Lecture Notes in Electrical Engineering 869

Ripon Patgiri
Sivaji Bandyopadhyay
Malaya Dutta Borah
Valentina Emilia Balas *Editors*

Edge Analytics

Select Proceedings of 26th International Conference—ADCOM 2020

Springer

Lecture Notes in Electrical Engineering

Volume 869

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering - quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada:

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries:

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

More information about this series at https://link.springer.com/bookseries/7818

Ripon Patgiri · Sivaji Bandyopadhyay ·
Malaya Dutta Borah · Valentina Emilia Balas
Editors

Edge Analytics

Select Proceedings of 26th International Conference—ADCOM 2020

Springer

Editors
Ripon Patgiri
Department of Computer Science
and Engineering
National Institute of Technology Silchar
Silchar, Assam, India

Malaya Dutta Borah
Department of Computer Science
and Engineering
National Institute of Technology Silchar
Silchar, Assam, India

Sivaji Bandyopadhyay
National Institute of Technology Silchar
Silchar, Assam, India

Valentina Emilia Balas
Aurel Vlaicu University of Arad
Arad, Romania

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-19-0018-1 ISBN 978-981-19-0019-8 (eBook)
https://doi.org/10.1007/978-981-19-0019-8

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Contents

About the Editors

Ripon Patgiri is an Assistant Professor at the Department of Computer Science and Engineering, National Institute of Technology, Silchar. He received his Bachelor's Degree from the Institution of Electronics and Telecommunication Engineers, New Delhi, in 2009, M.Tech. degree from the Indian Institute of Technology Guwahati in 2012 and a Ph.D. degree from the National Institute of Technology (NIT), Silchar, in 2019. Dr. Patgiri has numerous papers published in reputed journals, conferences, and books. His research interests include bloom filters, networking, security, privacy, secrecy, and communication. He is a member of several professional bodies such as IEEE, ACCS (India), ACM, EAI, and IETE. He is Area Editor of the EAI Endorsed Transactions on the Internet of Things. Dr. Patgiri serves as Editor of several conference proceedings.

Sivaji Bandyopadhyay is the Director of NIT Silchar since December 2017. He is a Professor of the Department of Computer Science and Engineering, Jadavpur University, India, where he has been serving since 1989. He is also a Professor in the Department of Computer Science and Engineering at NIT Silchar. Dr. Bandhyopadhyay has over 300 journal publications and 2 books to his credit. His research interests are natural language processing, machine translation, sentiment analysis, and medical imaging. He has organized several conferences and has been Program Committee Member and Area Chair in several reputed conferences. He has completed international funded projects with France, Japan, and Mexico. At the national level, he has been the Principal Investigator of several consortium mode projects in machine translation, cross-lingual information access, and treebank development. He is presently the Principal Investigator of an Indo-German SPARC project with the University of Saarlandes, Germany, on Multimodal Machine Translation and the Co-PI of several other international projects.

Malaya Dutta Borah is an Assistant Professor at the Department of Computer Science and Engineering, NIT Silchar. She received her Ph.D. degree from Delhi Technological University. She has published several papers in reputed journals, conferences, and books. She has rich experience in conducting conferences and workshops.

Valentina Emilia Balas is currently a Full Professor in the Department of Automatics and Applied Software at the Faculty of Engineering, "Aurel Vlaicu" University of Arad, Romania. She holds a Ph.D. Cum Laude, in Applied Electronics and Telecommunications from the Polytechnic University of Timisoara. Dr. Balas authored over 350 research papers in refereed journals and International Conferences. Her research interests are in intelligent systems, fuzzy control, soft computing, smart sensors, information fusion, modeling, and simulation. She is the Editor-in-Chief to *International Journal of Advanced Intelligence Paradigms* (IJAIP) and to *International Journal of Computational Systems Engineering* (IJCSysE), Member of Editorial Board Member of several national and international journals, and is evaluator expert for national, international projects, and Ph.D. thesis.

An IoT-Based Intelligent Irrigation Management System

Mansi Sahi and Nitin Auluck

Abstract Recently, rainfall and climate change have been observed to be erratic. Hence, proper irrigation of fields has become extremely important. Inappropriate management of irrigation leads to poor quality and quantity of crops. If we irrigate the fields properly, the yield can increase significantly. This paper proposes a smart irrigation system using artificial intelligence, ICT, embedded systems and IoT. The proposed system takes into consideration external factors like soil condition and climatic conditions before suggesting any action. Our experiments show that the performance of the proposed method is quite encouraging. The accuracy recorded for the trained model while testing is 75%. This helps in using water resources efficiently and thus reducing water wastage significantly.

Keywords Smart farming · Artificial intelligence · Embedded systems · Machine learning · Data analytics · Wi-Fi

1 Introduction

Populations across the world are highly dependent on agriculture, and India is no exception. This is a major source of income for 70% of the Indian population and accounts for 27% of the GDP [4]. The population is increasing rapidly, but the agricultural land area remains constant. Thus, it is very important to increase the yield of the land in order to satisfy the growing needs. To a large extent, the agricultural productivity depends on the proper irrigation of fields. There is a need to manage the irrigation proactively depending on various external conditions such as air humidity, temperature and moisture content in the soil.

M. Sahi (✉) · N. Auluck
Department of Computer Science and Engineering, Indian Institute of Technology Ropar, Rupnagar, India
e-mail: 2017csz0010@iitrpr.ac.in

N. Auluck
e-mail: nitin@iitrpr.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_1

Automation in agriculture is the need of the hour. However, irrigating fields at regular intervals is not a good idea as the climatic conditions change drastically. It is better to irrigate the fields according to given climatic and soil conditions. The average yield can be increased by maintaining the desired soil condition using an automated irrigation system. Using these systems not only reduces the cost of production, but also improves crop quality. There are several places that are drought prone. So, judicious use of natural resources like water is required. Worldwide, about 85% of the fresh water is used in agriculture [16]. This fresh water consumption will continue to grow in the near future, as the population is increasing rapidly and so is the demand for food. In addition to increasing yield, these automated systems using information and communications technology (ICT) also help in optimizing fresh water usage. It is believed that by using IoT and AI technologies, the fresh water supply processes can be subsequently impacted globally. These automated systems proposed in [3, 7, 8] report significant water saving in comparison with traditional irrigation systems.

This paper combines embedded systems and artificial intelligence in building a smart irrigation system which can not only increase crop yield, but also use water resources efficiently.

2 Related Work

A framework proposed in [19] uses a wireless sensor network and a web interface in order to control and monitor fields remotely. Authors in [13] have proposed an automated system in which the farmer is intimated using GSM. The irrigation is controlled by a micro-controller-based system and a soil sensor. A drip irrigation system has been proposed in [17], which uses a sensor-based IoT solution. However, the field information cannot be accessed without the Internet. In [5], a system has been proposed by using Zigbee and a wireless sensor network that collects environmental parameters. Based on these parameters, an SMS is sent to the client's mobile. Photovoltaic panels have been used in [8] to provide power, transmitting data to the web and triggering actuators. In addition, a cellular Internet interface has been used to provide communication and scheduling irrigation. Communication in the proposed wireless network has been enabled using Bluetooth technology in [2]. When the moisture in the soil hits below some threshold, then a message is sent to a motor using Bluetooth. An automated system for drip irrigation has been proposed in [1]. In this system, an email is sent using a Python script running on a Raspberry Pi, which then sends a signal to an Arduino system for turning on the pump. In order to power up the system in [9], the authors have used solar panels. The moisture content is measured in the soil by using electrodes. The energy is supplied to motors by solar panels to turn ON/OFF based on the threshold value. The greenhouse activities are remotely controlled in [12] by using a GSM module, a micro-controller, actuators and sensors. Authors in [6] have proposed a GPS-based smart robot which can be controlled remotely to perform activities like spraying, monitoring and sensing

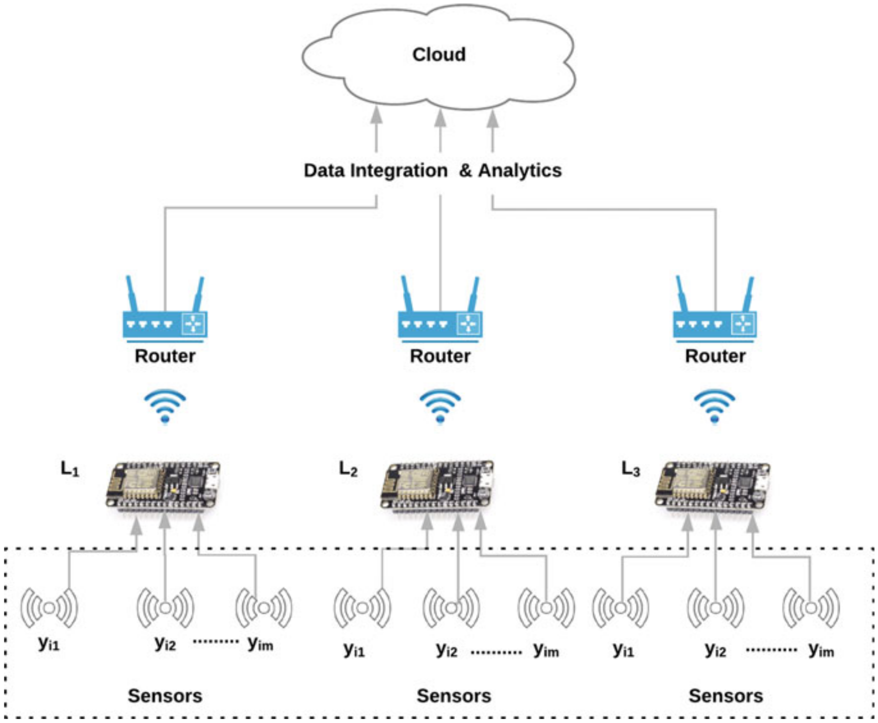


Fig. 1 Overall architecture of IoT-based irrigation management system

moisture. In [14], the authors have used Ethernet IEEE 802.3 for monitoring and controlling environmental conditions in an online farm. All the above systems are embedded, using various communicating protocols. However, they fail to provide the 'bigger picture' which can be obtained by analysing the data deeply. The water quantity is controlled by using a GPRS module and a micro-controller in the automated irrigation system proposed in [8]. The collaboration of artificial intelligence and embedded systems has proved to be beneficial in many areas of agriculture [10]. In [11], the authors use image processing in order to monitor fruit diseases. Authors in [18] have developed a monitoring system after analysing the previous harvest data and statistics. This is done in order to improve the decision-making efficiency of selecting a crop based on the environment. A greenhouse management system has been proposed in [15]. The data is acquired, and optimal environmental conditions are maintained accordingly.

3 Proposed Algorithms

The proposed smart irrigation architecture is shown in Fig. 1. The bottom-most layer comprises end devices like sensors and actuators. These end devices are connected with the fog device. The fog device senses the data sent by attached sensors. The sensed data from the fog devices is further pushed to the cloud for storage, computation and analysis. The data which is stored in the cloud is used for training the model. The model which is trained will be used for taking irrigation decisions in future. The dataset has 'n' instances and 'm' sensor inputs collected from various locations 'L'. The input matrix can be represented as follows:

$$Y = \begin{bmatrix} 1 & y_{11} & \dots & y_{1m} \\ 1 & y_{21} & \dots & y_{2m} \\ \vdots & \vdots & & \vdots \\ 1 & y_{n1} & \dots & y_{nm} \end{bmatrix} \tag{1}$$

Here, y_{vn} depicts value of the nth sensor input for vth instance and $y_{i0} = 1\,\forall i \in (0, 1, \dots, n)$.

The prognostic function for the training dataset D_{tr} can be computed using following equation:

$$f = \mu^T.Y \tag{2}$$

Here, μ represents the weight vector $[\mu_0, \mu_1, \dots, \mu_m]$. Now, μ_0, μ_1 and so on are the weights for each sensor input. The regression weights are initially assigned random values (Table 1).

The value of the prognostic function is calculated by performing matrix multiplication of sensor inputs (Y) with their corresponding initial weights (μ) represented by Eq. (2). The prognostic function for the jth instance of sensor inputs is calculated using the following equation:

$$f(y_j) = \mu_0 + \mu_1 y_{j1} + \cdots + \mu_m y_{jm} \tag{3}$$

Then, sigmoid function is applied upon the prognostic function. The sigmoid function has a 'squashing' behaviour, i.e. $0 \leq g(x) \leq 1$. When the value of x lies between -4 and 4, it implies linear behaviour, but when the value of x is above 5 and below -5, then it gives a maximum value of 1 and a minimum value of 0, respectively. This is done in order to prevent the model from being prone to outliers. The sigmoid function is represented using the following equation:

$$g(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{4}$$

After inserting the prognostic function, i.e. f in the sigmoid function, the equation may be represented as:

Table 1 Notations

μ	Vector of weights
μ_{op}	Vector of optimized weights
Y	Matrix of combined sensor data
y_{oi}	ith value in output vector of training data
g	Sigmoid function
f	Prognostic function
μ_0	Bias term
μ_i	Weight for ith sensor input
Pos	Correct prediction by model
Neg	Incorrect prediction by model
Acc	Accuracy of model
y_i	Vector of sensor data for ith instance
G_i	$gof(y_i)$ composition function
E_μ	Estimation function
P_μ	Penalty function
α	Learning rate

$$g(f_j) = \frac{1}{1 + \mathrm{e}^{\mu_0 + \sum_{i=1}^{m} \mu_i y_{ji}}} \tag{5}$$

Here, $f_j = f(y_j)$. Depending on the value of $g(f_j)$, the actuation $O(f_j)$ is performed as follows:

$$O(f_j) = \begin{cases} 1, & \text{if } g(f_j) \geq 0.5. \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Here, $O(f_j) = 1$ implies turning the pump ON and $O(f_j) = 0$ implies that no action is required.

The problem that we solve can be formulated as follows: given a cdc 'C' with the set of 'n' training instances gathered from 'L_i' locations (where 'i' depicts the location number), each having 'm' sensor inputs. The task is to obtain optimized parameters (weight vector) 'μ_{op}' that maximizes the test accuracy $= \frac{\text{pos}}{\text{pos} + \text{neg}} * 100$ and minimizes the overall penalty 'E_μ' paid by the model.

3.1 Training and Optimization Algorithm

Given Y training set with sensor inputs and y_o, i.e. output vector of training set where $y_o \in 0, 1$, we need to find weights of sensor inputs, i.e. μ such that our accuracy of predicting output maximizes. The estimation function can be represented as follows:

$$E_\mu(g(f_i), y_{oi}) = \frac{1}{n}\sum_{i=0}^{n} P_\mu(g(f_i), y_{oi}) \tag{7}$$

Here, $g(f_i)$ is the value which decides the actuation command $(O(f_i))$ and $P_\mu(g(f_i), y_{oi})$ is the penalty function. The log-loss penalty function is represented as follows:

$$P_\mu(g(f_i), y_o) = \begin{cases} -\log(g(f_i)), & \text{if } y_{oi} = 1. \\ -\log(1 - g(f_i)), & \text{if } y_{oi} = 0. \end{cases} \tag{8}$$

Here, $g(f_i)$ is the value which decides the predicted actuation command $(O(f_i))$ and y_{oi} depicts the actual actuation command. The penalty function for $y_{oi} = 1$ is depicted by the blue curve in Fig. 2. This indicates that when $g(f_i)$ is 1, then the penalty paid is 0. However, when $g(f_i)$ is 0 for the blue curve, then it pays a high penalty. Similarly, the red curve in Fig. 2 depicts penalty function for $y_{oi} = 0$. This indicates that when $g(f_i)$ is 0, then the penalty paid is 0. However, when $g(f_i)$ is 1 for the red curve, then it pays a high penalty. Equation 8 can be combined and rewritten as follows:

$$P_\mu(g(f_i), y_{oi}) = -y_{oi}.\log(g(f_i)) - (1 - y_{oi}).\log(1 - g(f_i)) \tag{9}$$

Therefore, the optimization equation is:

$$\mu_{op} = \arg\min_{\mu} E_\mu(g(f_i), y_{oi}) \tag{10}$$

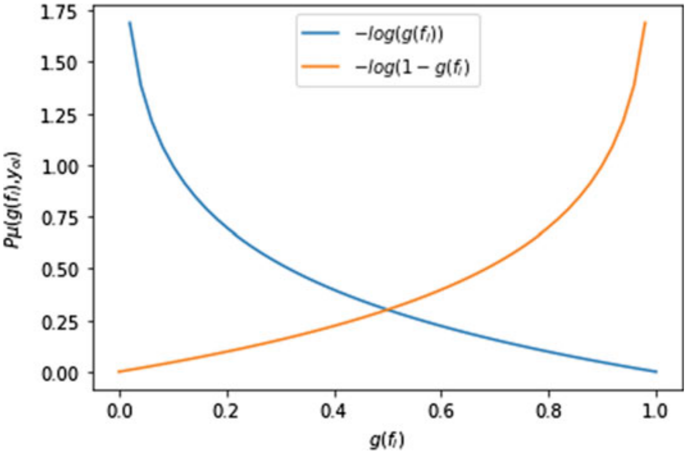


Fig. 2 Logistic loss penalty function

After replacing E_μ value from Eq. (7), we get:

$$\mu_{op} = \arg\min_{\mu} \frac{1}{n} \sum_{i=0}^{n} P_\mu(g(f_i), y_{oi}) \tag{11}$$

Algorithm 1: Optimization Algorithm

```
    Input: $\mathbf{Y} \in \Re^{nXm}$ , $y_{oi} \in \Re^m$ , $T = Number\_of\_iterations$
    Output: $\mu_{op} \in \Re_m$
initialize weights in $\mu : \mu_s \leftarrow 0, s \in (0, 1, \ldots, m)$;
$\mathbf{G}_i \leftarrow Calculate(\mu, Y)$
b=0
while $b \lnsim T$ do
    for $s \leftarrow 0$ to $m$ do
        $d_s \leftarrow \sum_{i=0}^{n} (G_i - y_{oi}) y_{is}$
        $\mu_s \leftarrow \mu_s - \alpha\ d_s$
    end
    $G_i \leftarrow Calculate(\mu_s, Y)$
    $b = b + 1$
end
```

Algorithm 2: Calculate(μ, Y)

```
    Input: $\mathbf{Y} \in \Re^{qxm}$ , $\mu \in \Re^m$ , $y_{oi} \in \Re^q$
    Output: $\mathbf{G}_i \in \Re^q$ , $acc$
initialize $\mathbf{G}_i \leftarrow 0, i \in (0, 1, \ldots, q)$, $pos = 0, neg = 0, acc = 0$;
i=0
while $i \lnsim q$ do
    $f(y_i) \leftarrow \mu_0 + \sum_{j=1}^{m} \mu_j y_{ij}$
    $Z_i \leftarrow f(y_i)$
    $g(Z_i) \leftarrow (1 + e^{-Z_i})^{-1}$
    $G_i \leftarrow g(Z_i)$
end
for $i \leftarrow 0$ to $m$ do
    if $Z_i > 0$ then
        $O_i.append(1)$
    else
        $O_i.append(0)$
    end
    if $O_i == y_{oi}$ then
        $pos = pos + 1$
    else
        $neg = neg + 1$
    end
end
$acc = \frac{pos}{pos + neg} * 100$
```

Plugging in the value of the penalty function from Eq. (9), we get the following optimization equation:

$$\mu_{op} = \arg\min_{\mu} \frac{1}{n} \sum_{i=0}^{n} -y_{oi}.\log(g(f_i)) - (1 - y_{oi}).\log(1 - g(f_i)) \tag{12}$$

The task here is to find the best μ (which is equal to the global minima) which minimizes the given penalty value. In order to find the minima, the equation given below must be satisfied:

$$\frac{\partial E_\mu(v, y_o)}{\partial \mu} = 0 \tag{13}$$

Here, $v = g(f_i)$, i.e. the sigmoid function. The derivative of E_μ w.r.t. μ_s can be computed by applying the chain rule, as shown in Fig. 3. The equation can be written as follows:

$$\frac{\partial E_\mu(v, y_o)}{\partial \mu_s} = \frac{\partial E_\mu}{\partial v}.\frac{\partial v}{\partial z}.\frac{\partial z}{\partial \mu_s} \tag{14}$$

Here, $z = f(x)$, i.e. the prognostic function, μ_s is the weight of the sth sensor input. The partial derivate of E_μ w.r.t. v, i.e. the sigmoid function, can be computed as follows:

$$\frac{\partial E_\mu}{\partial v} = -\frac{y_o}{v} + \frac{1 - y_o}{1 - v} \tag{15}$$

The partial derivate of v w.r.t. z, i.e. prognostic function, can be computed as follows:

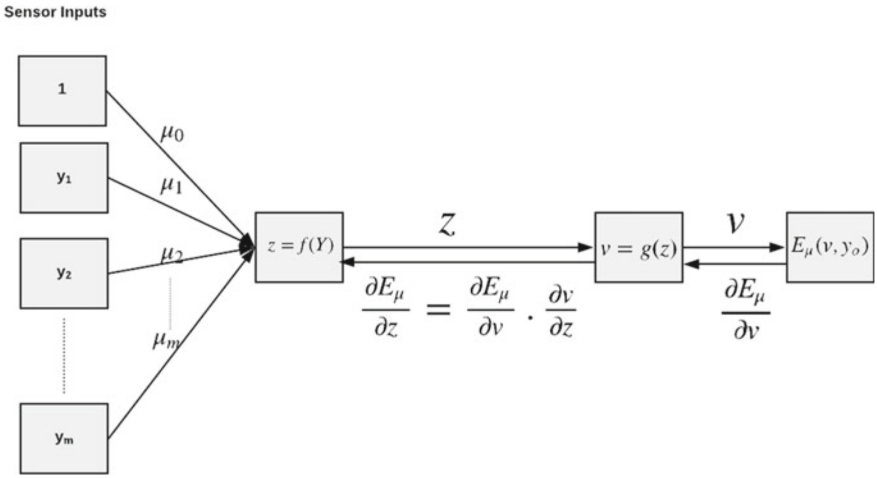

Sensor Inputs
1
y₁
y₂
yₘ
μ_0
μ_1
μ_2
μ_m
$z = f(Y)$
z
$v = g(z)$
v
$E_\mu(v, y_o)$
$\frac{\partial E_\mu}{\partial z} = \frac{\partial E_\mu}{\partial v} \cdot \frac{\partial v}{\partial z}$
$\frac{\partial E_\mu}{\partial v}$

Fig. 3 Back-propagation for computing optimal weights

$$\frac{\partial v}{\partial z} = v.(1 - v) \tag{16}$$

The partial derivate of z w.r.t. μ_s, i.e. the weight of sth sensor input for ith training example, can be computed as follows:

$$\frac{\partial z}{\partial \mu_s} = y_{is} \tag{17}$$

Putting values of Eqs. (15), (16) and (17) in Eq. (14), we get:

$$\frac{\partial E_\mu(v, y_o)}{\partial \mu_s} = (v - y_o).y_{is} \tag{18}$$

Here, v is the sigmoid function. So, $G_i = g(f_i), \forall i \in (0, 1, \dots, n)$. Therefore, Eq. (18) can be written as follows:

$$\frac{\partial E_\mu(v, y_o)}{\partial \mu_s} = \sum_{i=0}^{n} (G_i - y_{oi}).y_{is}, \ \forall s \in (0, 1, \dots, m) \tag{19}$$

In Algorithm 1, the weight optimization is done by using the derivative values computed by Eq. (19) depicted by d_s in the algorithm. Once the value of d_s is calculated $\forall s \in (0, 1, \dots, m)$, then the corresponding weights μ_s for each sensor input are updated using the following equation:

$$\mu_s = \mu_s - \alpha.\frac{\partial E_\mu(v, y_o)}{\partial \mu_s}, \ \forall s \in (0, 1, \dots, m) \tag{20}$$

Here, α is the learning rate. Inserting the value from Eq. (19) in Eq. (20), we get:

$$\mu_s = \mu_s - \alpha.\sum_{i=0}^{n} (G_i - y_{oi}).y_{is}, \ \forall s \in (0, 1, \dots, m) \tag{21}$$

After updating the weights, they are passed to Calculate() (Algorithm 2). Then, Z_i is computed using the prognostic function and G_i is computed using $G_i = g(Z_i)$ with updated weights. After that, if the value $G_i > 0.5$ or $Z_i > 0$, then we classify $O_i = 1$ (the server will give actuation to switch on the pump); otherwise, we classify $O_i = 0$ (no action is required).

Next, the predicted output is compared with the actual output in the training data. If it matches, then we correctly classify and then increment pos, else we increment neg. Then, the accuracy is calculated using the following equation:

$$\text{acc} = \frac{\text{pos}}{\text{pos} + \text{neg}} * 100 \tag{22}$$

Again weights will be updated using updated G_i [using Eq. (21)]. This procedure will be repeated till T iterations. Then, the optimization algorithm, i.e. Algorithm 1, returns the optimized weights μ_{op}. Then, the trained model is tested using these optimized weights (μ_{op}).

4 Experimental Results and Discussion

4.1 Experimental Set-Up

Units of DHT-11 (a humidity sensor), LM-35 (a temperature sensor) and FC-28 (a moisture sensor) were connected to an MCP3008 ADC which converts analog values to digital values. Next, the Arduino IDE was used to upload the code on to NodeMCU. The NodeMCU was connected to a 5 V supply. Using Wi-Fi, it sent the sensed data to the server through a router, as shown in Fig. 4. The data was stored in the server and then analysed in order to train the model for future predictions. The training of the machine learning model was carried out on an Intel core i7-8750H machine with a 2.20 GHz processor having 12 cores, 16 GB RAM, 64-bit OS and Graphics GTX 1050 Ti/PCIe/SSE2. Next, the actuation commands (i.e. predictions) were sent using HTTP to perform the necessary action of irrigating fields based on the environmental factors. The data was sensed after 10,000 ms from all sensors through the NodeMCU and sent to the server for storage. For training, 400 instances were randomly selected, such that 200 instances out of 400 were positive (labelled irrigation), while the remaining 200 instances belonged to the negative class (labelled no action). The evaluation metrics used in the experiments are as follows:

1. Accuracy (Acc.): This is described as the percentage of correctly classified instances vs. total number of instances. This is calculated using the formula $\frac{\text{pos}}{\text{pos+neg}} * 100$, where pos represents correctly classified and neg represents incorrectly classified instances.
2. Decision Boundary (DB): As we have 3 sensor inputs, so in order to separate the data points with two classes in 3D space plane (polynomial degree is 1) is required. If we increase the polynomial degree of the decision boundary to be greater than 1, this represents a hyperplane. So, DB represents the polynomial degree of decision boundary in our experiments and it ranges from 1 to 4.
3. Regularization Factor (λ): Regularization helps in trading off between overfitting (good performance on training data but poor performance on unseen data) and underfitting (poor performance on both training and unseen data) the model. This is done by tuning the regularization factor λ in the following equation:

$$E_{\mu}(g(f_i), y_{oi}) = \frac{1}{n}\sum_{i=0}^{n} P_{\mu}(g(f_i), y_{oi}) + \lambda\mu^T.\mu \tag{23}$$

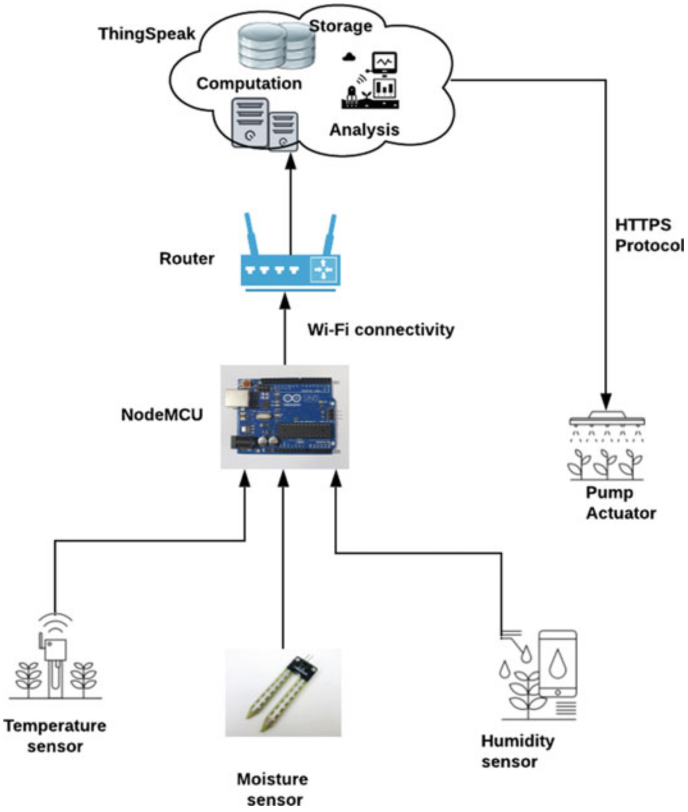


Fig. 4 Experimental set-up

4. Estimation function (E_μ): The estimation function imposes a penalty when instances are incorrectly classified by the model. This function is used to measure the error and to optimize the parameters so that the model learns to classify the instances correctly in future trails. In our experiments, we have evaluated the performance on two penalty functions, namely: square error loss function and log-loss. The square error loss penalty function is represented as follows:

$$P_\mu(g(f_i), y_{oi}) = \frac{1}{2}(g(f_i) - y_{oi})^2 \tag{24}$$

5. Learning Rate (α): This is the hyper-parameter which controls the updating of the weights of the sensor inputs based on the loss gradient. The relationship between the learning rate and updated weights is as follows:

$$\text{upd\_wts} = \text{curr\_wts} - \alpha * \text{loss\_gradient} \tag{25}$$

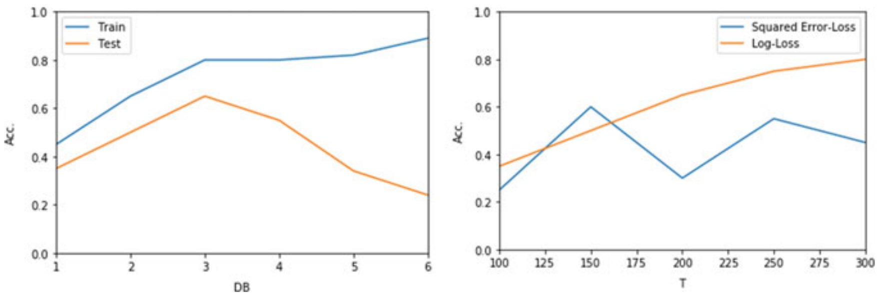
Here, upd_wts are the updated weight vectors, curr_wts are the current weight vectors and α is the learning rate in the gradient descent algorithm.

6. Number of iterations (T): This is the number of times the optimization algorithm must iterate in order to get the optimized weight vector μ_{op}.

4.2 Results and Discussion

Effect of Complex Decision Boundary on Accuracy As shown in Fig. 5a, the decision boundary with degree 1 (i.e. DB = 1) represents the plane in 3D space to classify the positive class (i.e. irrigate instances) with the negative class (i.e. no action instances). However, the train and test accuracy of the model is very low due to underfitting. This underfitting takes place because the model is too simple. The decision boundary with polynomial degree 3 (i.e. DB = 3) results in a hyperplane to classify instances. The accuracy recorded for DB = 3 is the maximum. This is because as we increase the degree of decision boundary, the model overfits the training data. Due to this overfitting, the trained model (complex hyperplane) learns the training data very well; therefore, the training accuracy is high. However, the test accuracy is low, because it fails to generalize the new sensed data for DB = 4 and DB = 5.

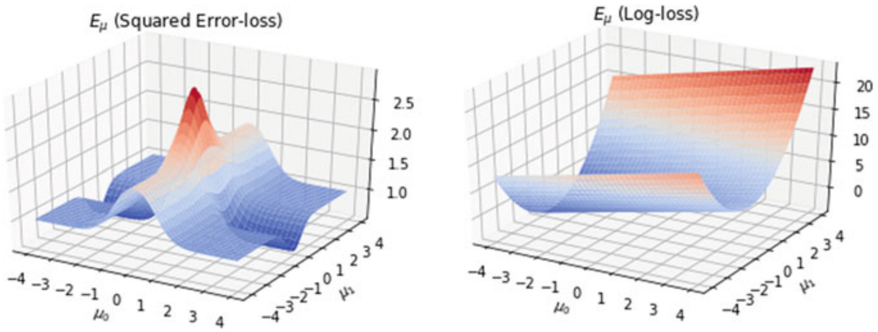
Effect of Penalty Function on Accuracy In order to minimize the penalty (E_μ), the model must be tuned to the best μ value, which is the global minima. We take the squared error as the penalty function [represented by Eq. (24)]. Please refer to Fig. 6a. When the estimation function for the squared error loss is plotted against the weights of two features (temperature and moisture), then the squared error loss penalty function results in a non-convex behaviour of the estimation function; i.e. there are a large number of local minimas. This happens since $g(x)$ (i.e. the sigmoid function) is a nonlinear function when plugged in the estimation function. So, it will not converge on the global minima when the gradient descent is applied. In contrast, when the log-loss penalty function is plugged into the estimation function, it results



(a) DB vs Accuracy

(b) Number of iterations vs Accuracy

Fig. 5 Effect of decision boundary (DB) and number of iterations on accuracy



(a) Squared Error-loss vs E_μ

(b) Log-loss vs E_μ

Fig. 6 Effect of squared error loss and log-loss on estimation function (E_μ)

in a smooth convex curve, as shown in Fig. 6b. Therefore, when we apply the gradient descent on the resultant estimation function, then it converges into the global minima. In Fig. 5b, we can see that as the number of iterations increases, the accuracy also increases for the log-loss penalty function. This is because when the log-loss penalty function is used, it results in a smooth convex curve of the estimation function with respect to the weights. Due to this, with the increasing iterations, the optimization algorithm converges to the global minima, which results in improving the accuracy of the model. However, when the squared error loss is used as a penalty function, then it demonstrates random behaviour as the number of iterations increases. This is because of the non-convex surface of the estimation function which might converge in a local minima in spite of the global minima with each iteration.

Effect of Regularization on Accuracy Regularization is done to address the overfitting issue. Penalizing a few features helps in providing simpler models. In Eq. (23), the goal of the first part of the equation is to fit the training data well. The goal of the second part (regularization) of the equation is to keep the parameter (weights) terms small. Therefore, Eq. (23) tries to trade off between both goals in order to avoid overfitting. As shown in Fig. 7, when $\lambda = 0.02$ (very less regularization), the training error (E_μ(log −loss)) is minimum. This is because in Eq. (24), the second part is negligible due to a small λ value. So, the first part of the equation which fits the training data plays the dominant role. Hence, this results in less error for training data but high error on test data or unseen data, leading to overfitting. When $\lambda = 0.64$ and $\lambda = 1.28$, the training error is observed to be very high. This is because more importance is given to the regularization part (second part) of Eq. (23), which assigns very low weights to the features. Due to this, the model becomes too simple and therefore results in underfitting. This underfit (simplistic) or highly biased model is unable to classify unseen data correctly, which leads to a high test error. In Fig. 7, we can see that the test error first decreases for $\lambda = 0.02$ and $\lambda = 0.04$. The minimum test error is reported at $\lambda = 0.08$. Then, for $\lambda > 0.08$, the test error starts increasing. So, the best accuracy (minimum error) is obtained at $\lambda = 0.08$. Moreover, as shown in

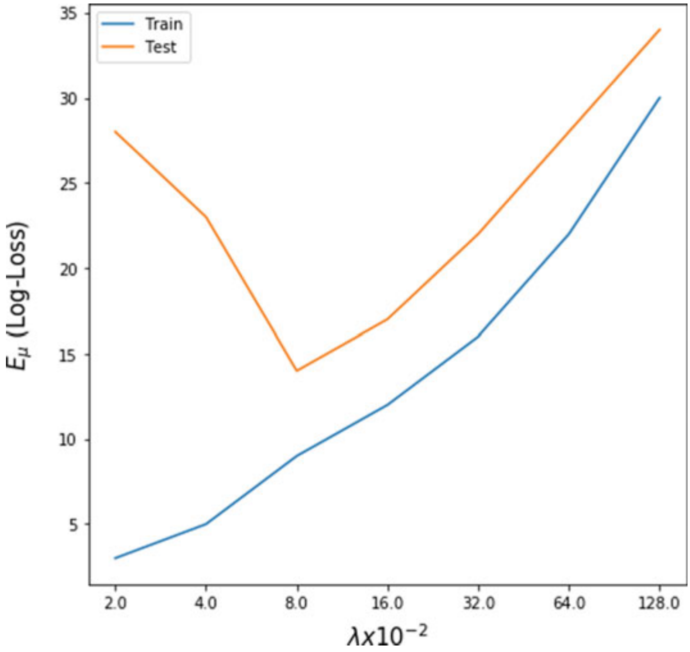
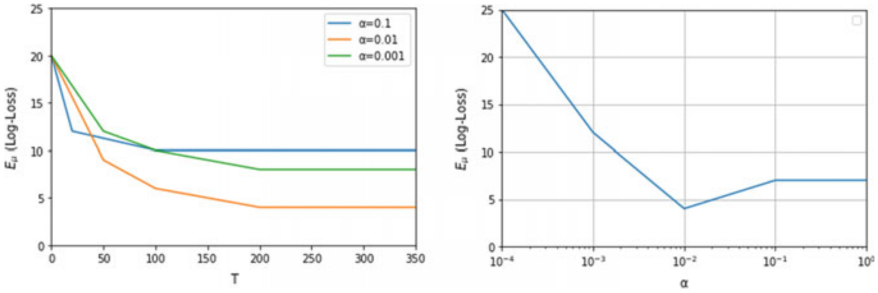


Fig. 7 Effect of regularization (λ) on estimation function

Fig. 5a, the model is overfitting. The train accuracy is 81% while the test accuracy is 62%. So, in order to resolve the problem of overfitting, a moderate value of $\lambda = 0.08$ gives the best result. Note that inserting $\lambda = 0.08$ in Eq. (23) maintains the balance between the overfitting and underfitting. Thus, by tuning the regularization factor to 0.08, the model significantly decreases the variance without substantially escalating the bias of the model. In fact, by using $\lambda = 0.08$, the test accuracy improves from 62 to 75%.

Effect of Learning Rate on Estimation Function The learning rate (α) determines how quickly (using lesser number of iterations) we obtain the optimized parameters (μ_{op}) using our *optimization* algorithm. This is a hyper-parameter (i.e. the best value is selected by trail and error). The value needs to be tuned in order to achieve the best accuracy in less time. As shown in Fig. 8a, when the value of $\alpha = 0.001$ (i.e. low learning rate), then it takes a large number of iterations to converge at the optimized parameters, which can minimize error (E_μ(log −loss)) or maximize accuracy. However, when the value of $\alpha = 0.1$ (i.e. high learning rate), then the error (E_μ) drops considerably in the first few iterations, but due to a large step-size overshoots the global minima for a large number of iterations. Thus, it takes a long time to converge to the optimized parameters. But, when the value of $\alpha = 0.01$, then the model converges to the minima gradually and smoothly. Also, at this value, the model records the minimum error with a very less number of iterations. Figure 8b shows the error



(a) Number of iterations vs E_μ

(b) Learning rate vs E_μ

Fig. 8 Effect of number of iterations and learning rate on estimation function

recorded for various learning rates after 200 iterations. It can be observed that with $\alpha = 0.01$, the minimum error is recorded. However, at $\alpha = 0.0001$, the error recorded is very high because with a very small value of α, it takes a large number of iterations to converge to the global minimum.

5 Conclusion

It is evident that due to their manual nature, the traditional methods used for irrigating fields prove to be quite inefficient. In order to efficiently and automatically monitor the fields in real time, the Internet of Things (IoT) can play a significant role. These IoT devices can be used to construct automated systems, which can optimize the usage of fresh water for irrigation. The use of artificial intelligence (AI) can assist in making these automated systems 'smart' by using the historical irrigation data in order to take future irrigation decisions. Using Wi-Fi, we collect data from various sensors at different locations and then aggregate this data in a web server. Next, the model is trained based on the acquired data by using an *optimization* algorithm. Finally, this trained model is used for making decisions in future. Experiments demonstrate that by using complex decision boundaries and log-loss as the penalty function, the test accuracy of the model approaches 75%. In addition, by introducing the regularization factor, the generalized performance (i.e. the decisions for unseen data) of the model is improved significantly, due to the minimization of the penalty paid by the trained model.

References

1. Agrawal N, Singhal S (2015) Smart drip irrigation system using raspberry pi and arduino. In: International conference on computing, communication and automation. IEEE, pp 928–932
2. Al-Ammri AS, Ridah S (2014) Smart irrigation system using wireless sensor network. Int J Eng Res Technol (IJERT) 3(1):2278-0181

3. Al-Ghobari HM, Mohammad FS (2011) Intelligent irrigation performance: evaluation and quantifying its ability for conserving water in arid region. Appl Water Sci 1(3–4):73–83
4. Basu PK (2011) Methods manual: soil testing in India. Department of Agriculture and Cooperation, Ministry of Agriculture Government of India New Delhi. Krishi Bhawan, New Delhi
5. Chavan CH, Karande PV (2014) Wireless monitoring of soil moisture, temperature and humidity using zigbee in agriculture. Int J Eng Trends Technol (IJETT) 11(10):493–497
6. Gondchawar N, Kawitkar RS et al (2016) IoT based smart agriculture. Int J Adv Res Comput Commun Eng 11(10):493–497
7. Gupta A, Mishra S, Bokde N, Kulat K (2016) Need of smart water systems in India. Int J Appl Eng Res 11(4):2216–2223
8. Gutiérrez J, Villa-Medina JF, Nieto-Garibay A, Porta-Gándara MÁ (2013) Automated irrigation system using a wireless sensor network and GPRS module. IEEE Trans Instrum measur 63(1):166–176
9. Harishankar S, Kumar RS, Sudharsan KP, Vignesh U, Viveknath T (2014) Solar powered smart irrigation system. Adv Electron Electr Eng 4(4):341–346
10. Jha K, Doshi A, Patel P, Shah M (2019) A comprehensive review on automation in agriculture using artificial intelligence. Artif Intell Agric 2:1–12. https://doi.org/10.1016/j.aiia.2019.05.004
11. Jhuria M, Kumar A, Borse R (2013) Image processing for smart farming: detection of disease and fruit grading. In: 2013 IEEE second international conference on image information processing (ICIIP-2013). IEEE, pp 521–526
12. Jin S, Jingling S, Qiuyan H, Shengde W, Yan Y (2007) A remote measurement and control system for greenhouse based on GSM-SMS. In: 2007 8th International Conference on Electronic Measurement and Instruments, pp 2-82-2-85. https://doi.org/10.1109/ICEMI.2007.4350806
13. Kansara K, Zaveri V, Shah S, Delwadkar S, Jani K (2015) Sensor based automated irrigation system with IoT: a technical review, p 12
14. Meena Kumari G, Devi V (2013) Real time automation and monitoring system for modernized agriculture. Int J Rev Res Appl Sci Eng 3
15. Mirabella O, Brischetto M (2010) A hybrid wired/wireless networking infrastructure for greenhouse management. IEEE Trans Instrum Measur 60(2):398–407
16. Naik P, Kumbi A, Katti K, Telkar N (2018) Automation of irrigation system using IoT. Int J Eng Manuf Sci 8(1):77–88
17. Parameswaran G, Sivaprasath K (2016) Arduino based smart drip irrigation system using Internet of things. Int J Eng Sci 5518
18. Rao RN, Sridhar B (2018) IoT based smart crop-field monitoring and automation irrigation system. In: 2018 2nd international conference on inventive systems and control (ICISC). IEEE, pp 478–483
19. Reshma S, Babu BASM (2016) Internet of things based automatic irrigation system using wireless sensor networks. Int J Mag Eng Technol Manag Res 3

Discerning Android Malwares Using Extreme Learning Machine

Anand Tirkey, Ramesh Kumar Mohapatra, and Lov Kumar

Abstract Android, being the most widely used mobile operating system of choice, also poses a security risk of mass privacy intrusion and sensitive data theft. Hence, as a mitigation effort, it is imperative to identify robust methods in identifying android malwares. In this paper, we extend our previous research work titled "Anatomizing Android Malwares" [20] and bring forward a novel method of identifying malicious android apps using object-oriented software metrics and supervised machine learning techniques. Initially, we retrieve object-oriented software metrics from the decompiled android app, and then this metrics tuple is tagged either as benign or malware using VirusTotal service. Finally, the set of metrics tuple acts as input features in machine learning algorithms. We evaluated the performance and stability of our forty-eight different machine-learned models against 5774 android apps collected from AndroZoo [1]. The discriminatory power of every machine-learned model is measured using its area under ROC curve (AUC), accuracy and F-measure values. Our method yields AUC, accuracy and F-measure of 1.0, 100% and 1.0, respectively.

Keywords Android · Malware detection · Machine learning · Object-oriented metrics

1 Introduction

Android OS market share in 2018 has been 85.1% and is expected to grow to 86.7% in 2019, according to International Data Corporation (IDC, USA). In 2018, Symantec intercepted and blocked an average of 10,573 malicious mobile apps per day. In May 2019, Google reported that 42.1% of android devices run unsupported versions of the OS. Meanwhile, Karstern Noh et al. [17] point out that the situation has worsened due to the fact that very few mobile handset vendors truly provide monthly patches of

A. Tirkey (✉) · R. Kumar Mohapatra
National Institute of Technlogy Rourkela, Rourkela, Odisha, India
e-mail: rkmohapatra@ieee.org

L. Kumar
BITS Pilani, Hyderabad Campus, Secunderabad, Telangana, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_2

the supported android OS released by Google. Even though Google releases monthly patches and a new version of their android OS every year, it ultimately depends upon the handset vendor to finally deliver the OS update. This has resulted in the android OS fragmentation, which potentially exposes millions of devices to malware threats.

Google has put various safeguards in place, in order to prevent such exploits such as android permissions and Google Play protect that proactively checks for malware apps installed via Google Play Store. Unlike the previous implementation of android permissions system, where all the required permissions were requested for approval prior to installation of apps, current implementation of android permissions system mandates every permission to be asked when the app truly requires it. Today almost every app asks for permissions in order to work, and this recurring habit of granting permission becomes the basis for the lapse in security because the device end users least understand the implications of granting unwanted android permissions to unsuspecting malware apps. Even after these safeguards, it is still possible to install or sideload apps from third-party sources and other marketplaces. Consequently, these safeguards seldom provide any reasonable security.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 presents the methods used to obtain object-oriented metrics-based dataset and subsequently to create models using this dataset. Section 4 discusses the model's performance and provides a comparison of our method with other authors. Section 5 mentions the future work and concludes the paper.

2 Related Work

The malware detection methods can broadly be grouped into two categories such as static analysis and dynamic analysis. Many authors have used static analysis such as Ma et al. [12] use API flows as features for malware detection. They obtain application program interface (API) information from a control flow graph retrieved from an apk, and they use this API information to build three API datasets, capturing different aspects of API flows. Neeraj Chavan et al. [3] use android permissions requests as their features in malware detection using SVM and ANN. Garg et al. [7] use API and permissions requests as static features and dynamic features such as battery temperature and network traffic. Finally, they validate the effectiveness of their model using machine learning techniques such as SVM, RIDOR, PART, MLP and their ensembles. Yen et al. [23] use apk source code visualization technique. They compute term frequency-inverse document frequency (TF-IDF) of the decompiled source code and transform that information into images, which is then fed as input to CNN for malware analysis. Martín et al. [13] have used Markov chains and dynamic analysis for malware classification. They have deployed DroidBox tool to gather run-time information, and this information is transformed into the first-order Markov model. From this model, transition probabilities and state frequencies are used as input data in deep learning algorithm, for malware classification. Saif et al. [18] use hybrid set of features retrieved from static analysis, dynamic analysis and set of system calls from both malware and benignware android apps. This hybrid set of features is then used as input for malware classification in deep belief networks. Martín et al. [14] have collected android apk malware information through the usage of 61 antivirus softwares. They have then used this information to group android apks into malware classes using graph community algorithms and hierarchical clustering. Finally, using these groups as their dataset, they have performed malware classification using logistic regression and random forest machine learning algorithms. Xiao et al. [22] considers that there exists some semantic information in the system calls within an android system. The sequences of android systems calls are considered a sentence, and this information is used as input feature in long short-term memory (LSTM) language model. Two separate LSTM machine-learned models are trained for benign and malware apps using system call sequences for the respective apps. Whenever a new app is encountered, its similarity score using app's system call sequence is calculated against two pre-trained machine-learned models. The encountered app is flagged as a malware, if the similarity score against malware LSTM model is greater than the score against benign LSTM model; otherwise, it is flagged as benign. Wang et al. [21] collect their features using Androguard [6] tool and use deep auto-encoder (DAE) and convolution neural network (CNN) model for android malware recognition. Initially, the high-dimensional input features are passed onto multiple CNN for android malware detection, followed by serial convolution neural network (CNN-S) with ReLu activation function to increase sparseness dropout, which helps in preventing data over-fitting. Consequently, for reducing training time of CNN model, it is pre-trained by applying DAE over CNN. Finally, the resultant model is used in detecting android malware. Zhu et al. [25] deploy an ensemble of machine learning techniques such as random forest and support vector machine (SVM) for android malware recognition. They obtain commonly available features such as allowed permissions, permission frequency, sensitive API and system events monitoring for each of the collected android apps. The ensemble of classifiers is then fine-tuned by judiciously selecting the features at each split and the number of decision trees. Yerima et al. [24] extract critical API calls and command sets using their custom build android analysis tool. These API calls and command sets form the input features to be used in ensemble learning (random forest, simple logistic, Naive Bayes) for android malware classification. Liang et al. [11] create general rules by observing the android permission combinations declared within an app's manifest file. Initially, they obtain the group of permission combinations frequently requested by malware apps but are seldom invoked by benign apps. Consequently, rules are made based on these observations that predict whether an incoming app is benign or malware. McLaughlin et al. [16] extract raw opcode sequence from decompiled android apps, and it is used as input feature over convolution neural network (CNN) for android malware recognition. This technique allows long sequence of opcodes to be run feasibly on a GPU for greater malware detection accuracy.

