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2228–2231, Portorož, Slovenia, May 2016.

[11] Francis Grégoire and Philippe Langlais. Extracting parallel sentences with bidirectional recurrent neural networks to improve machine translation. In

Proceedings of the 27th International Conference on Computational Linguistics, pages 1442–1453, Santa Fe, New Mexico, USA, 2018.

[12] Deepa Gupta, Vani Raveendran, and Rahul Yadav. Domain biased bilingual parallel data extraction and its sentence level alignment for english-hindi pair. *Research Journal of Applied Sciences, Engineering and Technology*, 7:1001–1012, 02 2014.

[13] Viktor Hangya and Alexander Fraser. Unsupervised parallel sentence extraction with parallel segment detection helps machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1224–1234, Florence, Italy, 2019.

[14] Akbar Karimi, Ebrahim Ansari, and Bahram Sadeghi Bigham. Extracting an English-Persian parallel corpus from comparable corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 1–5, Miyazaki, Japan, May 2018.

[15] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, 2017.

[16] Wang Ling, Luís Marujo, Chris Dyer, Alan W. Black, and Isabel Trancoso. Crowdsourcing high-quality parallel data extraction from twitter. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 426–436, Baltimore, Maryland, USA, June 2014.

[17] Pintu Lohar, Debasis Ganguly, Haithem Afli, Andy Way, and Gareth J. F. Jones. Fada: Fast document aligner using word embedding. *The Prague Bulletin of Mathematical Linguistics*, 106:169–179, 2016.

[18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002.

[19] Dana Ruiter. Online parallel data extraction with neural machine translation. Masters thesis, Saarland University, 2019.

[20] Rico Sennrich and Martin Volk. MT-based sentence alignment for OCR-generated parallel texts. In *The Ninth Conference of the Association for Machine Translation in the Americas*, pages 1–11, Denver, Colorado, USA, 2010.

[21] Brian Thompson and Philipp Koehn. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1342–1348, Hong Kong, 2019.

[22] Krzysztof Wolk, Emilia Rejmund, and Krzysztof Marasek. Multi-domain machine translation enhancements by parallel data extraction from comparable corpora. *Computing Research Repository*, abs/1603.06785, 2016.

AUTHOR INDEX