3 Proposed Work

We collected android apps from AndroZoo [1] and extracted the corresponding object-oriented metrics using CKJM extended tool [10] as illustrated in Fig. 1. These software metrics are used as features, to be used in various machine learning techniques.

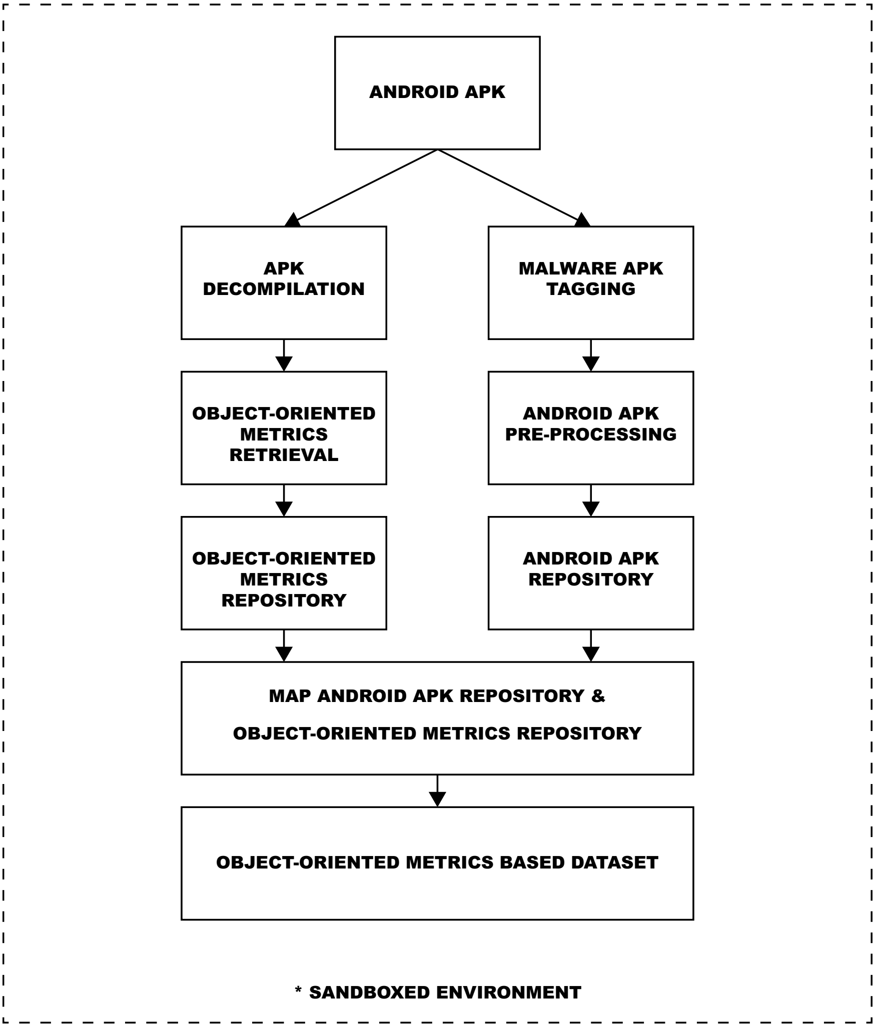

ANDROID APK
APK DECOMPILATION
MALWARE APK TAGGING
OBJECT-ORIENTED METRICS RETRIEVAL
ANDROID APK PRE-PROCESSING
OBJECT-ORIENTED METRICS REPOSITORY
ANDROID APK REPOSITORY
MAP ANDROID APK REPOSITORY & OBJECT-ORIENTED METRICS REPOSITORY
OBJECT-ORIENTED METRICS BASED DATASET
* SANDBOXED ENVIRONMENT

Fig. 1 Preparation of object-oriented metrics-based dataset

Table 1 Object-oriented metrics *p*-values (benign vs. malware)

Metrics	WMC	DIT	NOC	CBO	RFC	LCOM	Ca	Ce	NPM
p-value < 0.05	●	●	●	●	●	●		●	●
Metrics	LCOM3	LCO	DAM	MOA	MFA	CAM	IC	CBM	AMC
p-value < 0.05	●	●	●	●	●	●	●	●	

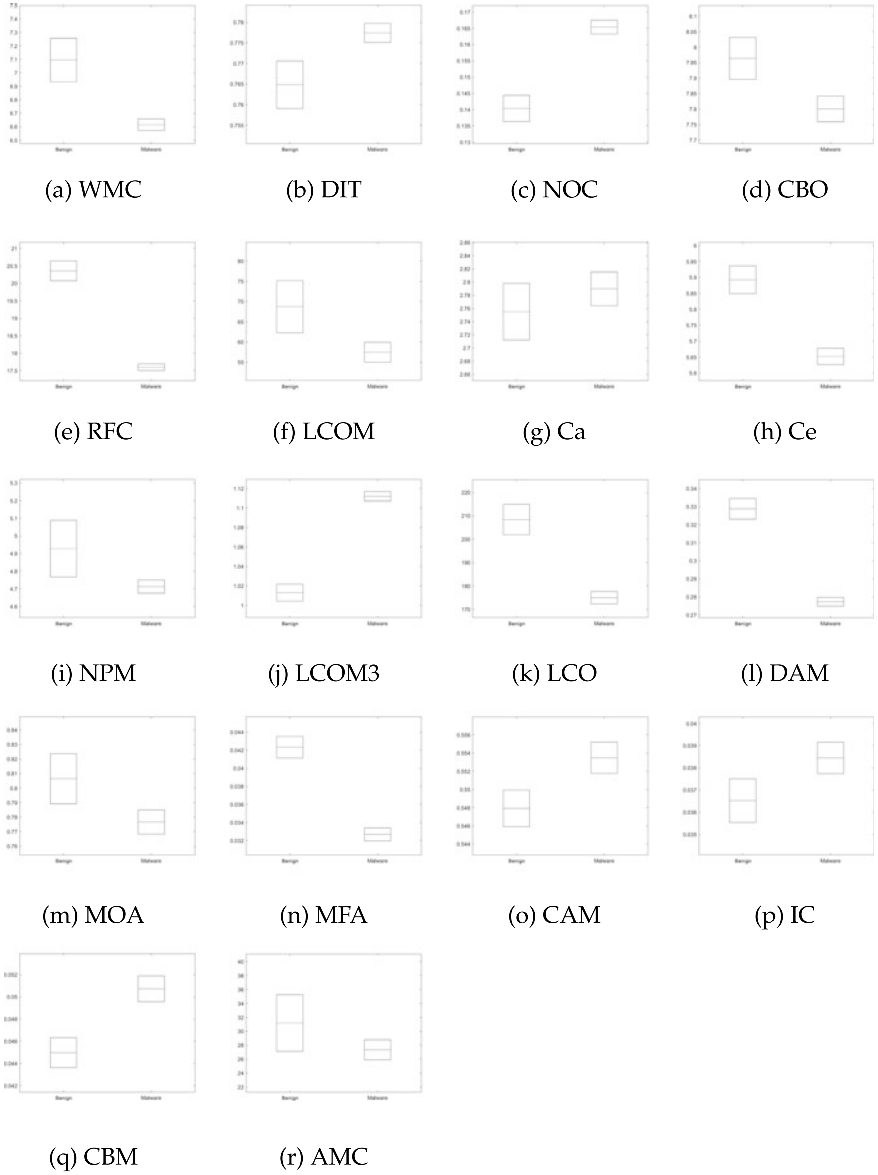
3.1 Object-Oriented Software Metrics

Object-oriented software metric is a measure of intangible software characteristics that are quantifiable and countable for any object-oriented paradigm-based programming language. In this study, 18 different object-oriented software metrics [2, 5, 8, 9, 15, 19] as defined by experts are retrieved using CKJM extended tool [10] and have been considered. These metrics are weighted methods per class (WMC), depth of inheritance tree (DIT), number of children (NOC), coupling between object classes (CBO), response for a class (RFC), lack of cohesion in methods (LCOM), afferent coupling (Ca), efferent coupling (Ce), number of public methods for a class (NPM), lack of cohesion in methods Henderson–Sellers version (LCOM3), lines of code (LCO), data access metric (DAM), measure of aggregation (MOA), measure of functional abstraction (MFA), cohesion among methods of class (CAM), inheritance coupling (IC), coupling between methods (CBM) and average method complexity (AMC).

Effectiveness of Object-Oriented Metrics Before proceeding with the experiment, it is necessary to ascertain the feasibility and effectiveness of using object-oriented metrics as features for detecting android malware. Boxplots for each of the 18 object-oriented metrics can be observed from Fig. 2. It is further observed that each subfigure under Fig. 2 has two boxplots depicting the distribution of metric values for benign and malware apps. Upon inspecting the inter-quartile range (IQR) from the boxplots, it is evident that the benign and malware IQRs for all of the object-oriented metrics do not overlap except for Ca and AMC metrics as shown in Fig. 2g, r, respectively. The *p*-values of object-oriented metrics both for benign and malware samples are shown in Table 1. Considering a significance level of 0.05, *p*-values less than 0.05 are denoted by symbol "●". It is observed from Table 1 that all metrics except for Ca and AMC are statistically significant. Hence, object-oriented metrics-based datasets can be effectively used in android malware detection using machine learning techniques.

$$\text{IQR} = (\text{Third Quartile} - \text{First Quartile}) \tag{1}$$

Object-Oriented Metrics Retrieval Object-oriented metrics-based dataset is obtained from android apk files collected from AndroZoo [1]. Then object-oriented metrics are retrieved from the decompiled Java classes and source codes using



(a) WMC (b) DIT (c) NOC (d) CBO

(e) RFC (f) LCOM (g) Ca (h) Ce

(i) NPM (j) LCOM3 (k) LCO (l) DAM

(m) MOA (n) MFA (o) CAM (p) IC

(q) CBM (r) AMC

Fig. 2 Object-oriented metrics boxplot

CKJM extended tool [10]. The list of selected object-oriented metrics is described in Sect. 3.1. After retrieving object-oriented metrics for every decompiled apk file, it is then stored in a local repository. Finally, object-oriented metrics-based dataset is obtained by mapping object-oriented metrics and android executable malware tag obtained from VirusTotal service. Feature vectors in this dataset are of the dimension (20×1), comprising of 18 object-oriented metrics along with its respective application package name and malware tag. In this study, all of the malicious app variants are clubbed together into a single category called malware, and the rest of the safe android apps are categorized as benign. This has been done in order to simplify the experiments toward binary classification.

3.2 Android Malware Detection

In this phase, six different supervised machine learning algorithms are deployed over eight different metrics-based datasets, in order to create ($6 \times 8 = 48$) forty-eight different machine-learned models from the metrics-based dataset. Consequently, these machine-learned models are used to classify the android apps either as benign or malware. The performance of these machine-learned models is evaluated and fine-tuned so as maximize the malware detection accuracy, AUC and F-measure. The process of android malware identification contains three sub-processes such as preprocessing of metrics-based dataset, creating machine-learned models from metrics-based dataset and classifying android as malware or benign through these models.

Preprocessing of Metrics-Based Dataset In order to create machine-learned models, preprocessing of the metrics-based dataset is required. Initially, the original dataset "OD" comprising of 5774 android apps is observed, and it is found that it contains only 1582 malware samples. Therefore, [4] synthetic minority oversampling technique (SMOTE) analysis is employed to counter the benign–malware sample imbalance. This class-balanced smote dataset derived from "OD" dataset is termed as "SMOTE". These two datasets, i.e., original dataset (OD) and class-balanced smote dataset (SMOTE), are used in analyzing android malwares. Four different feature selection algorithms have been used over OD and SMOTE datasets such as considering all features (ALL), wilcoxon signed-rank test (SIG), univariate logistic regression (ULR) and principal component analysis (PCA). Finally, we have a total of eight different metrics-based datasets such as original dataset taking all features (OD), class-balanced smote dataset taking all features (SMOTE), dataset obtained upon applying SIG over OD (OD-SIG), dataset obtained upon applying SIG over SMOTE (SMOTE-SIG), dataset obtained upon applying ULR over OD (OD-ULR), dataset obtained upon applying ULR over SMOTE (SMOTE-ULR), dataset obtained upon applying PCA over OD (OD-PCA) and dataset obtained upon applying PCA over SMOTE (SMOTE-PCA).

Each metric-based dataset D is split into two parts, X and Y, where X is the feature vector comprising T ($T \leq 18$, as the total number of object-oriented metrics is eighteen) object-oriented metrics from WMC through AMC and Y is the respective malware tag obtained from AndroZoo.

Then, each feature vector x_i in X having dimension (N×T) is min–max normalized using the formula as follows:

$$x_i = \frac{x_i - \min(X)}{\max(X) - \min(X)} \tag{2}$$

Subsequently, each malware tag y_i in Y having dimension (N×1) contains an integer value of 0 or more. Where the value $y_i = 0$ depicts the android app as benign and for values of $y_i > 0$ depicts the app as malicious. Here, the value of y_i shows the count of antivirus companies that have marked the android executable as a malware. In case, value of y_i is missing or invalid, then the android app is discarded from the local repository and consequently from any further experiments. In this study, y_i can either have a value of 0 or 1, where the value of $y_i = 1$ is considered a malware. Hence, each value of y_i in Y is processed using the formula as follows:

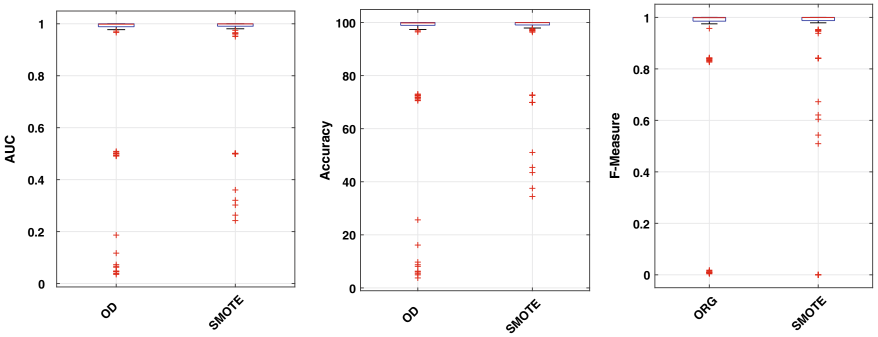
$$y_i = \begin{cases} 0 & y_i = 0 \\ 1 & y_i \geq 1 \end{cases} \tag{3}$$

Create Models from Dataset Using Machine Learning and Android Malware Detection In this study, six different classification algorithms as employed such as [26] weighted extreme learning machine with sigmoid activation function (WELM-Si), [26] weighted extreme learning machine with triangular basis activation function (WELM-triba), [26] weighted extreme learning machine with radial basis activation function (WELM-RBF), extreme learning machine with sigmoid activation function (ELM-Sig), extreme learning machine with linear activation function (ELM-lin) and extreme learning machine with radial basis activation function (ELM-RBF), over eight different metrics-based datasets to create ($6 \times 8 = 48$) forty-eight different machine-learned classification models. The classification strength for each of the machine-learned model is evaluated using its AUC, accuracy and F-measure values.

4 Experimental Results and Comparison

4.1 Analyzing Metrics-Based Datasets

In this experiment, a total of eight different metrics-based datasets are considered as described in Sect. 3.2. The boxplots depicting AUC, accuracy and F-measure values of the OD and SMOTE-based datasets are shown in Fig. 3a–c, respectively.



(a) (b) (c)

Fig. 3 Boxplots for metrics-based datasets

Table 2 Boxplot descriptive statistics and *p*-value for metrics-based datasets

	Min	Max	Mean	Median	$Q1$	$Q3$
OD	0.04	1.00	0.88	1.00	0.99	1.00
SMOTE	0.24	1.00	0.93	**1.00**	0.99	1.00

(a) AUC

	Min	Max	Mean	Median	$Q1$	$Q3$
OD	3.76	100.00	89.81	99.83	98.92	100.00
SMOTE	34.46	100.00	94.85	**99.94**	99.07	100.00

(b) Accuracy

	Min	Max	Mean	Median	$Q1$	$Q3$
ORG	0.00	1.00	0.90	1.00	0.99	1.00
SMOTE	0.00	1.00	0.93	**1.00**	0.99	1.00

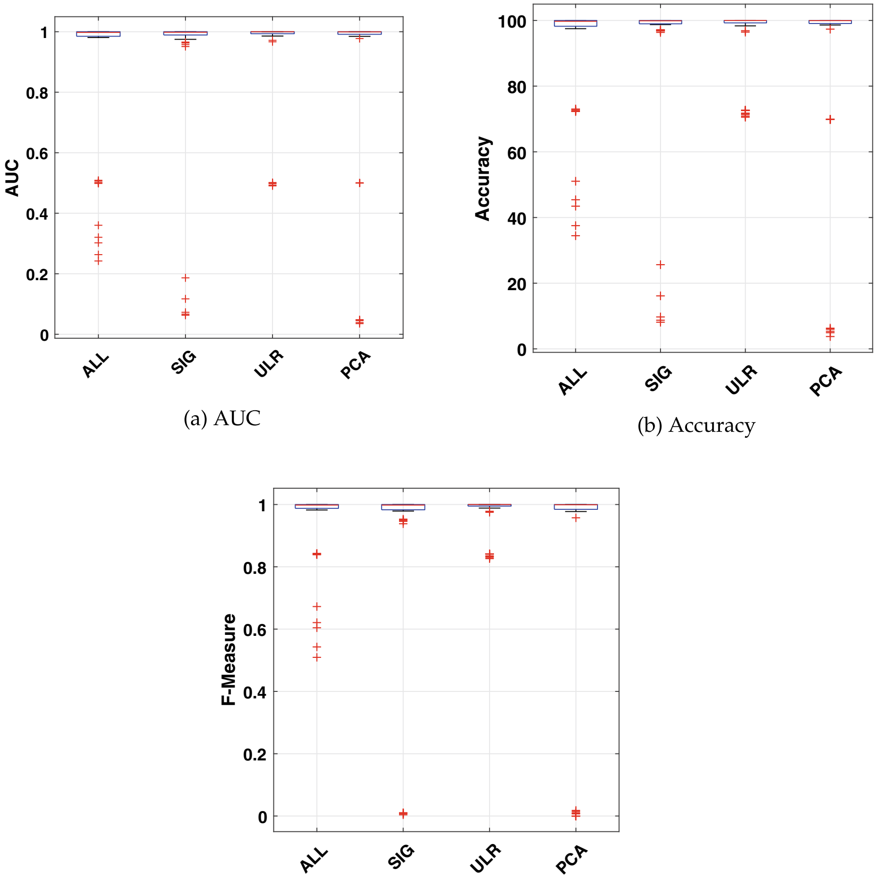
(c) F-measure

	OD	SMOTE
OD		●
SMOTE		

(d) p-value

Meanwhile, the corresponding boxplot descriptive statistics is shown in Table 2a–c, respectively. It is observed from these tables that SMOTE-based class-balanced datasets yield better median values for AUC, accuracy and F-measure with 1.0, 99.94 % and 1.0, respectively.

Considering two primary metrics-based datasets, i.e., OD and SMOTE, a total of ${}^{2}C_{2} = 1$ unique pair is possible. Analyzing the p-value of this unique pair at 0.05 significance level, we can reject a null hypothesis if and only if the p-value is less than $0.05/1 = 0.05$. In Table 2d, the p-values less than 0.05 are denoted by the symbol "●". It can be inferred from Table 2d that datasets based on OD are significantly different than datasets based on SMOTE.



(a) AUC

(b) Accuracy

(c) F-measure

Fig. 4 Boxplots for feature selection techniques

4.2 Analyzing Feature Selection Methods

Boxplots for four different feature selection algorithms depicting AUC, accuracy and F-measure values are shown in Fig. 4a–c, respectively. Meanwhile, the corresponding boxplot descriptive statistics is shown in Table 3a–c, respectively. It is observed from these tables that datasets applying ULR feature selection algorithm yield better median values for AUC, accuracy and F-measure with 1.0, 100.00% and 1.0, respectively.

Taking four different feature reduction algorithms as described in Sect. 3.2, a total of ${}^4C_2 = 6$ unique pairs are possible, and upon scrutinizing the result at 0.05 significance level, we can reject an null hypothesis if and only if the p-value is less

Table 3 Boxplot descriptive statistics and *p*-value for feature selection techniques

	Min	Max	Mean	Median	$Q1$	$Q3$
ALL	0.51	1.00	0.95	1.00	0.99	1.00
SIG	0.00	1.00	0.91	1.00	0.98	1.00
ULR	0.83	1.00	0.97	**1.00**	0.99	1.00
PCA	0.00	1.00	0.83	1.00	0.98	1.00

(a) AUC

	Min	Max	Mean	Median	$Q1$	$Q3$
ALL	34.46	100.00	92.52	99.74	98.22	100.00
SIG	8.12	100.00	92.28	99.89	98.98	100.00
ULR	70.54	100.00	95.10	**100.00**	99.26	100.00
PCA	3.76	100.00	89.42	99.94	99.06	100.00

(b) Accuracy

	Min	Max	Mean	Median	$Q1$	$Q3$
ALL	0.51	1.00	0.95	1.00	0.99	1.00
SIG	0.00	1.00	0.91	1.00	0.98	1.00
ULR	0.83	1.00	0.97	**1.00**	0.99	1.00
PCA	0.00	1.00	0.83	1.00	0.98	1.00

(c) F-measure

	ALL	SIG	ULR	PCA
ALL		•	•	•
SIG			•	•
ULR				•
PCA				

(d) p-value

than $0.05/6 = 0.0083$. In Table 3d, the *p*-values less than 0.0083 are denoted by the symbol "•". It can be inferred from Table 3d that all feature selection algorithms are significantly different and unique among themselves.

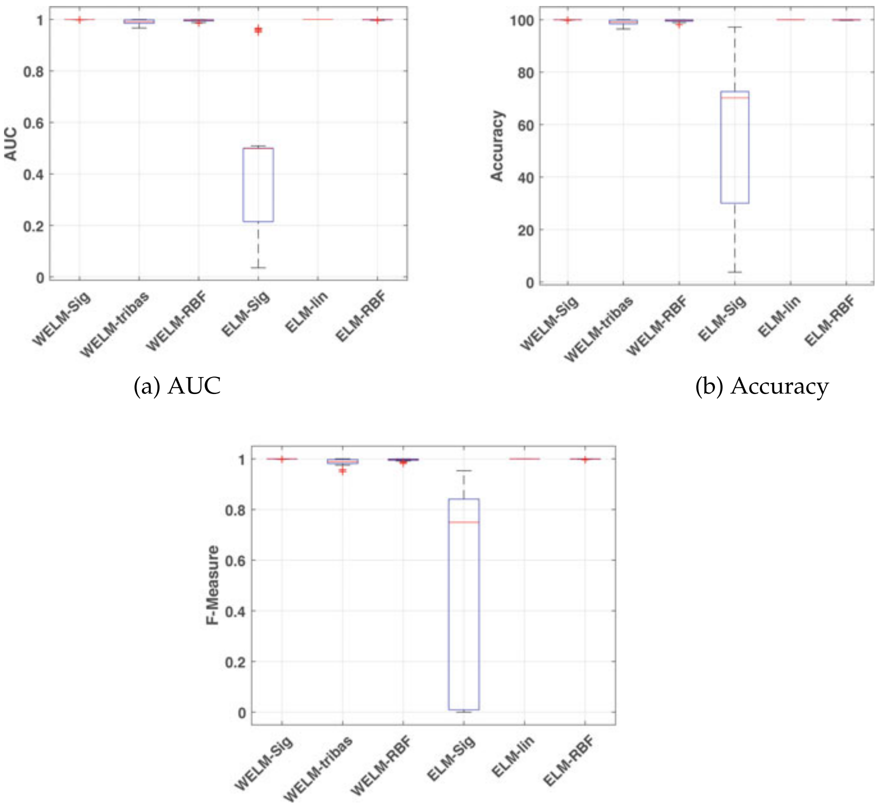
4.3 Analyzing Machine Learning Algorithms

Boxplots for six different machine learning algorithms depicting AUC, accuracy and F-measure values are shown in Fig. 5a–c, respectively. Meanwhile, the corresponding boxplot descriptive statistics is shown in Table 4a–c, respectively. It is observed from these tables that machine-learned models employing WELM-Sig classifier and ELM-lin classifier both yield better median values for AUC, accuracy and F-measure with 1.0, 100.00% and 1.0, respectively.

Considering six different machine learning algorithms as described in Sect. 3.2, a total of ${}^6C_2 = 15$ unique pairs are possible, and upon scrutinizing the result at 0.05 significance level, we can reject a null hypothesis if and only if the *p*-value is less than $0.05/15 = 0.0033$. In Table 4d, the *p*-values less than 0.0033 are denoted by the symbol "•". It can be inferred from Table 4d that out of fifteen pairs, fourteen pairs of classification algorithms are significantly different (p-value < 0.0033) and unique among themselves.

4.4 Analyzing Machine Learning Algorithms Over Metrics-Based Datasets

Upon analyzing Sects. 4.1, 4.2 and 4.3, it is observed that SMOTE-based class-balanced datasets applying ULR feature selection algorithm and classified using either WELM-Sig or ELM-lin yield better results. This observation is confirmed and validated by analyzing the AUC, accuracy and F-measure values shown in Tables 5,



(a) AUC

(b) Accuracy

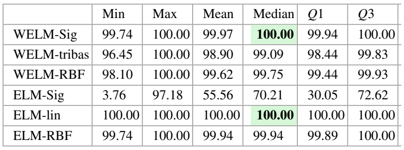
(c) F-measure

Fig. 5 Boxplots for classification techniques

Table 4 Boxplot descriptive statistics and *p*-value for classifiers

	Min	Max	Mean	Median	$Q1$	$Q3$
WELM-Sig	1.00	1.00	1.00	**1.00**	1.00	1.00
WELM-tribas	0.97	1.00	0.99	0.99	0.98	1.00
WELM-RBF	0.99	1.00	1.00	1.00	0.99	1.00
ELM-Sig	0.04	0.97	0.42	0.50	0.21	0.50
ELM-lin	1.00	1.00	1.00	**1.00**	1.00	1.00
ELM-RBF	1.00	1.00	1.00	1.00	1.00	1.00

(a) AUC

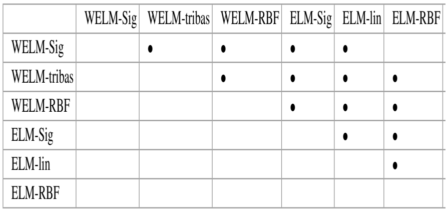


	Min	Max	Mean	Median	$Q1$	$Q3$
WELM-Sig	99.74	100.00	99.97	**100.00**	99.94	100.00
WELM-tribas	96.45	100.00	98.90	99.09	98.44	99.83
WELM-RBF	98.10	100.00	99.62	99.75	99.44	99.93
ELM-Sig	3.76	97.18	55.56	70.21	30.05	72.62
ELM-lin	100.00	100.00	100.00	**100.00**	100.00	100.00
ELM-RBF	99.74	100.00	99.94	99.94	99.89	100.00

(b) Accuracy

	Min	Max	Mean	Median	25%	75%
WELM-Sig	1.00	1.00	1.00	**1.00**	1.00	1.00
WELM-tribas	0.95	1.00	0.99	0.99	0.98	1.00
WELM-RBF	0.98	1.00	1.00	1.00	0.99	1.00
ELM-Sig	0.00	0.95	0.51	0.75	0.01	0.84
ELM-lin	1.00	1.00	1.00	**1.00**	1.00	1.00
ELM-RBF	1.00	1.00	1.00	1.00	1.00	1.00

(c) F-measure



	WELM-Sig	WELM-tribas	WELM-RBF	ELM-Sig	ELM-lin	ELM-RBF
WELM-Sig		•	•	•	•	
WELM-tribas			•	•	•	•
WELM-RBF				•	•	•
ELM-Sig					•	•
ELM-lin						•
ELM-RBF						

(d) p-value

Table 5 Classifier AUC against different datasets applying various feature selection techniques

	WELM-Sig	WELM-tribas	WELM-RBF	ELM-Sig	ELM-lin	ELM-RBF
ORG	1	0.994607549	0.995805872	0.84263449	1	0.999402985
SMOTE	1	0.98280543	0.998171846	0.007290401	1	0.99908341
ORG-SIG	1	0.97833935	0.997004194	0.832064942	1	0.999402985
SMOTE-SIG	1	0.957295374	0.987318841	0.010471204	1	1
ORG-ULR	0.999402985	0.982434888	0.999402985	0.54280279	1	0.999402985
SMOTE-ULR	**1**	0.949868074	0.995417049	0.945895522	**1**	0.999081726
ORD-PCA	1	0.999403697	0.999402985	0.841365462	1	1
SMOTE-PCA	1	0.997255261	1	0	1	1

Table 6 Classifier accuracy against different datasets applying various feature selection techniques

	WELM-Sig	WELM-tribas	WELM-RBF	ELM-Sig	ELM-lin	ELM-RBF
ORG	100	99.22077922	99.39393939	72.9004329	100	99.91341991
SMOTE	100	98.95027624	99.88950276	9.723756906	100	99.94475138
ORG-SIG	100	96.87771032	99.56634866	71.29228101	100	99.91326973
SMOTE-SIG	100	97.34660033	99.22609176	5.970149254	100	100
ORG-ULR	99.91334489	97.48700173	99.91334489	37.52166378	100	99.91334489
SMOTE-ULR	**100**	96.84908789	99.7236042	96.79380873	**100**	99.94472084
ORD-PCA	100	99.91334489	99.91334489	72.6169844	100	100
SMOTE-PCA	100	99.83416252	100	69.87285793	100	100

Table 7 Classifier F-measure against different datasets applying various feature selection techniques

	WELM-Sig	WELM-tribas	WELM-RBF	ELM-Sig	ELM-lin	ELM-RBF
ORG	1	0.993649443	0.995823389	0.506309148	1	0.999403341
SMOTE	1	0.990923633	0.999208861	0.072741341	1	0.999084249
ORG-SIG	1	0.966675741	0.997013142	0.49300924	1	0.998417722
SMOTE-SIG	1	0.977359627	0.994462025	0.047418273	1	1
ORG-ULR	0.999403341	0.980725658	0.999403341	0.263281321	1	0.999403341
SMOTE-ULR	**1**	0.974843224	0.996978429	0.957226222	**1**	0.999082569
ORD-PCA	1	0.998417722	0.999403341	0.5	1	1
SMOTE-PCA	1	0.998813291	1	0.5	1	1

6 and 7, respectively. Therefore, SMOTE-ULR class-balanced datasets applied over WELM-Sig and ELM-lin classification algorithms both yield better AUC, accuracy and F-measure values of 1.0, 100.0% and 1.0, respectively.

Table 8 Comparison

Reference	Accuracy	AUC	F-Measure
Proposed work	**100.00**	**1.0**	**1.0**
Ma et al. [12]	–	–	0.9898
Chavan et al. [3]	97.0	0.9900	–
Garg et al. [7]	98.27	–	–
Yen et al. [23]	92.0	–	–
Martín et al. [13]	77.8	–	0.768
Saif et al. [18]	99.1	–	0.993
Martín et al. [14]	92.7%	–	0.841
Xiao et al. [22]	93.7	–	–
Wang et al. [21]	99.82	–	0.9986
Zhu et al. [25]	89.91	0.9031	–
Yerima et al. [24]	97.5	0.993	0.9742
Liang et al. [11]	–	–	0.8767
McLaughlin et al. [16]	69.0	–	0.71

4.5 Comparison

Upon comparing our proposed work with other authors as shown in Table 8, it can be observed that our technique yields better values of AUC, accuracy and F-measure with 1.0, 100.0% and 1.0, respectively. None of the authors have used weighted extreme learning machine with sigmoid activation function and extreme learning machine with linear activation function over object-oriented metrics-based SMOTE-ULR class-balanced dataset for android malware recognition.

5 Conclusion

In this paper, we present methods that identify malicious android apps. Initially, 5774 android apps were collected from various sources. These android packages are extracted for “classes.dex” file. This dex file is then decompiled, and its object-oriented metrics are retrieved using extended CKJM tool, which are used as features in machine learning. Meanwhile, SHA-256 hash information of android app is queried over VirusTotal service that provides an integer or malware tag that represents the number of antivirus companies that have marked the file as a malware. Using this information, the apps that have malware tag value as zero are deemed benign, while the rest are considered as malware. To the best of our knowledge, we are the first ones to create object-oriented software metrics-based dataset for android malware detection. The feasibility and effectiveness of using object-oriented metrics are validated using boxplots for each of the eighteen metrics against benign and malware apps. Out of 5774 android apps, 1582 are malware apps. In order to class balance this disparity, SMOTE technique is used over the original dataset (OD). Considering these two datasets, i.e., OD and SMOTE, four different feature selection algorithms, i.e., ALL, SIG, ULR and PCA, are applied to each of these datasets. Finally, we obtain a total of $(2 \times 4 = 8)$ eight different datasets, i.e., OD, SMOTE, OD-SIG, SMOTE-SIG, OD-ULR, SMOTE-ULR, OD-PCA and SMOTE-PCA. Consequently, six different machine learning algorithms, i.e., WELM-Sig, WELM-tribas, WELM-RBF, ELM-Sig, ELM-lin and ELM-RBF, to be used over our eight datasets. Now, a total of $(6 \times 8 = 48)$ forty-eight different machine-learned models are created, and its discriminatory power is evaluated using three parameters, i.e., AUC, accuracy and F-measure. These parameters are depicted using boxplots and its corresponding boxplot descriptive statistics. Our results show that SMOTE-ULR class-balanced datasets used over WELM-Sig and ELM-lin classification algorithms both yield better values of AUC, accuracy and F-Measure with 1.0, 100.0% and 1.0, respectively.

In the future, we would like to explore the possibilities of using additional object-oriented software metrics along with different machine learning algorithms. Currently, our experiment is a binary classification problem, where android apps are categorized either under benign or malware. Further, a multi-class classification model encompassing malwares of various categories will be considered.

References

1. Allix K, Bissyandé TF, Klein J, Le Traon Y (2016) Androzoo: Collecting millions of android apps for the research community. In: 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR). IEEE, pp 468–471
2. Bansiya J, Davis CG (2002) A hierarchical model for object-oriented design quality assessment. IEEE Trans software Eng 28(1):4–17
3. Chavan N, Di Troia F, Stamp M (2019) A comparative analysis of android malware. arXiv preprint arXiv:1904.00735
4. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority oversampling technique. J Artif Intell Res 16:321–357
5. Chidamber SR, Kemerer CF (1994) A metrics suite for object oriented design. IEEE Trans Software Eng 20(6):476–493
6. Desnos A, Gueguen G (2011) Android: from reversing to decompilation. In: Proceedings of Black Hat Abu Dhabi, pp 77–101
7. Garg S, Baliyan N (2019) A novel parallel classifier scheme for vulnerability detection in android. Computers Electr Eng 77:12–26
8. Halstead MH et al (1977) Elements of software science, vol 7. Elsevier, New York (1977)
9. Henderson-Sellers B (1995) Object-oriented metrics: measures of complexity. Prentice-Hall, New York
10. Jureczko M, Spinellis D (2010) Using object-oriented design metrics to predict software defects, monographs of system dependability, vol. Models and Methodology of System Dependability. Oficyna Wydawnicza Politechniki Wroclawskiej, Wroclaw, Poland, pp 69–81

11. Liang S, Du X (2014) Permission-combination-based scheme for android mobile malware detection. In: 2014 IEEE international conference on communications (ICC). IEEE, pp 2301–2306
12. Ma Z, Ge H, Liu Y, Zhao M, Ma J (2019) A combination method for android malware detection based on control flow graphs and machine learning algorithms. IEEE Access 7:21235–21245
13. Martín A, Rodríguez-Fernández V, Camacho D (2018) Candyman: classifying android malware families by modelling dynamic traces with Markov chains. Eng Appl Artif Intell 74:121–133
14. Martín I, Hernández JA, de los Santos S (2019) Machine-learning based analysis and classification of android malware signatures. Future Gener Computer Syst
15. Martin R (1994) Oo design quality metrics. An analysis of dependencies 12:151–170
16. McLaughlin N, Martinez del Rincon J, Kang B, Yerima S, Miller P, Sezer S, Safaei Y, Trickel E, Zhao Z, Doupé A et al (2017) Deep android malware detection. In: Proceedings of the Seventh ACM on conference on data and application security and privacy. ACM, pp 301–308
17. Nohl K, Lell K (2018) Mind the gap: Uncovering the android patch gap through binary-only patch level analysis. In: HITB security conference
18. Saif D, El-Gokhy S, Sallam E (2018) Deep belief networks-based framework for malware detection in android systems. Alexandria Eng J 57(4):4049–4057
19. Tang MH, Kao MH, Chen MH (1999) An empirical study on object-oriented metrics. In: Proceedings sixth international software metrics symposium (Cat. No. PR00403). IEEE, pp 242–249
20. Tirkey A, Mohapatra RK, Kumar L (2019) Anatomizing android malwares. In: 2019 26th Asia-Pacific Software Engineering Conference (APSEC). IEEE, pp 450–457
21. Wang W, Zhao M, Wang J (2018) Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network. J Ambient Intell Humanized Comput 1–9
22. Xiao X, Zhang S, Mercaldo F, Hu G, Sangaiah AK (2019) Android malware detection based on system call sequences and LSTM. Multimedia Tools Appl 78(4):3979–3999
23. Yen YS, Sun HM (2019) An android mutation malware detection based on deep learning using visualization of importance from codes. Microelectron Reliab 93:109–114
24. Yerima SY, Sezer S, Muttik I (2015) High accuracy android malware detection using ensemble learning. IET Inform Secur 9(6):313–320
25. Zhu HJ, Jiang TH, Ma B, You ZH, Shi WL, Cheng L (2018) HEMD: a highly efficient random forest-based malware detection framework for android. Neural Comput Appl 30(11):3353–3361
26. Zong W, Huang GB, Chen Y (2013) Weighted extreme learning machine for imbalance learning. Neurocomputing 101:229–242

Crime Analysis Using Artificial Intelligence

Muskan Raisinghani, Rahul Sawra, Omkar Dhavalikar, Pawan Chhabria, and Nupur Giri

Abstract The crime rate in India is increasing rapidly. According to the India Today report, the crime rate was over 540 in 2013, which increased to 581 in 2014 and then went up to almost 582 in 2015. The crime rate then dropped to 379.3 in 2016. Further, it increased to 388.6 in 2017. According to data provided by National Crime Records Bureau (NCRB), in 2018 a total of 50.74 lakh crimes were registered. Our project helps in automated crime detection and recognition of dangerous situations and crimes using three modules. Modules implemented by us help in the detection of weapons, posture analysis, and audio classification of various weapons, which further improves the process of crime detection and generates more accurate results. Detecting crimes like theft, robbery, and assault can be easily done and will increase the speed of solving crimes. Our project comprises algorithms that will notify the human administrator/director when any weapon like a gun, knife, or firearm is visible in the CCTV. Along with weapon detection, we are also implementing posture analysis and audio classification to avoid false alarms. A notification will be generated to the security personnel about any suspicious activity or a person committing a crime using real-time analysis. This idea will lead to quick and effective response times. Also, it will reduce the number of potential victims. The audio classification will help in the analysis of a crime in situations where CCTV footage is a blur or with less resolution.

Keywords Crime analysis · Automated crime detection · CCTV footage analysis · Object detection · Posture analysis · Sound classification

1 Introduction

Criminology is a procedure to recognize crimes, deviant behavior, and illegal acts. The criminals and how they behave can be determined with the available criminology techniques. Prediction of crime cannot be done since it is neither structured nor random. Also, modern technologies and hi-tech methods help criminals in attaining

M. Raisinghani (✉) · R. Sawra · O. Dhavalikar · P. Chhabria · N. Giri
Vivekanand Education Society's Institute Of Technology, Mumbai, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_3

their misdeeds. According to Crime Records Bureau crimes such as property theft, motor vehicle theft, and others were decreased, whereas crimes such as sexual assault, robbery, rape, murder, and so on were increased. Although we cannot predict that all may be victims of crime, we can predict a possible place for them to emerge. Finding patterns and trends in crime is a challenging factor. Detecting a pattern, crime analysts spend considerable time, scrutinizing the details to determine if a particular crime matches his known pattern. The information should be separated as a new pattern if it does not fit into an existing pattern. After finding a pattern, it can be used to predict, foresee, and prevent crime. Artificial intelligence-based crime analysis usually involves data collection, classification, pattern identification, prediction, and visualization. The primary objective of our project is to create a prediction model that can accurately predict crime and analyze it.

2 Related Work

Sathyadeva et al. [1] discussed the trends and patterns in crime. Their system uses data mining techniques to determine crime-prone areas and the probability of crime occurrence in specific regions.

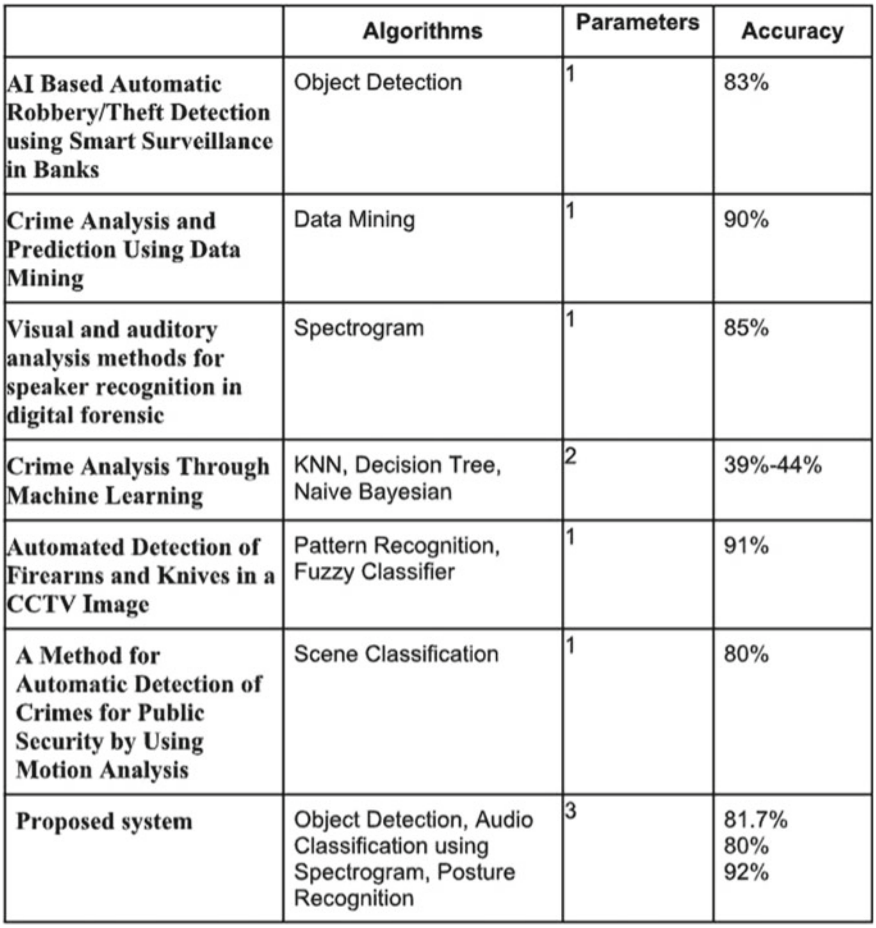
Kakadiya et al. [2] presented the use of CCTV-based theft detection using image processing. It uses object detection for real-time analysis.

Prabakaran et al. [3] discussed a number of data mining techniques which are used to analyze and subsequently predict crime. It uses various statistical models like hidden Markov model (HMM), classification model like naive Bayesian model and various clustering techniques like K-mean, K-mode clustering.

Lin et al. [4] presented a data-driven method to analyze crime data and thus predicting crime hotspots. It is based on broken windows theory and spatial analysis using various machine learning algorithms.

Kelion et al. [5] discussed that in 2019, around 14 police officials started utilizing the crime prediction program which has two kinds of software. The first software is "predictive mapping," and the other is "individual risk assessment." During predictive mapping, crime "hotspots" are arranged out, bringing about more surveillance in the zone. Whereas in the individual risk assessment, forces are trying to anticipate how likely a person is to carry out an offense or be a casualty of a criminal offense.

Faggella et al. [6] stated an article published by Daniel Faggella, it determines that there are five similar systems which uses artificial intelligence to spot, analyze, and prevent crimes. The first system by ShotSpotter helps in detecting the information about gunfire. They have used multiple sensors to collect data and have then used machine learning algorithms on that data to locate the exact location of the crime scene. One more approach that is discussed is the security cameras by Hikvision with AI which uses deep neural networks for the same. Using the camera they can perform facial recognition, scanning of license plates and also detect unattended bags in crowd. The next system is used to prevent the crime from happening in the first place. Predpol with the help of machine learning and big data tries to predict when



	Algorithms	Parameters	Accuracy
AI Based Automatic Robbery/Theft Detection using Smart Surveillance in Banks	Object Detection	1	83%
Crime Analysis and Prediction Using Data Mining	Data Mining	1	90%
Visual and auditory analysis methods for speaker recognition in digital forensic	Spectrogram	1	85%
Crime Analysis Through Machine Learning	KNN, Decision Tree, Naive Bayesian	2	39%-44%
Automated Detection of Firearms and Knives in a CCTV Image	Pattern Recognition, Fuzzy Classifier	1	91%
A Method for Automatic Detection of Crimes for Public Security by Using Motion Analysis	Scene Classification	1	80%
Proposed system	Object Detection, Audio Classification using Spectrogram, Posture Recognition	3	81.7% 80% 92%

Fig. 1 Comparison with existing systems

and where the crime will happen. According to them, they can predict the probable occurrence of crime at a particular place based on existing data of previous crimes. The fourth system by Cloud Walk Technology is trying to use gait analysis and face recognition techniques to predict the probability that an individual will commit crime and thus prevent it in advance. The last system in the article uses five years of criminal data to predict the level of threat that criminal holds to society and determine the probability of criminal committing another crime in future.

Grega et al. [7] proposed the automated knives and firearms detection using SVM classification and image recognition techniques (Fig. 1).

3 Crime Analysis Procedure

In general, police authorities are notified very late when a crime occurs, and even if they detect crime on time it becomes very difficult for them to analyze everything quickly. This is where our model comes into play. Our model can easily detect a crime and perform specific analysis which will make catching criminals very easy. In our system, we are detecting dangerous and deadly weapons which ensures that when a criminal is using a weapon our system will quickly identify it. Noises and postures, which are very common in crime scenes, are being detected too. In the end, using these three techniques of object detection, posture analysis, and sound analysis a report is generated too, which will help the authorities evaluate the situation very quickly and swiftly. The crime analysis can be performed using a procedure as shown in Fig. 2 which determines the task performed by each module for crime prediction. The input data is given to the preprocessor which performs the preprocessing. Once

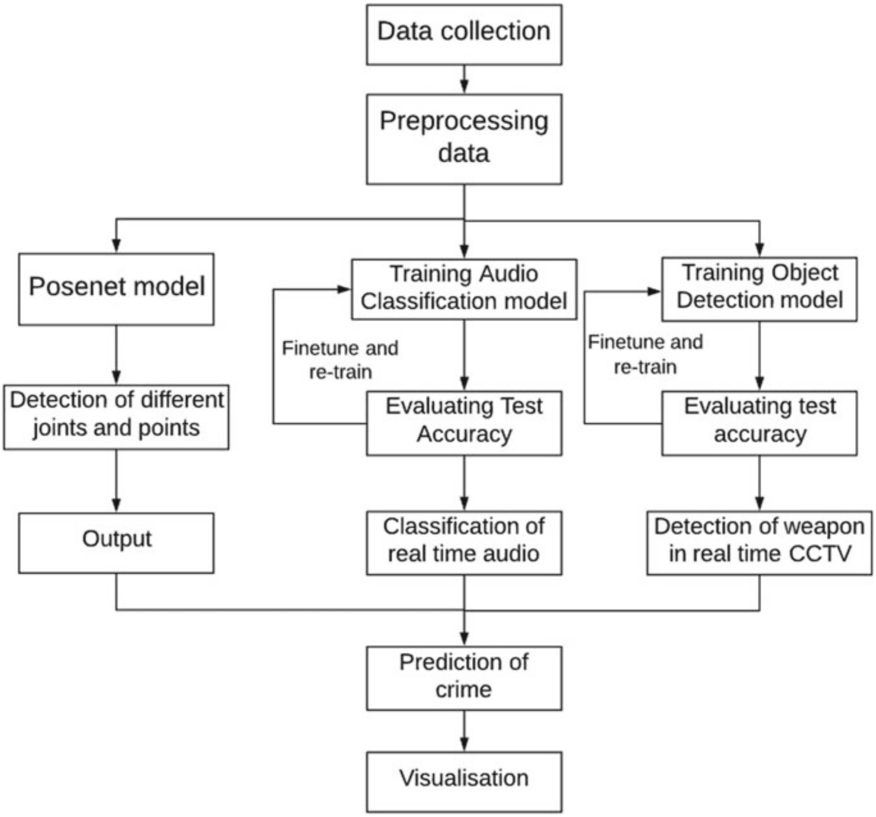

Data collection
Preprocessing data
Posenet model
Training Audio Classification model
Training Object Detection model
Finetune and re-train
Finetune and re-train
Detection of different joints and points
Evaluating Test Accuracy
Evaluating test accuracy
Output
Classification of real time audio
Detection of weapon in real time CCTV
Prediction of crime
Visualisation

Fig. 2 Block diagram

preprocessing is done, the preprocessed data for respective models are given as input as shown in Fig. 2. We have further explained the individual model in Sect. 4.

4 Methodology Used

To achieve the end result, we wanted to map out and execute an approach that was realistic, accurate, and efficient. To do so we divided our project into four sections which are object detection, audio analysis, posture detection, and report generation. The first three sections, i.e., object detection, audio analysis, and posture detection functions individually, and later analysis done by these three sections are combined for report generation during the end stages. More information about these sections is described as follows:-

4.1 TensorFlow Faster RCNN Object Detection

Here, we are using the TensorFlow faster RCNN object detection model to detect weapons that can be used during the crime. A good example of this can be pistols, rifles, and knives. There were many other models that were under consideration while selecting the most appropriate model like ResNet, MobileNet, and inception. Out of those we ended up selecting TensorFlow faster RCNN object detection model. The reason to use it was the accuracy that everyone gets from this particular model. It is really high compared to other models available for object detection. The downside of using this model is slow processing as it takes more time but this can be improved using higher processing power. In this model, first anchors (labeled boxes) are defined during the generation of the data set. Later, the data set is passed through region proposal network and region of interest algorithms where all the magic happens (Fig. 3).

From the above image, we can see that faster RCNN has been updated for faster processing compared to its predecessors. The data set contains 15,000 images

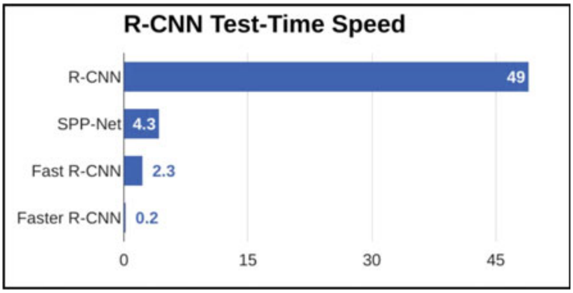


Fig. 3 Comparison of different object detection algorithm's test time speed (Rohith Gandhi, July 2018)

distributed among five classes, composed of 80:20 ratio for training and testing, respectively. The total number of steps for training was 60,000, and the accuracy we obtained was around 81.7%.

4.2 TensorFlow Inception V3

In this section, we are going to analyze the audio to detect if we can find a crime is happening. Few examples of sounds that we are going to detect are screams and gunshots. A spectrogram is visual representation audio in the form of images. Mel scale is used for a nonlinear transformation of the scale. This helps us segregate sounds that have small differences in frequency and which are hard to distinguish usually. Using mel scale, we can generate a mel-frequency cepstral coefficient (MFCC) spectrogram. Using the MFCC spectrogram, we can get the most accurate and unique visual representation of sound. We convert all our audio data sets to MFCC spectrogram images, and these images will be passed to an image classifier, which will eventually be used to detect voices related to crime. For image classification, TensorFlow Inception V3 model turned out to be appropriate from historical audio detection projects (Fig. 4).

The total number of steps for training was 5000, and the accuracy we obtained was around 80%.

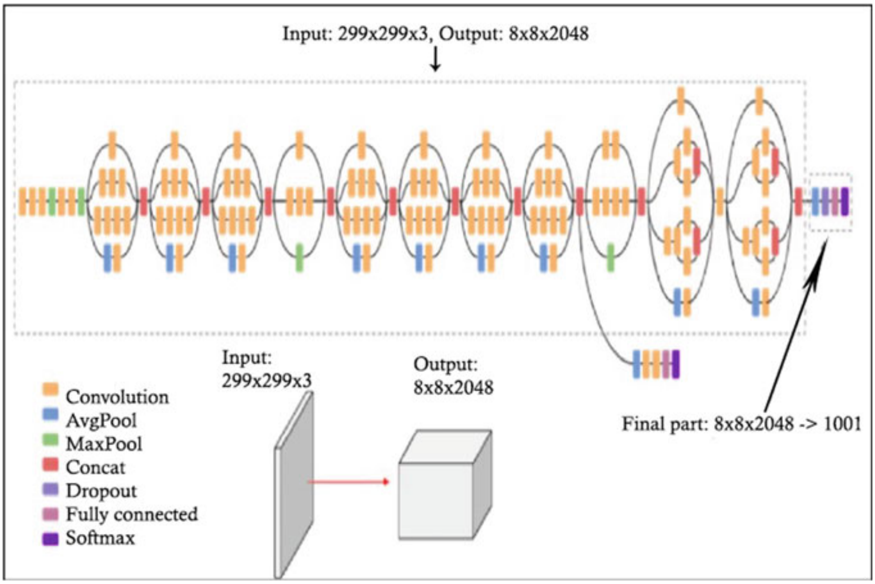


Fig. 4 A high-level diagram of TensorFlow Inception V3 model (Will Nowak, Oct 2018)

4.3 TensorFlow PoseNet

This section helps us to detect and analyze real-time postures that are common at crime scenes. A good example of this would be the hunchback position during bank robberies. Here, we are using TensorFlow PoseNet. In this model, seventeen unique joints can be detected on a human body. MobileNetV1 and ResNet50 are the two architectures supported by the model. The output obtained is the coordinates along with their score. For example,

```
{
        ‘‘position’’: {
          ‘‘y’’: 72.848854061245,
          ‘‘x’’: 263.08151234453
        },
        ‘‘part’’: ‘‘leftEar’’,
        ‘‘score’’: 0.8402985332129
     },
```

4.4 Report Generation

This is the last step of this project. Here analysis done by the above models is used to generate reports which can be read by the authorities to analyze the crime in a much quicker way as our model did most of the analysis very quickly compared to the time taken by a human to do such analysis. In the report generated by the model, the analyzed video is divided into three parts, namely pre-crime phase analysis, during-crime phase analysis, and after-crime phase analysis. This report will help in the future for the prediction of crimes.

5 Results

5.1 Weapon Detection

The below image of a gun is being detected using our object detection model. This module was trained on the TensorFlow faster RCNN model which can detect pistol, rifle, sword, stick, and knife on given images and real-time videos with 81.7% accuracy (Fig. 5).

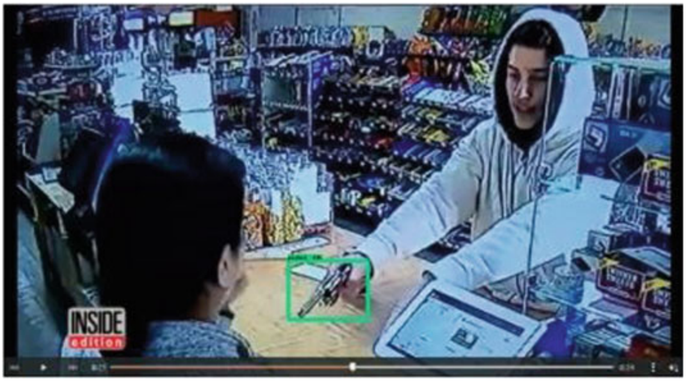


Fig. 5 Frame from the trained weapon detection model

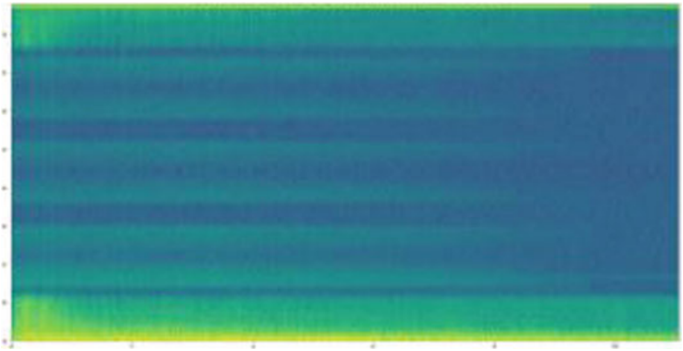


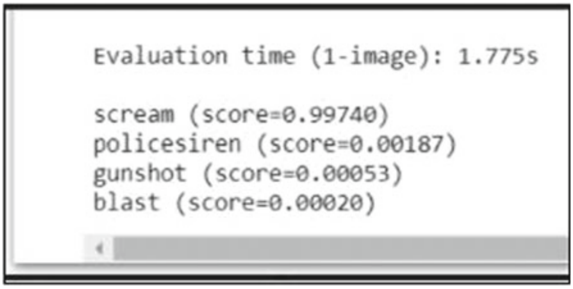
Fig. 6 Spectrogram of audio from the audio data set

5.2 *Audio - Spectrogram Conversion*

Figure 6 is an image of the MFCC spectrogram, which was generated using input audio. Such images are used for training and detecting audio. Figure 7 is the output of our model detecting scream audio. This module was trained by converting audio files of a gunshot, scream, blast, and police siren to MFCC spectrogram images and then training them on TensorFlow Inception V3 Classification model which was able to classify audio files with 80% accuracy.

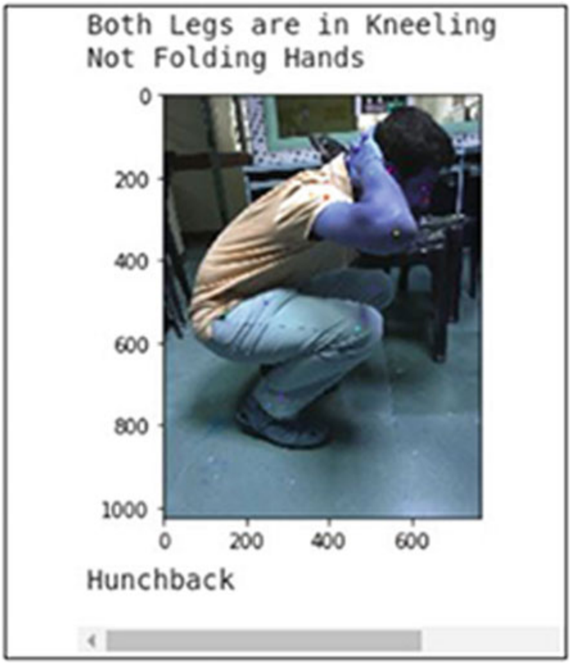
5.3 *Posture Analysis*

Figure 8 is an image of output from our posture detection model. Here, our model was able to detect a hunchback position which is very common in crime scenes including



```
Evaluation time (1-image): 1.775s

scream (score=0.99740)
policesiren (score=0.00187)
gunshot (score=0.00053)
blast (score=0.00020)
```

Fig. 7 Results from audio classification



```
Both Legs are in Kneeling
Not Folding Hands
```

```
Hunchback
```

Fig. 8 Position of hunchback is detected indicating the possibility of crime

hostages. This model was trained using the TensorFlow PoseNet model which can recognize hunchback, kneeling, hand folding, firearm position with 92% accuracy.

6 Conclusion

After rigorous analysis and considerations of certain required constraints, we were able to select proper machine learning models and technological stack which led us to make an efficient and accurate system. This system will help us to analyze crime using video and audio processing. The integration of three modules, i.e., object detection, posture analysis, and audio classification, has increased the accuracy of the system and thus will help to reduce the number of crimes. The whole cumbersome process of some authority analyzing crime by watching video evidence can be done way quickly by this model. For future scope, we can include profiling, criminal tracking, and facial recognition in this model. Criminals can be tracked using data obtained from previous analysis and by monitoring other CCTV cameras in the vicinity making the work of the police easier and thus generating the trail of the escape route by criminals. Facial recognition can be used to identify criminals and other individuals involved in a crime scene.

References

1. Sathyadevan S, Devan MS, Surya Gangadharan S (2014) Crime analysis and prediction using data mining. In: 2014 First international conference on networks & soft computing (ICNSC2014), Guntur, pp 406–412
2. Kakadiya R, Lemos R, Mangalan S, Pillai M, Nikam S (2019) AI-based automatic robbery/theft detection using smart surveillance in banks. In: 2019 3rd international conference on electronics, communication, and aerospace technology (ICECA), Coimbatore, India, pp 201–204
3. Prabakaran S, Mitra S (2018) Survey of analysis of crime detection techniques using data mining and machine learning. In: National conference on mathematical techniques and its applications (NCMTA 2018). IOP J Phys Conf Ser 1000
4. Lin YL, Yu LC, Chen TY (2017) Using machine learning to assist crime prevention. IEEE 6th International Congress on Advanced Applied Informatics (IIAI-AAI), Hamamatsu, Japan
5. Kelion L (ed) (2019) Crime prediction software. Adopted by 14 UK Police Forces. BBC News, 4 Feb 2019. https://www.bbc.com/news/technology-47118229
6. Faggella D (2019) AI for crime prevention and detection. 5 Current Applications [Web log post]. Retrieved 24 Oct 2020 from https://emerj.com/ai-sector-overviews/ai-crime-prevention-5-current-applications/
7. Michał G, Andrzej M, Piotr G, Mikołaj L (2015) Automated detection of firearms and knives in a CCTV image. Sensors 16:47. https://doi.org/10.3390/s16010047

A Novel Architecture for Binary Code to Gray Code Converter Using Quantum Cellular Automata

Mummadi Swathi and Bhawana Rudra

Abstract In CMOS, the channel length is sinking day by day which raises a lot of questions about its future. Quantum dot computation is an alternative solution to the CMOS technology, which has the strength to increase the speed of computations and reduce the power while performing those computations as well as it reduces the area when compared to CMOS technology. To perform computations using quantum, we generate arithmetic circuits where code converters play a significant role. In this paper, we are discussing 2-, 3-, and 4-bit binary to gray code converters that are designed with a minimum number of qubits using 0.0251, 0.0382, 0.06 µm^2 area respectively.

Keywords Quantum technology · Qubit · QCA · XOR · Majority gates · Code converters

1 Introduction

CMOS transistor works are based on the lithography and masks. Lithography is the main technology behind the transistor scaling and fails with the shrinking feature of CMOS. For the 180nm technology, the masks without optical proximity correction are good enough to pattern the devices using CMOS technology. As it moves below 100nm, the complexity increases and more advanced techniques are required for the development of masks. Although it is developed, how it works remains the future question. Some experiments were performed for developing below 20 nm using various techniques [1, 2] and proved to be difficult in mask making and pattern matching. 248 nm radiation was used to develop a mask of 9nm device but proved to be uneconomical. Not only these, the others include when moving toward the small channel are Physical, Power, Material, Economical and Technological Challenges. Some non-radiation techniques like nanoimprint lithography are attractive, and lack of investment made the experts to shift to the newer techniques that are cost effective.

M. Swathi (✉) · B. Rudra
National Institute of Technology Karnataka, Surathkal, India
e-mail: bhawanarudra@nitk.edu.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_4

A brief explanation of the Alternative technologies to the CMOS are as follows.

Single Electron Transistor technology (SET) is based on quantum effects like tunneling and Coulomb blockade effect [3]. It is used for charging sensor applications infrared radiation detector, ultra-sensitive microwave detector, etc., The limitations of SET are lithography and tunneling effect and consists of other problems too like high output impedance, low gain and background charges at room temperature.

Carbon Nanotubes are thin and long carbon cylinders that came into existence in 1991 by Sumio Iijima [4]. These have a big range of structural, thermal and electronic properties that depend on various types of nanotubes. It consists of multiple walls, i.e., cylinders inside the other cylinders. They are less than 100 nm in diameter and can be very thin as 1 or 2 nm. These can be applied in many areas like CNT fibers, energy storage. But this technology is based on equipment that is costly and hard to handle due to its small size.

Fin shaped Field Effect Transistor(FinFET) is a multigate device, built on a surface where the gate is laid on 2, 3, or 4 sides of the channel to form a double gate structure. These devices are called as "FinFETS" because the source or drain area forms fins on the silicon plane [5]. It offers higher performance over the currently existing planar devices. Current Density and Switching time of FinFET devices are high compared to CMOS technology.

Gallium Nitride (GaN) technology is of high cost due to material used in manufacturing. Currently, small signal MMIC and LNA markets are dominated by GaAs devices [6]. It will take some time due to its cost. Still, the research is in the process to reduce the cost. Once the cost is reduced it captures the market in various domains like wireless, medical, automobile, etc.

Gallium Arsenide (GaAs) technology is made up of 2 materials called Gallium and Arsenic. Gallium is rare to get and is a by-product of aluminum and zinc so costlier than gold whereas arsenic is poisonous. A single Crystal GaAs substrate is of high cost. These are more brittle and are of small size.

Floating Gate MOSFET(FG MOSFET) is electrically separated by developing a node in DC(Direct Current) and a number of secondary inputs or gates stored over the floating gate. It is enclosed by highly resistive material and also charges will not be changed for a long time [7]. **MESFET** The structure of MESFET is the Schottky metal gate, it uses a large number of enhancement mode transistors [8].

Silicon on Insulator (SOI) technology SOI Transistors are designed on a silicon layer on a buried oxide. The buried oxide makes total dose response more complex, and hardening compared to bulk silicon ICs [9]. If proper hardening of SOIs is considered then their significant performance and hardness will become an advantage that can be used in space and nuclear environments.

The major limitations of CMOS are power dissipation, sinking of the channel length in the usage; obstruct the momentum of microelectronics using regular circuit scaling made the digital industry to move toward alternative technologies. Shrinking the CMOS size and reducing power consumption became a very big challenge for designers that lead to finding a new technology, i.e., QCA, which gives high performance with low power consumption. This is an advanced access toward the era of nanotechnology and offers a new design with the use of logic gates. This proved to be effective in increasing speed, reduced power, and area consumption compared to CMOS technology. QCA allows the frequencies of THZ [10] which is not possible in available CMOS technologies. It is a rising technology for the future which consumes low power and gives effective switching speed and density structure. Douglas et al. [11] implemented basic logic gates, inverters, XOR and programmable logic gates using complex arrays and coupled Quantum dot cells. CMOS technology has some restrictions on the physical structure and it also increases the current leakage. As a solution to this problem [12–14] QCA technology was used with reversible gates. Authors implemented R, TR, NFT gates with smaller size using reversible logic. Similarly, authors designed structures for flip flops [15–18] and XOR gates [19–22]. XOR gates are useful in error detection and correction operations. Many authors implemented various combinational and sequential circuits like adders [23–31], memory circuits [32], Parity generator and checker circuits [33], Multipliers [34] and Shift registers [35] using QCA technology. QCA with Power and Energy dissipation [36, 37] gives better results. QCA with the majority gate results in the energy-efficient circuits. QCA in detail is explained in further section.

2 Fundamentals of QCA

Quantum dot computing is a cutting-edge technology, it not only alternates the CMOS technology but also improves the effects of parameters in all aspects. Devices which are designed by using quantum mechanical concepts increase the speed and reduce the size. The implementation of many quantum devices is same as classical devices, which encodes the data using voltage and current. The device architecture is conventional even though the operations are implemented using quantum physics. It is proven that using quantum designs in conventional structures leads to the problem due to extremely small quantum devices. The output of such devices will be of nanoamperes current which leads to the change in input voltage with several millivolts. Another problem is interrelated wires capacitance which influences the device functioning. Quantum Cellular Automata (QCA) [38–40] is a new cellular automata which is a combination of Coulomb-coupled quantum devices. We generally use quantum bits for information processing which are generally known as qubits. Qubits contains quantum dots which are interconnected by the electrons in each cell [12, 15, 19]. Electrons can shift to various quantum dots with the help of electron tunneling. Coulomb repulsion separates the electrons and moves to the corner area. The cell state is termed as polarization. Polarization is either ‘1’ or ‘0’ that is used to represent information in binary format. Qubit also has some similarities like binary representation of ‘0’ and ‘1’ but the difference is that the state of qubit is a superposition of both 1 and 0 and qubit stores more information. In other words, it will be in ‘ON’ and ‘OFF’ state at the same time. Based on the polarization angle, a spin of an electron in quantum information transmission takes place. QCA can be used for designing a combinational as well as a sequential circuit. Polarization is used to arrange the cells one after the another which leads to the quantum wire. This wire generates the output by transferring the signal from one end to another.

2.1 Superposition Theorem

In classical computer the data will be measured in the form of classical bits but in quantum computer it is measured in the form of qubits. A classical bit is either in 0 state or 1 state at a time but a qubit can be in both the states at a time; i.e., it is a superposition of both the states. Superposition of a qubit is represented as $|\phi\rangle = \alpha|0\rangle+\beta\ |1\rangle$. Here α, β are complex numbers. In general, the Bloch Sphere represents the superposition of a qubit which is shown in Fig. 1. Superposition is defined as the occurrence of both the quantum states at the same time.

The output of a qubit can be calculated based on the probability of the occurrence of the logical value based on rotation. As we know in qubits there are 2 states which are $|0\rangle$ state and $|1\rangle$ state which are called ‘basis vectors.’ These basis vectors are represented in matrix form, i.e., $|0\rangle=\binom{1}{0}$ and $|1\rangle=\binom{0}{1}$. One of the major applications of qubits is the way quantum dots are represented in the logical circuits using a group of cells. The electron spin is the deciding factor for output logic where the up spin represents logic 0 and down spin represents logic 1. The distance should be adequate for the electron to tunnel from one dot to another. Logic states in the qubit are decided by the polarization input shown in Fig. 2.

Fig. 1 Bloch sphere

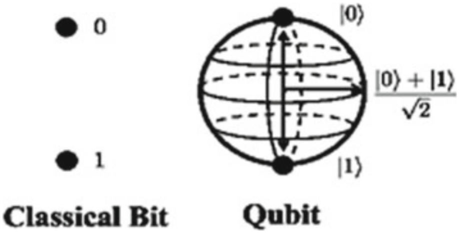
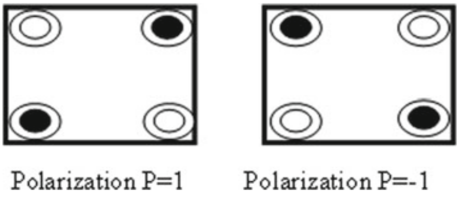


Fig. 2 Basic qubit cells



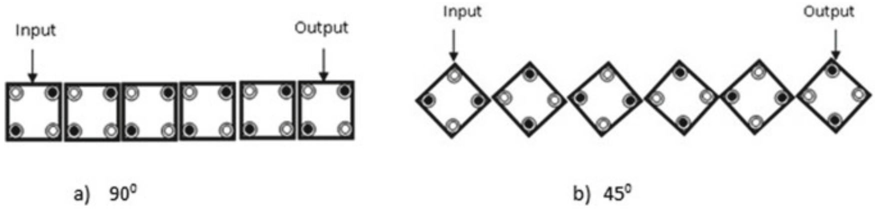


Fig. 3 QCA wire logic

2.2 QCA Wire Logic

It is a bundle of interconnecting cells. These are used to transfer polarization state. QCA wire can be made up of 90° cells and 45° cells shown in Fig. 3. For example if the input is 1 then the same value will be transferred to the output because the cells are connected in series. Hence the QCA 90°, 45° wire logic transfers the data from input to output.

The formal arrangement of QCA cell pattern is using a binary wire. For the electrostatic communications between the cells, the signal propagated from one side to another [41]. When the bias voltage is zero, dot is neutral and fermi levels of both source and drain are in equilibrium state [18]. When bias voltage, such as source–drain voltage (V_{ds}) and gate voltage (V_g) are applied, charge carriers will tunnel through tunnel junction from source to drain via dot [18, 23]. This event creates a charged dot because of its interaction with source (Q2), gate (Q3), and drain (Q1) [37]. Based on the polarization angle and spin of an electron, the quantum information transmission takes place. Code converters are efficient in the security department for devising and cracking codes.

2.3 QCA Implementation of Code Converters

In this paper we are implementing binary to gray converter. **Binary code** represents the data in the form of 0s and 1's. **Gray code** is a reflected binary code. In gray code, each value differs only by a single bit from the previous bit. Gray code can be used in simplifying fault correction, terrestrial television, some cable television system, analog-to-digital converter and peripheral apparatus. *Inverter logic:* The electrostatic force of attraction between the adjacent cells allows to implement the logic circuit by changing the placement of the cell in the layout. The inverters can be realized by placing the cells in corners to contact. The quantum dots in the cell are misaligned between the cells are inverted. The 45^0 displacement as in Fig. 4. of combined cells will generate the complement action of the input signal. In Fig. 5, the input A is sent to the 1st cell, and from 1st cell it is transferred to 4th, 5th cells and 8th, 9th cells. These cells will get into contact with the corners of the next cell and change

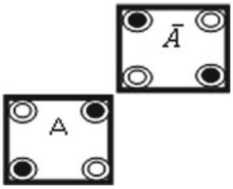


Fig. 4 Basic inverter using 2 Qubits

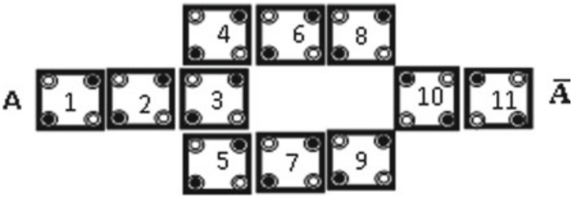


Fig. 5 Inverter representation using majority gate

its polarization because of 45° displacement and results inverted value, and this can be sometimes called as fork inverter.

QCA Majority Gate: Ali Newaz Bahar et al. [13, 42–46] have experimented on XOR, Half subtractor, Full subtractor and yielded better results than CMOS using majority gate. Majority gate works based on the majority of the inputs. It consists of 0's and 1's where we consider 2n+1 inputs (where n=1 to n) to decide one output.

3-Input Majority Gate (MV3): QCA MV3 is represented with five cells as shown in Fig. 8. The states of the top and bottom cells are fixed, while the center(Device Cell) and right cells will react to the fixed charge. In the particular case for input 110, two inputs are in the "1' state, and the other is in the "0' state. When we solve it for the ground state the majority will match to the state "1', which gives the result as true for all combinations of the 3 inputs. Similarly, for input 001, it gives result as 0 because of majority 0's. The logic symbol, circuit diagram and QCA design are represented in Figs. 6, 7 and 8. MV3 also works like 2-input AND gate if we place any input as logic 0, i.e., in MV3 if $C = 0$ then it works like logic AND, i.e., A.B and for C=1 it works like logic OR, i.e., A+B. Table 1 shows the different input combinations majority gate.

$$\begin{aligned} \text{MV3} &= AB + BC + CA \\ \text{MV}(A, B, 0) &= AB + 0 + 0 = AB \\ \text{MV}(A, B, 1) &= AB + B + A = B(A + 1) + A = B + A = A + B \end{aligned}$$

(Because of Annulment law A+1 = 1 and Cumulative law A+B = B+A)

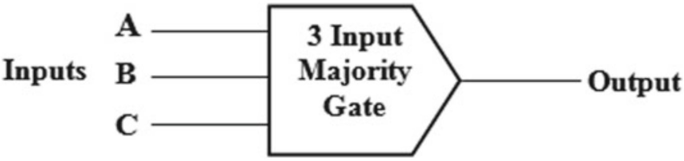


Fig. 6 MV3 logic diagram

Fig. 7 MV3 circuit diagram

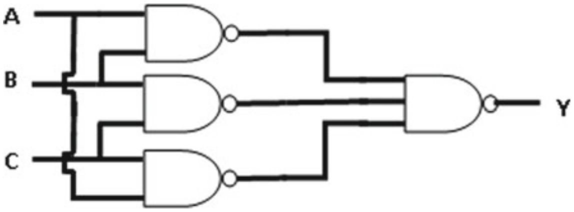
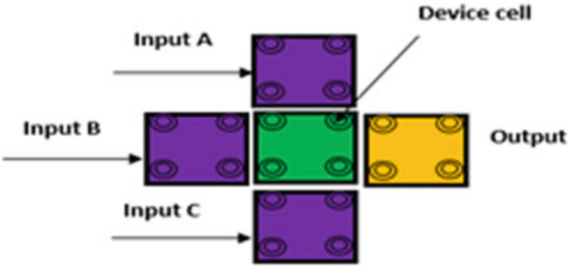


Fig. 8 MV3 QCA design



5 Input Majority Gate (MV5): As shown in Fig. 9, 5 inputs will be given to the gate and based on the majority of inputs it produces the output. The expression for 5 input majority gate is

Table 1 3-Input majority gate

S. No.	Inputs A B C	Majority gate output
1	0 0 0	0
2	0 0 1	0
3	0 1 0	0
4	0 1 1	1
5	1 0 0	0
6	1 0 1	1
7	1 1 0	1
8	1 1 1	1

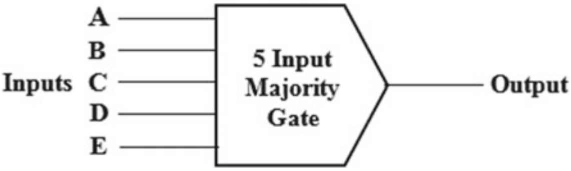


Fig. 9 MV5 logic symbol

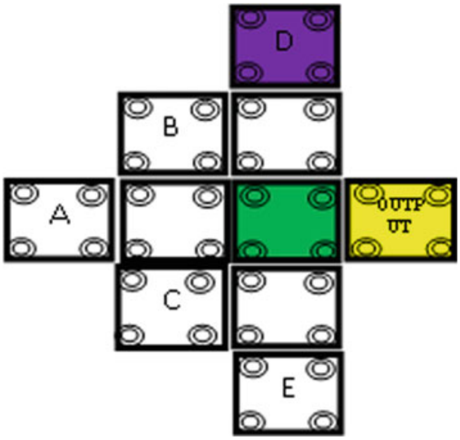


Fig. 10 MV5 QCA design

$$\begin{aligned} \text{MV5}(A, B, C, D, E) = {} & ABC + BCD + CDE + ABD + BDE \\ & + ABE + ACD + ADE + ACE + BCE \end{aligned}$$

In this expression, A, B, C, D, E are 5 inputs and based on the majority it produces the output for example if inputs are 1, 0, 1, 0, 0 then the output will be 0 because the majority of inputs are 0's. QCA design for MV5 gate is shown in Fig. 10.

3 Proposed Work

To design code converters we need XOR gates. If we implement XOR with fewer qubits then that can be used to implement efficient code converter circuits. In this paper, we implemented XOR gates using 3 and 5 input majority gates with less number of qubits and it is used to implement 2-, 3-, and 4-bit binary to gray code converter circuits.

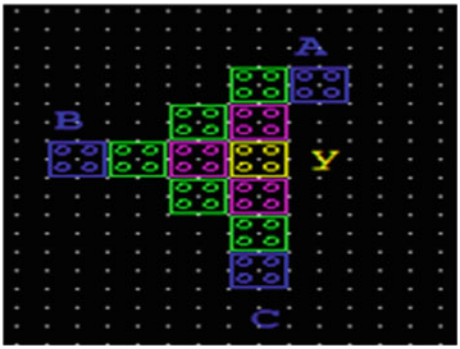


Fig. 11 3-Input XOR gate

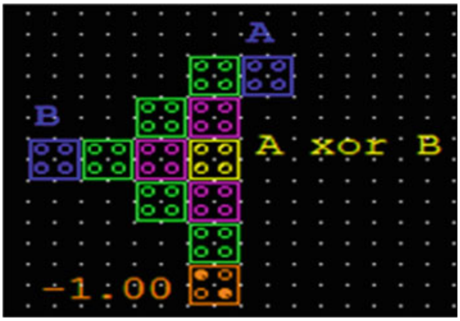


Fig. 12 2-Input XOR gate from 3-Input XOR gate

3.1 Novel XOR Gate

Designing a high efficient circuit that consumes less power is made easy with the help of XOR gates, it has many applications in digital circuits. The 3-input and 2-input XOR are highly efficient and very simple in structure. Due to its simple structure and efficient design, we used 2- and 3-input XOR gates to design our B to G converter. This design does not need any wire crossing and implemented on a single layer with only 12 cells. Figure 11 represents the 3-input XOR gate with three inputs A, B, C and one output Y. We can also perform 2-input XOR operation with the same design as shown in Fig. 12.

3.2 Binary to Gray Code Converter

Code converters are used to represent numbers in different ways. Binary to gray code converter is also one of the techniques where binary bits are converted into gray code format. 2-, 3-, and 4-bit binary to gray code converter circuit diagrams and truth tables are shown below.

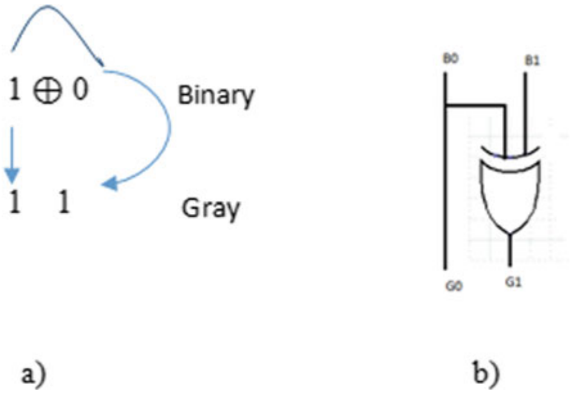


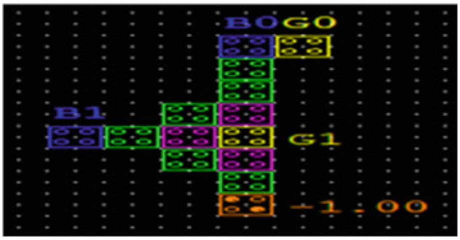
Fig. 13 **a** Code conversion procedure, **b** circuit diagram

2 bit Binary to Gray code converter: An example of 2 bit binary number 10 conversion into gray using XOR operation is shown in Fig. 13a. As represented in the figure, while converting binary to gray, MSB will be written as it is and for the next gray bit we need to perform XOR operation on successive bits, if both bits are same then the output of XOR operation will be 0 otherwise 1. For 3- and 4-bit binary to gray conversion also MSB will be written as it is and XOR operation will be performed on successive bits. As shown in Logic diagram Fig. 13b, to design 2 bit binary to gray converter we need one XOR gate. Here we used our novel XOR gate to reduce the number of cells for implementing the converter circuit. All possible combinations with 2 bits and its gray representation are shown in Table 2. In this paper, we implemented 2 bit binary to gray converter using 14 cells as shown in Figs. 14 and 15 which represent output waveform.

Table 2 2 bit code converter truth table

S. No.	Binary code B0 B1	Gray code G0 G1
1	0 0	0 0
2	0 1	0 1
3	1 0	1 1
4	1 1	1 0

Fig. 14 2 bit code converter QCA design



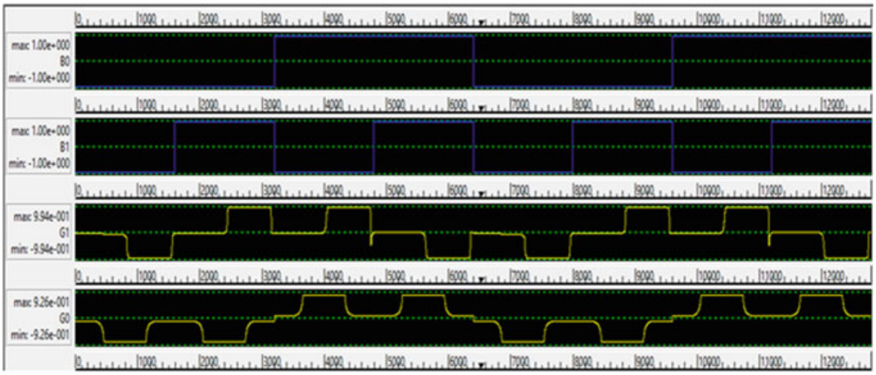
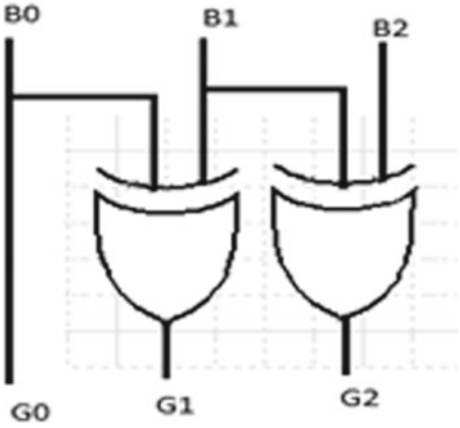


Fig. 15 2 bit code converter output waveform

Fig. 16 3 bit code converter circuit diagram



3 bit Binary to Gray code converter: 3 bit binary to gray converter required 3 inputs B0, B1, B2 which are passed to two XOR gates as shown in Fig. 16. and produces three outputs G0, G1, G2. Table 3 shows all possible 3 bit binary combinations and their gray representations. In this paper, we designed 3-bit binary to gray converter using 25 cells as shown in Fig. 17 and Output waveform represented in Fig. 18.

4-bit binary to gray code converter: In this 4-bit code converter, four inputs B0, B1, B2, B3 will be passed through three XOR gates and it produces the four outputs G0, G1, G2, G3 as shown in Fig. 19. All the possible combinations are shown in Table 4. Here we designed 4-bit binary to gray converter using 39 cells as shown in Fig. 20 and Output waveform represented in Fig. 21.

Table 3 3 bit code converter truth table

S. No.	Binary code B0 B1 B2	Gray code G0 G1 G2
1	0 0 0	0 0 0
2	0 0 1	0 0 1
3	0 1 0	0 1 1
4	0 1 1	0 1 0
5	1 0 0	1 1 0
6	1 0 1	1 1 1
7	1 1 0	1 0 1
8	1 1 1	1 0 0

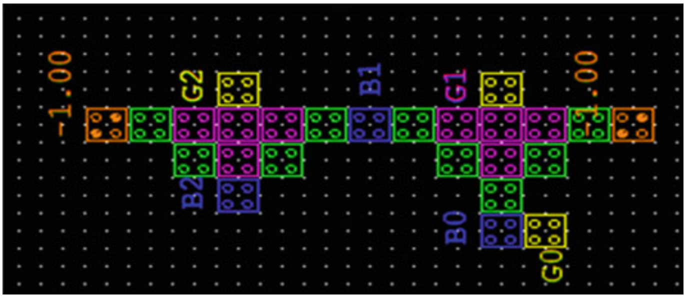


Fig. 17 3 bit code converter QCA design

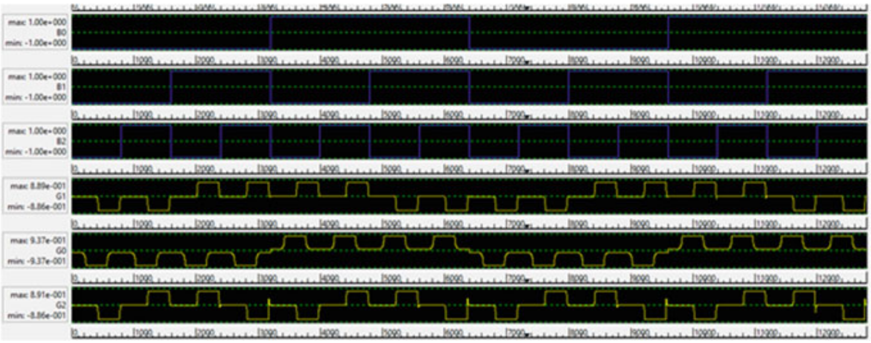


Fig. 18 3 bit code converter output waveform

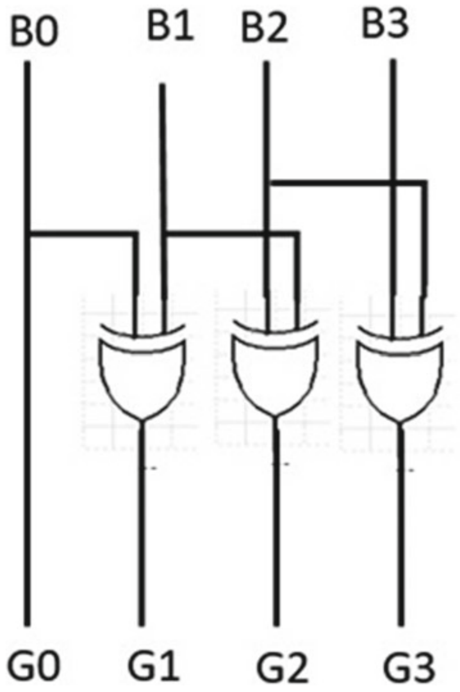


Fig. 19 4-bit code converter circuit diagram

Table 4 4-bit code converter truth table

S. No.	Binary B0 B1 B2 B3	Gray G0 G1 G2 G3
1	0 0 0 0	0 0 0 0
2	0 0 0 1	0 0 0 1
3	0 0 1 0	0 0 1 1
4	0 0 1 1	0 0 1 0
5	0 1 0 0	0 1 1 0
6	0 1 0 1	0 1 1 1
7	0 1 1 0	0 1 0 1
8	0 1 1 1	0 1 0 0
9	1 0 0 0	1 1 0 0
10	1 0 0 1	1 1 0 1
11	1 0 1 0	1 1 1 1
12	1 0 1 1	1 1 1 0
13	1 1 0 0	1 0 1 0
14	1 1 0 1	1 0 1 1
15	1 1 1 0	1 0 0 1
16	1 1 1 1	1 0 0 0

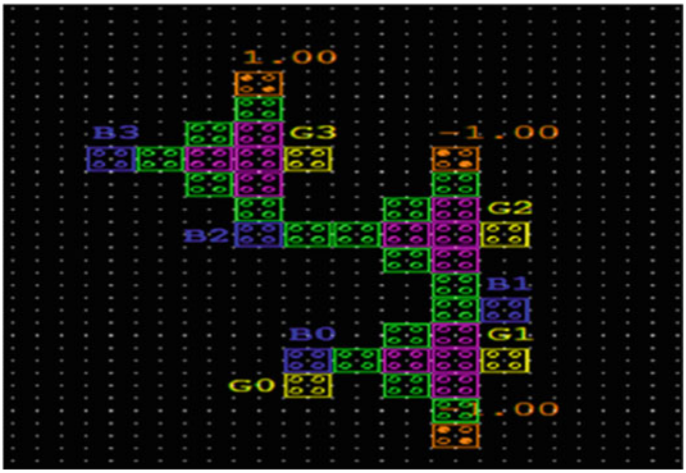


Fig. 20 4-bit code converter QCA design

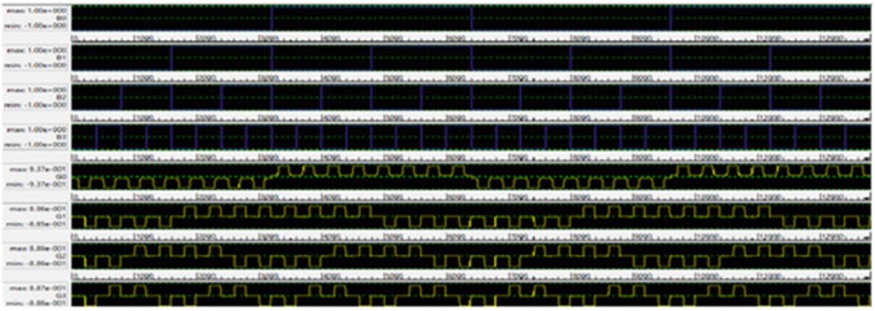


Fig. 21 4-bit code converter output waveform

4 Results

The proposed circuits have reduced cell count and area by using our novel XOR gate. Compared with all the parameters of existing work, the proposed design took less number of cells and achieved accurate results with less area for the conversion of binary to gray code. Parameter Comparison Charts and Tables are shown below.

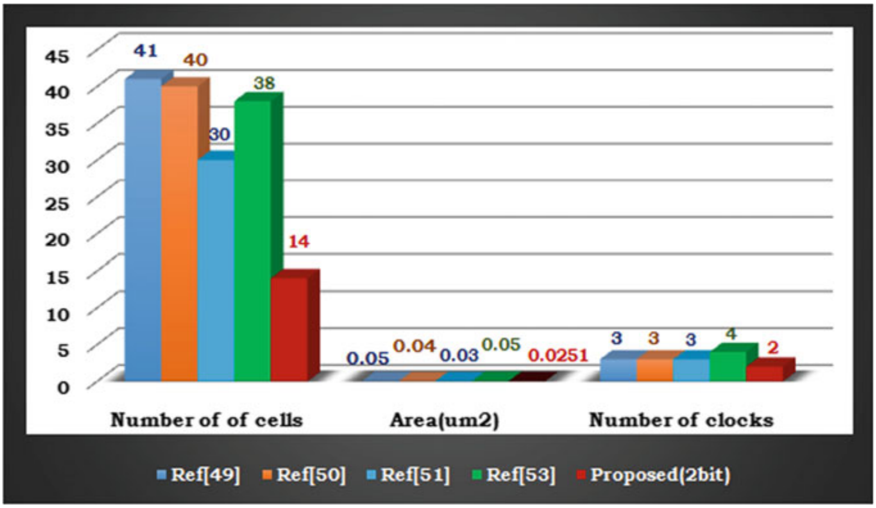


Fig. 22 2 bit code converter parameters comparison chart

4.1 2 Bit Binary to Gray Code Converter Parameter Comparison Chart

As shown in Fig. 22, the proposed 2 bit code converter designed with 14 cells, 0.0251μm^2 area which is very less compared to the existing designs [47–50].

4.2 3 Bit Binary to Gray Code Converter Parameter Comparison Chart

As shown in Fig. 23, the proposed 3 bit converter designed with 25 cells, 0.0382 μm^2 area which is very less compared to the existing designs [47–50].

4.3 4-Bit Binary to Gray Code Converter Parameter Comparison Chart

As shown in the following Fig. 24, the proposed 4-bit converter designed with 39 cells, 0.06μm^2 area which is very less compared to the existing designs [47, 48, 50, 51].

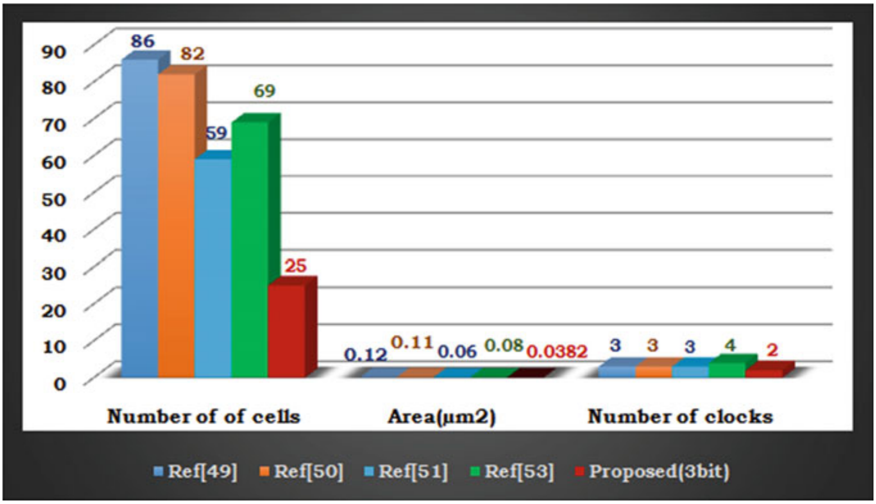


Fig. 23 3 bit code converter parameters comparison chart

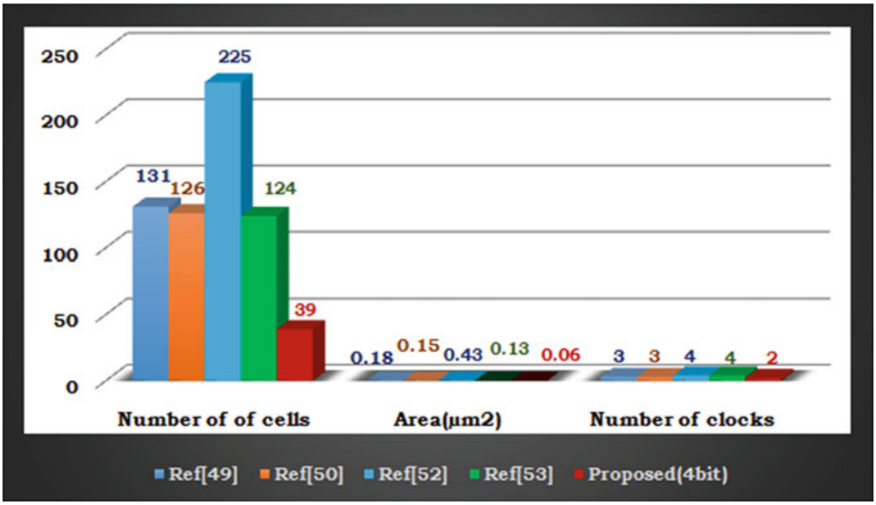


Fig. 24 4-bit code converter parameters comparison chart

5 Conclusion

We presented Novel XOR gate and QCA based 2, 3 and 4-bit binary to gray code converters. The proposed designs reduce the number of cells and area and increase switching speed. A comparative study has been performed and proved that the number of clocks for 2, 3 and 4-bit binary to gray code converter is less compared to the available results. The proposed designs can also be used in quantum computing, nanotechnology and digital signal processing to implement power and area-efficient circuit architectures.

References

1. Moore GE (1965) Cramming more components onto integrated circuits, Reprinted from Electronics 38(8). EEE SSCS NEWSLETTER, Sept 2006.https://doi.org/10.1109/N-SSC.2006.4785860
2. Ali Razavieh PZ, Nowak EJ (2019) Challenges and limitations of CMOS Scaling for FinFET and beyond architectures. IEEE Trans Nanotechnol. https://doi.org/10.1109/TNANO.2019.2942456
3. Kumar O, Kaur M (2010) Single electron transistor: applications & problems. Int J VLSI Design Commun Syst 1(4). https://doi.org/10.5121/vlsic.2010.1403
4. Pitroda J, Jethwa B, Dave SK (2016) A critical review on carbon nanotubes. Int J Constr Res Civil Eng (IJCRCE) 2(5). https://doi.org/10.20431/2454-8693.0205007
5. Bibin Lawrence R, Jency Rubia J (2015) Review of Fin FET technology and circuit design challenges. Int J Eng Res Appl 5(12). ISSN: 2248-9622
6. Vecchia MD, Ravyts S, Van den Broeck G, Driesen J (2019) Gallium-nitride semiconductor technology and its practical design challenges in power electronics applications: an overview. Energies 12. https://doi.org/10.3390/en12142663
7. Thomsen A, Brooke MA (1994) Low control voltage programming of floating gate MOSFETs and applications. IEEE Trans Circ Syst I Fundamental Theory Appl 41(6)
8. Fukui H (1979) Channel current limitations in GaAs MESFETs. Solid-State Electron 22(5). https://doi.org/10.1016/0038-1101(79)90157-6
9. Singh RK (2011) Silicon on insulator technology review. Int J Eng Sci Emerg Technol 1(1). ISSN: 2231-6604
10. Tougaw PD, Lent CS (1994) Logical devices implemented using quantum cellular automata. J Appl Phys 75(3)
11. Lent CS, Tougaw PD (1997) A device architecture for computing with quantum dots. Proc IEEE 85(4)
12. Bahar AN, Waheed S, Hossain N (2015) A new approach of presenting reversible logic gate in nanoscale. SpringerPlus 4
13. Bahar AN, Waheed S, Habib MA (2015) An efficient layout design of Fredkin gate in quantum-dot cellular automata (QCA). Duzce Universite Bilimve Technol Dergisi 3
14. Islam, MS, Abdullah-Al-Shafi M, Bahar AN (2016) A new approach of presenting universal reversible gate in nanoscale. Int J Comput Appl 134
15. Shamsabadi AS, Ghahfarokhi BS, Zamanifar K, Movahedinia N, Applying inherent capabilities of quantum-dot cellular automata to design: D flip-flop case study. J Syst Archit 55
16. Sheikhfaal S, Navi K, Angizi S, Navin AH (2015) Designing high speed sequential circuits by quantum-dot cellular automata: memory cell and counter study. Quantum Matter 4
17. Yang X, Cai L, Zhao X, Zhang N (2010) Design and simulation of sequential circuits in quantum-dot cellular automata: falling edge-triggered flip-flop and counter study. Microelectron J 41
18. Xiao L, Chen X, Ying S (2012) Design of dual-edge triggered flipflops based on quantum-dot cellular automata. J Zhejiang Univ Sci C 13
19. Beigh MR, Mustafa M, Ahmad F (2013) Performance evaluation of efficient XOR structures in Quantum-dot cellular automata. Circ Syst. 04
20. Hashemi S, Farazkish R, Navi K (2013) New quantum-dot cellular automata cell arrangements. J Comut Theor Nanosci 10
21. Chabi AM, Sayedsalehi S, Angizi S, Navi K (2014) Efficient QCA exclusive-or and multiplexer circuits based on a nanoelectronic compatible designing approach. Int School Res Notices. https://doi.org/10.1155/2014/463967

22. Angizi S, Alkaldy E, Bagherzadeh N, Navi K (2014) Novel robust single layer wire crossing approach for exclusive OR sum of products logic design with quantum-dot cellular automata. J Low Power Electron 10. https://doi.org/10.1166/jolpe.2014.1320
23. Azghadi M, Kavehei O, Navi K (2007) A novel design for quantum-dot cellular automata cells and full adders. J Appl Sci 7
24. Cho H, Swartzlander Jr EE (2009) Adder and Multiplier design in quantum dot cellular automata. IEEE Trans Comput 58
25. Cho H, Swartzlander Jr. EE (2009) Adder and multiplier design in quantum-dot cellular automata. IEEE Trans Comput 58
26. Wang W, Walus K, Jullien GA (2003) Quantum-dot cellular automata adders. In: Third IEEE conference on nanotechnology, vol 2
27. Vetteth A, Walus K, Dimitrov VS, Jullien GA (2002) Quantumdot cellular automata carry-look-ahead adder and barrel shifter. In: Proceedings of the IEEE emerging telecommunications technologies conference
28. Kim K, Wu K, Karri R (2007) The robust QCA adder designs using composable QCA building blocks. IEEE Trans Comput AidedDes Integr Circ Syst 26
29. Zhang R, Walus K, Wang W, Jullien GA (2005) Performance comparison of quantum-dot cellular automata adders. In: IEEE International symposium on circuits and systems 3
30. Hanninen I, Takala J (2008) Binary adders on quantum dot cellular automata. J Sign Process Syst Sign Image Video Technol 58
31. Hänninen I, Takala J (2010) Binary adders on quantum-dot cellular automata. J Signal Process Syst 58(1)
32. Vankamamidi V, Ottavi M, Lombardi F (2008) A serial memory by quantum-dot cellula-rautomata (QCA). IEEE Trans Comput 57
33. Swathi M, Gnaneshwara Chary U (2018) Design of parity generator and parity checker using quantum dot automata. In J Pure Appl Math 118(24)
34. Cho H, Swartzlander E (2009) Adder and multiplier design in quantum-dot cellular automata. IEEE Trans Computers 58(6)
35. Automata (QCA) shift register and analysis of errors. IEEE Trans Electron Devices 50(9) (2003)
36. Sheikhfaal S, Angizi S, Sarmadi S, Hossein Moaiyeri M, Sayedsalehi S (2015) Designing efficient QCA logical circuits with power dissipation analysis. Microelectronics J 46. https://doi.org/10.1016/j.mejo.2015.03.016
37. Singh G, Sarin RK, Raj B (2016) A novel robust exclusive-OR function implementation in QCA nanotechnology with energy dissipation analysis. J Comput Electron 15. https://doi.org/10.1007/s10825-016-0804-7
38. Lent CS, Tougaw PD, Porod WD, Bernstein GH (1993) Quantum cellular automata. Nanotechnology 4(1)
39. Zhang R, Walus K, Wang W, Jullien GA (2008) A method of majority logic reduction for quantum cellular automata. IEEE Trans Nanotechnol 3(4)
40. Tougaw PD, Lent CS, Porod W (1993) Bistable saturation in coupled quantum-dot cells, J Applied Phys 74(5)
41. Cho H, Swartzlander EE (2007) Adder designs and analyses for quantum-dot cellular automata. Nanotechnology IEEE Trans 6(3)
42. Bahar AN, Waheed S (2016) Design and implementation of an efficient single layer five input majority voter gate in quantum dot cellular automata. SpringerPlus 5. https://doi.org/10.1186/s40064-016-2220-7
43. Islam S, Farzana S, Bahar SAN (2014) Area efficient layout design of multiply complements logic (MCL) gate using QCA technology. Global J Res Eng 14
44. Abdullah-Al-Shafi M, Shifatul M, Bahar AN (2015) A review on reversible logic gates and its QCA implementation. Int J Comput Appl 128. https://doi.org/10.5120/jca2015906434
45. Bahar AN, Waheed S, Habib MA (2014) A novel presentation of reversible logic gate in Quantum-dot Cellular Automata (QCA). In: International conference on electrical engineering and information communication technology (ICEEICT). https://doi.org/10.1109/ICEEICT.2014.6919121

46. Sarker A, Bahar AN, Biswas PK, Morshed M (2014) A novel presentation of Peres gate (PG) in quantum-dot cellular automata (QCA). Eur Sci J 10
47. Shifatul Islam Md, Abdullah-Al-Shafi Md, Bahar AN (2015) Implementation of binary to gray code converters in quantum dot cellular automata. J Today's Ideas – Tomorrow's Technol 3(2). https://doi.org/10.15415/jotitt.2015.32010
48. Abdullah-Al-Shafi Md, Bahar AN (2016) Novel binary to gray code converters in QCA with power dissipation analysis. Int J Multimedia Ubiquitous Eng 11(8)
49. Bhowmik D, Saha AK, Dutta P (2016) A novel design and implementation of binary to gray code converters up to 4-bit by quantum dot cellular automata. Int J Control Theory Appl 9(41)
50. Chakraborty R, Banerjee A, Mahato DK, Choudhuri S, Mandal NK (2018) Design of binary to gray code converter for error correction in communication systems using layered quantum dot cellular automata. In: 2nd International conference on electronics, materials engineering & nano-technology. IEEE
51. Guleria N (2017) Binary to gray code converter implementation using QCA. IEEE
52. Schlesinger TE (2001) Gallium arsenide. Science and Technology, ScienceDirect, Encyclopedia of Materials
53. Azghadi MR, Kavehei O, Navi K (2007) A novel design for quantum-dot cellular automata cells and full adders J Appl Sci 7(22)

Algorithmic Analysis on Spider Monkey Local Leader-based Sea Lion Optimization for Inventory Management Integrated with Block Chain in Cloud

C. Govindasamy and A. Antonidoss

Abstract In process industry, the supply chain includes more intermittent production procedures and it is considered as the significant model triggered due to more processing time. This paper plans to develop an algorithmic analysis of multi-objective inventory management in block chain technology under cloud sector. The analysis focuses on the Spider Monkey Local Leader based Sea Lion Optimization Algorithm (SMLL-SLnO). Here, SMLL-SLnO is adopted for the inventory management, which intends to minimize the multi-objective functions that concern transaction cost, inventory holding cost, transportation cost, shortage cost, and time cost. Moreover, the algorithmic analysis of the proposed SMLL-SLnO is done by varying the value of perturbation rate from 0.05 to 0.30. Hence, the analysis clearly reveals the effect of perturbation rate on SMLL-SLnO for maintaining the inventory management.

Keywords Inventory management · Supply chain network · Block chain technology · Cloud sector · Spider monkey local leader-based sea lion optimization · Perturbation rate

1 Introduction

The inventory management is necessary in multiple domains like supply chain management and production. Block chain technique represents the distributed database, which holds the continuous growing of data records list that is protected from revision and tampering. This includes the batches of blocks maintaining each transaction [1, 2]. Each block consists of time and an association related to the earlier block [3, 4]. In supply chain, the block chain is used for enabling accurate and transparent end-to-end tracking. The conventional business methods hold the complete actions history in one centralized database, which is vulnerable to cyberattack. To all the users, the databases are distributed by the block chain technology that develops

C. Govindasamy (✉) · A. Antonidoss
Department of Computer Science and Engineering, Hindustan Institute of Technology & Science, Padur, Chennai, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_5

the theory of consensus method as it is quite complex for attacking many databases concurrently [5]. Moreover, the block chain model is seemed to be transparent and secured. In block chain model, the consensus feature removes any concern that one centralized organization might control the transaction data or request more amounts for indispensable services [6, 7].

For the operations performed on International Space Station (ISS), Inventory Management System (IMS) [8] employs near-distance technologies and this subsystem is named as Barcode Inventory Tracking System (BITS). In order to enhance situational awareness and IMS autonomy, Toyoda et al. [9] introduced the employment of wireless technologies together with middleware. At the current stage, the BITS is the efficient tracking model, for all the onboard missions of ESA Columbus. Anyhow, the optimization model [10] based on simulation has some conflicts for stochastic inventory management. As many imitations are necessitated to eliminate the noise present in the obtained result, the simulation is generally computationally expensive. Moreover, the simulation offers a "what-if" response, when it has no gradient data [11]. Although there are more efficient models to clear the simulation-based issues like Simulated Annealing (SA) and Genetic Algorithm (GA), these are the meta-heuristic algorithms that won't ensure the quality of the solutions [12, 13]. In order to make use of surrogate approaches, a significant branch of simulation-based optimization resorts, in which black-box functions have been sampled and forecasted using the analytical approximations that lie in between linear regression and adaptive non-linear methods [14, 15]. However, the inventory control optimization has discrete and unfavorable features.

In inventory management, the cloud is an effective way to maintain the right stock level accurately. Also, it reduces the processing time, incorrect, and missing order information. In inventory management, to provide the security to the cloud will keep the stored data secured in the long run. Hence, in this paper, the SMLL-SLnO is proposed for multi-objective inventory management model with block chain in cloud.

The major contribution of the present paper is.

- To analyze the developed SMLL-SLnO on multi-objective inventory management model in block chain technology under the cloud sector.
- The main purpose of the developed SMLL-SLnO is to reduce the multi-objective functions, which include transportation cost, inventory holding cost, transaction cost, shortage cost, and time cost.
- Here, the analysis is done by varying the perturbation rate between 0.05 and 0.30 for validating the performance of the proposed block chain technology under cloud sector.

The remaining paper is structured in the following manner: Sect. 2 elaborates the literature review of inventory management. The architectural view of inventory management in block chain under cloud is provided in Sect. 3. Section 4 specifies the optimal inventory management using proposed SMLL-SLNO. The results and discussions of the entire paper are shown in Sect. 5. In Sect. 6, the conclusion of the entire paper is described.

2 Literature Review

In 2018, Zhao and Wang [16] have concentrated on a hybrid supply chain procedure for multi-product three-echelon inventory control method in simulation basis. A feedback control law was modeled for inventory control on the basis of the concept of control engineering. To the developed method, many mixed inventory control mechanisms were implemented. For modifying the inventory control mechanisms, the proportional plus integral control algorithm was employed. Moreover, a simulation-based optimization algorithm was implemented on the basis of the proposed simulation model for three-echelon inventory control model. The outcomes have demonstrated that the complete entropy ratio and customer satisfaction's reciprocal were optimally tuned using the mixed inventory control mechanism.

In 2017, Demey and Wolff [17] have embellished a new model-based searching technique named Semantic Inventory Management for International Space Station (SIMISS), in which the probable positions of lost items was computed on the basis of three-dimensional contextual features such as "spatial, temporal, and human." This has included machine learning algorithms, databases, ubiquitous client applications, and ontologies. The implementation and testing of SIMISS were done using the sample data taken from IMS, onboard short-term plan, and operation data files tests were performed in simulation scenario group.

In 2016, Ye and You [18] have suggested an effective method in order to decrease the whole operation cost by fulfilling the constraints for service level. By using kriging methods in a region-wise process, the performances of every inventory were evaluated, which majorly decreased the processing time while optimization and sampling. The aggregated surrogate methods were tuned using the trust-region model, in which a method recalibration procedure was employed for solution's validity assurance. To clear the problems of supply chain with uncertain demand, multi-sourcing capability, stochastic lead time, and asynchronous ordering, the suggested model was employed, which was verified using two case studies.

In 2020, Golsefidi and Jokar [19] have introduced the Mixed Integer Linear Programming model (MILP) in order to clear the issue that occurred during the production-inventory routing by consistent pickup and delivery when the products flow was considered in reverse order. The aim of the developed model was to decrease the whole system's cost concerning the quantities of the production–reproduction, supplier inventory management, production–reproduction setups, retail inventory management under the Vendor-Managed Inventory (VMI) policy. Moreover, a new MILP formulation was introduced when many uncertainty conditions consisting of the amount of each retailer's demand, reproduction cost, and the number of defective products obtained from each retailer in the certain period were considered. In addition, two developed models such as GA and SA algorithms were introduced. The analysis of the developed models was conducted and compared over numerical examples in three sizes such as large, small, and medium as well as for each size three states comprised of high inventory cost, standard, and high transportation cost.

Table 1 Review of existing inventory management optimization approaches

Author [citation]	Methodology	pros	Cons
Zhao and wang [16]	genetic algorithm	• High efficiency and feasibility	• Computationally expensive
Demey and wolfs [17]	fact-based modeling (FBM)	• To extract the data from plausible facts in the provided domain	• Poor performance
Ye and You [18]	Trust-region algorithm	• To solve the black-box optimization	• In one iteration, the quadratic trust region sub-problem needs to solve many times
Golsefidi and jokar [19]	genetic algorithm (GA)	• Ability to acquire optimal solutions • Has high performance	• If it is not used in the best way, it doesn't converge an optimal solution
Zhao et al. [20]	Fluid model analysis	• High performance	• It consumes more time for large models

In 2019, Zhao et al. [20] have considered the joint optimization of production and inventory for stochastic customer orders in order to increase the throughput. In the inventory department, the customer order's demand was dynamically arrived and each order comprised of many kinds of products with random workloads. Some specific amounts of usual raw materials were required for processing the workloads, and that must be drawn from the inventory section. In the inventory section, the order of the customer was missed when there were not sufficient raw materials. In order to improve the efficient throughput with proper management of the production and inventory sections, the suggested model was employed.

Although there are many optimization models for inventory management, still there are some issues with the existing models so that the new model needs to be introduced with effective performance integrated with block chain technology under cloud. Some of the advantages and disadvantages of existing inventory management optimization models are shown in Table 1.

3 Architectural View of Inventory Management in Block Chain Under Cloud

3.1 Proposed Model

The outcomes of the optimal division of the entire enterprise provide the accurate inventory cost for each enterprise still shuns away the complete model of supply chain. In actual fact, the supply chain method includes "suppliers, manufacturers, distributors, and transportation". This kind of individual team to maintain the organizations cannot split the entire model instead it is appropriate for exploration of two inventories of echelon supply chain. The knowledge extraction concerning the transportation over the systems and the earlier transportation costs assists for including these models. This contribution employs the block chain technique in the cloud environment for performing the inventory management effectively. In inventory management, to provide the security to the cloud will keep the stored data secured in the long run. The block chain is mainly used to improve the data security, traceability, and efficiency across a cloud computing environment. Also, it improves the processing speed of any process. Moreover, it creates a permanent transaction records. The diagrammatic representation of the developed inventory management model using block chain technology is shown in Fig. 1. Here, three-echelon supply chain inventory model is taken and performed the product transportation on the basis of manufacturers between the distributors and suppliers. The overall details related to the distributors are maintained in the block chain technique under cloud environment. Therefore, the data secured in distributor 1 is not known to the remaining distributors. Block chain is one of the famous financial techniques that completely transfigure the transactions related to business. With the help of this method, several cryptographic methods are performed, which is supported using the decentralized network. This type of reliable

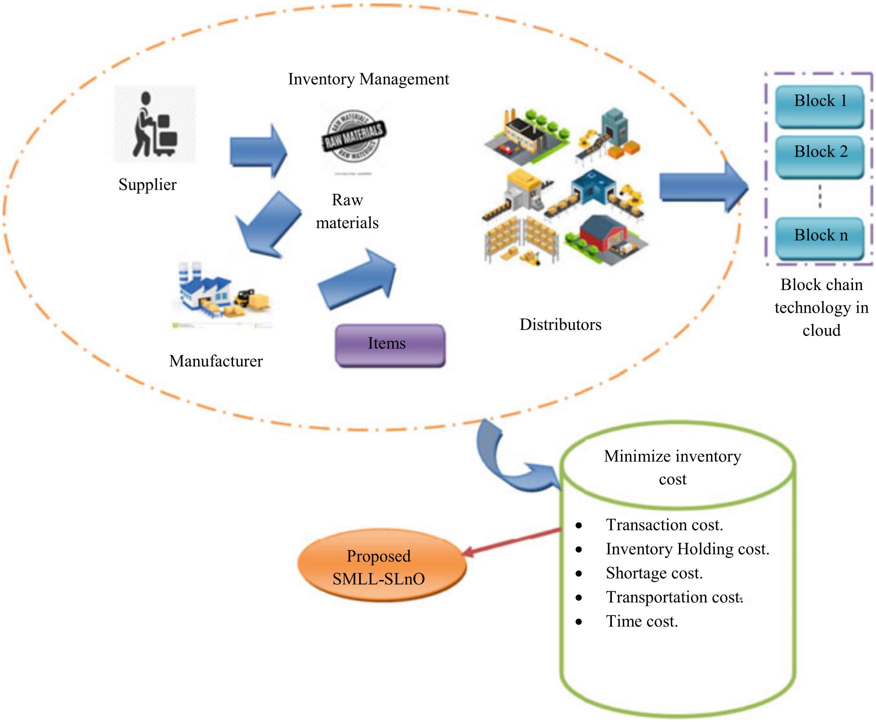


Fig. 1 Inventory management framework using block chain technology under cloud

and powerful storage of the distributor's information in block chain technique is merged with the cloud computing for better system performance.

This model is designed for tuning the five parameters in the inventory management that are combined using the block chain technique in the cloud. The five parameters such as "real time inventory of raw material at supplier, real time inventory of final product at manufacturer, real time inventory of final product at receiver, delayed transit time of raw material from 2 supplier, and delayed transit time of final product from manufacturer" that are optimally tuned using the developed SMLL-SLnO algorithm. The developed algorithm updates the solution that is suitable by considering "transaction cost, inventory cost, shortage cost, transportation cost, and time cost" for inventory management. Once all costs are reduced with respect to each distributor, the last optimal solution is maintained in the block chain under cloud environment that is secured and not known to the remaining distributors. Therefore, the developed SMLL-SLnO algorithm acquires the best solutions by minimum supply chain cost that helps for maintaining the inventory costs of the entire supply chain network.

3.2 Formulation of Problem

Consider that the term $\mathrm{hc}_{\mathrm{sb}}^{1}$ denotes raw material holding cost per unit b at supplier s, and the term $\mathrm{hc}_{\mathrm{mc}}^{2}$ represents the final product holding cost per unit c at manufacturer m. The final product c at distributor d, per unit holding cost is given by $\mathrm{hc}_{\mathrm{dc}}^{3}$. The raw material transportation cost per unit b from supplier to manufacturer is given by $\mathrm{tc}_{\mathrm{smb}}^{1}$, whereas the term $\mathrm{tc}_{\mathrm{mdc}}^{2}$ denotes the cost of transportation of the finishing product c from manufacturer to distributor. Similarly, the term $\mathrm{sh}_{\mathrm{dc}}$ denotes final product shortage cost per unit c for distributor d. The raw material fixed order cost b from manufacturer m to supplier s is denoted as $\mathrm{or}_{\mathrm{msb}}^{1}$ and the term $\mathrm{or}_{\mathrm{dmc}}^{2}$ represents the final product fixed order cost c from distributor d to manufacturer m. In addition, the term $\mathrm{dmn}_{\mathrm{dc}}(\mathrm{it})$ refers the raw materials demand from the manufacturer through time it.

Moreover, the raw material real time inventory b at supplier s during time it is denoted as $B_{\mathrm{sb}}^{1}(\mathrm{it})$, whereas the term $B_{\mathrm{mc}}^{2}(\mathrm{it})$ denotes the final product real-time inventory c at manufacturer m during time it, and the final product real time inventory c at distributor d through time it is denoted as $B_{\mathrm{dc}}^{3}(\mathrm{it})$.

Assume that the late raw material transportation cost b from supplier s to manufacturer m is expressed as $\mathrm{dtr}_{\mathrm{smb}}^{1}$, whereas the late final product transportation cost c from manufacturer m to distributor d is denoted as $\mathrm{dtr}_{\mathrm{mdc}}^{2}$. The term $\mathrm{dt}_{\mathrm{smb}}^{1}$ denotes the delayed transmit time of raw material b from supplier s to manufacturer m, whereas the term $\mathrm{dt}_{\mathrm{mdc}}^{2}$ refers the delayed transit time of the last product c from manufacturer m to distributor d. In addition, the index of the raw material is denoted as b,$b = \{1, 2, \ldots, A\}$ and the index of the final product is given by c,$c = \{1, 2, \ldots, B\}$. The index of the time period is given by it, $\mathrm{it} = \{1, 2, \ldots, \mathrm{IT}\}$. The term s, $s = \{1, 2, \ldots, S\}$ denotes the index of the supplier and the distributor inventory index count is denoted as d,$d = \{1, 2, \ldots, D\}$.

3.3 Objective Function

The main intent of the developed inventory management in the block chain under cloud sector is to reduce the multi-level inventory cost [21].

(a) Transaction cost: This contains the transaction cost with the manufacturers and suppliers, and the distributors and manufacturers. The mathematical equation is denoted in Eq. (1).

$$\mathrm{TC} = \sum_{s=1}^{S} \sum_{\mathrm{ab1}}^{A} \mathrm{TC}^{1}_{\mathrm{smb}} + \sum_{d=1}^{D} \sum_{c=1}^{B} \mathrm{TC}^{1}_{\mathrm{mdc}} \tag{1}$$

(b) Inventory holding cost: This consists of the cost linked with the inventory holding of manufacturer, suppliers, and distributors. Equation (2) refers to the mathematical equation of inventory holding cost.

$$\mathrm{IHC} = \sum_{s=1}^{S} \sum_{b=1}^{A} \mathrm{hc}^{1}_{\mathrm{sb}} \cdot B^{1}_{\mathrm{sb}}(\mathrm{it}) + \sum_{c=1}^{B} \mathrm{hc}^{2}_{\mathrm{mc}} \cdot B^{2}_{\mathrm{mc}}(\mathrm{it}) + \sum_{d=1}^{D} \sum_{c=1}^{B} \mathrm{hc}^{3}_{\mathrm{dc}} \cdot B^{3}_{\mathrm{dc}}(\mathrm{it}) \tag{2}$$

(c) Shortage cost: By using the shortage cost $\mathrm{sh}_{\mathrm{dc}}$, it is determined and the raw materials demand from the manufacturer $\mathrm{dmn}_{\mathrm{dc}}(\mathrm{it})$ and the final product real-time inventory $B^{3}_{\mathrm{dc}}(\mathrm{it})$ and the corresponding equation is given in Eq. (3).

$$\mathrm{SH} = \sum_{d=1}^{D} \sum_{c=1}^{B} \mathrm{sh}_{\mathrm{dc}} \cdot \left[\mathrm{dmn}_{\mathrm{dc}}(\mathrm{it}) - B^{3}_{\mathrm{dc}}(\mathrm{it})\right] \tag{3}$$

(d) Transportation cost: This contains the transportation cost among the manufacturers and suppliers, as well as the distributors and manufacturers. The numerical formula of transportation cost is denoted in Eq. (4).

$$\mathrm{TR} = \sum_{s=1}^{S} \sum_{b=1}^{A} \mathrm{tc}^{1}_{\mathrm{smb}} \cdot B^{1}_{\mathrm{sb}}(\mathrm{it}) + \sum_{d=1}^{D} \sum_{c=1}^{B} tc^{1}_{\mathrm{mdc}} \cdot B^{2}_{\mathrm{mc}}(\mathrm{it}) \tag{4}$$

(e) Time cost: This contains the time cost among the manufacturer and the supplier, as well as the manufacturer and distributor, which is mathematically expressed in Eq. (5).

$$\mathrm{TMc} = \sum_{s=1}^{S} \sum_{b=1}^{A} \mathrm{dt}^{1}_{\mathrm{smb}} \cdot \mathrm{it}^{1}_{\mathrm{smb}} + \sum_{d=1}^{D} \sum_{c=1}^{B} \mathrm{dt}^{2}_{\mathrm{mdb}} \cdot \mathrm{it}^{2}_{\mathrm{mdb}} \tag{5}$$

In this, the supplier transporting delay of the raw material to the manufacturer and the delay of the final product in transportation are considered. The multi-echelon supply chain inventory method final objective function is represented in Eq. (6). Here, the inventory and time cost weight coefficients are denoted as α and β, respectively. In order to solve the objective model, Eq. (7), (8) and (9) provide few constraints, which must be taken into consideration.

$$\mathrm{Min\,IC} = \alpha(\mathrm{TC} + \mathrm{IHC} + \mathrm{SH} + \mathrm{TR}) + \beta\mathrm{TMc} \tag{6}$$

$$\alpha + \beta = 1,\ \beta = \alpha - 1, \tag{7}$$

$$\mathrm{itq}^{1}_{\mathrm{smb}}(\mathrm{it}) \geq \mathrm{dt}_{\mathrm{dc}}(\mathrm{it})e_{\mathrm{cb}} \tag{8}$$

$$\frac{\mathrm{dt}_{\mathrm{dc}}(\mathrm{it}) - B^{2}_{\mathrm{mc}}(it)}{\mathrm{dt}_{\mathrm{dc}}(\mathrm{it})} < 7\% \tag{9}$$

In order to combine these two sections in the objective, Eq. (7) is used as the inventory and the time cost is having different dimensions. Equation (8) certifies that the raw material demand is not exceeding the supply, where the bill of the material is given by e_{cb} demand of the final c on the raw material b. The raw material supply b from supplier s during time it is denoted as $q^{1}_{\mathrm{smb}}(\mathrm{it})$. Similarly, the shortage cost is not exceeding 7% as in Eq. (9) [21].

4 Optimal Inventory Management Using Proposed SMLL-SLnO

4.1 Solution Encoding

Some parameters like "real time inventory of raw material at supplier $B^{1}_{\mathrm{sb}}(\mathrm{it})$, real time inventory of final product at manufacturer $B^{2}_{\mathrm{mc}}(\mathrm{it})$, real time inventory of final product at receiver $B^{3}_{\mathrm{dc}}(\mathrm{it})$, delayed transit time of raw material from supplier $\mathrm{dt}^{1}_{\mathrm{smb}}$, and delayed transit time of final product from manufacturer $\mathrm{dt}^{2}_{\mathrm{mdc}}$" are taken into account for obtaining the optimal inventory management in block chain under cloud sector. To reduce the cost of multi-echelon supply chain inventory model, the accurate optimization of these parameters needs to be performed. The diagrammatic representation of the solution encoding is shown in Fig. 2.

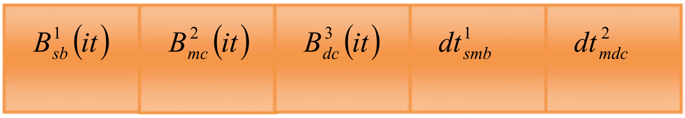


Fig. 2 Solution encoding

4.2 Proposed SMLL-SLnO

The optimization methods have performed discrete changes and improvements in order to solve complex issues. Meta-heuristic search methods in several implementations are seemed to be appropriate and attained best results [22, 23]. The inspiration of conventional SLnO [24] is the hunting behavior of sea lions and its whiskers, which are employed for prey recognition. The conventional algorithm represents for finding the positive regions of design space, and employs the best one. This has the ability to generate competitive results and it has the best exploration ability. However, there are specific disadvantages like it is complex for acquiring the global optimal solution as of randomness. A meta-heuristic model called SMO [25] is integrated with the conventional SLnO for enhancing the performance of traditional SLnO algorithm. It is able to deal with the complicate real-world optimization issues because of high efficiency generated by SMO. The traditional SMO is combined with the SLnO for producing a novel algorithm called SMLL-SLnO algorithm.

Sea lions are seen in thousands of members, which live in vast colonies [24]. The sea lions are classified by their sex, age, and function. The major features of sea lions are determined by the speedy behaviour of the movements of the fish. The prey can be found easily to their eyes. The pupils are majorly opened for more amount of light for passing by in order has a clear underwater vision. The most significant feature of sea lions is whiskers that assist for locating the accurate prey's location. The other feature is their ability for moving speed and accurate over water. The significant stages of sea lions hunting behavior is chasing and tracking the prey using their whiskers, encircling and attacking the prey.

Detecting and tracking phase: In order to hunt and chase the prey, the sea lion that identifies the prey's location and invites the subgroups. The numerical formula of the detection and tracking phase is expressed in Eq. (10), in which the distance between the target prey and the sea lions is given by Dst. The position vector of sea lions and the target prey is denoted as pv(tm), the current iteration is represented as tm, and the random vector is denoted as E, which lies in between 0 and 1 and it is multiplied by 2 for improving the search space that is helpful for the search agents in order to verify the optimal solution. The position of the target prey is modified in the successive iteration and the behaviour is numerically modeled as per Eq. (11).

$$\mathrm{Dst} = |2E.\mathrm{pv(tm)}\text{ - }\mathrm{SeLi(tm)}| \tag{10}$$

$$\mathrm{SeLi(tm+1)} = \mathrm{pv(tm)} - \mathrm{Dst}.E \tag{11}$$

In the above equation, the term E is reduced from 2 to 0 until the final iteration, and the next iteration is given by tm + 1.

Vocalization Phase: The sea lion's sound is traveled four times faster in water over air. Many ways of vocalization helps the sea lion for promoting the communication over each other in hunting and chasing. This lions have SpMoall ears, which has the ability to identify the sounds both above and under water. Thus, if the prey is found by the lion, it passes the information to others for attacking. This type of feature is numerically expressed in Eq. (12), (13) and (14).

$$\mathrm{SSL}_{\mathrm{ldr}} = \left| \frac{(G_1(1 + G_2))}{G_2} \right| \tag{12}$$

$$G_1 = \sin\theta \tag{13}$$

$$G_2 = \sin\varphi \tag{14}$$

In the above equations, the speed of the sound of the sea lion is indicated by $\mathrm{SSL}_{\mathrm{ldr}}$, and the speed of sounds in air and water is given by G_2 and G_1, respectively. The sound of the sea lion is denoted as $\sin\varphi$ that is reflected to the other medium, which is the air for communicating with the remaining lions that are present in the shore. The sea lion that makes the sound that is refracted at the similar medium in order to communicate with other members under water is represented as $\sin\theta$.

Attacking Phase: The new search agent is capable of encircling, detecting the best prey. The hunting behavior of sea lions is mathematically represented in two phases, which are mentioned below.

Dwindling encircling method: It is based on the value of E and it is given in Eq. (11), which is reduced linearly from 2 to 0 in a period of iterations.

Circle updating position: The sea lions track the bait ball of fishes initially from the edges and chase them, which is denoted in Eq. (15).

$$\mathrm{SeLi}(\mathrm{tm} + 1) = |\mathrm{pv}(\mathrm{tm}) - \mathrm{SeLi}(\mathrm{tm})|.\cos(2\pi g) + \mathrm{pv}(\mathrm{tm}) \tag{15}$$

In Eq. (15), the distance between the search agent and the finest optimal solution is given by $|\mathrm{pv}(\mathrm{tm}) - \mathrm{SeLi}(\mathrm{tm})|$, the absolute value is given by | |, the random number is given by g, which lies in between -1 and 1, and the sea lion that swims in the prey region in circular path is given by $\cos(2\pi g)$.

Searching for prey: The search agents locations are updated on the basis of the sea lion, which is chosen at random. SLnO algorithm evaluates the global search agent for finding the global optimal solution if E is exceeding 1. The mathematical representation of searching for the prey is expressed in Eq. (16) and (17).

$$\mathrm{Dst} = |2E.\mathrm{SeLi}_{\mathrm{rnd}}(\mathrm{tm}) - \mathrm{SeLi}(\mathrm{tm})| \tag{16}$$

$$\mathrm{SeLi}(\mathrm{tm}+1)=\mathrm{SeLi}_{\mathrm{rnd}}(\mathrm{tm})-\mathrm{Dst}.E \tag{17}$$

In the above equation, the randomly selected sea lion, which is selected from the current iteration, is given by $\mathrm{SeLi}_{\mathrm{rnd}}$.

LLD of SMO: The position of the small group members is updated by the information accumulated from local and global leaders or by initializing them at random when the position of the local is not updated to the constant threshold called Local Leader Limit (LLLimit), which is shown in Eq. (18).

$$\begin{aligned}\mathrm{SeLi}_{\mathrm{newij}}=\mathrm{SeLi}_{\mathrm{ij}}+\mathrm{rnd}[0,1](\mathrm{GlLe}_{j}-\mathrm{SeLi}_{\mathrm{ij}})\\+\mathrm{rnd}[0,1](\mathrm{SeLi}_{\mathrm{ij}}-\mathrm{LoLe}_{\mathrm{pj}})\end{aligned} \tag{18}$$

In Eq. (18), the term $\mathrm{SeLi}_{\mathrm{ij}}$ indicates the jth dimension of ith SMO [21], and the random number is denoted as rnd, which lies in between 0 and 1. The jth dimension of the position of the global leader is given by GlLe_{j} and the jth dimension of the pth local group leader position is represented as $\mathrm{LoLe}_{\mathrm{pj}}$.

In general, the position of the current search agent is updated by exploration phase in traditional SLnO algorithm as per Eq. (17). Here, in the developed SMLL-SLnO algorithm, the position of the current search agent is updated by LLD phase of SMO as shown in Eq. (18). By integrating different optimization rules, the hybrid optimization algorithm is introduced. It is reported as the promising one for specific search issues. It acquires the advantages of various optimization methods for fast convergence. The convergence behavior of the developed SMLL-SLnO algorithm is seemed to be best over other algorithms [26]. The pseudo-code representation of developed SMLL-SLnO algorithm is depicted in Algorithm 1. In this algorithm, the perturbation rate PR, limit of LLL LLLimit, limit of GLLL GLLimit are initialized. The term PR is the one of the main control parameters of SMO. In standard SMO, the perturbation rate is varied in between 0.1 and 0.9. Here, the perturbation rate PRis varied from 0.05 to 0.30 for analysis.

Algorithm 1: Proposed SMLL-SLnO Algorithm

```
Start
Population initialization
Initialize $PR$ , $LLLimit$ , $GLLimit$
Choose $SeLi_{rnd}$
Compute fitness function for each search agent)
The best candidate search agent who have best fitness is the $SeLi$
while t<maximum number of iterations
    Compute $SSL_{ldr}$ using Eq. (12)
    if( $SSL_{ldr} < 0.25$)
        if($|E| < 1$)
            Update the position of the current search agent by Eq. (10)
        Else
            Update the position of the current search agent using the LLD
            phase of SMO by Eq. (18)
    Else
        Update the position of the current search agent by Eq. (15)
    Compute the fitness function for each search agent
    Update $SeLi$ if there exists any best solution
```

5 Results and Discussions

5.1 *Experimental Setup*

The implemented inventory management in block chain technique under the cloud sector was developed using MATLAB 2018a, and the evaluations were done. The simulation setup of the developed method is given in Table 2. In this, the performance of the developed method was evaluated using three test cases, which are shown in Table 3. The population size considered for analysis is 10 and the total count of iterations is considered as 1000. The performance of the developed model was compared with by varying the value of perturbation rate from 0.05 to 0.30.

Table 2 Simulation setup

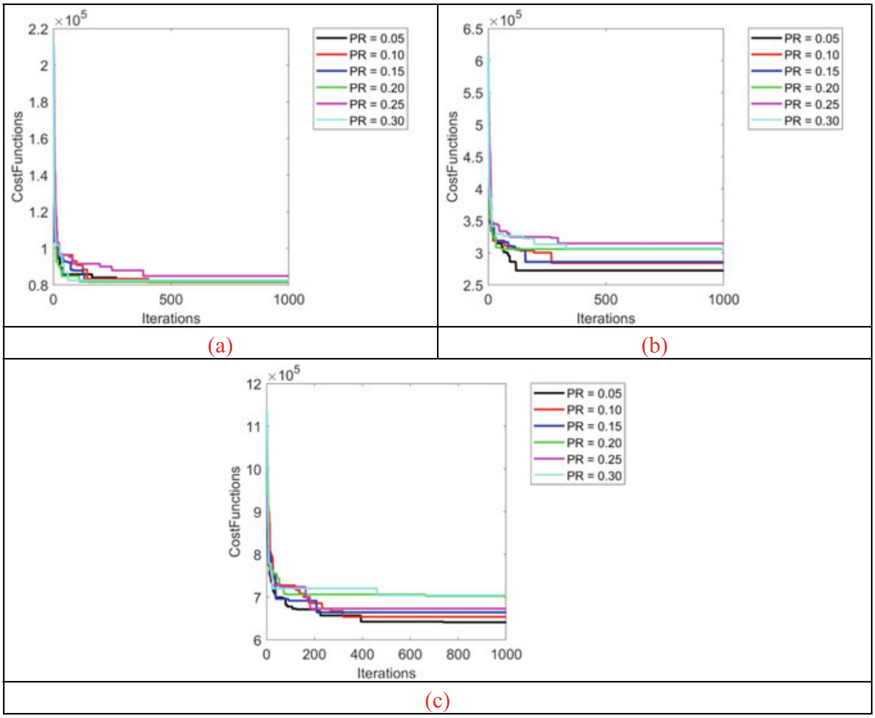
Parameter	Value
Population size	10
Total iterations	1000
Perturbation rate PR	0.05–0.30

Table 3 Details of test cases

Test cases	Suppliers	Distributors	Finished work	Raw material
Test case 1	5	4	1	4
Test case 2	8	6	2	4
Test case 3	10	8	3	4

5.2 Analysis of Proposed SMLL-SLnO Algorithm by Varying Perturbation Rate

The analysis of the proposed SMLL-SLnO algorithm concerning iterations by varying perturbation rate for all the test cases is shown in Fig. 3. From Fig. 3(a), the cost function of the developed SMLL-SLnO algorithm is acquiring the minimum cost when PR = 0.30 and when PR = 0.25 is having the maximum cost at 1000th iteration. When considering the 50th iteration, the cost function of the presented SMLL-SLnO is minimum when PR = 0.30 followed by PR = 0.20. Later, PR =



(a)

(b)

(c)

Fig. 3 Algorithmic analysis of the proposed SMLL-SLnO for inventory management in block chain under cloud sector by varying perturbation rate for **a** Test case 1, **b** Test case 2, and **c** Test case 3

0.05, PR = 0.15, PR = 0.25 is having the minimum cost function, respectively. The cost function of the developed SMLL-SLnO at 50th iteration when PR = 0.30 is 3.5% advanced than PR = 0.20, 4.6% advanced than PR = 0.05, 8.8% advanced than PR = 0.15, and 16.3% advanced than PR = 0.25. Based on the degree of improvement, the cost function of the developed model at all PR's is reduced until 1000th iteration. Figure 3(b) shows the cost function analysis of the implemented SMLL-SLnO algorithm with respect to iterations for test case 2. In Fig. 3(b), the cost function of the introduced SMLL-SLnO at 1000th iteration is having minimum value when PR = 0.05. It is 3.5% upgraded than PR = 0.10, 5.2% upgraded than PR = 0.15, 11.4% upgraded than PR = 0.20, 12.9% upgraded than PR = 0.30, and 14.2% upgraded than PR = 0.25. For test case 3 in Fig. 3(c), the cost function of the developed SMLL-SLnO algorithm at 800th iteration is minimum when PR = 0.05. At 800th iteration, the cost function of the SMLL-SLnO algorithm when PR = 0.05 is 1.5% better than PR = 0.10, 3.0% better than PR = 1.5, 4.4% better than PR = 0.25, 8.5% better than PR = 0.20, and 9.2% better than PR = 0.30. Thus, it is concluded that the proposed SMLL-SLnO algorithm is acquiring the best performance for inventory management in block chain technology.

5.3 Analysis on Final Cost

The statistical analysis on final cost of the developed SMLL-SLnO algorithm for all the three test cases by varying PR is shown in Table 4. For Test case 1, the standard deviation of the developed SMLL-SLnO is having the best performance when PR = 0.05. It is 82.5% superior to PR = 0.10, 78% superior to PR = 0.15, 82.9% superior to PR = 0.20, 74.2% superior to PR = 0.25, 90.4% superior to PR = 0.30. The mean of the introduced SMLL-SLnO for test case 2 is acquiring the best performance when PR = 0.05. Here, the standard deviation of the implemented SMLL-SLnO when PR = 0.20 is 5.5% improved than PR = 0.05, 31.5% improved than PR = 0.10, 33.7% improved than PR = 0.15, 56.1% improved than PR = 0.25, and 73.7% improved than PR = 0.30. Moreover, the standard deviation of the developed SMLL-SLnO is performing well when PR = 0.10. It is 26.8% enhanced than PR = 0.05, 53% enhanced than PR = 0.15 45.4% enhanced than PR = 0.20, 29.5% enhanced than PR = 0.25, and 47% enhanced than PR = 0.30. Therefore, it is confirmed that the developed SMLL-SLnO is performing well in inventory management model in block chain under cloud sector.

5.4 Analysis on Time Complexity

Table 5 shows the computational time of the developed method for various test cases, such as Tast case 1, Test case 2, and Test case 3, in which the Test case 1 has a minimum computational time of 1.0289 s.

Table 4 Statistical analysis of the proposed inventory management models in block chain under cloud sector for all test cases

Test case 1						
Measures	PR = 0.05	PR = 0.10	PR = 0.15	PR = 0.20	PR = 0.25	PR = 0.30
Best	81,161	81,453	81,613	79,609	82,224	78,252
Worst	82,220	86,777	86,927	86,434	86,620	90,751
Mean	81,503	84,055	84,003	82,538	84,830	84,187
Median	81,390	84,478	84,054	81,669	84,907	83,879
Standard deviation	435.43	2500.9	1985	2558	1694.1	4550.8
Test case 2						
Measures	PR = 0.05	PR = 0.10	PR = 0.15	PR = 0.20	PR = 0.25	PR = 0.30
Best	2.64×10^5	2.76×10^5	2.86×10^5	2.94×10^5	2.91×10^5	2.71×10^5
Worst	2.76×10^5	2.95×10^5	3.04×10^5	3.06×10^5	3.15×10^5	3.14×10^5
Mean	2.70×10^5	2.85×10^5	2.94×10^5	2.98×10^5	3.02×10^5	3.03×10^5
Median	2.73×10^5	2.85×10^5	2.92×10^5	2.97×10^5	2.99×10^5	3.10×10^5
Standard deviation	4937.9	6811.7	7040.9	4664.1	10,631	17,790
Test case 3						
Measures	PR = 0.05	PR = 0.10	PR = 0.15	PR = 0.20	PR = 0.25	PR = 0.30
Best	6.15×10^5	6.54×10^5	6.64×10^5	6.64×10^5	6.73×10^5	6.69×10^5
Worst	6.41×10^5	6.72×10^5	7.08×10^5	7.03×10^5	7.00×10^5	7.04×10^5
Mean	6.33×10^5	6.61×10^5	6.88×10^5	6.82×10^5	6.88×10^5	6.92×10^5
Median	6.36×10^5	6.59×10^5	6.88×10^5	6.82×10^5	6.87×10^5	6.92×10^5
Standard deviation	10,327	7558.1	16,102	13,861	10,734	14,265

Table 5 Computational time

Test case	Computational time (S)
Test case 1	1.0289
Test case 2	1.3577
Test case 3	1.5552

6 Conclusion and Future work

This paper has developed an algorithmic evaluation of multi-objective inventory management in block chain technique under cloud sector. The assessment was done on the proposed SMLL-SLnO algorithm. In this paper, SMLL-SLnO algorithm was used for inventory management that aimed for minimizing the multi-objective functions namely “transaction cost, inventory holding cost, shortage cost, transportation cost, and time cost." By varying the perturbation rate from 0.05 to 0.30, the assessment of the proposed SMLL-SLnO was performed. For Test case 1, the standard deviation of the developed SMLL-SLnO was having the best performance when PR = 0.05. It was 82.5% superior to PR = 0.10, 78% superior to PR = 0.15, 82.9% superior to PR = 0.20, 74.2% superior to PR = 0.25, 90.4% superior to PR = 0.30. Thus, the effect of perturbation rate on SMLL-SLnO was effectively proved from optimal inventory management. In the future, the performance of the proposed method will be further analyzed by using large number of test cases. Also, more comparison techniques will be used to compare the performance of the developed method.

References

1. Yu R et al (2017) Authentication with block-chain algorithm and text encryption protocol in calculation of social network. IEEE Access. 5:24944–24951
2. Ewen H, Mönch L, Ehm H, Ponsignon T, Fowler JW, Forstner L (2017) A testbed for simulating semiconductor supply chains. IEEE Trans Semicond Manuf 30(3):293–305
3. Nakamoto S (2009) Bitcoin: a peer-to-peer electronic cash system, cryptography mailing list
4. Kim HM, Laskowski M (2018) Toward an ontology-driven blockchain design for supply-chain provenance. Intellig Syst Account Fin Manage 25(1):18–27
5. Xiao L, Xiao Y, Yang P, Liu J, Li S, Xiang W (2017) Space-time block coded differential spatial modulation. IEEE Trans Veh Technol 66(10):8821–8834
6. Chang J, Katehakis M, Melamed B, Shi J (2018) Blockchain design for supply chain management. SSRN Electron J
7. Chen Z, Chen S, Xu H, Hu B (2018) A security authentication scheme of 5G ultra-dense network based on block chain. IEEE Access. 6:55372–55379
8. Lee S, Kim YJ, Cheong T, Yoo SH (2019) Effects of yield and lead-time uncertainty on retailer-managed and vendor-managed inventory management. IEEE Access. 7:176051–176064
9. Toyoda K, Mathiopoulos PT, Sasase I, Ohtsuki T (2017) A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain. IEEE Access 5:17465–17477
10. Liu R, Xie X, Yu K, Hu Q (2018) A survey on simulation optimization for the manufacturing system operation. Int. J. Modell. Simul. 38(2):116–127
11. Wickert J et al (2016) GEROS-ISS: GNSS REflectometry radio occultation, and scatterometry onboard the international space station. IEEE J Select Topics Appl Earth Observ Remote Sens. 9(10):4552–4581
12. AfshinMansouri S (2006) A simulated annealing approach to a bi-criteria sequencing problem in a two-stage supply chain. Comput Ind Eng 50(1–2):105–119
13. Mele FD, Guillén G, Espuña A, Puigjaner L (2006) A simulation-based optimization framework for parameter optimization of supply-chain networks. Ind Eng Chem Res 45:3133–3148
14. Cozad A, Sahinidis NV, Miller DC (2014) Learning surrogate models for simulation-based optimization. AIChE J
15. Wang G, Shan S (2011) Review of metamodeling techniques for product design with computation- intensive processes. J Mech Design
16. Zhao W, Wang D (2018) Simulation-based optimization on control strategies of three-echelon inventory in hybrid supply chain with order uncertainty. IEEE Access 6:54215–54223
17. Demey YT, Wolff M (2017) SIMISS: a model-based searching strategy for inventory management systems. IEEE Internet Things J 4(1), 172–182(2017).
18. Ye W, You F (2016) A computationally efficient simulation-based optimization method with region-wise surrogate modeling for stochastic inventory management of supply chains with general network structures. Comput Chem Eng 87:164–179

19. Golsefidi AH, Jokar MRA (2020) A robust optimization approach for the production-inventory-routing problem with simultaneous pickup and delivery. Comput Indus Eng 143
20. Zhao Y, Xu X, Li H (2019) Inventory-constrained throughput optimization for stochastic customer orders. IEEE Trans Autom Sci Eng 16(3):1218–1231
21. Wang Y, Geng X, Zhang F, Ruan J (2018) An immune genetic algorithm for multi-echelon inventory cost control of IOT based supply chains. IEEE Access 6:8547–8555
22. Presbitero A, Krzhizhanovskaya V, Mancini E, Brands R, Sloot P (2016) Immmune system model calibration by GA". Procedia Comput Sci 101:161–171
23. Wang L-L, Zhang T, Guo Y (2014) Under the environment of agile supply chain inventory control research. Adv Mater Res 10(4028):3179–3182
24. Masadeh RMT, Mahafzah BA, Sharieh AA-A (2019) Sea lion optimization algorithm. Int J Adv Comput Sci Appl 10(5):388–395
25. Agrawal V, Rastogi R, Tiwari DC (2018) Spider monkey optimization: a survey. Int J Syst Assur Eng Manage 9:929–941
26. Park Y-B (2001) A hybrid GA for the scheduling problem with due times and time deadlines". Int J Prod Econ 73(2):175–188

Energy-Efficient Wireless Communications Using EEA and EEAS with Energy Harvesting Schemes

Anupam Das, Mohammad Ali Akour, Abdullah Bahatab, and Qin Zin

Abstract Goal: In this work, we are trying to introduce two algorithms and implemented the same for achieving the energy efficiency in wireless communications using energy harvesting technique and efficient clustering. During the study, we proposed two algorithms: energy-efficient algorithm (EEA) and energy-efficient algorithm with security (EEAS). The energy node structure with its energy harvest components is also crafted in this work. The work is implemented by using NS2, and enough testing is done for checking its results. The results are also tested with different network size and with heterogeneous networks along with the IoT. Finally, the performance is tested for the network considering the QoS parameters like throughput, packet-delivery ratio (PDR), end-to-end delay (E2E) and energy consumption.

Keywords Energy efficiency algorithm (EEA) · Energy efficiency algorithm with security (EEAS) · Energy harvest · Soft clustering throughput · Packet–delivery ratio (PDR) · End-to-end delay (E2E) · Energy consumption · Phonetic distance

1 Introduction

In wireless communications, the consumption of power is always being an important issue. With IoT environment, the wireless communications become more and more popular, and hence, the problem of making power consumption at the efficient way becomes a burning research issue. In this work, we try to focus on the various issues of power consumption and suggest some new approaches which will make the

A. Das (✉)
Associate professor, Royal Global University, Guwahati, India

M. Ali Akour
Al Sharqyat University, Ibra, Oman

A. Bahatab
VTC, Riyadh, Saudi Arabia

Q. Zin
University of the Faroe Islands, Torshavn, Faroe Islands

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_6

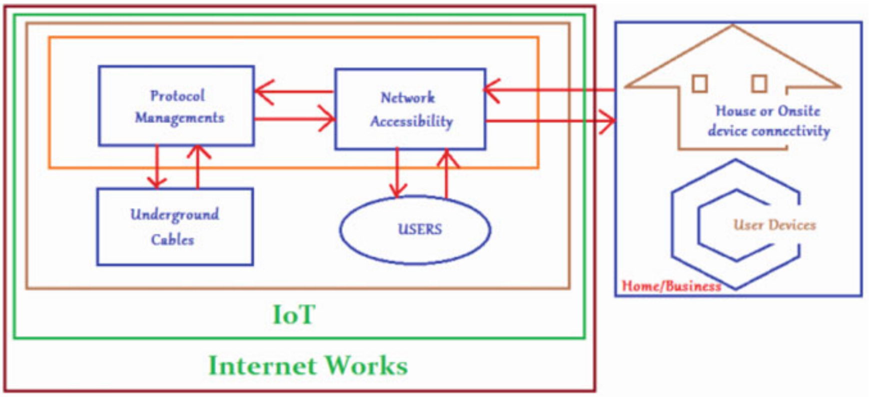


Fig. 1 Block diagram of communications architecture

consumption level of power lower than the existing way of use. Let us first discuss the issues by which we can deal with our consumption level of wireless communications in IoT. Figure 1 shows the basic building block of communications.

The structure of any network consists of some devices, and the devices are wired as well as wireless. All the activities of the devices consume certain level of power in their activities, and it is inevitable. Now in case of wireless communications, the following issues are major in case of any architecture with or without IoT.

a. Antenna
b. Transmitting/receiving signals
c. Clustering of notes

Let us discuss the above-listed issues one by one.

a. Antenna: antenna plays an important role for signal transmitting and receiving. The transmission distance can be improved by improving the sensitivity of receiver and the power of the transmitter. This issue can be dealt with the manufacturing level, so it is an issue mainly opened for antenna manufacturers. During manufacturing of the modern antenna, certain characteristics can be considered for optimizing the quality of the antenna:

i. Frequency level
ii. Impedance
iii. Antenna gain
iv. Antenna polarization
iv. Antenna radiation pattern
v. Short-distance communication (indoor and outdoor both)
vi. Long-distance communication (indoor and outdoor both)
vii. Accommodating in single antenna coverage
ix. Special long distance antenna (Fig. 2).

b. Transmitting/Receiving Signals: signal transmitting is directly related to the output power, and the increase in the output power results in the increase in the transmission range keeping all other parameters constant. The generic structure for wireless energy transmission is given in Fig. 3.

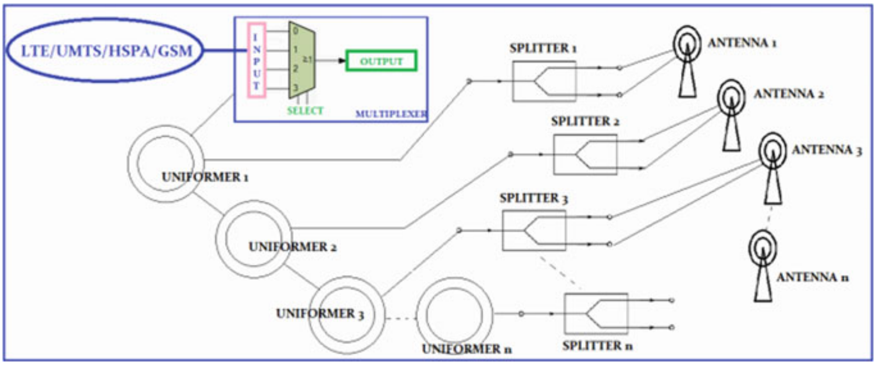


Fig. 2 Antenna transmit/receive signal

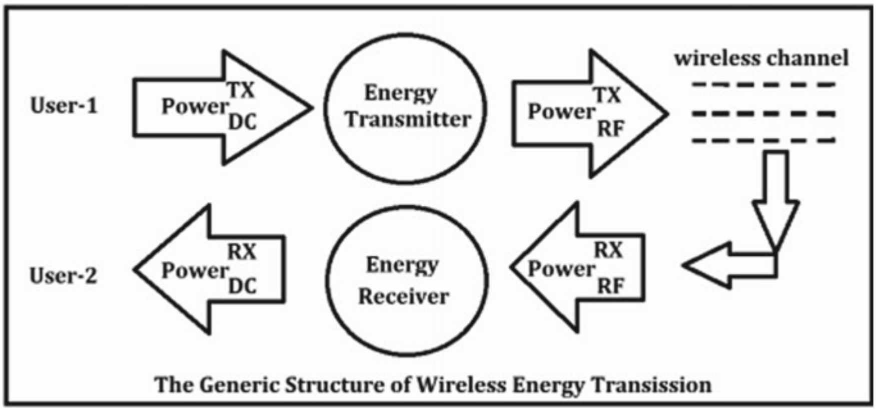


Fig. 3 Generic structure of wireless energy communications

The efficiency of the end-to-end (E2E) power transfer can be calculated as follows (Table 1):

$$E = \frac{Power^{RX}_{DC}}{Power^{TX}_{DC}} = \frac{Power^{TX}_{RF}}{Power^{TX}_{DC}} = \frac{Power^{RX}_{RF}}{Power^{TX}_{DC}} = \frac{Power^{RX}_{DC}}{Power^{RX}_{RF}} \tag{1}$$

Table 1 E2E power transfer

Power_DC_TX	Power_DC_RX	Power_RF_TX	Power_RF_RX	Energy_Transfer
12	12.05467	12.0567	12.05782	1.004555833
11	11.05467	11.0567	11.05782	1.00497
13	13.05467	13.0567	13.05782	1.004205385
14	14.05467	14.0567	14.05782	1.003905
12.5	12.55467	12.5567	12.55782	1.0043736
11.8	11.85467	11.8567	11.85782	1.004633051
12.83333333	12.88800333	12.89003333	12.89115333	1.00426
12.96190476	13.01657476	13.01860476	13.01972476	1.004217744
13.09047619	13.14514619	13.14717619	13.14829619	1.004176319
13.21904762	13.27371762	13.27574762	13.27686762	1.004135699
13.34761905	13.40228905	13.40431905	13.40543905	1.004095862
13.47619048	13.53086048	13.53289048	13.53401048	1.004056784
13.6047619	13.6594319	13.6614619	13.6625819	1.004018446
13.73333333	13.78800333	13.79003333	13.79115333	1.003980825
13.86190476	13.91657476	13.91860476	13.91972476	1.003943902
13.99047619	14.04514619	14.04717619	14.04829619	1.003907658

The Gaussian noise can be calculated as follows (Table 2):

$$G(n) = \frac{1}{n_0} + \exp\left(\frac{n}{n_0}\right) \tag{2}$$

where n is greater than equal to 0 and n_0 is the average SNR depending on communication distance.

Let us consider 'm' number of nodes of uniform field on a specific region with density 'd' on sufficiently large circular region with radius 'R', and the distribution function of the distance between some nodes obeys the Poisson law, i.e.

$$F_m = \frac{a^m}{m!} \exp(-a) \tag{3}$$

where 'a' is the mathematical expectation of number of nodes. The nearest region of each node can be calculated by

Table 2 Gaussian noise in existing method

Existing model				
Signal power (S)	Noise power (N) in dB	SNR	n	GN
52	12	6.368220976	10	0.032659568
67	11	7.846821175	20	0.009962658
55	10	7.403626895	30	0.00234836
47	9	7.178553485	40	0.000529673
77	8	9.834007382	50	0.000629715
45	11	6.118198286	60	9.00189E−06
57	3	12.78753601	70	0.000327982
55	5	10.41392685	80	4.42717E−05
56	7	9.03089987	90	5.20216E−06
61	9	8.310873256	100	7.15707E−07
64.1	11	7.654653444	110	7.5009E−08
67.2	11.5	7.666714327	120	2.07864E−08
70.3	13	7.330119727	130	2.70805E−09
73.4	13.83333333	7.247692179	140	5.63318E−10
76.5	14.83333333	7.124226789	150	1.00746E−10
79.6	15.83333333	7.013407128	160	1.76326E−11
82.7	16.83333333	6.913353862	170	3.02671E−12
85.8	17.83333333	6.822547605	180	5.10524E−13
88.9	18.83333333	6.739745679	190	8.47563E−14
92	19.83333333	6.663921163	200	1.38697E−14

$$F_1(R) = 1 - e^{-\pi R^2 d} \tag{4}$$

c. Clustering of Nodes: there are many clustering techniques that can be implemented for efficient cluster formations. In modern research, energy aware fuzzy clustering can be one of the good choices. Energy aware fuzzy clustering is based on cognitive technique. The following assumptions can be made:

1. Random selection of nodes
2. Sensor nodes must be static
3. Sensor deployment can be fairly random
4. Sensor should be self-automotive
5. The distance between any two sensor nodes is calculated based on signal strength.

Clustering Formation: in the cluster formation, any two nodes are selected at random, and the distance between them based on the signal strength is calculated. The process continues till a group of cluster heads are elected from the given set of nodes. A threshold value is sent to every node, and every nodes also generated a number for comparing their threshold with given threshold. If the number generated by the node is greater than the threshold, then the node declares itself a cluster head. All cluster heads are selected based on

i. Residual energy
ii. Two-hop node degree
iii. Cluster head (CH)—centre based on low-energy consumption during data aggregation and flooding.

2 Literature Review

In the previous works based on the energy in wireless communications, there are lot of schemes implemented, and various energy efficiency techniques are applied. Muthukumaran et al. [14] tried to attempt bettering the energy efficiency at node level and also increase the lifetime of the network by introducing a hierarchical routing scheme based on energy-efficient cluster (ENEFC). In the article, Bozorgi et al. [2] proposed a method that uses a unique technique in clustering that makes node clusters in a network not using the data from neighbours. The hybrid clusters are introduced to transmit signals in the nodes with the this clustering [2]. The energy harvesting is one of them. The harvesting energy is the need of the hour. The energy harvesting is a novel scheme by which we can harvest energy in various ways. There are many ways to convert energy into harvest energy like mechanical energy to harvest energy [12, 23], piezoelectric energy to harvest energy [23], electrostatic energy to harvest energy [13, 17], electromagnetic energy to harvest energy [13, 23, 24], photovoltaic energy to harvest energy [3, 15], thermal energy to harvest energy [6], RF energy to harvest energy [1, 8], resonant energy to harvest energy [5, 9, 16], wind energy to harvest energy [4, 10, 12, 18, 21, 22], biochemical energy to harvest energy [20, 26], acoustic energy to harvest energy [7, 19], etc.

3 Proposed Method for Achieving Clustering

The objective of the work: in this work, we will try to optimize the clustering using soft clustering adding the security aspect to the wireless communications and energy efficiency with energy harvesting (Fig. 4).

In the existing method, the lifetime of wireless sensor node increased by applying proficient energy-based cluster head selection method. In this work, it is considered the following:

1. Cluster head selection is based on modern soft clustering with energy aware method.
2. Energy aware method increases the lifetime of a node.

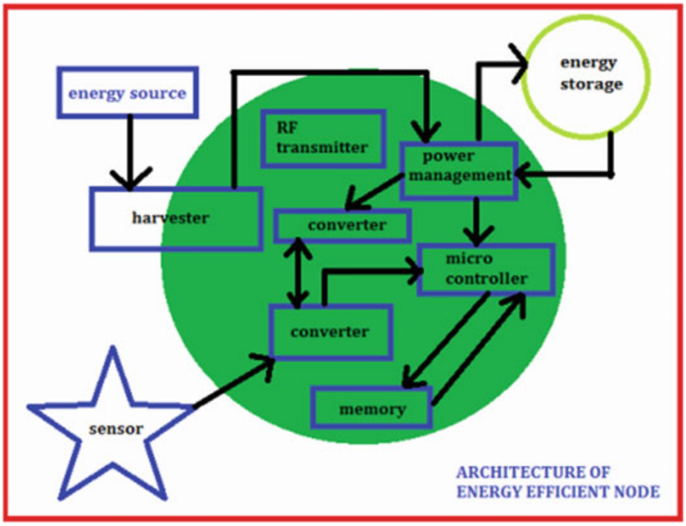


Fig. 4 Energy harvest-efficient node structure

3. Congestion aware routing is applied for efficient communication and minimizing E2E delay.
4. Use the energy harvesting algorithm for optimizing cluster behaviours in energy consumption.
5. Applying energy budget to the network.

The above schemes are applied to the proposed algorithms:

1. Algorithm-1: Energy-Efficient Algorithm (EEA)
2. Algorithm-2: Energy-Efficient Algorithm with Security (EEAS).

3.1 Algorithm-1 Energy-Efficient Algorithm (EEA)

In the algorithm-1, the network is initiated, then calculates the value of k, and the results are compared with the taken threshold. Then signal strength is calculated, thus the cluster head is selected, and the entire network is scanned till the all clusters are created. The detail algorithm is given below:

i. Initiate the network structure with a set of nodes ‘S’
ii. Check for conduciveness of all the nodes and compute the constant ‘k’ and compare with threshold ‘T’
iii. Calculate the signal strength with beacon
iv. Compute for all nodes remaining that is $(m - 1)$ in ‘S’
v. If a node wins as CH, then mark it
vi. Repeat step ii to step v until all the nodes are checked.

3.2 *Algorithm-2 Energy-Efficient Algorithm with Security (EEAS)*

Here the clusters communicate with their nodes and also to communicate with inter-cluster nodes as well as intra-cluster nodes depending on the communication requests generated. The algorithm is given below:

i. All the selected cluster heads communicate to every other node with a security check for authenticating the member for ‘S’
ii. If CH successes its security for data communication, it forwards data to the concerned node, else warning every node for security check including all other CHs
iii. Repeat step i to step ii, whenever a request for data communication crops in the network (Fig. 5).

In a network, the nodes are divided into three categories: the full active nodes, semi-active nodes and the sleeping nodes. The transitions among these nodes are shown in Fig. 6.

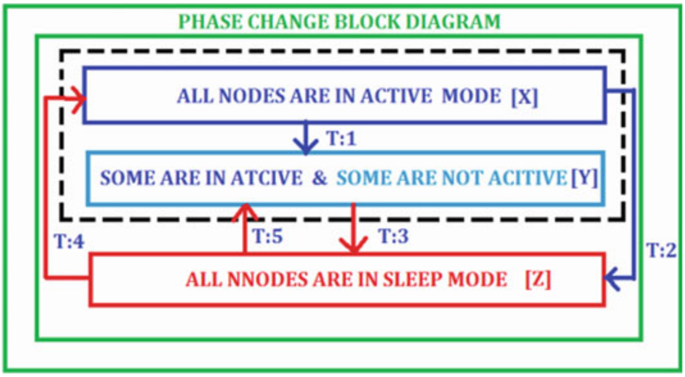

PHASE CHANGE BLOCK DIAGRAM
ALL NODES ARE IN ACTIVE MODE [X]
T:1
SOME ARE IN ATCIVE & SOME ARE NOT ACITIVE [Y]
T:4
T:5
T:3
T:2
ALL NNODES ARE IN SLEEP MODE [Z]

Fig. 5 Phase transition in the network

The set X is denoted for full active nodes, Y is denoted for semi-active nodes, and Z is sleep nodes. Now the phase transitions among these nodes are shown in Fig. 7.

The model is designed with some parametric assumptions as given below:

1. Average duration in full active mode is reciprocal of k_1
2. Average duration in sleep mode is reciprocal of $_2$
3. Average duration in semi-active mode is reciprocal of k_3
4. Average time in packet formation = k_4
5. Average time in packet managing = k_5
6. Average time in packet transmission = k_6
7. Energy consumption in $X = E_P^X$
8. Energy consumption in $Y = E_P^Y$
9. Energy consumption in managing pkt. in $X = E_{P'}^X$
10. Energy consumption in managing pkt. in $Y = E_{P'}^Y$
11. Energy consumption in mode change in X to $Y = E_{P'}^{XY}$
12. Energy consumption in mode change in X to $Z = E_{P'}^{XZ}$
13. Energy consumption in mode change in Y to $Z = E_{P'}^{YZ}$
14. Energy consumption in mode change in Z to $X = E_{P'}^{ZX}$.

The nodes in the network consume a steady level of energy consumption during its steady mode, but this can be calculated by using the traditional probabilistic approach with the assumed set of nodes in the network as X, Y, Z as follows: $P(X_i)$, $P(Y_i)$ and $P(Z)$. The mathematical computations of these probabilistic values are summarized as follows:

$$P(X_i) = f(x)\eta p_1^i, \quad i = 0, 1, 2, 3, \tag{5}$$

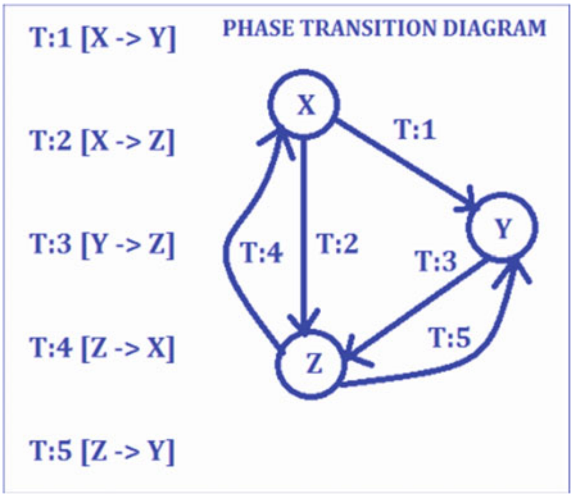


Fig. 6 Phase transition

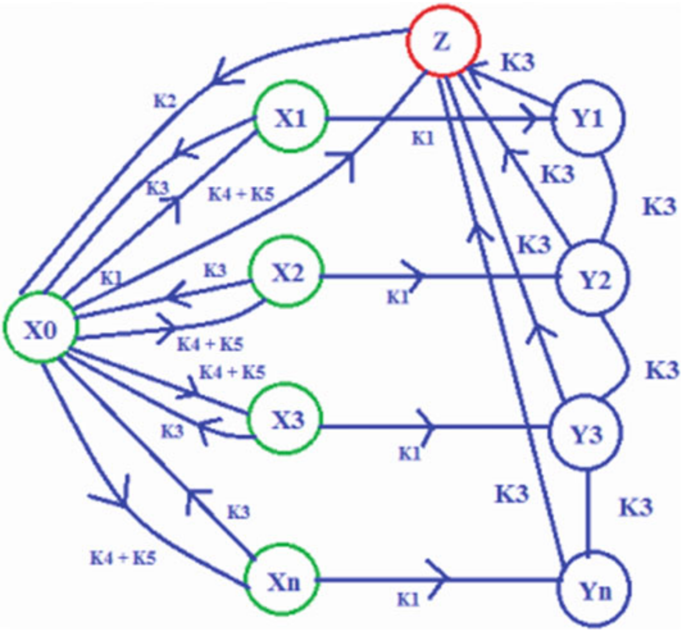


Fig. 7 Phase transitions among all the nodes

$$P(X_i) = f(x)\eta p_1^{i-1}, \quad i = 0, 1, 2, 3, \tag{6}$$

$$P(Z) = f(x)(k_1 + k_3 p_2) \tag{7}$$

where $f(x) = \frac{(i-p_1)}{\eta(1+p_1)+(1-p_1)+(k_1+k_2k_3)}$, $p_1 = \frac{1}{2k_3}g(k)$, $p_2 = \frac{k_1p_1}{k_3(1-p_1)}$ and $g(k) = \bar{k} - \sqrt{\bar{k}} - 4k_3(k_4 + k_5)$.

Using Eqs. (5)–(7), the energy consumptions can be measured; with our test bed for the network, the instances are tested for measuring the energy with the energy computing expressions shown below for different situations of the network

$$E_{\mathrm{TX}}^{X} = \frac{f(x)\eta E_P^X}{(1 - P_1)^2} \tag{8}$$

$$E_{\mathrm{TX}}^{Y} = \frac{f(x)\eta p_2 E_P^X}{(1 - P_1)^2} \tag{9}$$

$$E_C^{\mathrm{XZ}} = f(x)\eta k_1 E_{P'}^{\mathrm{XY}} \tag{10}$$

Table 3 Values all the constants when eta = 0.3798

η	p_1	p_2	k_1	k_2	k_3	k_4	k_5	$f(x)$
0.3798	10	0.09	0.1	0.12	0.312	20	30	10.93589
0.3798	20	0.0977	0.134	0.222	0.355	25	33	5.239235
0.3798	30	0.1054	0.211	0.1311	0.398	29	37	4.020895
0.3798	40	0.1131	0.259333	0.1688	0.441	33.66667	39.66667	3.096606
0.3798	50	0.1208	0.314833	0.17435	0.484	38.16667	42.66667	2.560614
0.3798	60	0.1285	0.370333	0.1799	0.527	42.66667	45.66667	2.183997
0.3798	70	0.1362	0.425833	0.18545	0.57	47.16667	48.66667	1.903726
0.3798	80	0.1439	0.481333	0.191	0.613	51.66667	51.66667	1.686577
0.3798	90	0.1516	0.536833	0.19655	0.656	56.16667	54.66667	1.513189
0.3798	100	0.1593	0.592333	0.2021	0.699	60.66667	57.66667	1.371453
0.3798	110	0.167	0.647833	0.20765	0.742	65.16667	60.66667	1.253379

$$E_C^{\text{XY}} = \frac{f(x)\eta k_1 p_1 E_P^X}{(1 - P_1)^2} \tag{11}$$

$$E_C^{\text{ZX}} = f(x)(k_1 + k_3 p_2)\eta E_{P'}^{\text{ZX}} \tag{12}$$

Also delay is an important QoS parameter of any communications and is a very much inevitable factor, and the energy consumption during this delay process is thus again an important quantity to be measured. The packet management is sometimes delayed by some factors, and for that, the process needs the energy in that additional activity. The mathematical expression we get after considering all the factors due to which a delay may occur is summarized as follows:

$$E_{\text{pktmanagement}}^{\text{Delay}} = \frac{f(x)\eta(p_1 + p_2)}{(1 - p_1)^2} \tag{13}$$

Throughput is another QoS parameter where again some energy is consumed so the efficient calculation of this energy component is also important. The mathematical expression for this can be as follows(Tables 3, 4, 5, 6, 7, 8 and 9; Figs. 8, 9, 10, 11 and 12):

$$E_{\text{Process}}^{\text{Throughput}} = (1 - k_1 + k_2 + k_3 p_2)k_3 \tag{14}$$

$$E_{\text{Total}} = E_{\text{TX}}^X + E_{\text{TX}}^Y + E_C^{\text{XZ}} + E_C^{\text{XY}} + E_C^{\text{ZX}} + E_{\text{pktmanagement}}^{\text{Delay}} + E_{\text{Process}}^{\text{Throughput}} \tag{15}$$

Table 4 Power consumptions when eta = 0.3798

E_X_P	sqr_(1-pl)	E_Y_P	E_XZ_P	E_XY_P″	E_ZX_p″
0.121	81	0.213	0.213	0.162	0.133
0.133	361	0.162	0.472	0.317	0.155
0.214	841	0.317	0.627	0.213	0.211
0.249	1521	0.213	0.213	0.472	0.244333
0.2955	2401	0.472	0.452	0.627	0.283333
0.342	3481	0.627	0.511	0.213	0.322333
0.3885	4761	0.213	0.57	0.452	0.361333
0.435	6241	0.452	0.629	0.213	0.400333
0.4815	7921	0.511	0.627	0.472	0.439333
0.528	9801	0.57	0.213	0.627	0.478333
0.5745	11,881	0.629	0.452	0.213	0.517333

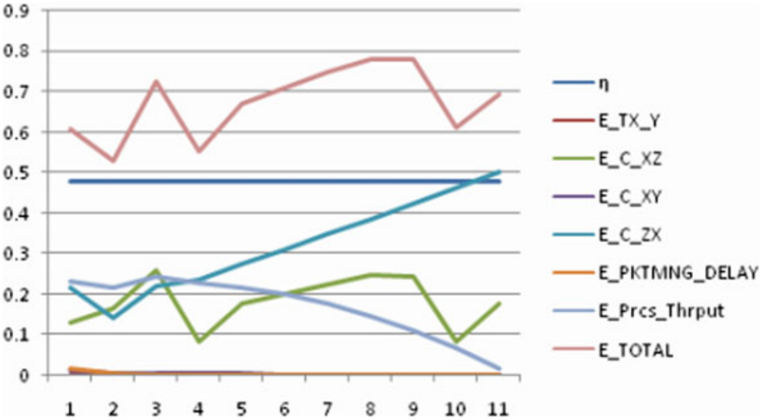


Fig. 8 Graphical analysis of the energy consumptions when eta = 0.3798

4 Findings

1. The energy harvesting enhances the power management in the nodes.
2. The soft clustering handles to form the efficient cluster heads to achieve the more energy saving by reducing the noise members in the network.
3. The performance in terms of energy is distinctly visible in case of EEA and EEAS, though the EEAS is a bit energy consuming but here the security aspect is considered.

Table 5 Final energy consumptions in transmission when eta = 0.3798

E_TX_X	E_TX_Y	E_C_XZ	E_C_XY	E_C_ZX	E_PKTMNG _DELAY	E_Prcs _Thrput	E_TOTAL
0.009035	0.00143138	0.12882422	0.01209617	0.214729	0.014799143	0.23459904	0.606479
0.000955	0.000113653	0.16395462	0.00610048	0.141261	0.002464024	0.216307358	0.530201
0.000499	7.78385E−05	0.2592034	0.00314107	0.217945	0.001055221	0.245148418	0.726571
0.000245	2.37418E−05	0.08280174	0.00482539	0.236036	0.000536045	0.230197399	0.55442
0.000152	2.93476E−05	0.17586304	0.00508021	0.272428	0.000317609	0.218937142	0.672655
0.000103	2.43555E−05	0.19913332	0.0014307	0.309672	0.00020489	0.201338857	0.711804
7.48E−05	5.58324E−06	0.22240404	0.00259302	0.347414	0.000140303	0.17731712	0.749874
5.65E−05	8.45422E−06	0.24559803	0.00106608	0.385494	0.000100448	0.146786508	0.779053
4.42E−05	7.11451E−06	0.24485655	0.00209435	0.423827	7.445E−05	0.109661596	0.780521
3.55E−05	6.10604E−06	0.08315434	0.00249748	0.462367	5.67374E−05	0.065856961	0.613939
2.91E−05	5.32402E−06	0.17633054	0.00076932	0.501085	4.42395E−05	0.015287179	0.693521

Table 6 Values all the constants when eta = 1

η	p_1	p_2	k_1	k_2	k_3	k_4	k_5	$f(x)$
1	10	0.09	0.1	0.12	0.312	20	30	61.24115
1	20	0.0977	0.134	0.222	0.355	25	33	6.450102
1	30	0.1054	0.211	0.1311	0.398	29	37	4.443249
1	40	0.1131	0.259333	0.1688	0.441	33.66667	39.66667	3.276185
1	50	0.1208	0.314833	0.17435	0.484	38.16667	42.66667	2.657135
1	60	0.1285	0.370333	0.1799	0.527	42.66667	45.66667	2.242085
1	70	0.1362	0.425833	0.18545	0.57	47.16667	48.66667	1.941472
1	80	0.1439	0.481333	0.191	0.613	51.66667	51.66667	1.712515
1	90	0.1516	0.536833	0.19655	0.656	56.16667	54.66667	1.53179
1	100	0.1593	0.592333	0.2021	0.699	60.66667	57.66667	1.38525
1	110	0.167	0.647833	0.20765	0.742	65.16667	60.66667	1.263898

Table 7 Power consumptions when eta = 1

E_X_P	sqr_(1-p1)	E_Y_P	E_XZ_P	E_XZ_P″	E_ZX_p″
0.121	81	0.213	0.213	0.162	0.133
0.133	361	0.162	0.472	0.317	0.155
0.214	841	0.317	0.627	0.213	0.211
0.249	1521	0.213	0.213	0.472	0.244333
0.2955	2401	0.472	0.452	0.627	0.283333
0.342	3481	0.627	0.511	0.213	0.322333
0.3885	4761	0.213	0.57	0.452	0.361333
0.435	6241	0.452	0.629	0.213	0.400333
0.4815	7921	0.511	0.627	0.472	0.439333
0.528	9801	0.57	0.213	0.627	0.478333
0.5745	11,881	0.629	0.452	0.213	0.517333

5 Future Work

The work is though a better version to the earlier works, but there are ample opportunities to do farther better incorporating some more aspects in terms of energy spending in a network communication. Our next work is also in the process where we will introduce a novel technique to achieve more efficiency in the energy consumption in the network including IoT environment using deep learning

Table 8 Final energy consumptions in transmission when eta = 1

E_TX_X	E_TX_Y	E_C_XZ	E_C_XY	E_C_ZX	E_PKTMNG_DELAY	E_Prcs_Thrput	E_TOTAL
0.091484	0.01449374	1.30443658	0.1224823	1.043221	0.149851812	0.23459904	2.869084
0.002376	0.000282793	0.40795603	0.01517936	0.168644	0.006131047	0.216307358	0.814501
0.001131	0.000176524	0.58782845	0.00712341	0.237146	0.002393059	0.245148418	1.079816
0.000536	5.18897E−05	0.18096988	0.01054628	0.247517	0.001171569	0.230197399	0.670454
0.000327	6.31001E−05	0.37812265	0.01092294	0.281041	0.000682889	0.218937142	0.88977
0.00022	5.18942E−05	0.42429282	0.0030484	0.31658	0.00043656	0.201338857	0.945749
0.000158	1.18301E−05	0.47124373	0.00549426	0.353192	0.000297282	0.17731712	1.007556
0.000119	1.78476E−05	0.51847867	0.00225059	0.390466	0.000212055	0.146786508	1.058212
9.31E−05	1.4981E−05	0.51559224	0.00441005	0.428197	0.000156769	0.109661596	1.058033
7.46E−05	1.28336E−05	0.17477289	0.00524918	0.466269	0.00011925	0.065856961	0.71228
6.11E−05	1.11744E−05	0.3700954	0.00161471	0.504612	9.2853E−05	0.015287179	0.891713

Table 9 Comparison among other algorithms

Performance analysis

Energy consumption	Methods	No. of nodes						
		25	50	75	100	125	150	175
	EEA	2.22	2.16	2.11	2.03	1.92	1.89	1.78
	EEAS	2.31	2.21	2.17	2.11	2.05	1.99	1.89
	AEEC	2.37	2.442	2.21	2.32	2.06	2.11	2.42
	LEMA	2.44	2.56	2.32	2.42	2.16	2.21	2.54
	LEACH	2.65	2.78	2.49	2.91	2.29	2.52	2.52

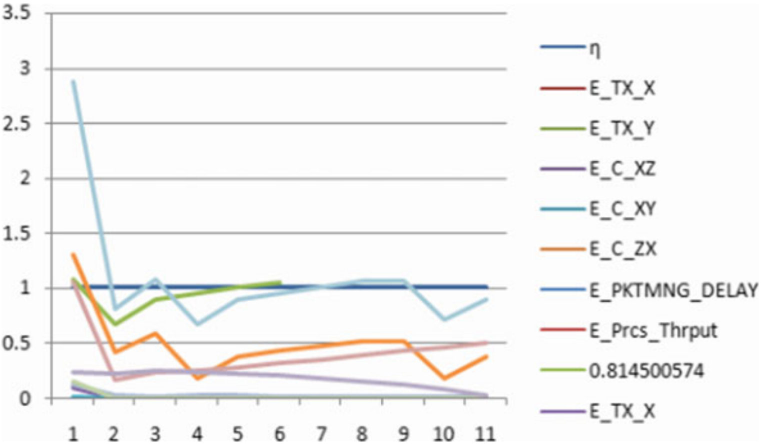


Fig. 9 Graphical analysis of the energy consumptions when eta = 1

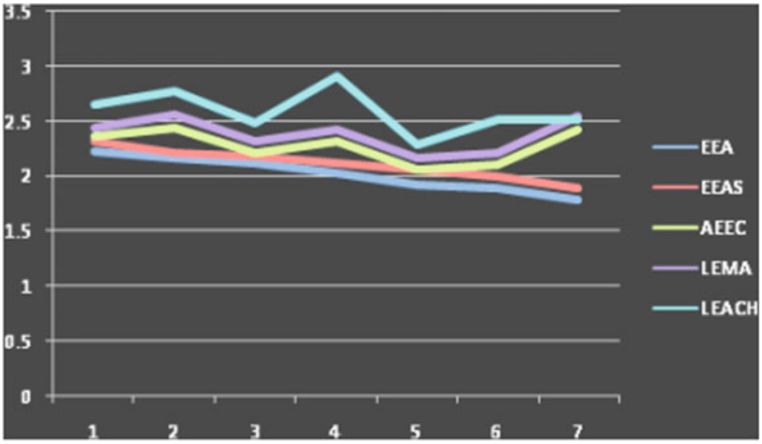


Fig. 10 Graphical comparisons among other algorithms

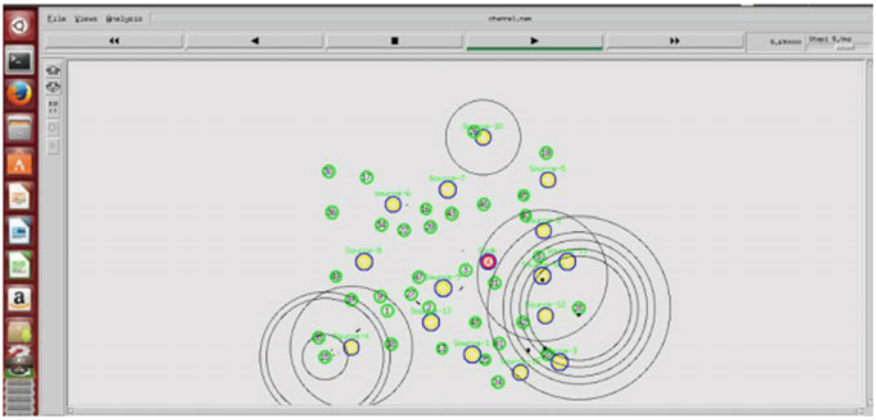


Fig. 11 Active nodes with their respective clusters

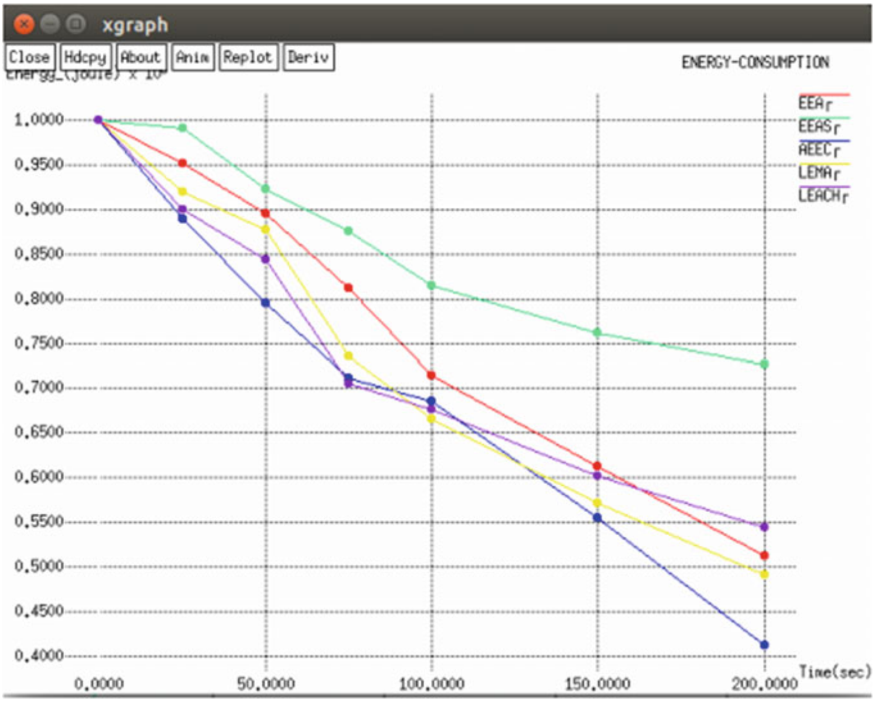


Fig. 12 Energy spending in different algorithms

6 Conclusion

The work is achieved few milestones like it achieves energy efficiency along with a secured communications. The work also achieved efficient clustering using soft clustering. The QoS is tested using different network sizes, and it is observed that the performance is quite satisfactory with all the QoS parameters considered.

References

1. Abbaspour R (2010) A practical approach to powering wireless sensor nodes by harvesting energy from heat in room temperature. In: Proceedings of IEEE, ICUMT 2010, pp 178, 181
2. Bozorgi SM, Bidgoli AM (2018) Wireless Netw. https://doi.org/10.1007/s11276-018-1744-x
3. Chalasani S, Conrad JM (2008) A survey of energy harvesting sources for embedded systems. In: Proceedings of the IEEE Southeastcon 2008, pp 442–447
4. Fei F, Mai JD, Li WJ (2012) A wind-utter energy converter for powering wireless sensors. Sens Actuators A: Phys 173(1):163–171
5. Heer R, Wissenwasser J, Milnera M, Farmer L, Hopfner C, Vellekoop M (2010) Wireless powered electronic sensors for biological applications. In: Proceedings of IEEE EMBC 2010, pp 700–703
6. Hudak NS, Amatucci GG (2008) Small-scale energy harvesting through thermoelectric, vibration, and radio frequency power conversion. J Appl Phys 103(10):1–24
7. Liu F, Phipps A, Horowitz S, Ngo K, Cattafesta L, Nishida T, Sheplak M (2008) Acoustic energy harvesting using an electromechanical Helmholtz resonator. J Acoust Soc Am 123(4):1983–1990
8. Lu X, Yang S-H (2010) Thermal energy harvesting for WSNs. In: Proceedings of IEEE SMC 2010, pp 3045, 3052
9. Mandal S, Turicchia L, Sarpeshkar R (2010) A low-power, battery-free tag for body sensor networks. IEEE Pervas Comput 9(1):71–77
10. Matova SP, Elfrink R, Vullers RJM, van Schaijk R (2011) Harvesting energy from air own with a michromachined piezoelectric harvester inside a Helmholtz resonator. J Micromech Microeng 21(10):1, 6
11. Merrett GV, Weddell AS, Lewis AP, Harris NR, Al-Hashimi BM, White NM (2008) An empirical energy model for supercapacitor powered wireless sensor nodes. In: Proceedings of IEEE ICCCN 2008, St. Thomas, US Virgin Islands, 3–7 Aug 2008, pp 1–6
12. Mitcheson PD, Yeatman EM, Rao GK, Holmes AS, Green TC (2008) Energy harvesting from human and machine motion for wireless electronic devices. Proc IEEE 96(9):1457, 1486
13. Moghe R, Yang Y, Lambert F, Divan D (2009) A scoping study of electric and magnetic field energy harvesting for wireless sensor networks in power system applications. In: Proceedings of IEEE ECCE 2009, San Jose, CA, 20–24 Sept 2009, pp 3550–3557
14. Muthukumaran K, Chitra K, Selvakumar C (2017) Energy efficient clustering in wireless sensor networks. In: 2017 international conference on inventive computing and informatics (ICICI), Coimbatore, 2017, pp 351–355
15. Raghunathan V, Kansal A, Hsu J, Friedman J, Srivastava MB (2005) Design considerations for solar energy harvesting wireless embedded systems. In: Proceedings of ACM/IEEE IPSN 2005, Los Angeles, CA, 25–27 April 2005, pp 457, 462
16. Reinisch H, Gruber S, Unterassinger H, Wiessecker M, Hofer G, Pribyl W, Holweg G (2011) An electro-magnetic energy harvesting system with 190 nW idle mode power consumption for a BAW based wireless sensor node. IEEE J Solid-State Circ 46(7):1728, 1741
17. Roundy SJ (2003) Energy scavenging for wireless sensor nodes with a focus on vibration to electricity conversion. Ph.D. thesis, University of California at Berkeley, Berkeley, CA

18. Sardini E, Serpelloni M (2011) Self-powered wireless sensor for air temperature and velocity measurements with energy harvesting capability. IEEE Trans Instr Meas 60(5):1838–1844
19. Sherrit S (2008) The physical acoustics of energy harvesting. In: Proceedings of IEEE IUS 2008, pp 1046, 1055
20. Sue C-Y, Tsai N-C (2012) Human powered MEMS-based energy harvest devices. Appl Energy 93:390–403
21. Tan YK, Panda SK (2011) Energy harvesting from hybrid indoor ambient light and thermal energy sources for enhanced performance of wireless sensor nodes. IEEE Trans Ind Electron 58(9):4424–4435
22. Tan YK, Panda SK (2011) Self-autonomous wireless sensor nodes with wind energy harvesting for remote sensing of wind-driven wide spread. IEEE Trans Instr Meas 60(4):1367–1377
23. Torres EO (2010) An electrostatic CMOS/BiCMOS Li ion vibration-based harvester-charger IC. Ph.D. thesis, Georgia Institute of Technology
24. Torres EO, Rinc on-Mora GA (2005) Long-lasting, self-sustaining, and energy harvesting system-in-package (SiP) wireless micro-sensor solution. In: Proceedings of INCEED 2005, Charlotte, NC, 24–30 July 2005
25. Webster JG (1998) The measurement, instrumentation, and sensors handbook. The electrical engineering handbook series. CRC Press
26. Xu C, Pan C, Liu Y, Wang ZL (2012) Hybrid cells for simultaneously harvesting multi-type energies for self-powered micro/nanosystems. Nano Energy 1(2):259–272

Optimizing View Synthesis Approach for TerraSAR-X Image Registration Using Decision Maker Framework

B. Sirisha, B. Sandhya, J. Prasanna Kumar, and T. Chandrakanth

Abstract Image registration using iterative view synthesis algorithm is experimentally shown to solve a wide range of registration problems, albeit at the cost of additional memory and time to be spent on generation of views and feature extraction across all the views. This paper tries to optimize the iterative algorithm of image registration using prior knowledge of possible deformations between the images. The proposed approach incorporates a trained classifier model into the registration pipeline. This classifier model can predetermine the possibility of registering the input image pairs and also predict the minimum number of synthetic views necessary to register the input image pair. Hence for the images that have been registered, an additional time requirement for the proposed approach is tolerable. However for the images that are not registered, the gain in time because of classifier model is extremely significant.

Keywords Terra SAR-X · Image registration · Feature detection · Feature description · View synthesis

1 Introduction

Image registration is a classical research problem that has been extensively studied. Image registration or alignment is a fundamental task in remote sensing, which spatially transforms the sensed image coordinates into the reference image coordinates. However, finding a robust fully automatic image registration approach has been a challenge. Most of the existing registration approaches are limited to the designed application, sensor and characteristics of imaged region. Hence, TerraSAR-X image registration needs further focus. In addition, the rapid development of TerraSAR-X image acquisition devices and also growing number and diversity of images have

B. Sirisha (✉) · B. Sandhya · J. P. Kumar
MVSR Engineering College, Hyderabad, India

T. Chandrakanth
Hexagon, Hyderabad, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_7

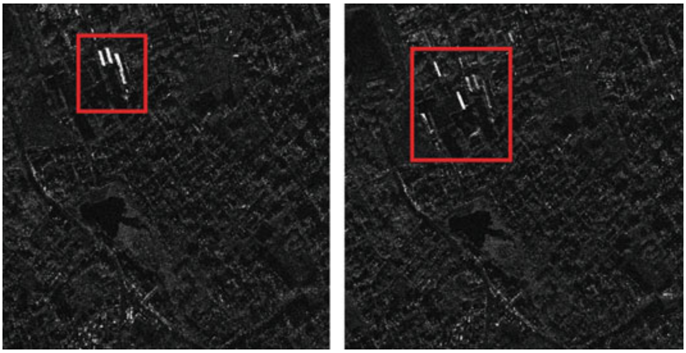


Fig. 1 TerraSAR-X images acquired at different look angles

posed need and challenges for research on automatic TerraSAR-X image registration [1, 2]. The problem is to address the

1. Challenges due to look angle and incidence angle parameter

    - (1) Incidence angle of the TerraSAR-X sensor alters the characteristics of the backscattered signal significantly. Figure 1 shows two TerraSAR-X images acquired at different look angles, it can be observed that the backscattering patterns of the identical objects of the same scene appear different, and the idea of detecting the same feature/control points becomes more challenging. (2) The geometric distortion is a function of the incidence angle of the radar beam; hence, TerraSAR-X image pairs emerging from varying angles contain a lot of geometric distortion. (3) It is extremely difficult to precisely locate the control points (CPs) as large look/view angle images are prone to blurring and resolution change.

2. Inherent complexities of TerraSAR-X images like

    - (1) Side-ward looking geometry and non-intuitive nature of TerraSAR-X imaging sensor. (2) Multiplicative nature of the inherent speckle noise. (3) Complex backscattering within images.

Automation in TerraSAR-X image registration has been extensively studied and formulated in the past few years, but the exponential growth in TerraSAR-X imaging sensors and diversity of TerraSAR-X images have led to challenges for fully automatic TerraSAR-X image registration [3, 4]. In the literature, image registration approaches are mostly classified into area and feature-based methods [5, 6]. In this paper, the focus is on feature-based image registration combined with view synthesis approach for addressing contemporary challenges in TerraSAR-X image registration. The standard feature-based registration approach includes the extraction of the local feature points, generation of matches and their geometric validation with the computed homography [7]. This approach is mainly suitable for simple registration problems, where the deformation complexity between the input image pair is very minimal. Iterative registration approach using view synthesis is widely implemented for registering complex registration problems, where the deformation complexity between the input image pair is extremely high [8]. In this approach, synthetic view images are gradually increased as the iterations progress until the images are registered. The time taken to register images increases with the increase in the iterations. Hence, the focus of the paper is to optimize the iterative image registration algorithm using prior knowledge of possible deformations between the images by predetermining minimum number of views needed to register the input image pairs. Applications like registration aided navigation system, the error in navigation is reduced by combining the current navigation information (obtained through INS) and the position update coordinates information through the registration of a real-time sensed TerraSAR-X image with respect to geo-referenced image [3]. In this scenario, the deformation information obtained from the trajectory helps in prediction. The contribution made in this respect is twofold: **(1) Incorporation of knowledge into the feature-based registration algorithm by building a trained classifier model that can be used for real-time registration of input image pairs. (2) An effort has been made to identify the feature attributes to train and build a classifier model to predict the synthetic views required for registering the images**. This paper is organized as follows: Sect. 2 deals with related work in iterative registration using view synthesis (IRVS), Sect. 3 describes the iterative registration algorithm and its technical challenges, Sects. 4 and 5 present the proposed approach, evaluation parameters, results, and finally conclusion is provided in Sect. 6.

2 Related Work

To improve the invariance toward larger deformation, iterative image registration approaches have been proposed. Lepetit [9] has demonstrated that the robustness of feature matching is enhanced by the amalgamation of supplementary synthetic views of a given single, frontal parallel view of an object. Morel [10] has integrated the view synthesis with DOG feature detector and scale invariant feature transform matching. This approach is called affine SIFT, effectively matches the challenging optical image pairs with orientation differences upto 80 °C, albeit at the cost of an additional memory and time to be spent on the generation of views and feature extraction from all the views. Pang [11] has substituted SIFT feature extraction [12] by SURF feature extraction [13] in the Affine SIFT approach to decrease the computational time. The resulting feature matching approach is called FAIR SURF(FSURF). Dmytro [8, 14] has proposed two-view feature matching approach, which integrates the view synthesis with Hessian affine and MSER [15, 16] feature detectors and employs the root SIFT [17] feature matching. This feature matching approach, is called as MODS—matching with on-demand view synthesis, can handle the orientation differences up to 160 °C. Mishkin [18] has integrated the view synthesis with multiple detectors like MSER [19] and Hessian affine [20] and multiple descriptor combination root SIFT and half SIFT [21] for feature matching. The resulting feature matching approach,

Table 1 Comparative study of the various iterative feature-based image registration

Approach	Feature detector	Feature descriptor	Feature matching	Geometric verification	Can handle orientation difference up to (°)	Year
ASIFT	SIFT	SIFT	Second closest ratio	ORSA	80	2009
FAIR SURF	SURF	SURF	Second closest ratio	ORSA	80	2012
MODS	Combined MSER + Hessian affine	Root SIFT	First geometric inconsistent rule. First-to-second closest ratio	DEGEN SAC	160	2013
WxBS	Combined MSER + Hessian affine	Combined root SIFT and half SIFT	First geometric inconsistent rule. First-to-second closest ratio	DEGEN SAC	170	2015
Image matching	Handcrafted features	Learned descriptors	Symmetrical nearest neighbor ratio	DEGEN SAC	240	2020

called WxBS: wide baseline stereo generalizations, effectively matches images that vary in illumination, viewpoint and the type of sensor. D. Mishkin has experimentally analyzed that structure from the motion algorithm improves the performance compared to both contemporary and classical methods [22] A comparison of iterative feature-based image registration approaches employing the view synthesis is presented in Table 1. The information used for comparison is as follows: (1) feature detector, (2) feature descriptor, (3) matching technique, (4) geometric verification and (5) observations regarding the extent of handling orientation difference between the reference and the sensed images. It is observed that research paper which uses view synthesis on Terra SAR-X images does not exist. Improvements over ASIFT [10] for optical images can be observed in [11, 14, 18]. Improvements over ASIFT are not with respect to predicting no. of tilts they are with respect to matching strategy. Our focus is to predetermine the possibility of registering the input image pairs and also predict the minimum number of synthetic views necessary to register the input image pair.

It is observed that the view synthesis can address a large range of deformations between the images effectively. However, the time taken to register images increases with the increase in the iterations. Iterative image registration approaches using view synthesis like ASIFT and MODS, and all these algorithms have to iterate through all the stages and then conclude that the given input image pair could not be registered. Hence, the proposed approach can be improved by (1) predetermining the possibility of registering the input image pairs before going through the iterations of registration and (2) predetermining the exact iteration required to register the input image pairs.

3 Image Registration Using View Synthesis (IRVS)

To register TerraSAR-X images with variations in look angle and resolution, an iterative feature-based registration approach has been implemented. In this approach, initially features are extracted from reference and sensed TerraSAR-X images using Hessian affine feature detector and SIFT feature descriptor. Hessian affine detector along with SIFT descriptor has been proven to be effective among several scale and affine invariant feature detectors and descriptors. Once features are extracted, feature correspondences are identified using NNR matching technique. The corresponding features are fed to RANSAC algorithm to transform the sensed image coordinates to reference image coordinates. In this stage, we calculate inlier ratio and keypoint error. If inlier ratio >0.1 and keypoint error <10, the images are registered, else they enter into view synthesis stage. Figure 2 shows the iterative registration approach using view synthesis for TerraSAR-X image registration.

1. Stage:1—Synthetic view generation
   - Apply a set of rotations to source and target images.
   - Apply in continuation a set of simulated tilts to rotated source and target images.
2. Stage:2—Feature extraction (detector and descriptor)
   - Detect and extract feature points in each view of sensed and reference image pairs.
   - Extracted features are stored in a vectored array.
3. Stage:3—Feature correspondences
   - Nearest neighbor ratio matching is used to find the correspondences between the pair of images.
4. Stage:4—Homography estimation
   - Using RANSAC algorithm, inliers and outliers are detected, and homography is estimated using the inlier points: error estimation and loop iteration.
   - Error is computed between the transformed and target image.
   - If the error is above a predefined threshold, the process is iterated with increasing the number of views.

In Stage:1—The synthetic views are generated using affine camera model, which uses three major parameters, i.e., latitude (θ), longitude (Φ) and scale (Ψ). The synthetic views generation steps are as follows:

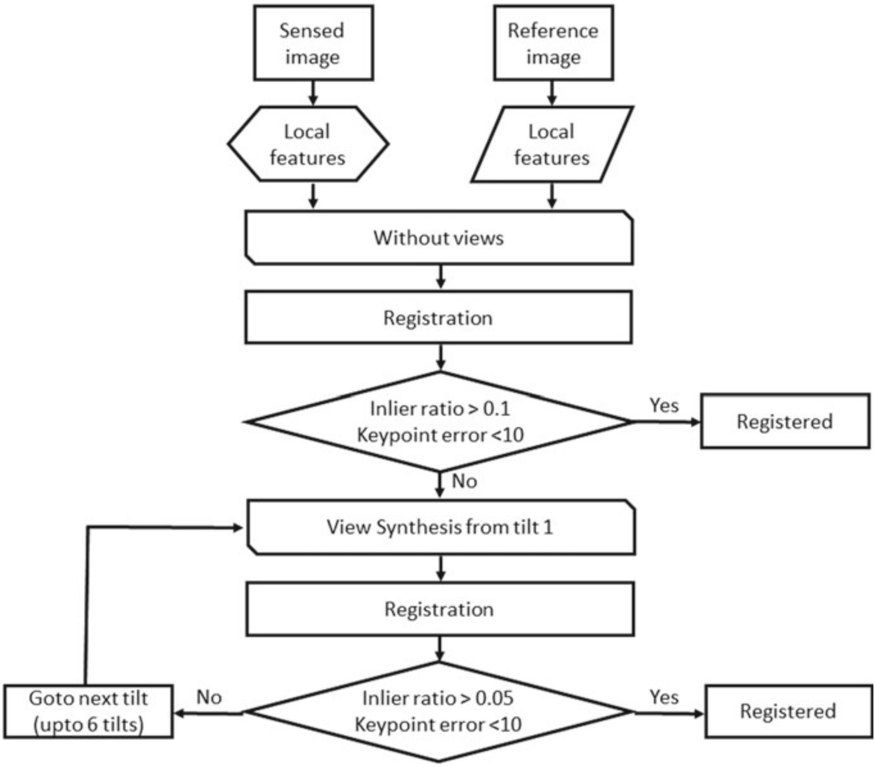


Fig. 2 Image registration using view synthesis (IRVS)

1. The input image scale space is constructed with Gaussian whose variance must satisfy $\sigma - \sigma_{\text{base}} < 1$.
2. The obtained scale space image from step:1 is rotated with the help of longitude parameter ϕ with step size $\Delta\phi = \Delta\phi_{\text{base}/t}$.
3. The rotated image obtained in step:2 is convolved with a Gaussian filter in vertical as well as horizontal directions with $(\sigma = \sigma_{\text{base}})$ and $(\sigma = t.\sigma_{\text{base}})$.
4. Finally, tilt is applied on the image by shrinking with a factor-t along horizontal direction.

In Stage:2, detect and extract feature points in each view of sensed and reference image pairs using Hessian affine detector and SIFT descriptor. Extracted features are stored in a vectored array. In Stage:3, nearest neighbor ratio matching is used to find the correspondences between the pair of images. Finally, in Stage:4 using RANSAC algorithm, inliers and outliers are detected, and homography is estimated using the inlier points[23, 24]. Error is computed between the transformed and target image. If the error is above a predefined threshold, the process is iterated with increasing the number of views. IRVS is implemented to provide robust and accurate registration. This method has been proved in finding more precise and correct feature correspondences over a data set of 540 TerraSAR-X images. It is experimentally tested that out of 540 TerraSAR-X image pairs, standard feature-based image registration approach could register only 197 image pairs, and IRVS approach could register 502 image pairs. **Technical challenges of IRVS**: View synthesis approach of image registration works by generating synthetic views of source and target images. Synthetic view images are gradually increased as the iterations progress until the images are registered. Hence, the space and time complexities of the approach which directly depend on the number of views generated exponentially increase when one has to execute several iterations. Since the number of synthetic views to be generated depends on the kind and amount of deformation, the iterative approach can be optimized in terms of time and space if the deformation between the images is known prior to the registration.

4 Optimization of View Synthesis Approach

The iterations in the view synthesis approach can be reduced if the number of views is known a priori resulting in optimization of time and space required for the registration of source and target images. The number of views is usually a function of geometric variations existing across the images which can be derived from knowing reference image characteristics and source image capture conditions. This paper aims to optimize the iterative algorithm of image registration using prior knowledge of possible deformations between the images. We try to propose this as a machine learning problem by trying to predict the number of views using the classifier with inputs from trajectory information and source, reference images. Proposed approach is shown in Fig. 3.

The classifier model is designed with features extracted from reference image characteristics and deformation information. Reference image characteristics are captured using global image descriptors: color and Hu moments: (10 values), GLCM (4 values), local binary pattern (LBP) (59 values) and local derivative pattern (LDP) (56 values). Deformation information: overlap, scale and orientation angle (3 element vector). The view synthesis approach can be optimized by developing a trained classifier model, and this includes the following activities: **(1) attributes generation for the decision maker, (2) attribute selection and (3) training data generation**. Global features from reference image and deformation information are fed as the attributes for the classifier. The feature attributes are selected using information gain ratio; out of eight attributes overlap, scale, angle and local derivative pattern(LDP) are selected as the key attributes for the classifier model. For learning a classifier, training data is essential. We have generated a data set using look angle varied TerraSAR-X images and induced deformations in terms of scale and rotation. The source images have varying degrees of overlap (Fig. 4; Table 2).

Identifying class label: For training the model, data needs to be generated with the actual value for class label, i.e., angles at which images have been registered. Large

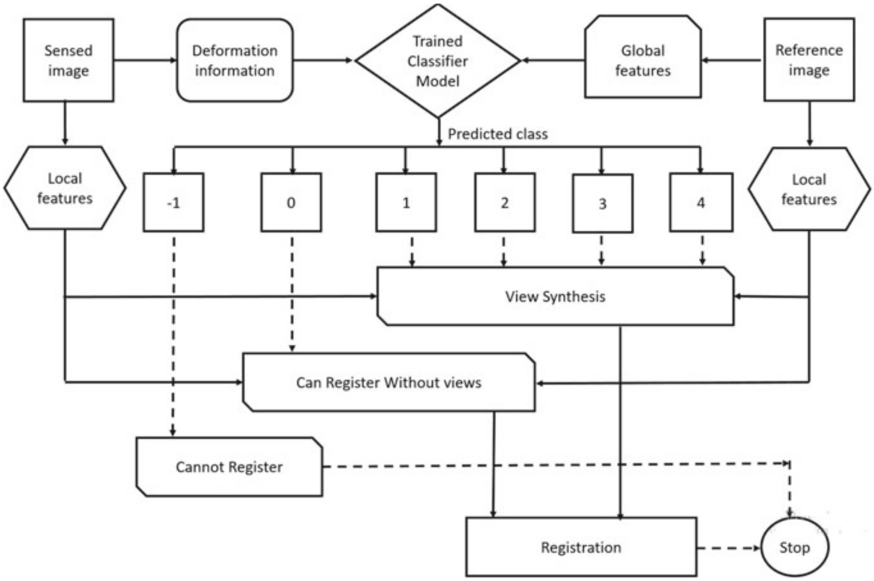


Fig. 3 Proposed approach: optimization of view synthesis approach

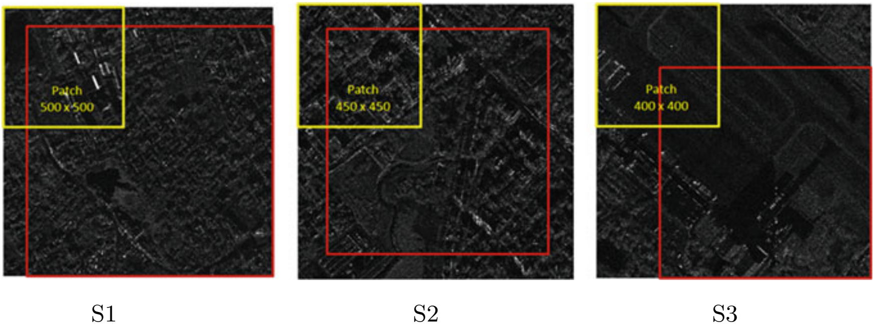


Fig. 4 Generated data set images

Table 2 Generated data sets details

S. No.	Size of ref image	Size of source image	No. of images	Deformation
S1	1000 × 1000	500 × 500 (stride: 25 × 25)	13,133	Rotation 90–270° step 5°
S2	800 × 800	450 × 450 (stride: 50 × 50)	9873	Scale: 0.75–1.25 step 0.25 rotation: 90–270° step 5°
S3	700 × 700	400 × 400 (stride: 100 × 100)	2257	Scale 0.75–1.25 step 0.25 rotation 90–270° step 5°

Table 3 Number of views generated for each tilt

Class	$\triangle\theta$	Θ	No. of views and view angles ϕ
Class:−1			Could not register
Class:0			Could register without views
Class:1	$72/\sqrt{2}$	$\theta < 180$	4; (0°, 51°, 102°, 153°)
Class:2	72/2	$\theta < 180$	5; (0°, 36°, 72°, 108°, 144°)
Class:3	72/2	$\theta < 180$	8; (0°, 25°, 50°, 75°, 100°, 125°, 150°, 175°)
Class:4	$72/2\sqrt{2}$	$\theta < 180$	Views = 17; (0°, 25°, 50°, 75°, 100°, 125°, 150°, 175°)

data set of TerraSAR-X images addressing various deformations is generated. Pairs of TerraSAR-X images are registered using iterative view synthesis approach. Seven thousand and four hundred and sixty-four training images are registered using view synthesis approach and class label saved for each pair of the images. The process does not involve any subjective evaluation as the class is assigned based on the registration error measured and inlier ratio. Attributes such as color and Hu moments, GLCM, LBP, LDP, overlap, scale and orientation are evaluated and saved along with the corresponding class label. Deformation information is manually entered for each pair of reference and source images. A class label is assigned to each vector by the objective analysis of the result of registration; class:−1 being 'Cannot register', class: 0 being 'Can register without view synthesis', class:1 being 'Can register using View Synthesis—4 views at angles 0, 51, 102, 153', class:2 being 'Can register using View Synthesis; 5 views at angles 0, 36, 72, 108, 144' and class:3 being 'Can register using View Synthesis; 8 views at angles 0, 25, 50, 75, 100, 125, 150, 175'. Class:4 being 'Can register using View Synthesis; 5 views at angles 0,51,102,153'. Support vector machine (SVM) classifier is trained with the data generated. The output of the trained classifier model is one of the six classes. Based on the output of the decision maker, the registration approach exits successfully in the case of class:0, class:1 or continues the registration using view synthesis in all other classes. Table 3 shows the view angles at which the image pairs would be registered for class-1,2 and 3. Given a pair of images, to be registered, attributes needed for the classifier are generated. Saved model predicts the view at which images have to be registered. Synthetic views of sensed image are generated corresponding to the view and registered. All the modules are implemented in C++ using OpenCV libraries.

5 Experimental Results

The following section describes the effectiveness of the trained classifier model and time analysis of proposed optimized approach.

Table 4 Results for trained classifier model (9250 instances) using SVM

Classifier	Class	TP rate	FP rate	Precision	Recall	ROC area	Accuracy
SVM	−1	0.695	0.044	0.897	0.795	0.894	88.682%
	0	0.955	0.035	0.969	0.955	0.982	
	1	0.923	0.041	0.956	0.923	0.909	
	2	0.893	0.051	0.762	0.781	0.876	
	3	0.736	0.059	0.81	0.656	0.916	
	4	0.692	0.071	0.79	0.636	0.876	

Table 5 Time taken for registration in two registration models for reference (a) and sensed (7) images (6 class)

Registration type	Tilt	Time (s)
View synthesis	3	1344.107
No iteration	3	965.394

Implementation details and parameter setting of the algorithm ***Feature detection parameters***: max iterations = 16; initial sigma = 1.6f; convergence threshold = 0.05; patch size = 41; smm window size =19; up-scale input image = 0; number of scales = 3; initial sigma = 1.6f; threshold = 16.0f/3.0; edge Eigen value ratio = 10.0f; border = 5. **Feature description parameters** are contrast threshold: 0.04, edge threshold:10, Gaussian sigma: 1.6. **Feature matching parameters** are two-way NNR ratio: 1.1, Bhattacharya distance. **Transformation estimation: RANSAC parameters** are max no. of trails: 2000, probability that one random sample is free from outlier: 0.99.

Effectiveness of Trained Classifier Model: Nine thousand and two hundred and fifty TerraSAR-X image pairs are generated from thirteen reference images that are used for generating the training data. Support vector machine classifier is trained on 9250 training records with six classes. Class distribution is divided as [class-0 = 1881, class-1 = 1263, class-2 = 3707, class-3 = 863, class-4 = 708 and class-5 = 830]. The performance of the support vector machine model is verified using tenfold cross-validation. The trained SVM classifier model's effectiveness is assessed with the help of accuracy measure. Table 4 shows the true positive rate, false positive rate, precision, recall and ROC area of each class. It can be observed that ROC area is uniform across the classes. The trained classifier model accuracy obtained with tenfold cross-validation is about 88.682%.

Execution time of proposed approach Time of optimized approach is compared with the iterative image registration approach. Figures 5 and 6 show the results of registering two images using view synthesis and proposed optimized model. As can be observed, image could be registered in the third tilt directly because of the prediction from classifier (Table 5)).

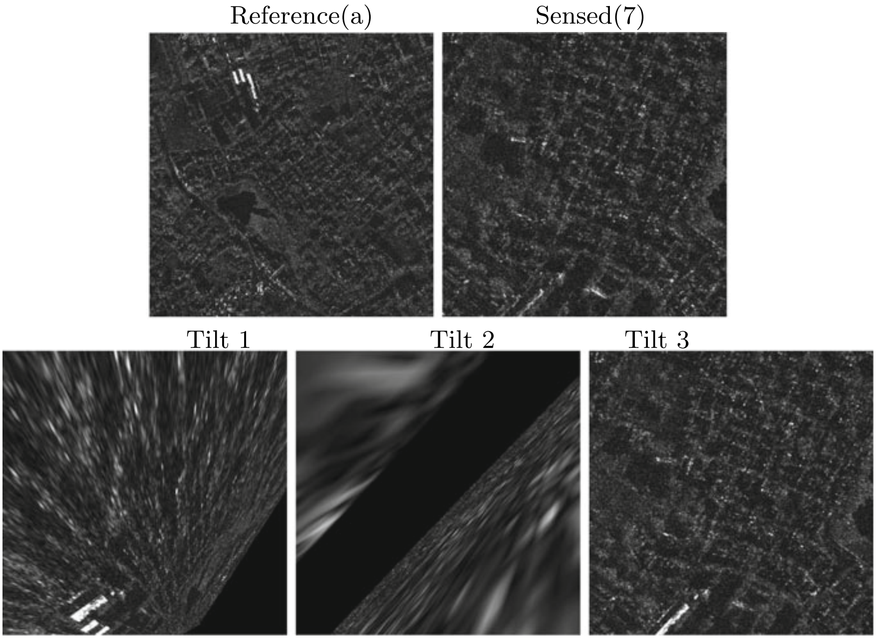


Fig. 5 View synthesis model output

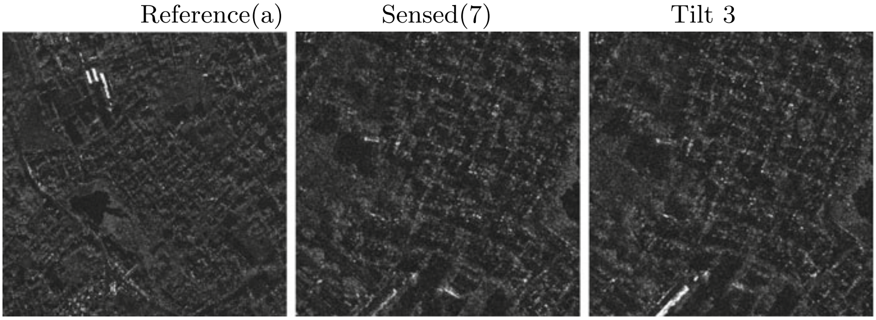


Fig. 6 No iteration model (6 class) output

Table 6 shows the time taken for registration for 11 sample image image pairs. Figure 7 shows the corresponding graph. It is seen that for images which cannot be registered (class-1), time saved is huge as the algorithm quite without going through the iterations of view synthesis. For images which can be registered in tilt 3 (class-3), save in time is quite considerable (one example shown in Table 5).

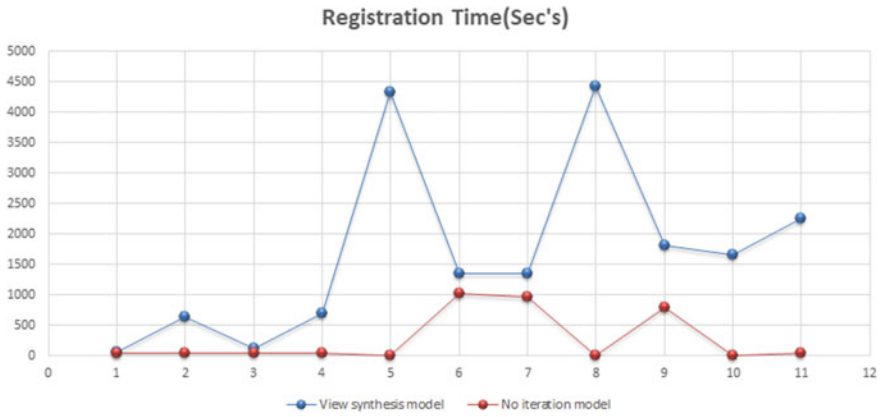


Fig. 7 Graph showing time taken for registration in two registration models (6 class)

Table 6 Time taken for registration in two registration models (6 class)

Sensed image	View synthesis model		No iteration model	
	Tile of registration	Time in sec's	Predicted class	Time in sec's
1	1	60.484	1	47.972
2	2	632.282	1	44.838
3	1	108.097	1	47.227
4	2	692.608	1	44.965
5	−1	4335.374	−1	0.006
6	3	1351.677	3	1017.756
7	3	1344.107	3	965.394
8	−1	4425.593	−1	0.006
9	3	1806.001	3	789.554
10	−1	1653.9065	−1	0.001
11	−1	2254.7615	1	36.512

6 Conclusion

In the proposed approach, we have adapted the view synthesis by incorporating a decision maker model which helps to predetermine the possibility of registering two images and also predict the iteration at which the image pair will be registered, without actually registering them. It is observed that the average time taken by the decision maker is significant compared to the time taken by any other process of image registration such as feature extraction, feature matching and transformation estimation. Hence for the images that have been registered, an additional time requirement for the proposed approach is tolerable. However for the images that are not registered, the gain in time because of DM is extremely insignificant. The proposed approach is meant to improve the execution performance (time and space) unlike iterative view synthesis approach. The proposed approach is found to be efficient in addressing the extreme geometric deformations, which are not registrable with previous state of the art. The main drawback of iterative view synthesis approach is the algorithm is fast for simple registration problems and consumes space and time for extremely hard registration problems because generation of synthetic views is done until a valid geometric estimate is achieved. For image pairs which cannot be registered, the iterative approach has to iterate through all the tilts and then concludes that the given image pair could not be registered. This setback is overcome by the proposed algorithm, and in our approach, the algorithm exits dynamically when a valid geometric estimate is not attained.

References

1. Oliver C et al (2004) Understanding synthetic aperture radar images. SciTech Publishing, Herndon, VA
2. Cumming G et al (2005) Digital processing of synthetic aperture radar data: algorithms and implementation. Artech House, Norwood, MA
3. Ghaffary BK et al (1983) A survey of new techniques for image registration and mapping. Proc SPIE Appl Digital Image Process 432:222239
4. Fonseca LMG et al (1996) Registration techniques for multisensor remotely sensed images. Photogram Eng Rem Sens 1049–1056
5. Brown LG (1992) A survey of image registration techniquess. ACM Comput Surv 24:325–376
6. Zitov B et al (2003) Image registration methods: a survey. Image Vis Comput 21(11):977–1000
7. Eastman RD, Netanyahu NS, Le Moigne J (2011) Survey of image registration methods. Image Registration Remote Sens 21:35–76
8. Mishkin D et al (2013) Two-view matching with view synthesis revisited. In: IVCNZ, pp 436–441
9. Lepetit V et al (2006) Keypoint recognition using randomized trees. IEEE Trans Pattern Anal Mach Intell 28(9):1465–1479. https://doi.org/10.1109/515TPAMI.2006.188
10. Morel J-M et al (2009) Asift: a new framework for fully affine invariant image comparison. SIAM J Imag Sci 2(2):438–469
11. Pang Y et al (2012) Fully affine invariant surf for image matching. Neurocomputing 85:610. https://doi.org/10.1016/j.neucom.2011.12.006
12. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110
13. Bay H et al (2008) Speeded-up robust features (SURF). Comput Vis Image Underst 110:346–359
14. Mishkin D et al (2015) Mods: fast and robust method for two-view matching. CoRR abs/1503.02619
15. Mikolajczyk K et al (2004) Scale and affine invariant interest point detectors. Int J Comput Vis 60(1):63–86
16. Mikolajczyk K et al (2001) Indexing based on scale invariant interest points. In: Proceedings of the 8th international conference on computer vision, pp 525–531
17. Arandjelovic R et al (2012) Tree things everyone should know to improve object retrieval. In: CVPR
18. Mishkin D et al (2015) WxBS: wide baseline stereo generalizations. In: Proceedings of the British machine vision conference. BMVA

19. Matas J et al (2002) Robust wide baseline stereo from maximally stable extrema regions. In: BMVC, pp 384–393
20. Mikolajczyk K, Schmid C (2002) An affine invariant interest point detector. In: Heyden A, Sparr G, Nielsen M, Johansen P (eds) Computer vision — ECCV 2002. ECCV 2002. Lecture notes in computer science, vol 2350. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-47969-4_9
21. Kelman A, Sofka M, Stewart CV (2007) Keypoint descriptors for matching across multiple image modalities and non-linear intensity variations. In: 2007 IEEE conference on computer vision and pattern recognition, pp 1–7. https://doi.org/10.1109/CVPR.2007.383426
22. Jin Y et al (2020) Image matching across wide baselines: from paper to practice. arXiv preprint arXiv:2003.01587
23. Hartley RI, Zisserman A (2004) Multiple view geometry in computer vision, second edition, Cambridge University Press, ISBN: 0521540518
24. Lebeda K et al (2012) Fixing the locally optimized RANSAC. In: BMVC

Partitioning Attacks Against RPL in the Internet of Things Environment

Rashmi Sahay, G. Geethakumari, and Barsha Mitra

Abstract Billion of physical devices are connected by the Internet of Things (IoT), leveraging the advancements in embedded technologies like sensors and RFIDs. Embedded devices have limited energy resources, memory, computational power, and radio range. A network of such embedded devices is often termed as the low power and lossy networks (LLNs). These present routing constraints that are satisfied by the IPv6 routing protocol for LLNs (RPL). However, security is a significant concern in RPL supported IoT-LLNs. RPL organizes LLNs as destination-oriented acyclic graphs (DODAGs). Constituent devices in the DODAG connect to the intended IoT application through the root (sink) node. Hence, any node in the DODAG should have an active connection to the sink. A malicious node instigating a partitioning attack disconnects a live node from the root node. In the present work, we investigate partitioning attacks against RPL in IoT-LLNs and suggest a mechanism based on analyses of messages exchanged among the sensor nodes to detect the attack.

Keywords Internet of things · RPL · Topological attacks · Partitioning attacks

1 Introduction

Recent past has witnessed the Internet of things connect billions of physical entities to the Internet by virtue of the networks of embedded devices. Such network of constrained embedded devices is called the low power and lossy networks (LLNs). Numerous IoT applications involve the deployment of LLNs in large scale, which makes routing a mandatory requirement. Also, LLNs in IoT have specific routing

R. Sahay (✉) · G. Geethakumari · B. Mitra
Department of CSIS, Hyderabad Campus Birla Institute of Technology and Science,
Pilani, India
e-mail: p20160009@hyderabad.bits-pilani.ac.in

G. Geethakumari
e-mail: geetha@hyderabad.bits-pilani.ac.in

B. Mitra
e-mail: barsha.mitra@hyderabad.bits-pilani.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_8

requirements like varied traffic patterns, parametric constrained routing, energy constraints, autonomous configuration, scalability, and security. To satisfy all such routing requirements, IPv6 routing protocol for low power and lossy networks (RPL) was suggested by IETF [1]. RPL has certain security features in the form of preinstalled and authentication keys that a node must acquire before it joins the IoT-LLN. Though this protects RPL against external attacks, it remains vulnerable to several routing attacks instigated by internal malicious actors. The authors in [2, 3] have classified routing protocol attacks against RPL in IoT-LLN into three categories, namely *resource attacks, topological attacks, and traffic attacks*. On the one hand, resource attacks aim to deplete the network resources, and on the other hand, traffic attacks aim to disrupt the normal path of data packets or eavesdrop packet and LLN information. The goal of attack against network topology is to degrade the optimal performance of the network by false route advertisement or dropping packets.

The network topology is significant factor in ensuring efficient network operation. A consistent topology helps in the reduction of operational cost and maintenance. Topology also affects the utilization of the network resources, traffic patterns, reliability, and throughput. Attacks against network topology invariably snowball to traffic attacks. Hence, topological integrity and availability of LLNs will be the most important driving force of secured IoT applications. Existing literature classifies the topological attack as sub-optimization and isolation attacks [2, 3]. On the one hand, malicious node instigating a sub-optimization attack forces the nodes in its radio range to choose sub-optimal paths to connect to the root node. This degrades the network performance and leads to a delay in packet delivery. Examples of such attacks are sinkhole attack, worst parent attack, and DAO inconsistency attacks [4–6]. On the other hand, nodes instigating isolation attacks drop the received data packets from the nodes in their subtree. As a result, the network suffers data packet loss and reduced packet delivery ratio. Examples of such attacks are blackhole attack, greyhole attack, and selective forwarding attack [7]. Our first proposition is "*RPL is also susceptible to partitioning attack.*"

- **Partitioning attack**: It can be defined as an attack on the network topology by a malicious node that aims to decompose the network into disjoint clusters. As a consequence, nodes in one cluster cannot communicate with the nodes in another cluster. This may lead to the complete disassociation of some of the victim nodes from the sink node.

1.1 Major Contributions

In the present work, we study the phenomenon of network partitioning in RPL-based IoT-LLNs. We investigate the circumstances which lead to the partitioning of LLNs in the IoT environment. Our study also guided us to arrive at a novel mechanism to instigate a partitioning attack in RPL-based IoT-LLNs. The paper makes the following significant contributions:

- A novel taxonomy of topological attacks in RPL-based IoT-LLNs is presented in the work. Here, we propose a two-way classification of the topological attacks based on the way the attack is instigated and also the goal of the attack.
- We present the existing study on the partitioning against RPL, and based on the study, we propose a novel partitioning attack against RPL.
- We propose an algorithm that detects partitioning in RPL-based IoT-LLNs and determines the malicious devices responsible for partitioning.

The remaining paper is structured in six sections. Section 2 presents the exiting work on partitioning in RPL-based IoT-LLNs. Section 3 explains the problem statement. Section 4 presents a novel taxonomy of the topological attacks in RPL-based IoT-LLNs. Section 5 explains partitioning in IoT-LLN and presents a novel partitioning attack against RPL. Section 6 presents the proposed mechanism to detect the IoT network partitioning attack. Section 7 summarizes the findings of the present work.

2 Related Work

RPL organizes sensor nodes as directed acyclic graphs (DAG). DAGs may consist of multiple destination-oriented directed acyclic graphs (DODAGs) with independent sink (root) nodes federated by a backbone network [1]. This facilitates autonomous partitioning of urban networks for efficiency and creating security gaps through the use of appropriate routing metrics [8]. However, within a DODAG, all the nodes should be connected to the root node either directly or via a parent node as the root node is the sole point of connectivity between the sensor nodes and the IoT application platform. In a circumstance where a live node has no link to the root node, the live node is said to be partitioned. A partitioned node continues to forward data packets to its parent node if the communication link to the parent is alive [9]. Karim Fathallah et al. in their work [10] proposed a partition aware routing process for agricultural application where subtrees belonging to particular farmland within a DODAG are termed parcels. The authors defined a special link termed as bridge to establish connection to the root node. In such applications, failure of the bridge link would completely disconnect a farmland pocket from the sink node.

Partitioning of nodes in RPL supported IoT-LLNs has found mentioned in a few research papers. In [11], the authors have proposed a novel DIO suppression attack in which the attacker replays an old DIO. As a result, victim nodes suppress their DIO as they believe there is no update in the configuration to share. Though Pericle et al. stated that DIO suppression attack may lead to network partitioning, they did not further explore it. The authors in [12] suggested that the sensor nodes due to memory constraints do not maintain the list of unreachable nodes. As a result, a discarded node may be chosen as a preferred parent leading to partitioning. Topological attacks like decreased rank attack [5] or sinkhole attack [4] surge the traffic toward a target node. This may deplete node energy leading to occasional partitioning in the absence of an alternate path. Similarly, the resource attack may also lead to partitioning in IoT-LLN.

In the following section, we present our problem statement.

3 Problem Statement

Recent literature describes an in-depth study of the impact of topological sub-optimization and isolation attacks against RPL in IoT-LLN. However, the study of partitioning attacks against RPL is missing in the existing literature. Also, as mentioned in Sect. 1, partitioning attacks against RPL are missing from the existing taxonomies. The specific objectives of our work are as follows:

1. To present a novel taxonomy of topological attacks against RPL which includes partitioning attacks.
2. To explore ways in which an attacker may instigate partitioning attacks against RPL in IoT-LLNs.
3. To present an algorithm to detect partitioning attacks in RPL supported IoT-LLNs.

4 Proposed Taxonomy of Topological Attack Against RPL

We present a novel depiction of the taxonomy of topological attacks against RPL in Fig. 1. In the currently available literature, classification of attacks against RPL is based on the goal of the attack. However, we present our taxonomy based on the way the attack is instigated and map it to various goals of the topological attack. This helps in better understanding of the attack and guides in devising appropriate detection and mitigation mechanisms.

As observed from Fig. 1, while packet dropping leads to isolation attacks, false advertisement and replay of old control messages lead to sub-optimization attacks. Partitioning attack is a result of a malicious node skipping a routing operation. Partitioning attack may be caused due to a node which does not output any DAO message. A detailed explanation of this is presented in Sect. 5. It can also be observed from Fig. 1 that DIO suppression attack may also lead to network partition. In the case of DIO suppression attack, as old DIO messages are replayed, this may occasionally lead to partitioning, as new available paths may remain unknown to the victim nodes.

5 Partitioning Attacks in RPL

As mentioned before, RPL configures sensor nodes as single or multiple DODAGs. A single DODAG has the following categories of nodes:

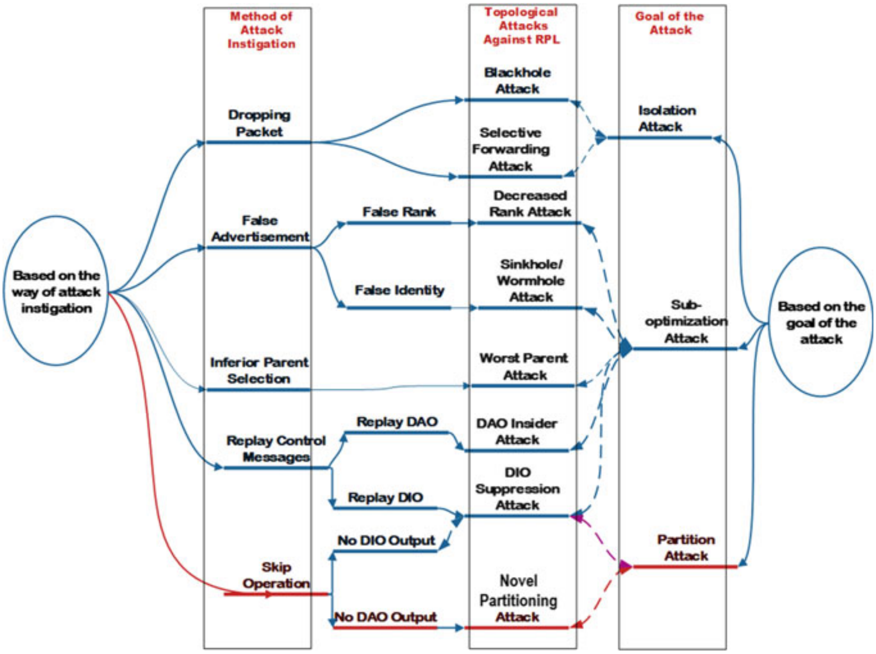
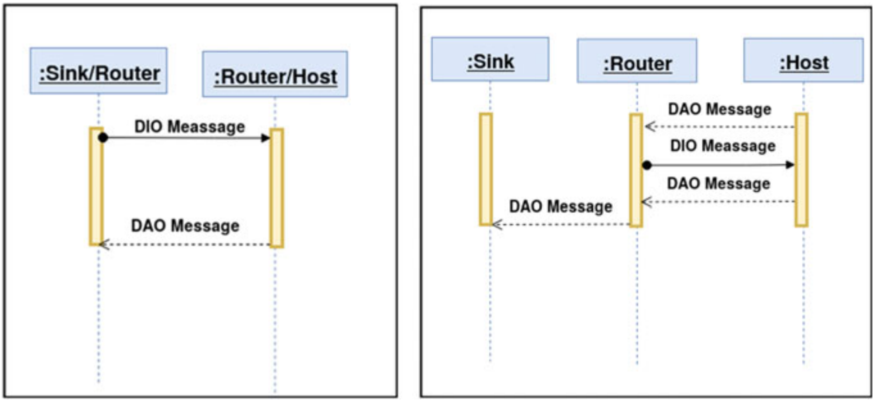


Fig. 1 Proposed taxonomy of topological attacks against RPL in IoT-LLNs

1. **Sink**: It connects devices in the DODAG to the rest of the IoT network. It is usually a gateway or a border router responsible for the topological organization and maintenance of IoT-LLNs.
2. **Router**: Sensors or actuators with routing capabilities.
3. **Host**: Sensor or actuator nodes.

The goal of the sensor and actuator devices in the IoT-LLN is to transmit data or receive command from an IoT application. The sensor devices fail to achieve this goal, if they are not connected to the sink either directly or via the intermediate nodes. Nodes organize them as DODAGs by exchanging three types of control messages. The sequence diagram in Fig. 2 depicts the order of the exchange of the control messages among the nodes in order to organize themselves as DODAGs. As observed from Fig. 2a, the sink node broadcasts DODAG Information Object (DIO) messages at time periods controlled by the trickle timer [13], and the nodes respond with a DODAG advertisement object (DAO) message. DIO messages carry DODAG configuration parameters that are used by the nodes to construct DODAGs and join them. The DAO messages are forwarded up to the sink node that uses it to update its routing table, as represented in Fig. 2b. New nodes multicast DODAG Information Solicitation (DIS) messages to the neighboring nodes to obtain the network configuration parameters. In response, the neighboring nodes send the DIO messages, including their specific rank and the network configuration parameters. A node chooses its parent by responding with a DAO message. The parent node is supposed to forward



(a) DODAG Maintenance (b) Node Joining DODAG

Fig. 2 Flow of control messages for DODAG maintenance and formation

this DAO message to the sink node to allow the sink to view this new node. Hence, we can infer that a sink node can reach or view a specific node only if it receives a DAO message from that node. Figure 2b depicts the message exchange scenario involved in this process.

Partitioning in IoT-LLNs can occur due to correlated failure of multiple nodes or a single node described as follows:

- Border Router Failure: In case of border router failure, the entire DODAG is partitioned from the IoT application. This may happen in rare circumstances as border router is computationally more powerful devices but are not unseen events. The authors in [14] have explored the family of malwares responsible for a plethora of DDoS attacks which are capable of bringing down powerful servers.
- Funneling: Owing to the structure of DODAGs, nodes near to the sink node handle more data and control traffic. As a result, such nodes often suffer from energy depletion resulting in partitioning of their sub-DODAGs. The authors in [15] have proposed an enhanced RPL version incorporating load balancing and address prefix translation techniques to overcome funneling.
- Partitioning Attacks: A malicious node may intentionally execute mechanisms to partition one or multiple nodes from the sink node. In order to meet the requirements like high scalability and fast convergence, RPL allows the formation of floating and grounded DODAG as shown in Fig. 3. A node sets or resets a flag variable “G” in its DIO message to indicate if it is a part of the grounded or floating DODAG. As nodes in IoT-LLNs have limited radio range, it is impossible for all the nodes to be directly connected to the sink node. Hence, nodes are dependent on neighboring nodes closer to the sink for acquiring configuration information. Therefore, nodes can easily become victims of false information. For example, a malicious node may falsely advertise being part of a grounded DODAG.

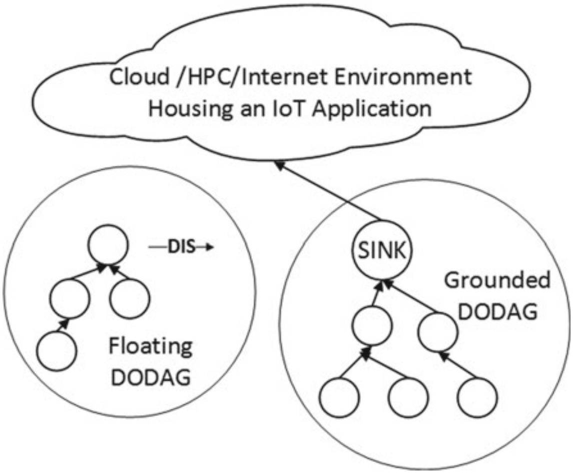


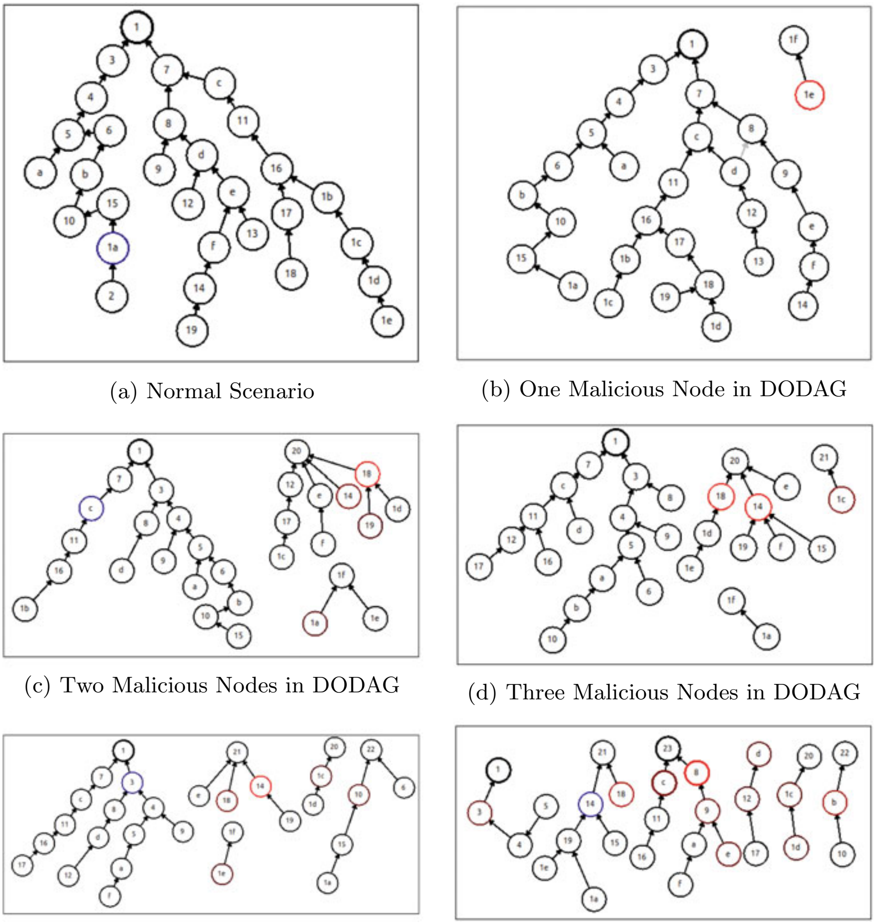
Fig. 3 Grounded and floating DODAGs

5.1 Novel Partitioning Attack

Based on the above understanding, we present a novel partitioning attack scenario against RPL. As observed in Fig. 2, the sink node gets the view of any node "*n*" in the DODGA only after it receives a DAO message from the node "*n*". In the proposed network partitioning attacks, a malicious node after receiving the DIO messages from neighboring nodes never responds with a DAO message. Thus, its registration process is incomplete and the sink node has no view of this malicious node. In essence, the malicious node blocks its DAO output. However, as the malicious node has already received the configuration information, it starts broadcasting the same and falsely advertises itself as being part of a grounded DODAG. Consequently, neighboring nodes join the malicious node without being aware that they are not connected to the sink node. Thus, the malicious node partitions its sub-DODAG from the sink. The malicious node could successfully execute the partitioning attack because RPL does not define a fixed sequence of emission of control messages [6]. Though an undefined sequence ensures increased scalability and reduces the volume of the control message, it makes RPL prone to the proposed partitioning attack.

5.2 Penetration Testing of Novel Partitioning Attack

We performed penetration testing of the novel partitioning attack by simulating the attack scenario in the Cooja simulator. To analyze the attack, we performed a 30-node simulation with the number of rouge nodes instigating the partitioning attack varying from 1–5. Sky motes with 50 m of radio range were used in the simulation.



(a) Normal Scenario

(b) One Malicious Node in DODAG

(c) Two Malicious Nodes in DODAG

(d) Three Malicious Nodes in DODAG

(e) Four Malicious Nodes in DODAG

(f) Five Malicious Nodes in DODAG

Fig. 4 DODAG under partitioning attack

The routing metric used in the experiments is a function of the link object expected transmission count. Figure 4a depicts the topological formation of the DODAG under normal scenario, and Fig. 4b–f depicts the topological state of the DODAG in the presence of 1–5 malicious nodes. In all cases, Node 1 is the sink node. From Fig. 4b, it can be observed that malicious node 1f segregates Node 1e from the DODAG. Similarly, from Fig. 4c, it can be observed that there are three DODAGs formed with one rooted at the sink node (Node 1) and the other two rooted at the two malicious nodes (Nodes 20 and 1f). We can observe from Fig. 4c that the two malicious nodes instigating the partitioning attack could segregate the 11 fair nodes from the sink node. With an increase in the count of malicious nodes present in the DODAG, the number of nodes connected to the sink drops drastically, as can be observed from

Fig. 4f. Only three nodes are connected with the sink, and the rest of the victim nodes are in the subtrees of one of the malicious nodes. The victim nodes remain unaware that they are unavailable to the sink node.

6 Proposed Detection Mechanism of Partitioning Attack

From the penetration testing performed in the previous section, we observed that the partitioning attack results in unavailability of nodes in the DODAG due to its disconnectivity from the sink. The unique feature of the proposed partitioning attack is that, nodes segregated from the sink are unaware of the disconnectivity. Hence, they do not probe the neighboring nodes continuously like in the case of resources attacks. The nodes which are available handle lesser traffic and hence consume less energy. The unavailable nodes become part of the malicious node's subtree. In the case of packet-dropping attacks, malicious nodes drop received data packets instead of forwarding them to the sink. Consequently, data packets received by the sink node drops in number. However, the number of available nodes and the approximate power consumption remain the same. This leads to our second proposition that "In order to identify a partitioning attack scenario, the number of available nodes to the sink in the DODAG and the average power consumption by the available nodes should be monitored." If the number of available nodes drop below the acceptable limit and the power consumed by the available nodes is reduced, we initiate a partitioning attack detection mechanism to check for the presence of disconnected clusters of nodes and identify the malicious nodes instigating the partitioning attack. Algorithm 1 depicts the attacks detection process. It takes as input two parameters, (a) the desirable percentage of available nodes, x and (b) the acceptable drop in the average power consumption of the available nodes. The algorithm also uses a flag variable, namely $detection\_phase$ which is initially set to $false$. If the values of these two parameters drop below the desirable limit, the flag $detection\_phase$ is set to $true$ and the algorithm calls the **Detect and Identify Malicious Node** procedure (Algorithm 2).

Algorithm 1 :Initiate Partitioning Attack Detection

```
INPUT :$x$ - Desirable percentage of available nodes
INPUT :$y$ - Acceptable drop in avg. power consumed by available nodes
$detection\_phaseFlag = False$
if $Available\_No\_of\_Nodes \leq x\%$ then
    if $Avg\_Power\_per\_node \leq y\%$ then
        $detection\_phase = True$
    end if
end if
while $detection\_phase == True$ do
    call Detect and Identify Malicious Nodes
end while
```

Algorithm 2 :Detect and Identify Malicious Nodes

```
Output: Malicious Node IDs
Require :$Mal\_ID[]$ - Array Variable to store malicious node IDs
Require :$Part\_Attack$ - Set to True if Partitioning Attack is detected
1: call Build Info Table
2: for $i = 0$ to $INFO\_TABLE.size$ do
3:     if $INFO\_TABLE[i][DAO] == 0$ & $INFO\_TABLE[i][DIO] > 0$ then
4:         $Mal\_ID[j++] = INFO\_TABLE[i][node\_id]$
5:     end if
6: end for
7: Build subtrees rooted at the malicious nodes
8: if all unavailable nodes ∈ subtrees rooted at a malicious nodes then
9:     $Part\_Attack = True$
10: end if
11: if $Part\_Attack = True$ then
12:     Alert_Message(“DODAG is Under Partitioning Attack")
13: end if
14: return $Mal\_ID$
```

Algorithm 3 :Build Info Table

```
INPUT :$PCAP$ - Files containing captured network traffic
Require :$INFO\_TABLE$ - Table to store number of emitted control messages by each node
for $each\ enrty\ in\ PCAP$ do
    Extract Node_ID
    if $NewNodeID$ then
        $INFO\_TABLE[NewRow][0] = Node\_ID$          ▷ Add new row in $INFO\_TABLE$
    end if
    for $i = 0$ to $INFO\_TABLE.Size$ do
        if $INFO\_TABLE[i][0] = Node\_ID$ then
            Update Row $i$ of $INFO\_TABLE$
        end if
    end for
end for
return $INFO\_TABLE$
```

Algorithm 2 detects the presence of the partitioning attack, identifies the attacker nodes, and saves their IDs in an array variable Mal_ID. The algorithm also uses a flag variable $Part\_Attack$ which is initially set to $false$. To accomplish detection, Algorithm 2 calls Algorithm 3 which constructs a node information table named, $INFO\_TABLE$, by extracting the packet capture information available in radio logs. Contiki OS provides a tool named “Radio Messages” to capture the radio logs which are pcap (Packet Capture) files. The table keeps a track of the sum total of the DIS, DIO, and DAO messages emitted by each node and their parent node ID. Since a malicious node skips the route registration step but allows nodes to join its subtree, its DAO count should be equal to zero and DIO count should be at least greater than 1 as depicted through lines 2–5. Once the malicious nodes are identified, we construct subtrees rooted at the malicious nodes. If the unavailable nodes are in the subtrees of the identified malicious nodes, the $Part\_Attack$ flag is set to $true$ which initiates the generation of an alert message to state that the DODAG is under Partitioning Attack.

The experimental results of the partitioning attack detection are presented in Table 1. The same simulation setup explained in Sect. 5.2 is used to validate the detection algorithm. Initially, the DODAG is simulated normally without any malicious

Table 1 Observation made during partitioning attack detection

Simulation timestamp	200	400	500	600	700	800
No. of node available in DODAG	30	29	21	18	17	9
Avg. power consumption of the DODAG (mW)	2.72	2.52	2.08	2.02	2.25	2.14
Value of detection phase	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
Nodes with DIO > 0 and DAO = 0	NIL	1f	1f, 20	1f, 20, 21	1f, 20, 21, 22	d, 20, 21, 22, 23
IDs of available nodes	{1, 3, 7, 4, 8, c, 5, 9, d, 11, a, 6, 12, e, 16, b, f, 13, 17, 1b, 10, 14, 18, 1c, 15, 19, 1d, 1a, 1e, 2}	{1, 3, 7, 4, 8, c, 5, 9, d, 11, a, 6, 12, e, 16, b, f, 13, 17, 1b, 10, 14, 18, 1c, 15, 19, 1d, 1a, 2}	{1, 3, 7, 4, 8, c, 5, 9, d, 11, a, 6, 16, b, 13, 1b, 10, 15, 1a, 2}	{1, 3, 7, 4, 8, c, 5, 9, d, 11, a, 6, 12, 16, b, 13, 1b, 10, 15, 2}	{1, 3, 7, 4, 8, c, 5, 9, d, 11, a, 6, 12, 16, 17, f}	{1, 3, 4, 5}
DODAG (Grounded + Floating)	{1, 3, 7, 4, 8, c, 5, 9, d, 11, a, 6, 12, e, 16, b, f, 13, 17, 1b, 10, 14, 18, 1c, 15, 19, 1d, 1a, 1e, 2}	{1, 3, 7, 4, 8, c, 5, 9, d, 11, a, 6, 12, e, 16, b, f, 13, 17, 1b, 10, 14, 18, 1c, 15, 19, 1d, 1a, 2} {1f, 1e}	{1, 3, 7, 4, 8, c, 5, 9, d, 11, a, 6, 16, b, 13, 1b, 10, 15, 1a, 2} {20, 12, e, 14, 18, 17, f, 19, 1d, 1c} {1f, 1e}	{1, 3, 7, 4, 8, c, 5, 9, d, 11, a, 6, 12, 16, b, 13, 1b, 10, 15, 2} {20, e, 14, 18, 1d, 19, f, 15, e, 1e} {1f, 1a} {21, 1c}	{1, 3, 7, 4, 8, c, 5, 9, d, 11, a, 6, 12, 16, 17, f} {20, 1c, 1d} {1f, 1e} {21, e, 18, 14} {22, 10, 6, 15, 1a}	{1, 3, 4, 5} {20, 1c} {21, 18, 14, 19, 15, 1e, 1a} {22, b, 10} {23, c, 8, 11, 9, 16, a, e, f} {d, 12, 17}
DODAG under attack	No	Yes	Yes	Yes	Yes	Yes

node. Later, five malicious nodes are inserted in the DODAG one by one at gaps of 100 timestamps. The detection mechanism sets the $detection\_phase$ as true, and the malicious nodes are identified. The available and the unavailable nodes are represented in the form of a tree traversal, i.e., in the set $\{d, 12, 17\}$, d is the root, 12 is the left child of d and 17 is either the left child of 12 or right child of d. The results depicted in Table 1 show that, with a decrease in the number of available nodes, the average power consumed by the DODAG also decreases.

7 Conclusion and Scope of Future Work

In this research, we investigated a novel partitioning attacks scenario against RPL in IoT-LLNs. In IoT-LLNs, sensor nodes are connected to the IoT applications via the sink node. Hence, each sensor node in the DODAG should have an active connection to the sink node. For a better understanding of this phenomenon, we proposed a novel taxonomy of topological attacks in RPL supported IoT-LLNs based on the method of attack instigation. This taxonomy aids in understanding various topological attacks against RPL. We further focused on the partitioning attacks and presented a novel partitioning attack scenario against RPL. We also performed the penetration testing of the attack and based on the test we presented a mechanism to detect the attack.

In future, we plan to propose a mitigation mechanism to make RPL resilient of partitioning attacks. A further research direction could be to analyze the partitioning attack in collaborative attack scenarios.

References

1. Winter T, Thubert P, Brandt A, Hui J, Kelsey R, Pister K, Struik R, Vasseur JP, Alexander R (2016) RPL—IPv6 routing protocol for low-power and lossy networks. In: RFC 6550
2. Wallgren L, Raza S, Voigt T (2013) Routing attacks and countermeasures in the RPL-based Internet of things. Int J Distrib Sens Netw 9(8):794326, 1–11
3. Mayzaud A, Badonnel R, Chrisment I (2016) A taxonomy of attacks in RPL-based Internet of things. Int J Netw Secur 18(3):459–473
4. Cervantes C, Poplade D, Nogueira M, Santos A (2015) Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of things. In: Proceedings of the IFIP/IEEE international symposium on integrated network management (IM), pp 606–611
5. Sahay R, Geethakumari G, Modugu K (2018) Attack graph-based vulnerability assessment of rank property in RPL 6LOWPAN in IoT. In: Proceedings of the 2018 IEEE world forum on internet of things, 05–08 Feb 2018, pp 308–313
6. Ghaleb B, Al-Dubai A, Ekonomou E, Qasem M, Romdhani I, Mackenzie L (2018) Addressing the DAO insider attack in RPL's Internet of things networks. IEEE Commun Lett 23(1):68–71
7. Sahay R, Geethakumari G, Mitra B, Goyal N (2019) Investigating packet dropping attacks in RPL-DODAG in IoT. In: Proceedings of the 5th IEEE international conference for convergence in technology, 2019 (IEEE I2CT-2019), India, 29–31 Mar 2019, pp 1–5
8. Vasseur JP, Kim M, Pister K, Dejean N, Barthel D (2012) Routing metrics used for path calculation in low-power and lossy networks. In: RFC 6551
9. Paszkowska A, Iwanicki K (2018) The IPv6 routing protocol for low-power and lossy networks (RPL) under network partitions. In: EWSN, 12 Feb 2018, pp 90–101
10. Fathallah K, Abid MA, Hadj-Alouane NB (2018) PA-RPL: a partition aware IoT routing protocol for precision agriculture. In: Proceedings of the 14th IEEE international wireless communications and mobile computing conference (IWCMC), 25 June 2018, pp 672–677
11. Perazzo P, Vallati C, Anastasi G, Dini G (2017) DIO suppression attack against routing in the Internet of things. IEEE Commun Lett 21(11):524–527
12. Clausen T, Herberg U, Philipp M (2011) A critical evaluation of the IPv6 routing protocol for low power and lossy networks (RPL). In: Proceedings of the 7th IEEE international conference on wireless and mobile computing, networking and communications (WiMob), 10 Oct 2011, pp 365–372

13. Levis P, Patel N, Culler D, Shenker S (2004) Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In: Proceedings of the 1st USENIX/ACM symposium on networked systems design and implementation, vol 25, 29 Mar 2004, pp 1–14
14. Kambourakis G, Kolias C, Stavrou A (2017) The mirai botnet and the IoT zombie armies. In: Proceedings of the IEEE military communications conference (MILCOM), 23 Oct 2017, pp 237–272
15. Foubert B, Montavont J (2019) Sharing is caring: a cooperation scheme for RPL network resilience and efficiency. In: Proceedings of the IEEE symposium on computers and communications, 29 June 2019, pp 1–6

Medical Sign Shaped Compact Wideband 2 × 2 MIMO Antenna for 5G Application

Prachi and Vishal Gupta

Abstract The paper reports a new wideband compact 2 × 2 MIMO antenna fed with microstrip line for fifth generation in millimeter wave frequency. The need for the upcoming mobile wireless technology should be consisting of high gain, large bandwidth so as to accommodate large number of applications, compact size along with maintaining highly efficiency. There is a tradeoff among antenna size and mutual coupling. The antenna design is implemented on Rogers 5880 with the total substrate dimension as $12.4 \times 6.25\ \text{mm}^2$. The proposed antenna consists of two radiating slots placed at 90° to each other forming a medical shaped plus sign etched on the radiating patch which is providing a gain of 7.4 dB with a remarkable bandwidth of 1.8 GHz resonating at 38 GHz in millimeter wave spectrum. The MIMO attains a S_{11} of less than -10 dB as well as the mutual coupling among the two radiating elements is found to be less than -20 dB at the operating frequency. Additionally, the MIMO antenna provides the promising results for the diversity gain, mean effective gain, and envelope correlation coefficient with accommodating high radiation efficiency of 93% at the operating frequency.

Keywords ECC · MEG · Diversity gain · Multi–input–Multi–output · Fifth generation

1 Introduction

The advancement of wireless communication from 1 to 4G have great impact on lifestyle of mankind in the past decades. In order to encounter the future demands of high data rate as well as high bandwidth, sufficient wireless technology needs to be evolved. Present wireless communication technology is still unable to accommodate few challenges such as high energy consumption, network area coverage, and data rates. The mobile communication system is facing a rapid growth in Internet speed along with mobility acquired by new functions. To overcome these issues and to address the high speed as well as large bandwidth requirement for users, the future

Prachi (✉) · V. Gupta
The ICFAI University, Dehradun, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_9

communication system expected to boom the speed of data rates which is supposed to be deployed in 2020. By this, the fifth generation will enable high speed network, wide area coverage, low latency, versatility, reliability, and reduced power consumption [1]. To surpass the bandwidth requirement, the ITU has standardized mm-wave frequency for 5G spectrums ranging from 24.25–27.5, 37–40 and 66–76 GHz [2]. By keeping in mind the frequency ranges, various researchers presented their innovative ideas for antennas resonating at 28, 38 and 60. Multipath fading degrades the signals transmitted by antennas which further reduces link reliability and capacity.

In this work, antenna is proposed with two slots in the center of the patch that resonates at 38 GHz discussed in Sect. 3. The slots are placed in 90° to each other forming a medical sign shaped slot resulting in improving the performances in the aspect of return loss, gain, impedance matching, etc. Further, a 2 × 2 MIMO antenna is designed using the same element in a way to obtain reduced mutual coupling between the elements. The two symmetrical elements consisting of two ports are placed on the same side. In Sect. 4, simulated results of both single as well as MIMO antenna are discussed and in Sect. 5, a comparative analysis has been tabulated with previous literatures. Finally, in Sect. 6, a conclusion of the proposed work is given.

2 Literature Survey

In order to overcome the degradation of signal, Multiple Input and Multiple Output (MIMO) techniques provide the favorable results such as increased data rate and reliability which can be achieved by deploying several antennas at both transmitter and receiver end. The designing of MIMO antennas are challenging for the researchers as they have to consider various MIMO parameters such as Envelope Correlation Coefficient (ECC), Mutual coupling (MC), Channel Capacity Loss (CCL), Mean Effective Gain (MEG), and Diversity Gain (DG) within the acceptable values [3]. In Kumar et al. [4], a triple-band operation is obtained by a MIMO antenna which consists of one T-shaped and two L-shaped stubs loaded on rectangular monopoles. In Wong et al. [5] wideband circularly polarized patch antenna for MIMO applications is presented. Various techniques are employed by the researchers to improve isolation parameters in MIMO antennas like space decoupling, decoupling structures reported in [6–8]. In Ruswanditya et al. [9] MIMO 8 × 8 antenna array is introduced with two H-slotted rectangular patches array at 15 GHz with a 5G radio access system to increase gain and feed for transmission speed. Optimization of new MIMO and 5G system patch antenna was presented in Faleh and Tahar [10] in which a new type of MIMO presented which provides a multiple frequency band operations based on rectangular slots. The work in Sundhari and Veeramani [11] presents 1 × 4 MIMO Antenna based on spatial diversity with slits and slots for multiband operations in 5G. A triple-band antenna with grounded stub is presented in Chaudhari and Gupta [12] for 5G mobile communication.

3 Design Procedure for Single and MIMO Antenna

3.1 Single Element

Single element is designed for 5G with a suitable substrate Rogers 5880 with a height of 0.254 mm. The single element resonates at 38 GHz frequency that is for Ka-band in electromagnetic spectrum. The entire dimension of the 50 Ω microstrip line fed single antenna is 6 × 6.25 × 0.254 mm^3. The ground of the antenna is having 0.017 mm thickness made up of copper. Two slots in the shape of plus sign are etched in the center of the patch that are optimized to operate the design in the desired frequency. Figure 1 depicts the design single element and parameters of the design are shown in Table 1.

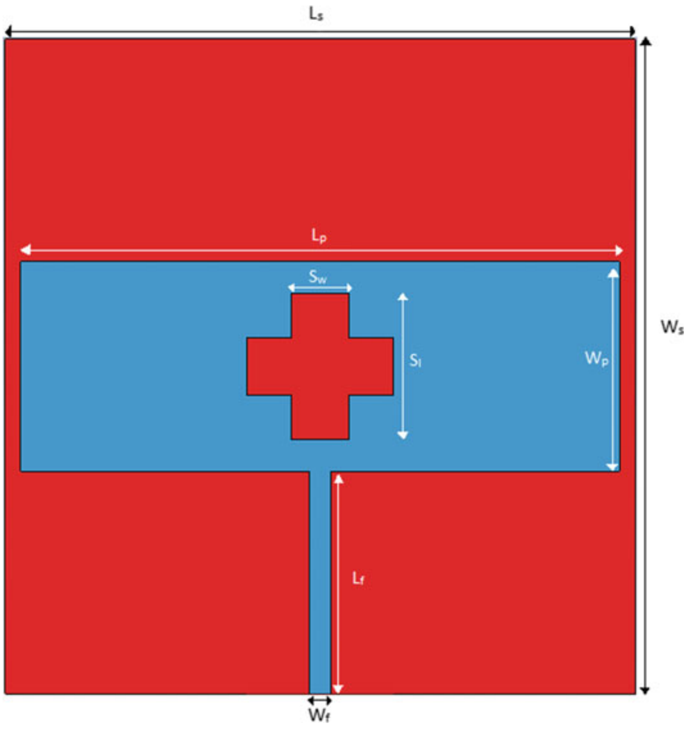


Fig. 1 Single element for MIMO

Table 1 Single element design parameters

Parameters	Description	Values (mm)
SUBSTRATE		
L_s	Length	6
W_s	Width	6.25
t_s	Thickness	0.254
PATCH		
L_p	Length	*5.7*
W_p	Width	*2*
t_p	Thickness	*0.017*
SLOT		
S_L	Length	*1.4*
S_W	Width	*0.55*
FEED		
Lf	Length	*2.13*
Wf	Width	*0.2*

3.2 MIMO Antenna

A 2 × 2 MIMO antenna using single element is made with inter element spacing of 0.81λ having overall size of antenna 12.4 × 6.25 × 0.254 mm^3. The inter element spacing is adjusted in such a way to obtain high isolation by using in built optimizer in the software. The 2D and 3D view of proposed MIMO antenna is depicted in Fig. 2a and b. CST Microwave studio ver. 18.0 is utilized to model the proposed antenna and optimize its dimension (Table 2).

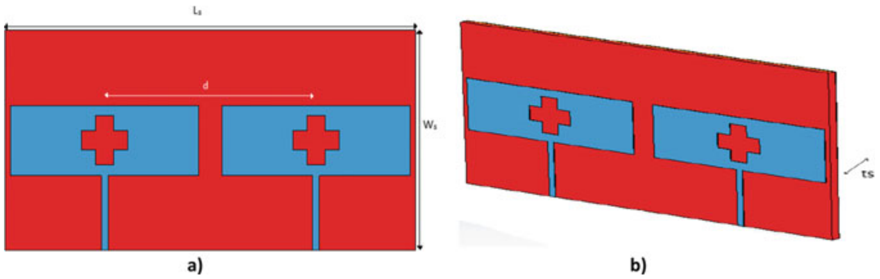


Fig. 2 **a** 2D Design of 2 × 2 MIMO antenna and **b** 3D view of MIMO antenna

Table 2 MIMO antenna parameters

Parameters	Description	Values (mm)
L_s	Length of substrate	12.4
W_s	Width	6.25

4 Simulated Results

4.1 S11 and S21 Parameters

S11 and S21 are the important parameters in determining the relationship between the powers at the ports. S11 represents the return loss whereas S21 represents the transmission coefficient.

The proposed MIMO design is having the perfect symmetry ($|S_{11}| = |S_{22}|$ and $|S_{21}| = |S_{12}|$), thus presenting only two parameters $|S_{11}|$ and $|S_{21}|$.

The single element has S_{11}(impedance matching) of −31 dB achieving 1.7 GHz bandwidth whereas 2 × 2 MIMO has S_{11} of −23 dB with 1.8 GHz of bandwidth operating at 38 GHz. S_{21} (Mutual coupling) of MIMO design is −21.4 dB. Figure 3a depicts the graphical representation of S_{11} of both single and MIMO antenna and Fig. 3b shows the S_{21} of MIMO antenna that shows the good isolation between the elements.

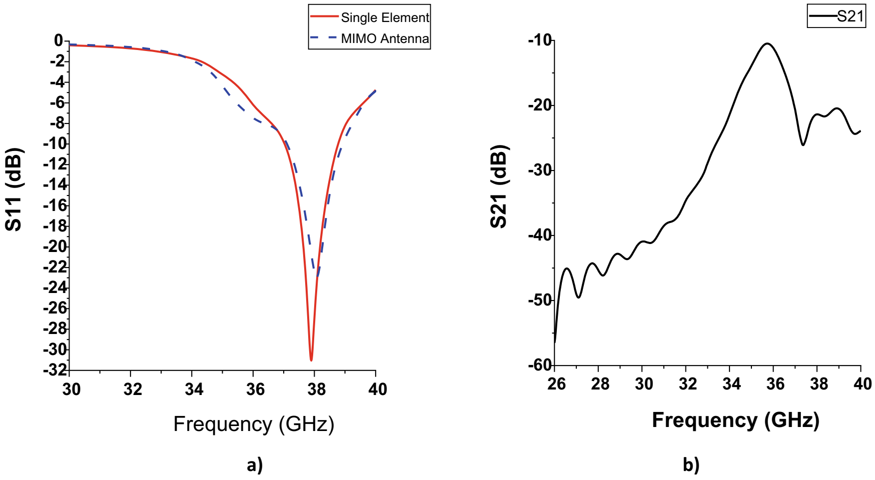


Fig. 3 **a** S11 of both single and MIMO antenna and **b** S21 of MIMO antenna

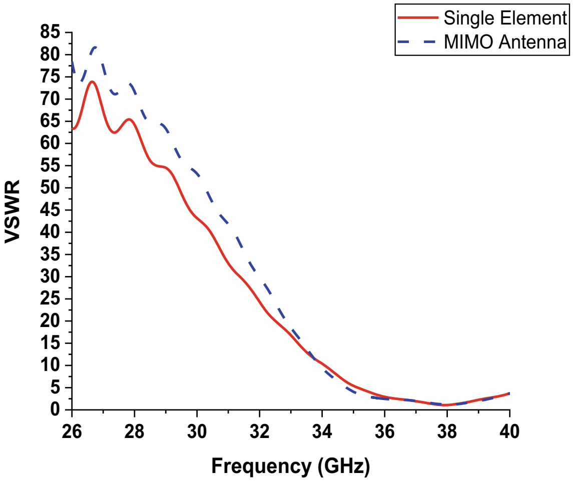


Fig. 4 VSWR of both single and MIMO antenna

4.2 VSWR

VSWR is abbreviated as Voltage Standing Wave Ratio and it is a function of return loss. The reported single element and 2 × 2 MIMO antenna gives the VSWR as 1, respectively, which implies the perfect matching. The graphical representation of single element and MIMO Antenna are shown in Fig. 4.

4.3 Gain

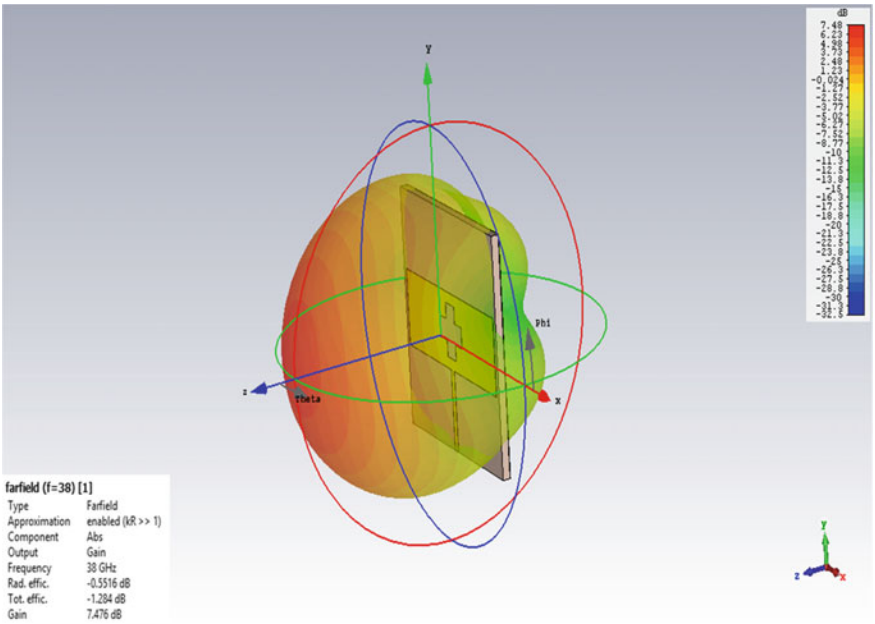
The 3D radiation pattern of single antenna is obtained in order to analyze the radiation characteristics which shows gain of about 7.47 dB depicted in Fig. 5.

4.4 Analysis of MIMO Performance Parameters

To ensure the performance of MIMO of reported 2 × 2 design, various performance metrics such as ECC, DG, and MEG are analyzed to characterize the MIMO antenna system which are not essential for single element.

4.4.1 Envelope Correlation Coefficient and Diversity Gain

ECC denoted by (ρ) defines the correlation between elements and Diversity Gain are one of the essential diversity parameters of MIMO system. The method of calculating ECC and DG is given in Marcus [2] that shows the acceptable value of $\rho \leq 0.5$ and

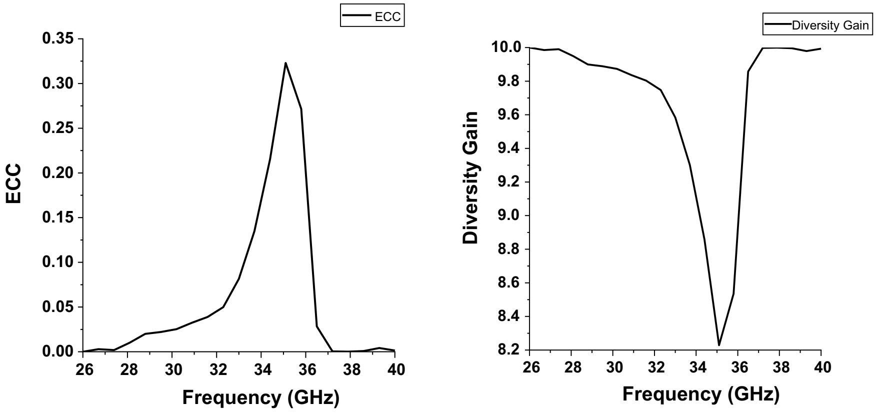


farfield (f=38) [1]

Type	Farfield
Approximation	enabled (kR >> 1)
Component	Abs
Output	Gain
Frequency	38 GHz
Rad. effic.	-0.5516 dB
Tot. effic.	-1.284 dB
Gain	7.476 dB

Fig. 5 Gain of single element antenna

DG ≥ 9.95 dB and the proposed work has a value of ECC is 0.0002 and DG is 9.99, respectively, in the 2 × 2 MIMO Antenna. The graphical representations of evaluated MIMO parameters are depicted in Fig. 6a and b that shows the MIMO antenna performance is high.



a)

b)

Fig. 6 **a** ECC of MIMO antenna and **b** DG of MIMO antenna

Table 3 Comparison with previous literatures

Cited work	Substrate ε_r	Antenna size (mm$^{3)}$	Resonating frequency (GHz)	Isolation (dB)	S11 (dB)	Efficiency (%)	Gain (dB)
[11]	FR-4	26.6 × 3.2505 × 1.6	28, 37, 41 & 74	–	−22, −37, −10 & −19	–	4, 5.2, 7.2 & 12.5
[13]	Rogers 5880	55 × 110 × 0.508	27.946 & 37.83	−30	−27.84 & −18.35	91.24 & 89.63	7.18 & 9.24
[14]	FR-4	15.2 × 7.3 × 1.524	21.06–29.7	−10.52	−22	72.5	6.38
Proposed work	Rogers 5880	12.4 × 6.25 × 0.254	38	−21.38	−23	93	7.4

4.4.2 Mean Effective Gain (MEG)

For good diversity performance in MIMO antennas, the acceptable value of MEG is ≤ 3 dB that shows the mean received power. In this reported work the value of MEG for the 2 × 2 MIMO is found to be −3.01 dB.

5 Comparative Analysis

The characteristics of the reported MIMO antenna is compared with previous works in terms of various performance parameters such as size, efficiency, gain, and return loss is represented in Table 3. It has been observed that proposed work shows better isolation and efficiency of about 93% along with maintaining gain of 7.4 dB at the operating frequency, i.e., 38 GHz for 5G. The proposed MIMO antenna satisfies performance metrics of MIMO but also occupies very compact physical area as compared to other literatures.

6 Conclusion

The represented 2 × 2 MIMO antenna in this paper is planned for fifth generation in order to function at 38 GHz which are standardized for 5G applications. However, the MIMO antenna does not only consist of sufficient center-to-center spacing of 0.81λ but also a compact structure having a dimension of 12.4 × 6.25 mm^2 with the height of 0.254 mm. The proposed design successfully achieves a high gain of 7.4 dB along with impressive wide band of 1.8 GHz at the said frequency. The proposed design is simulated using an electromagnetic simulation tool known as CST MW Studio. The antenna procures the S_{11} than -10 dB with the mutual coupling less than -20 dB between the two radiating elements along with the high radiation efficiency of 93%. Further in this work, MIMO parameters such as ECC and mean effective gain are evaluated to analyze the proposed antenna. The results procured by the 2 × 2 MIMO antenna are within the acceptable range and hence it offers its candidature for fifth generation. In future, the proposed MIMO antenna design can be amended into a Massive MIMO which can be tested and fabricated that shows immense potentials to cater the demand of high data rate for fifth-generation communication systems.

References

1. Haraz OM et al (2015) Design of a 28/38 GHz dual-band printed slot antenna for the future 5G mobile communication networks. In: 2015 IEEE international symposium on antennas and propagation & USNC/URSI national radio science meeting, Canada
2. Marcus MJ (2015) 5G and IMT for 2020 and beyond [spectrum policy and regulatory issues]. IEEE Wireless Commun 22(4):2–3
3. Sharawi MS (2013) Printed multi-band MIMO antenna systems and their performance metrics. IEEE Antennas Propag Mag 55(5):218–232
4. Kumar A, Sharma MM, Jhanwar D (2017) A compact triple-band planar MIMO diversity antenna for WiMAX/WLAN applications. In: 2017 international conference on computer, communications and electronics (Comptelix), pp 1–5
5. Wong KL, Chou CM, Yang YJ, Wang KY (2019) Multipolarized wideband circular patch antenna for fifth-generation multi-input -multi-output access-point application. IEEE Antennas Wireless Propag Lett 18:2184–2188
6. Rajkumar S, Sivaraman NV, Murali S et al (2017) Heptaband swastika arm antenna for MIMO applications. IET Microw. Antennas Propag. 11(9):1255–1261
7. Ramachandran A, Mathew S, Rajan V et al (2017) A compact triband quad-element MIMO antenna using SRR ring for high isolation. Antennas Wirel Propag Lett 16:1409–1412
8. Wang SM, Whang LT, Lee CJ et al (2015) MIMO antenna design with built-in decoupling mechanism for WLAN dual-band applications. Electron Lett 51(13):966–968
9. Ruswanditya SA, Wahyu Y, Wijanto H (2017) MIMO 8×8 antenna with two H-slotted rectangular patch array for 5G access radio at 15 GHz. In: 2017 international conference on control, electronics, renewable energy and communications (ICCREC), Yogyakarta, pp 221–226 (2017)
10. Faleh S, Tahar BJ (2017) Optimization of a new structure patch antenna for MIMO and 5G applications. In: 2017 25th international conference on software, telecommunications and computer networks (SoftCOM), split, pp 1–5
11. Sunthari PM, Veeramani R (2017) Multiband microstrip patch antenna for 5G wireless applications using MIMO techniques. In: 2017 first international conference on recent advances in aerospace engineering (ICRAAE), pp 1–5
12. Chaudhari AA, Gupta RK (2018) A simple tri-band MIMO antenna using a single ground stub. Prog Electromag Res C 86:191–201
13. Marzouk HM et al (2019) Novel dual-band 28/38GHz MIMO antennas for 5G mobile applications. Prog Electromagn Res C 93:103–117
14. Waghmare GP, Gehlod K, Shakya A, Malviya L (2019) 2×2 Wideband array MIMO antenna for 5G spectral band. In: 2019 IEEE 5th international conference for convergence in technology (I2CT) Pune, India, pp 29–31

Uniform Channel Decomposition-Based Hybrid Precoding Using Deep Learning

B. Rajarajeswarie, Aravind Raj, and R. Sandanalakshmi

Abstract The massive multiple-input multiple-output (MIMO) is an emerging solution for advanced wireless communication due to the accomplishment of high data rate and system throughput in the wireless network. Precoding is used to allocate independent and appropriate weights for the data streams of the antenna arrays such that the link throughput is maximized at the receiver side. However, in fully digital precoding schemes, each and every antenna requires a dedicated radio frequency (RF) chain. To overcome these, hybrid precoding is found to be the most promising technique, as base station generates huge data, and it is difficult to fully achieve the spatial information. Hence, deep learning (DL)-enabled mm-Wave massive MIMO framework is constructed for active hybrid precoding. In Downlink, DL framework used the geometric mean decomposition (GMD)-based approach that suffers a capacity loss at low SNR. Hence, uniform channel decomposition (UCD)-based approach is adopted in the proposed framework. The proposed work performs better when compared with GMD-based hybrid precoding in terms of minimizing the bit error ratio (BER) and also enhances the spectrum efficiency.

Keywords Massive MIMO · mm-Wave · Deep learning · Hybrid precoding · Uniform channel decomposition

1 Introduction

The fifth-generation (5G) network is looking forward to supporting high throughput and low latency [1]. To meet the vital principle of future communication networks, many techniques have been proposed in which most of the researchers focused on the mm-Wave communication along with a massive MIMO system to enhance the system capacity by implementing numerous antennas at the base station [2]. The base station has to generate and sense huge data for communication from the environment

B. Rajarajeswarie (✉) · A. Raj · R. Sandanalakshmi
Department of Electronics and Communication Engineering, Pondicherry Engineering College, Puducherry, India
e-mail: rajarajeswarie.b@pec.edu

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_10

[3]. However, a practical massive MIMO system encounters many challenges, such as computational complexity and energy consumption [4]. These issues could be addressed by deep learning (DL) technique and hybrid precoding techniques. Deep learning technique could be able to fulfill and meet future requirements as it mimics the human brain in terms of learning and decision-making power [5, 6]. To realize the full capabilities of the MIMO system, each antenna of the array requires a dedicated radio frequency (RF) chain that would burden the network resource. Hence, the current research work is focused on the hybrid precoding technique to reduce multi-user channel interference and shorten the complexity of the receiver in order to enhance the channel capacity. The key idea of hybrid precoder is decomposed into two stages in which the first stage is baseband precoding and the second stage is RF precoding. In the pioneering work [7], hybrid precoding is realized by a successive interference cancelation, the issues are non-convex constraints and several sub rate optimizations. In singular value decomposition (SVD)-based hybrid precoding suffers from bit allocation problem. Therefore, it needs separate signal-to-noise ratios (SNRs) for different sub-channel [8]. To overcome this geometric mean decomposition (GMD) is implemented with identical SNR in [9, 10]. Addressing the non-convex constraint brings a great challenge to exploit spatial information. Therefore, uniform channel decomposition (UCD) is proposed because of its maximal diversity gain, and capacity loss is less at any signal-to-noise ratio (SNR) [11, 12]. Based on the limitations of the precoding scheme mentioned in the literature, the presented work has been summarized as follows.

1. Design a framework that integrates deep learning into hybrid precoding. Where multiple layer perceptron modeled with the aid of activation function to optimize the layers and make the corresponding mapping relations.
2. In the proposed work, the channel is decomposed by UCD to overcome the capacity loss at low SNR and also to achieve bit error rate and spectral efficiency.

Remaining of the paper is organized as follows: in Sect. 2, a conventional mm-wave massive MIMO system model has been presented. Section 3, provides an idea of implementing a proposed hybrid precoding scheme using deep learning. Simulation and the performance analysis are provided in Sect. 4, followed by the conclusion and future works in Sect. 5.

2 System Model

This section describes a multi-user massive MIMO system with conventional mm-wave channel model.

2.1 Massive MIMO System Model

In the proposed work, a typical mm-Wave massive MIMO system is considered, where a base station (BS) equipped with N_t transmitting antennas that send N_S independent data streams to users and N_{RF} chains to serve the K single-antenna users. We assume that the knowledge of the channel is known to the BS and $N_{RF} = K$. First, BS transmits the data stream to the digital precoder F_{BB} that is processed in baseband which is followed by an analog precoder F_{RF} in RF; it is used to realize the phase changes as it controls only the phase. The received signal at the user is represented by,

$$Y = HXs + n \tag{1}$$

Here, Y represents the received signal vector, and X is the hybrid precoding matrix that consists of analog precoder and the digital precoder.

$$X = F_{RF}F_{BB} \tag{2}$$

$$Y = HF_{RF}F_{BB}s + n \tag{3}$$

Here, H denotes the channel matrix with H= $[h_1, h_2, \ldots, h_k]^H$ being the channel vector between the BS and the Kth user $H \in \mathbb{C}^{NtxK}$. The hybrid precoding matrix is decomposed into the baseband digital precoding matrix (F_{BB}) and RF analog precoding matrix (F_{RF}) [13]. For all K users, s is the transmitted signal vector by satisfying $E[ss^H] = I_K$. The n indicates the additive white Gaussian noise vector of size $K \times 1, n \sim (0, \sigma^2)$ with zero mean and variance σ^2. The noise power is denoted by ρ^2. Where ρ is the total transmit power constraints.

$$\rho = \|F_{BB}F_{RF}\|^2 \tag{4}$$

2.2 Channel Model

Conventionally, in mm-wave massive MIMO, the channel is modeled in two ways: the first model describes the channel element with transmitting and receiving antenna. This type of model is known as an analytical model that is used for theoretical analysis by channel transfer function matrix. The second model describes the physical characteristics of the channel by electromagnetic wave propagation. The delay of multi-path components, angle of arrival (AoA), angle of departure (AoD), the direction of arrival (DoA), and direction of departure (DoD). The conventionally used channel model in the MIMO system is the Rayleigh fading model which is inaccurate for mm-wave, but it is traditionally used for MIMO analysis. To capture accurately the mathematical structure, Saleh-Valenzuela model has been utilized in the mm-wave channel. In this paper, Saleh-Valenzuela channel model [13, 14] has been adopted to capture the channel characteristics such as the angle of departure and the angle of arrival. The channel matrix is denoted by,

$$H = \sqrt{\frac{N_t K}{P}}(\alpha_0 a_t(\theta_0^t) a_r(\theta_0^r) + \sum_{p=1}^{P} \alpha_p a_t(\theta_p^t) a_r(\theta_p^r)) \tag{5}$$

Here, the number of non-lines of sight (NLos) component is represented as p. The array response of the user, as well as the BS, is defined in the form of steering vectors $a_r(\theta_p^r)$ and $a_t(\theta_p^t)$, respectively, [15]. The antennas in BS are of Uniform linear array (ULA) $a_r(\theta_p^r)$ and $a_t(\theta_p^t)$ can be expressed as:

$$a_r(\theta_p^r) = \frac{1}{\sqrt{N_r}}[1, \mathrm{e}^{-j2\pi i(\frac{d}{\lambda})} \sin \theta_p^r, \ldots, \mathrm{e}^{-j2\pi i(\frac{d}{\lambda})(N_r-1)} \sin \theta_p^r]^T \tag{6}$$

$$a_t(\theta_p^t) = \frac{1}{\sqrt{N_t}}[1, \mathrm{e}^{-j2\pi i(\frac{d}{\lambda})} \sin \theta_p^t, \ldots, \mathrm{e}^{-j2\pi i(\frac{d}{\lambda})(N_t-1)} \sin \theta_p^t]^T \tag{7}$$

Here, d represents the space between the antennas, and λ denotes the wavelength of the carrier frequency.

3 Proposed Hybrid Precoding Scheme

This section provides a DNN architecture and the learning policy to map the hybrid precoder for the efficient performance in the mm-wave massive MIMO system.

3.1 DNN Architecture

DNN has many hidden layers when compared to artificial neural network (ANN) to enhance the learning and mapping abilities [16]. This architecture is mainly used to improve the non-linear model through the activation function. The various activation functions used in MLP are the rectified linear unit (ReLU), softmax, tanh, Sigmoid, and so on. In most cases, ReLU and Sigmoid are used for non-linear operation, and it is defined as $\mathrm{ReLU}(x) = \max(0, x)$ and $\mathrm{Sigmoid}(x) = 1/(1 + \mathrm{e}^{-x})$, where x is the argument. In [15], the mapping operation can be expressed as:

$$Z = f(a, w) \tag{8}$$

where a denotes the input data. The weight of the neural network is represented by w. Figure 1 describes the DNN framework to realize the hybrid precoding, in which the dimension describes the length of each training sequence. The input data features are captured by the input layer with 128 units. The two hidden layers compress into 400 units and 256 units as shown in Figure 1. The output of the hidden layer is distributed

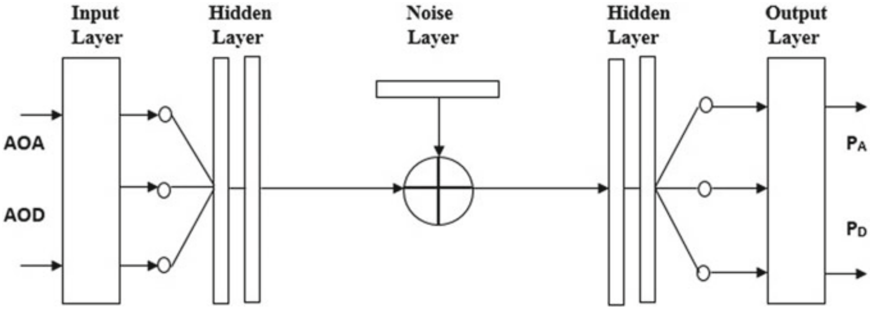


Fig. 1 DNN architecture of the proposed scheme

by the noise signal (noise layer), i.e., additive white Gaussian noise (AWGN), which consists of 200 units. Two more hidden layers are designed for decoding operation with 128 and 64 units. The output layer provides the required output signal of the system. The activation function used in the framework is the ReLu function.

3.2 Learning Policy

The learning policy is mainly used to map into the hybrid precoding. The network learns the features by capturing the spatial features in the angle domain. The UCD method is adopted to decompose the complex channel matrix in mm-wave massive MIMO. This approach can efficiently decrease the complication of the receiver. The UCD-based channel matrix H is formulated by,

$$H = QRP^* \tag{9}$$

$$Y = R_1 s + Q_1^H n \tag{10}$$

Here, R represents the upper triangular matrix with an identical diagonal element, P is the precoder, and Q is the combiner. The received signal based on UCD is represented by Y. Using DNN framework, the autoencoder is constructed, which is given by,

$$P_1 = f(P_A P_D; \Omega) \tag{11}$$

The mapping relationship and the data set samples are denoted by $f(.)$ and Ω, respectively. P_A and P_D represent the analog and digital precoder, respectively, and it is based on the UCD.

Algorithm 1 The proposed Hybrid Precoding algorithm

```
Input: The angle of arrival (AoA) $a_r(\theta_p^r)$ angle of departure (AoD) $a_t(\theta_p^t)$
Output: The optimized Precoder $P_1$
Initialization: $i \leftarrow 0$ and $w \leftarrow 0$, then set $P_A \leftarrow 0$ and $P_D \leftarrow 0$. The error threshold $\tau \leftarrow 10^{-7}$
Generate the AoA and AoD randomly
DNN framework constructed
Simulate the wireless channel with noise
while error $\geq \tau$ do
    optimize the DNN $\rightarrow$ stochastic gradient descent method
    Update the $P_A$ and $P_D$.
    Obtain the bias between $P_1$ and update the output layer of the DNN network.
end while
Return: $P_1$
```

4 Simulation Results and Analysis

The system performance of the proposed algorithm has been evaluated using MATLAB2018a and compared with GMD-based precoding. Table 1 describes the system parameters used for the analysis. The proposed approach investigates and evaluates the BER and spectrum efficiency performance in the mm-Wave massive MIMO system. The proposed hybrid precoding is compared with those of GMD-based hybrid precoding and fully digital GMD-based precoding. The BER performance is evaluated with various batch sizes of the training data set. The number of samples that are passed to the neural network on one occasion is known as batch size. The network is trained for 45,000 iterations.

Figure 2 shows the bit error rate (BER) performance of the proposed hybrid precoding scheme compared with those of the GMD-based fully digital precoding, GMD-based hybrid precoding, and proposed UCD-based hybrid precoding. It is observed that for GMD-based hybrid precoding, and proposed UCD-based hybrid precoding, there is an 8-dB improvement at BER of 10^{-2}. This implies that UCD-based hybrid precoding solves the non-convex problem with the aid of deep learning.

In Fig. 3, the performance of the proposed hybrid precoding using UCD is evaluated through spectrum efficiency and compared with the other conventional hybrid precoding. It has been observed that as SNR increases, the spectrum efficiency is also

Table 1 System parameters and its value

Parameters	Value
N_t	128
K	64
NLoS	3
Carrier frequency	28 GHz
Iterations	45,000

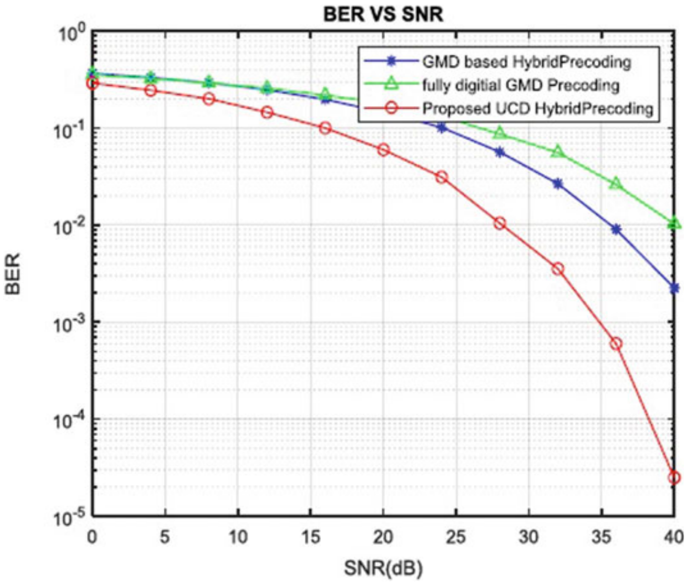


Fig. 2 BER performance versus SNR

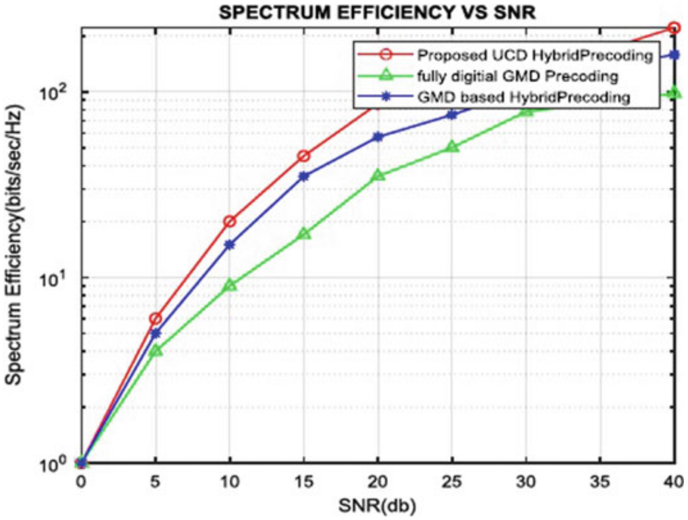


Fig. 3 Spectrum efficiency versus SNR

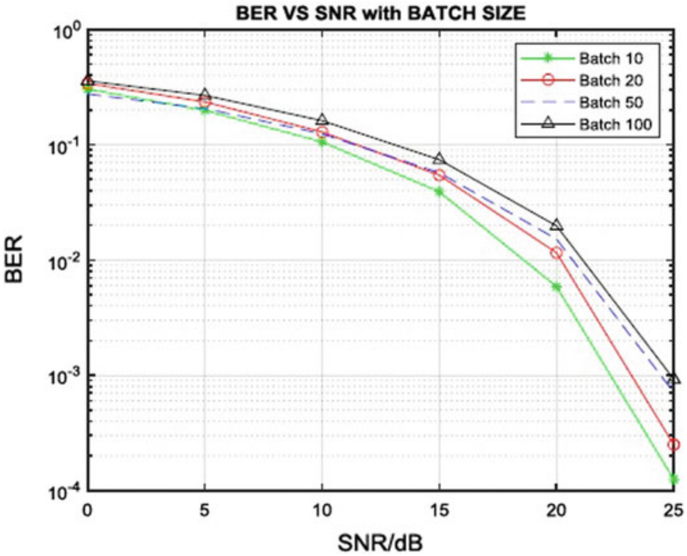


Fig. 4 BER versus SNR with various batch size

improved in all the methods. From Fig. 3, it shows the proposed hybrid precoding performs better than the other method by the excellent mapping of deep learning. Also noted that the performance gap of the proposed method and other method is large as SNR increases.

Fig. 4 shows the BER performance for various batch sizes against the SNR. It is observed that as the size of the batch increases, the performance of deep learning-based approach dismisses in terms of BER. Therefore, the size of the batch must be chosen carefully to achieve the optimal performance for the proposed precoding scheme. It is well understood that the proposed method based on a deep learning approach provided superior performance compared with other methods.

5 Conclusion

In this paper, we proposed innovative hybrid precoding for mm-Wave massive MIMO to decrease the computational complexity using deep learning. The DNN framework is built with a proposed UCD method as learning policy to improve BER performance and spectrum efficiency. The proposed algorithm achieves a near-optimal performance in comparison with the traditional algorithm. The forthcoming work is to apply the intelligent technique for radio resources management in the mm-Wave massive MIMO system.

References

1. Health RW, Gonzalez-Prelcic N, Rangan S, Roh W, Sayeed A (2016) An overview of signal processing techniques for millimeter wave MIMO systems. IEEE J Sel Top Sig Process 10(3):436–453
2. Rappaport TS, Sun S, Mayzus R, Zhao H, Azar Y, Wang K, Wong GN, Schulz JK, Samimi M, Gutierrez F (2013) Millimeter wave mobile communications for 5G cellular: it will work! IEEE Access 1:335–349
3. Zoha AI, Abu A (2014) Challenges in 5G: how to empower SON with big data for enabling 5G. IEEE Netw 28:27–33
4. Larsson GE, Edfors O, Tufvesson F, Maezetta LT (2014) An overview of massive MIMO: benefits and challenge. IEEE J Sel Top Sig Process 8:742–758
5. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554
6. Kato N et al (2017) The deep learning vision for heterogeneous network traffic control: proposal, challenges, and future perspective. IEEE Wirel Commun 24(3):146–153
7. Gao X, Dai L, Han S, Chih-Lin I (2016) Energy-efficient hybrid analog and digital precoding for mm wave MIMO systems with large antenna arrays. IEEE J Sel Areas Commun 34(4):998–1009
8. Jin J, Zheng YR, Chen W, Xiao C (2018) Hybrid precoding for millimeter wave MIMO systems: a matrix factorization approach. IEEE Trans Wirel Commun 17(5):3327–3339
9. Jiang Y, Li J, Hager WW (2005) Joint transceiver design for MIMO communications using geometric mean decomposition. IEEE Trans Sig Process 53(10):3791–3803
10. Chen CE, Tsai Y-C, Yang C-H (2015) An iterative geometric mean decomposition algorithm for MIMO communications systems. IEEE Trans Wirel Commun 14(1):343–352
11. Yi J, Li J, Hager WW (2005) Uniform channel decomposition for MIMO communication. IEEE Trans Sig Process 53(11)
12. Chen R, Li J, Chen D, Hui L (2012) Robust uniform channel decomposition and power allocation for MIMO systems with imperfect CSI. Wirel Pers Commun 761-781, (2012)
13. Gao Z et al (2015) Mm-wave massive-MIMO-based wireless back haul for the 5G ultra dense network. IEEE Wirel Commun 22(5):13–21
14. Xie T et al (2018) Geometric mean decomposition-based hybrid precoding for mm wave massive MIMO systems. China Commun 15(5):229–238
15. Huang H, Song Y, Yang J, Gui G, Adachi F (2019) Deep-learning-based millimeter-wave massive for hybrid precoding. IEEE Trans Veh Technol 68:3027–3032
16. Fadlullah ZM et al (2017) State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE Commun Surv Tutor 19(4):2432–2455

An Effective Scheme to Mitigate Blackhole Attack in Mobile Ad Hoc Networks

Mukul Shukla and Brijendra Kumar Joshi

Abstract MANET refers to a mobile ad hoc network, which is self-configured without having a fixed base. It has been used in various fields for various purposes like military, local conferences, and the movable of information. However, due to the lack of built-in security, safety is a significant concern in the MANET. There are various types of attacks which are possible on MANETs. One of the attack is a Blackhole attack. It is an active attack in which a malicious node shows itself as the shortest route and absorbs the packet just like a blackhole does in-universe. In this paper, a proposed technique has used a trust-based fuzzy method based on auditing of energy, the Trust of a neighbouring node, check for the integrity of packets, and authentication of the Node Member. Trust values in the fuzzy logic range from 0 to 1. If node trust value is higher than or equal to 0.6, then the node is trusted, and its type of node taken in our scenario for communication between source to destination. If the node trust rate on the routing table is less than 0.6, that means the node is a blackhole node, and its type node does not consider a safe route, in this paper proposed a method, i.e., Trust-based Fuzzy Ad hoc On-Demand Distance Vector (TFAODV), to an attack scenario. This paper is improved results in terms of throughput, packet delivery ratio, end-to-end delay, shows the throughput improvement, and found to be 1441 kbps, packet delivery ratio enhancement of 57.10%, and delay decrease of 52% from Blackhole Ad hoc On-Demand Distance Vector (BAODV). Therefore, the proposed protocol covers the way and possesses the potential to secure the MANET.

Keywords Blackhole attack · MANET · AODV · Trust authentication · Trust of a neighbouring node

M. Shukla (✉)
Department of Information Technology, Shri G. S. Institute of Technology & Science, Indore, India
e-mail: mukul@sgsits.ac.in

B. K. Joshi
Electronics & Telecommunication and Computer Engineering, Military College of Telecommunication Engineering, MHOW 454431, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_12

1 Introduction

MANET fall in that category of networks that can operate without the help of any infrastructure. MANETs are helpful in that place where we require fast deployment, and there is no wired link available.

It requires an expensive investment because it has a base station and mobile nodes. Many tasks are taken care of like network monitoring, discovering routes, sending packets, and making communication secure. The most important one is to prevent the network from attacks, and for that purpose, we have designed an algorithm that takes care of it robustly [1]. Intermediate nodes in MANET are responsible for routing in a multi-hop fashion. In the case of a wired network, we face route failure; the same problem exists in the wireless network also [2]. The leading cause of route failure in MANET is the limited battery power and mobility of nodes.

Figure 1 expressions show the underlying architecture of MANET. Due to the high movement of nodes, routing is one of the exceptions in the MANET [3].

In MANET, the protocol for routing should be adaptive; it means that they should have the capability to fight with the worst condition of path and link break between source and destination node during communication [4].

In MANETS, majorly, two attacks are seen, i.e., active attacks and passive attacks. Here, in this research work, we emphasize active attacks. In this paper, we have stressed of Blackhole attack [5].

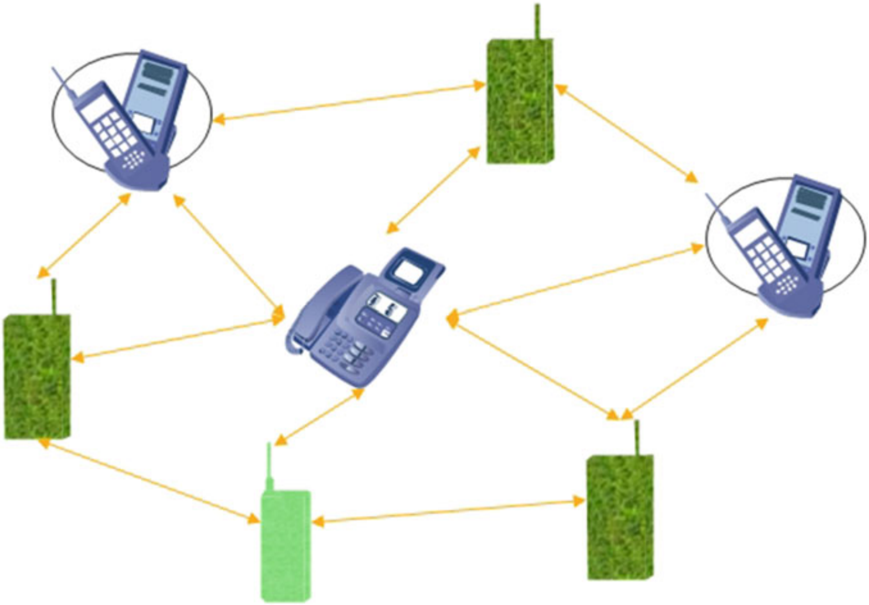


Fig. 1 General architecture of MANET

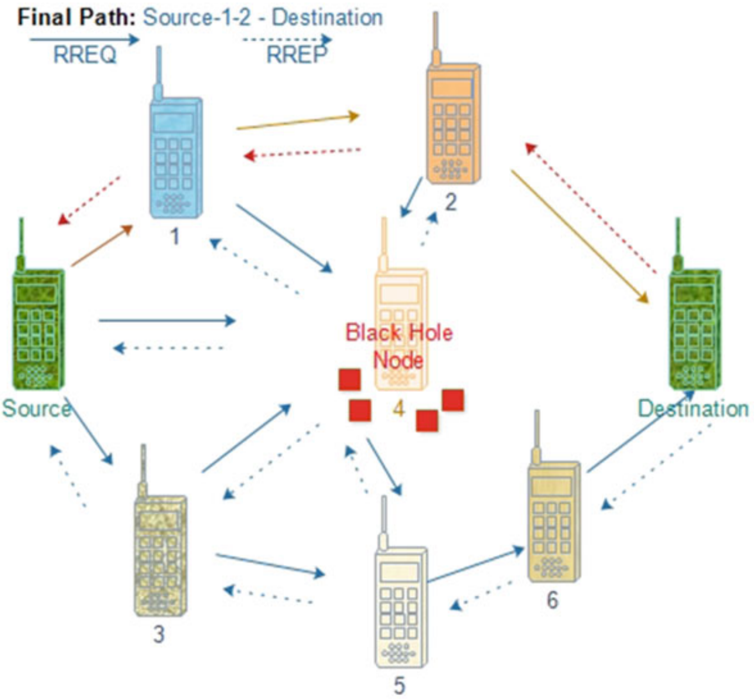


Fig. 2 Network diagram representing various attacks

(a) **Blackhole Attack (BHA)**: The BHA in MANET is the attack in which the malicious node sends the wrong information to the source node about the shortest path in between source and destination. It sends the RREP packet. In Fig. 2, the attack between the node source and destination is the blackhole attack. This type of attack is most prominent in the ad hoc network and needs attention. To avoid the blackhole attack, our algorithm detects the malicious nodes and prevents this type of attack in the system [5].

In Fig. 2, BHA is shown in between Node source and destination. One malicious node represented a blackhole node that exists, so the attack caused by them is called Blackhole Attack.

The paper contributes majorly in the following:

1. This paper consists of a proposed approach that helps to find the shortest route and provide security to the network.
2. Our approach consists of the combination of BAODV and Trust-based Fuzzy AODV (TFAODV) that helps to find the shortest distance between the nodes.
3. Once the shortest route is detected, to ensure the network's security, we will perform a trust authentication. This trust value ensures the reliability of the node.

4. Finally, the efficiency of the proposed approach is computed by using evaluation parameters like throughput, e-to-e delay, and packet delivery ratio. Outcomings states that among three protocols BAODV, and TFAODV. The proposed TFAODV progresses the effectiveness and safety of the network.

Organization of the paper: Sect. 2 of the literature review presents a detailed study of previous research conducted in the same field. Section 3 consists of a comprehensive survey of the proposed approach and performance analysis. Section 4 gives implementation details: simulation and setup, the results, and the impact of various factors on parameters. The last Sect. 5 concludes the paper.

2 Related Work

Li et al. [6] have presented an article, which was used for the detection of glitch. It has two main stages: the first is to train, and the second is glitch detection, which is capable of the discovery of abnormal occurrences, and the specific attack modes are provided to the operators of the network. The results obtained for the anomaly detection are 96% precision, and 7 clusters are given for the attack nodes.

Keerthika et al. [7] have worked for securing from the blackhole attacks on MANET. For preventing BHA, the authors have presented a secure routing mechanism that is trust based and is AODV routing. They have applied a hybrid Weighted Trust-based Artificial Bee Colony 2-Opt algorithm. The use of 2opt for the local search is done for the implementation of the hybridization. The parameters such as PDR, E-to-E Delay, and hop sinking are used for analysis.

Kavitha et al. [8] have applied PSO for feature optimization and classification by using Neural Networks, which detects the malicious node. Authors have worked upon the parameters like PDR, communication delay, and energy consumption for the identification of malicious node and isolation of the intruder.

Gurung et al. [9] has given the issues related to the BHA on MANET, and apart from this, various natures of nodes are also stated by them. The author has been provided with the techniques for dealing with the BHA, and the classification of these methods is done based on the essential operation of this method.

Tourani et al. [10] have proposed that Information-Centric Networking (ICN) and a survey is performed on the previous literature present in terms of the security and the privacy for the information-centric networking also they have presented an open question related to it. They are considering it shorter the authors have categorized it into three areas that are the threats of security and the risks seen in privacy and the enforcement mechanisms for the access control.

Ochola et al. [11] have discussed the blackhole attack on MANET. They have simulated their proposed algorithm with AODV and DSR protocol on simulators NS2 and NS3. They have got results based on parameters PDR; throughput is decreased on the blackhole scenario as the data packets are lost from the malicious nodes. There is also seen the decrement in the E to E delay as the malicious node acts as the correct node, and route discovery becomes faster.

Sankara Narayanan et al. [12] have discussed that MANET's Blackhole attack is one of the significant threats seen. Blackhole attack has the property to create a tunnel between two nodes in a given scenario. Their proposed method detects active and passive attacks and got better results based on network parameters compared to actual work.

Yaseen et al. [13] have proposed an algorithm that selects the best route in the scenario when more than one path exists between source and destination. They had better outcomes in terms of delay and routing over the head when they simulated their algorithm.

Cai et al. [14] have proposed the ESCT algorithm to detect and prevent attackers on MANET. The ECST works as the exchanging the information of Trust between the nodes and will do the cognitive judgment to analyse the knowledge of Trust. Their proposed algorithm works best in terms of Mobile Ad hoc network parameters.

3 Proposed Scheme

The application of the fuzzy logic rule prediction method has been discussed in this section. For this, trust value management has been implemented between two nodes where each node keeps a value of Trust with its neighbour node. To existing AODV protocol in MAENT, route trust is calculated after trust value computation. Packets are transmitted further only if the nodes are found valid. The trust value calculation is as follows:

$$T_j(j) = \alpha T_{i(\text{self})}(j) + \beta T_{i(\text{neighbour})}(j) \tag{1}$$

denoted.

$T_j(j) =$ states the worth of Trust for node i to its neighbour node j.

$T_{i(\text{self})}(j)$ states value of Trust for node i and node j.

$T_{i(\text{neighbour})}(j)$ it is the value of Trust that node i is having on node j, the increment factor is given by α and β, where the value is represented as $\alpha + \beta = 1$.

While the transmission of packets takes place, the algorithm for this is the list of source nodes, i.e., S_List is maintained as in the Dishonest node, and the packet from the source node is observed.

```
Begin ()
{
    If ((fwd) $node_j$ and (S_List Contain Dishonest node))
        {
            (fwd) $node_j$++;
            (To fwd) $node_j$++;
            (fwd) $node_j \geq$Limit
        }
    Else
        {
            Repeat (trust value computation);
        }
Exit ();
}
```

In case the immortal node fails in the task of updating and forwarding, also the count for node *j* the count of forward, then it is detected as blackhole node. If not, then the transmission is considered as secure.

3.1 Auditing of Energy

Whenever data is transmitted to neighbour nodes, energy is consumed. The selfish nodes are recorded to have the maximum energy conservation as they are only meant to receive data and not transmit it. Trusted nodes are required to obtain and forward data packets, hence consume more data. Depending on the initial node configuration, each node has a variable energy calculation (E_v) [15]. The setup relies upon parameters like transmission consumption of power at reception and control at ideal. An energy supervisor monitors the energy consumption in MANET whenever any transmission takes place.

All nodes usually behave like a selfish node to conserve energy because of the limited availability of resources. The energy supervisor also regulates the packets transmitted over a node.

$$E_{v=\sum(\text{Packetreceived}+\text{Packetforwarded}+\text{Batterpower})/\text{Node}} \tag{2}$$

3.2 Trust Manager

In our presented method of the neighbourhood relationship verification, we allow the detection of the route with the blackhole node present in it. When the node N_i gets the packet of RREQ or RREP from any other node N_j and in case, this node is not neighbours. Then for the route discovery, there must have a hidden mode blackhole node.

Definition: It is to be stated that if two nodes lie within the radius of transmission, then this node is neighbours (N_i, N_j). Practically the neighbourhood of both nodes is defined only when d (N_i, N_j) $\leq$ min ($R_{N_i} - R_{N_j}$) here, R is the max radius of transmission, and d is the Euclidean distance that can be calculated from Eq. (3) [16].

$$d\left(N_i,\, N_j\right) = \sqrt{\left(X_{N_i} - X_{N_j}\right)^2 + \left(Y_{N_i} - Y_{N_j}\right)^2} \tag{3}$$

Here $X_{N_\delta} - Y_{N_\delta}$ are the coordinates of the location for the node N_δ.

The behaviour of the neighbour node helps determine the trust value. A dog mechanism is employed to keep a check on the neighbour node's activities. There is a passive observation for the detection of delayed packets, dropped packets and forwarded packets. Any undesired action is recorded. On the commencement of communication, the total trust value is recorded based on all the observations and documented in the trust table.

$$T_v = \text{Node\_ index} + \text{Direct\_ trust} \tag{4}$$

i. The trust request recommendation is referred by the origin node S to the node(s) N.
ii. For the source node, the value of Trust is checked.
iii. The Trust recommendation Reply is sent in case the S is having the trust value directly on the D.
iv. When the direct Trust is not found, then the trust recommendation request is directly discarded.
v. When the reply is obtained from the recommendation trust, then the fuzzy logic method is applied, and this shows the largest value of direct Trust for every node.
vi. After obtaining all the direct Trust, we will compute the indirect value of Trust for all the neighbouring nodes.

3.3 Check for the Veracity of Packets

The changed message from the middle node can be rejected to ensure the integrity of packet communication. In case of any modification, the packet veracity cost (PVC) value decreases from positive. There are digital signatures using the node's private key with every message that is generated. These signatures are used to decrypt messages and to authenticate messages from the neighbour node.

Identically, the message is authenticated by all intermediate nodes and is forwarded to the neighbourhood. There is single point documentation in the PVC value in case of any alteration. Digital signatures are verified using the RSA algorithm.

This verification of packet is performed to save the tampering of the control packet from the blackhole node at the time when the hop to hop relaying is being performed. The blackhole nodes do the content altering in the Participation Mode. The work done here by any node (N_i) is the verification of the integrity of any of route request (RREQ) or route reply (RREP) from the source or any other node (N_j). The verification of the packet is found. Firstly, the sender node N_j hash the fields of RREQ or RREP, and after this, the encryption is done with the use of the private keys, as stated in Eq. 5. The value encrypted is saved in the RREQ or RREP packet checking value field before sending it to any other node N_i.

$$P.\mathrm{VC} = \mathrm{En}\,(H\,(P.\,\mathrm{Fields}\,\{\mathrm{VD}\}),\,k_{n_j} \tag{5}$$

Here P is the packet (RREQ or RREP),

Definition: The N_{center} will provide the certification of membership automatically to all the nodes, and at the time of the route discovery, this certification is included in the RREQ or RREP packets. For the calculation of the Membership Certification for any node N_δ hash value is encrypted at first with the node address and the public key $k_{N_\delta}+$ along with the private key of the N_{center}. Then the encryption is performed with the encrypted result obtained from step first with the private key of N_δ and this can be seen in Eq. 6.

$$\mathrm{MC}_{N_\delta} = \mathrm{En}(\mathrm{En}(H(IP_{N_\delta}, k_{N_\delta}+), k_{N_{center}}-), k_{N_\delta}-) \tag{6}$$

3.4 Authentication of the Node Member

As seen in the cryptography techniques applied as a solution, every node presented has a private and public key. Here the proposed method certifies that the node that is participating in the route discovery should be certified, and the neighbour node will certify this as per our presented method. This node resulting with the public key of the node N_{center}. This is certified by matching the hash value of the address of the node N_δ after the decryption is done, then the validity of N_δ node is confirmed. Else the packet dropping is done. It has to be noted that after performing the integrity of node RREQ or RREP, then only we can perform the authentication process ($\mathrm{AN}_{\text{value}}$).

3.5 Final Trust Manager

All the four aspects studied earlier are used for the computation of the final value of Trust. The energy of node, trust rate, type, packet veracity cost, and authentication of a node is recorded in the trust table.

For every node, the computation of the final value of Trust is done as

$$\mathrm{FT}_{\mathrm{value}} = E_{\mathrm{value}} + T_{\mathrm{value}} + \mathrm{PVC}_{\mathrm{value}} + \mathrm{AN}_{\mathrm{value}} \tag{7}$$

3.6 Certificate Authority (CA)

The node with the highest value of Trust is selected as the certificate authority node. The Trust table assists such a selection. The certificate authority ensures a secure transmission and node segregation. If the node is selected, the node requires renewal. The centralized control certifies the source and the destination. The packet uses the public key for encryption and forward transmission. The intermediate nodes are not expected to view the message in the middle of the process by decryption.

The proposed SHA1 algorithm uses hash packets for their simplicity and minimal cost of energy. Algorithm for the authentication certificate:

Notation of Algorithm	
SSK	Shared Key at Source
DSK	Shared Key at destination
PBSK, PBDK	for source and destination, the Public Keys
PRSK, PRDK	for source and destination Private Keys
SID	ID for the source node
DID	ID for the destination node

```
Begin ();
{
    Init (SSK);         // initiation of SSK
    If (communication is started E [SREQ (SIS, DID, Ftv) SSK])
      {
          S → CA
      }

    The certified authority of Sreq does the decryption, and it finds out the Sid and the
    repository of the ID
    If (SID == ID)
      {
         CA → Verification (Sid and the DID); // verification of the Sid and the DID is
          done by the certificate authority
      }
    Generate () → PBSK, PBDK, PRSK, PRDK, DSK;   //Generate PBSK, PBDK,
    PRSK, PRDK, DSK
    CERT A = SID, PBSK, PBDK, PRSK, PRDK, Ftv, TS  }
    CERT B = DID, PBDK, PRDK, Ftv, TS              }  // authentication certificate
    is given

    If (authority is permitted)
      {
        Send → e [(CERT A) SSK] →  starting point
        Send → e [(CERT B) DSK] → end point
      }
    Else
      {
        Print ("B denial of transmission");
      }
Exit ();
}
```

3.7 Fuzzy-Based Analyzer

A positive trust level experience represents node reliability, which decreases with a negative value—trust values in the fuzzy logic range from 0 to 1.

We have selected values subjectively, i.e., $a1$, $a2$, $a3$. The addiction function states the addiction values I state the description of the Trust from the available options of the values x on the model of parameters as the i (x, p). The computed values such as E_v, FT_v, T_v, PVC, and helps calculate the trust value of the node. Based on the fuzzy logic algorithm, all these values are taken as the input to the fuzzy algorithm, and the marking of a node is done as the Blackhole or the trusted node.

Whenever the communication is set up for the data packet exchange, it automatically calls the fuzzy logic algorithm. The node is marked as malign if the fuzzy value is found lower than the threshold value. On initiation of any communication between two nodes, a fuzzy analyzer is involved once the certified authority is requested. An alarm message is generated by the certified authority in case it detects any blackhole node. Secure transmission is made for medium, high, and very high fuzzy values. Trust node requests for node renewal when the node certificate expires.

4 Implementation and Result

In this section, experiments are conducted to verify the effectiveness of the proposed protocols named BAODV and TFAODV. The proposed protocol validates on NS2 simulation tool. The algorithm outcomes are compared with BAODV and TFAODV protocols based on five performance matrices: throughput, E to E delay, packet delivery ratio (PDR).

4.1 NS2 Simulation Study

Here, the discussion of the results is given, and the comparison is made for the results obtained, and this result is carried out on the experiments done on the NS2 simulator.

We have added the concept of the proposed algorithm on NS2, and we change directory NS2.35 on file like Makefile, priqueue.cc, packet.h. Cmu-trace.h. Cmu-trace.cc, ns-packet.TCL, ns-lib.TCL, ns-agent.tcl and ns-mobile node.tcl. After adding the concept of both proposed algorithms on the above file, then we execute the command step by step. /configure, /make clean, /make and. /install.

4.2 NS2 Simulation Parameters

This subsection presents a comparison of the proposed algorithm along with BAODV [17] and TFAODV protocols for throughput, E to E Delay, and packet delivery ratio of routing performance metrics.

Table 1 shows the simulation parameters. The network simulator is version 2.35, Simulation time 900in Seconds, Network size is 900 m * 900 m, 3 m/s and 5/s m are maximum node speed, the data rate is CBR, source and destination are random, Data packet size is 512 bytes, Protocols are BAODV and TFAODV, Phy/MAC Protocol in IEEE 802.11, Two-ray ground model of propagation, mobility model is a random waypoint, wireless channel, Omnidirectional model of antenna, and languages are Tcl, oTcl, C++, AWK Scripting. For simulation, we have considered the 45 mobile nodes and 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 enter into the network as malicious nodes, which results in the Blackhole. In our network, it is mobile nodes.

Table 1 Simulation Parameters

Parameters	Specification	Parameters	Specification
Network simulator	NS-2, Version 2.35	PHY/MAC Protocol	IEEE 802.11
Network size	900 m×900 m	Propagation model	Two-ray ground
Connection protocol	UDP/CBR	Mobility model	Random way point
Data Type	Constant bit rate (CBR)	Channel type	Wireless channel
Source/Destination	Random	Antenna Model	Omnidirectional
Data packet size	512 bytes	Simulation time (Second)	900
Simulation protocol	BAODV, TFAODV	Language	Tcl, oTcl, C++, AWK Scripting
Simulation scenario (No. of Mobile Nodes)	45	No of malicious nodes	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

4.3 *Result*

This section includes results that are obtained by the proposed approach. For evaluating the results, evaluation parameters are used; these considerations are throughput, e to e delay, PDR. This result section shows the overall result of the considered scenarios includes a total node as 45 in-network.

The parameters are as follows:

1. Throughput: the data retrieval at the destination node in any unit of the time interval is termed as throughput [18].

$$\text{Throughput} = \frac{\text{receivedbytes} * 8}{\text{timeofsimulation} * 1024}\text{kbps}$$

2. 2Avg E-to-E Delay: The time utilized by a packet to reach source to destination is called as the end-to-end delay [18]

$$\text{AvgEEdelay(ms)} = \frac{1}{N}\sum_{n=1}^{N}(R_n - S_n)$$

3. PDR: The data packets' ratio sent to the data packets received is termed as the PDR [18]. Mathematically, it can be defined as

$$\text{PDR}(\%\ ) = \frac{\text{packetsrecieved}}{\text{packetssent}}$$

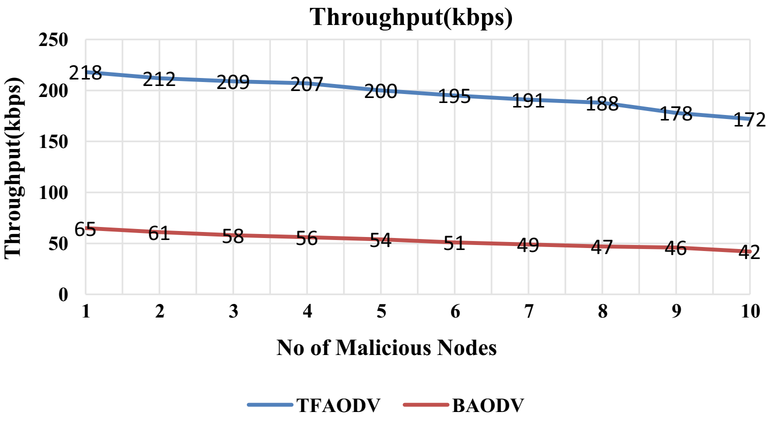


Fig. 3 Throughput based on No. of malicious nodes

4.3.1 Throughput

Throughput is the parameter that keeps track of several packets delivered successfully per unit of time. From Fig. 3, it has been seen that Trust-based fuzzy AODV (TFAODV) returns better results with the enhancement of 1441 kbps as compare to (Blackhole attacked) BAODV.

We can observe from the above graph that when we have introduced an attack in the network, the throughput is reduced. But with the application of our proposed method, TFAODV, the performance increased.

4.3.2 End-to-End Delay

The time utilized by a packet to reach source to the destination is called as the end-to-end delay. Figure 4 shows that with the increase in the number of nodes, the value TFAODV also increases means. Even though in this case, TFAODV shows better performance compared to BAODV.

From the above graph, it can be easily observed that the application of the proposed method TFAODV reduced the overall delay from source to the destination when compared with the attacked scenario.

4.3.3 Packet Delivery Ratio

The ratio between packet received by destination and packets transmits by the source is known as PDR. Figure 5 shows that the TFAODV shows an improvement of 57.10% compared to the attacked network with minimum alteration of actual BAODV of network based on PDR.

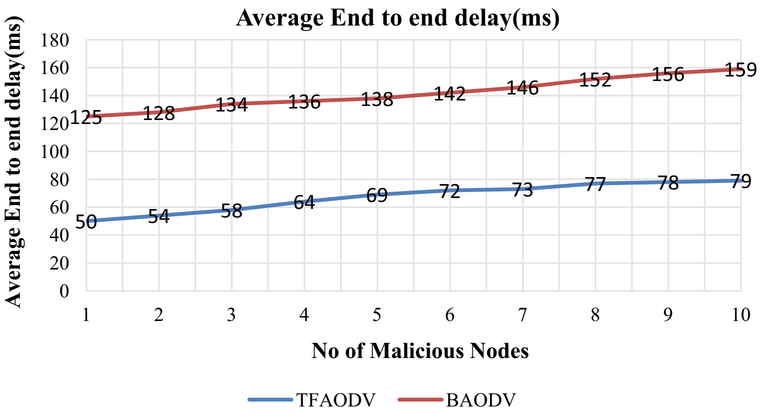


Fig. 4 End-to-End Delay based on No. of Malicious Nodes

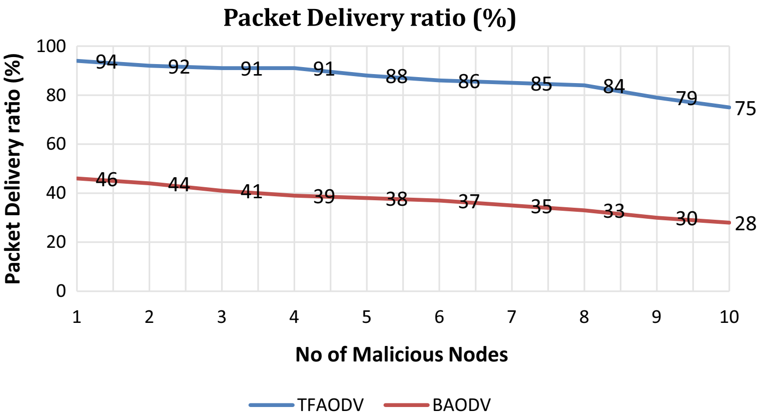


Fig. 5 Packet delivery ratio percentage based on No. of Malicious nodes

In this network's packet delivery ratio is low when introduced the malicious nodes. By applying the proposed method, we can see that the packet delivery ratio is increased.

5 Conclusion

In this research paper, we have designed an algorithm that takes care of a blackhole. We have proposed a trust-based fuzzy technique based on auditing of energy, the Trust of a neighbouring node, Check for the integrity of packets, and Authentication of the Node Member. Trust values in the fuzzy logic range from 0 to 1. If node trust value is higher than or equal to 0.6, we assume node is trusted and its type of node we have taken in our scenario for communication between source to destination. If the node trust value is less than 0.6, that means the node is a blackhole node, and its type node we do not consider in our safe route. We have applied our proposed method, i.e., TFAODV, to an attack scenario. Improved results in terms of throughput, packet delivery ratio, and end-to-end delay show the throughput improvement of 1441 kbps, packet delivery ratio enhancement of 57.10%, and delay decrease of 52% from BAODV. Therefore, the proposed protocol paves the way and possesses the potential to safeguard the MANET.

References

1. Elmahdi E, Yoo SM, Sharshembiev K (2020) Secure and reliable data forwarding using homomorphic encryption against blackhole attacks in mobile ad hoc networks. J Inf Secur Appl 51:102425. https://doi.org/10.1016/j.jisa.2019.102425
2. Moudni H, Er-Rouidi M, Mouncif H, El Hadadi B (2019) Black hole attack detection using fuzzy based intrusion detection systems in MANET. Procedia Comput Sci 151:1176–1181. https://doi.org/10.1016/j.procs.2019.04.168
3. Kalkha H, Satori H, Satori K (2019) Preventing black hole attack in wireless sensor network using HMM. Procedia Comput Sci 148:552–561. https://doi.org/10.1016/j.procs.2019.01.028
4. Vinayagam J, Balaswamy C, Soundararajan K (2019) Certain investigation on MANET security with routing and blackhole attacks detection. Procedia Comput Sci 165:196–208. https://doi.org/10.1016/j.procs.2020.01.091
5. Tsiota A, Xenakis D, Passas N, Merakos L (2019) On Jamming and black hole attacks in heterogeneous wireless networks. IEEE Trans Veh Technol 68(11):10761–10774. https://doi.org/10.1109/TVT.2019.2938405
6. Li T, Ma J, Pei Q, Song H, Shen Y, Sun C (2019) DAPV: Diagnosing anomalies in MANETs routing with provenance and verification. IEEE Access 7:35302–35316. https://doi.org/10.1109/ACCESS.2019.2903150
7. Keerthika V, Malarvizhi N (2019) Mitigate black hole attack using hybrid bee optimized weighted trust with 2-Opt AODV in MANET. Wirel Pers Commun 106(2):621–632. https://doi.org/10.1007/s11277-019-06182-8
8. Kavitha T, Geetha K, Muthaiah R (2019) India: Intruder node detection and isolation action in mobile ad hoc networks using feature optimization and classification approach. J Med Syst 43(6). https://doi.org/10.1007/s10916-019-1309-2
9. Gurung S, Chauhan S (2019) A survey of blackhole attack mitigation techniques in MANET: merits, drawbacks, and suitability. Wirel Networks 1–31. https://doi.org/10.1007/s11276-019-01966-z
10. Tourani R, Misra S, Mick T, Panwar G (2018) Security, privacy, and access control in information-centric networking: a survey. IEEE Commun Surv Tutorials 20(1):556–600. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/COMST.2017.2749508
11. Ochola EO, Mejaele LF, Eloff MM, Van Der Poll JA (2017) Manet reactive routing protocols node mobility variation effect in analyzing the impact of black hole attack. SAIEE Africa Res J 108(2):80–91. https://doi.org/10.23919/saiee.2017.8531629
12. Sankara Narayanan S, Murugaboopathi G (2020) Modified secure AODV protocol to prevent black hole attack in MANET. Concurrency Comput 32(4). https://doi.org/10.1002/cpe.5017
13. Yaseen QM, Aldwairi M (2018) An enhanced AODV protocol for avoiding black holes in MANET. Procedia Comput Sci 134:371–376. https://doi.org/10.1016/j.procs.2018.07.196

14. Cai RJ, Li XJ, Chong PHJ (2019) An evolutionary self-cooperative trust scheme against routing disruptions in MANETs. IEEE Trans Mobile Comput 18(1):42–55. https://doi.org/10.1109/TMC.2018.2828814
15. Arulkumaran G, Gnanamurthy RK (2019) Fuzzy trust approach for detecting black hole attack in mobile ad hoc network. Mobile Networks Appl 24:386–393. https://doi.org/10.1007/s11036-017-0912-z
16. Vo TT, Luong NT, Hoang D (2019) MLAMAN: a novel multi-level authentication model and protocol for preventing blackhole attack in mobile ad hoc network. Wirel Network 25:4115–4132. https://doi.org/10.1007/s11276-018-1734-z
17. https://tools.ietf.org/html/rfc3561
18. Uddin M, Taha A, Alsaqour R, Saba T (2017) Energy-efficient multipath routing protocol for a mobile ad-hoc network using the fitness function. IEEE Access 5:10369–10381. https://doi.org/10.1109/ACCESS.2017.2707537

HeuristiX: Adaptive Event Processing Using Lightweight Machine Learning Approaches on Edge Devices

Sanket Mishra, Shubham Arawkar, Sashikant Singh, Ankit Shibu, and Chittaranjan Hota

Abstract Recent times have witnessed a significant proliferation of cognitive applications in smart cities. Internet of things (IoT) is instrumental in gathering data that can be processed for creating intelligent scenarios. For developing such IoT applications, heterogeneous sensors can be deployed across various regions of a city to collect voluminous data. This data needs to be processed for predictions with minimum latency so as to formulate predictions in real time. In this paper, we present HeuristiX, a framework using lightweight machine learning solutions that can be deployed at edge devices. The work employs machine learning approaches to predict congestion events in a smart city scenario. By using Openwhisk, data is processed in minimum latency on resource-constrained devices for predicting congestion events in the city of Madrid. We augment the insights generated by models deployed on edge devices with a complex event processing engine, Apache Flink that is deployed on a server (cloud). The CEP forms complex events indicating congestion in specific areas across the city. We implement forecasting approaches and clustering approaches on IoT boards for processing traffic data for enriching the capabilities of Flink, i.e., prediction of future congestion events and generation of adaptive thresholds for CEP queries. Our proposed framework, HeuristiX, depicts that processing raw IoT at the edge is efficient as it minimizes latency and also provides a distributed setup for processing of data for better management of resources.

S. Mishra (✉) · S. Arawkar · S. Singh · A. Shibu · C. Hota
BITS Pilani Hyderabad Campus, Hyderabad, Telangana, India
e-mail: h20191030502@hyderabad.bits-pilani.ac.in

S. Singh
e-mail: h20191030022@hyderabad.bits-pilani.ac.in

A. Shibu
e-mail: f20170297@hyderabad.bits-pilani.ac.in

C. Hota
e-mail: hota@hyderabad.bits-pilani.ac.in

S. Mishra
Vellore Institute of Technology, Amaravati, Andhra Pradesh, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_13

Keywords Tree regression · Internet of things · Intelligent transportation systems · Complex event processing · Edge computing

1 Introduction

Presently, sensors can range from generic sensing devices attached to embedded IoT boards to in-built sensors present in the smartphone. Availability of low-cost hardware and sensing devices with a splurge in Internet connectivity has propelled the process of development of complex IoT applications in this area. The captured data from these embedded devices is voluminous in nature and can be analyzed for building intelligent applications in the domain of intelligent transportation systems (ITS) [6, 7], smart buildings [4], precision agriculture [1], etc. In ITS scenarios, data is voluminous and large scale in nature and thus poses a big data problem. ITS applications gather large data from multiple sources, and a need arises to process and extract patterns from these data streams. Depending upon the underlying application, the complex patterns can be transformed as complex events using a complex event processing (CEP) engine. CEP [5] involves data processing, fusion and generation of complex events from various univariate data streams for real-time congestion prediction in ITS scenarios. In this paper, we devise lightweight regression approaches for predicting future data points that can be amalgamated in CEP engine using event processing rules (EPL) to form complex events. These approaches are tuned by multi-bandit optimization for finding optimal hyper-parameters and furnishing predictions at a faster rate. However, the EPL rules trigger a complex event only when a threshold is exceeded. Generally, thresholds are provided by developers while developing the CEP engine in EPL rule base. But the disadvantage is these thresholds are set by human experts who may or may not have entire understanding of congestion scenario of smart city. Data drift also causes thresholds to shift, and they need to be updated as traffic signatures vary at different times of the day. For updating the thresholds in EPL queries, we develop dynamic clustering approaches.

Akbar et al. [8] present an IoT analytics framework that employed Apache Spark to predict congestions on data streams. The work depicts the implementation of a midpoint-based k-means clustering approach. In this approach, the clustering algorithm computes the midpoint between two centroids of two clusters as a decision boundary for computing thresholds. These thresholds are inputs to the CEP engine for event generation. Puschmann et al. [7] propose an adaptive k-means clustering approach on IoT data streams based on symbolic aggregate approximation (SAX) discretized features. The work is implemented on the traffic data of Aarhus city and suggested forming the density estimation of each feature. It further divided the area under data distributions of each feature into equal regions for computing its midpoints. These initial points served as initial centroids to the proposed clustering approach. Any change in the distribution is identified as a “turning point” which marks a change in the statistical properties in the data (primarily due to data drifts) that triggers the retraining of the algorithm for congestion prediction. Akbar et al. [2]

developed a predictive CEP that uses adaptive moving window regression (AMWR) to predict simple events. The regression approaches ingest data using an adaptive window based on the mean absolute percentage error (MAPE) that increases or decreases to contain the error within 5–20%. The predicted data points were sent to the CEP engine which fuses them to form complex "congestion" events well before their occurrence. However, it is found in the presence of noise and non-stationarity in data, and the performance of the model is poor. The authors [3] employed a Bayesian belief network (BBN) for identifying causal relationships between traffic data, weather data, temporal data, social data and their conditional probabilities by counting the number of instances of the particular attribute for congestion and dividing that by a total count.

The major contributions of this paper are as follows:

1. We have proposed a real-time IoT framework "HeuristiX" for identifying congestion on IoT data streams. We infer patterns for complex event detection using complex event processing engine, such as Apache Flink.
2. We have integrated Apache Flink with forecasting approaches to predict future congestion events and thus take steps to avoid it. In this work, we also overcome the "de facto" nature of CEP engines in which rules are triggered when data instances exceed thresholds specified by domain experts. As the prediction takes place on streams, we devise a genetic algorithm-based clustering for adaptive computation of thresholds in CEP queries.
3. IoT data streams suffer from the issue of "data drift." To handle drifts in IoT streams, we proposed adaptive windows that is less data intensive. The work proposes novel mechanisms to retrain the model when the predictions are erroneous owing to data drifts.
4. We have devised a fallback mechanism in which once the communication between the IoT node and server is lost, the models can be executed on the IoT board directly.

2 Proposed Framework

Figure 1 depicts the proposed framework to predict congestion in ITS scenario. The various modules and their respective functionalities are outlined in the following sections.

2.1 Data Acquisition

We use Node-RED in the front end to acquire the data from the data sources and push it to a NoSQL database called InfluxDB. The data is taken from the city of Madrid from September 2019 to November 2019. The data is acquired through a representational

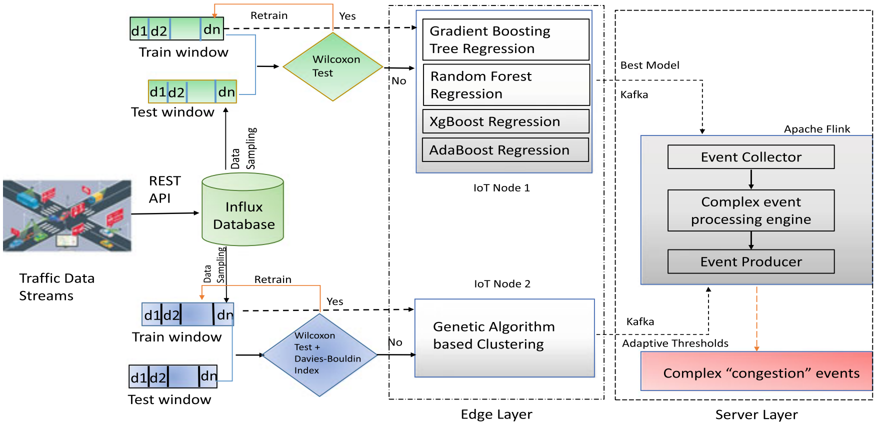


Fig. 1 Architecture of HeuristiX depicting adaptive CEP engine and augmented predictive analytics modules

state transfer (REST) application program interface (API).[1] The traffic administration of Madrid has installed various sensors to capture various traffic characteristics, such as traffic intensity, traffic velocity and occupancy. We collected the weather data of six regions from an API web service.[2] The weather information collected from the API consists of various attributes, such as temperature, precipitation, apparent temperature, dew point and humidity index. We also collected the Twitter data[3] to aggregate the count of tweets coming from a particular location. The data is collected in a interval of 1 min. The various traffic attributes with their respective descriptions are outlined in Table 1. The congestion is binary in nature, that is, 0 or 1 signifying no congestion and congestion, respectively. This ground truth is captured from a real-time traffic API, TomTom API[4] indicating congestion at a given time for a particular location.

The dataset comprises 24,000 data points, out of which 17,000 points were considered for training and 7000 points were considered for testing the model. The traffic data is captured from the REST API every 5 min. The weather data was downsampled to 5 min to be in tandem with the traffic data. The tweets are collected through Twitter API, and tweet counts represent the aggregated tweets every 5 min. The traffic, weather and social data is collected and stored in a time series database.

[1] http://informo.munimadrid.es/informo/tmadrid/pm.xml.

[2] https://api.darksky.net/forecast/.

[3] https://api.twitter.com/1.1/search/tweets.json.

[4] https://developer.tomtom.com/traffic-api.

Table 1 Traffic dataset

Attribute	Description
ID	Represents the identification number for various streets in Madrid
Timestamp	Represents the temporal parameters(time and date) for the data
Intensity	Signifies the total number of vehicles on the road at a particular location
Velocity	Represents the mean velocity of vehicles on the road at a particular time instant
Weather	Presents an estimation of the weather conditions of the particular location
Congestion	Ground truth signifying a binary outcome, i.e., 0 for no congestion and 1 for congestion

2.2 Machine Learning Approaches

In this work, we employ gradient boosting trees and random forests for forecasting on Madrid traffic data. Both the tree-based regressions are incremental in nature. Though both the approaches exhibit better performance,[5] there is a noticeable trade-off observed between training time and accuracy. In complex IoT applications, execution time is a vital evaluation criteria, and gradient boosting trees were found to be faster in comparison with random forest on training and testing times. For finding optimal hyper-parameters, i.e., number of lagged values, depth of trees, etc., we use HyperBand that uses an underlying bandit optimization policy in contrast to robust Bayesian optimization (RoBO). HyperBand enhances the process of discovering optimal hyper-parameters by utilizing a resource parameter to tune the model. A resource parameter is defined as a measure that can enhance the functionality of the approach by reducing the execution time of the approach. For example, the number of learners or trees in the regression approach can be modeled as a resource parameter that can tune the ensemble of trees and can further increase the overall predictive capability of the model. In this work, we employ symmetric mean absolute percentage error (SMAPE) for obtaining the set of optimal hyper-parameters in HyperBand. The additional functionality provided in this approach is its ability to run on multiple threads in order to reduce computation time which makes it more scalable.

2.3 Genetic Algorithms

Genetic algorithms (GA) belong to the class of evolutionary and swarm intelligence algorithms. It starts with a random population of chromosomes that contain probable solutions. GAs explore the “search space” in order to identify the most optimal solution for a concerned optimization problem. There is high probability that the solutions lying next to one another possess same features or characteristics. A sim-

[5] Experiments depicting training and testing times of random forest, XGBoost and AdaBoost were compared, and gradient boosting trees performed better but not depicted in this work.

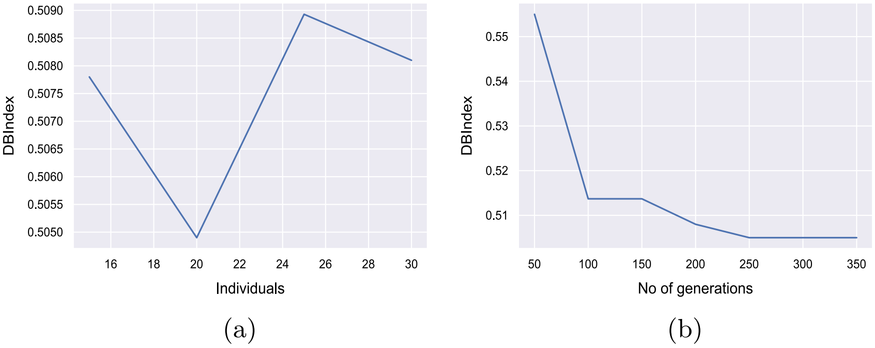


Fig. 2 **a** Plot depicting the number of individuals **b** Plot exhibiting the convergence of GA after 250 generations

ilarity metric is considered for searching the solutions in the "search space." The search continues through multiple generations in which new solutions are created through techniques like selection, crossover and mutation. A string of bits represents the solutions in a genetic algorithm. These strings are called as chromosomes.

We use Davies–Bouldin Index (DBI) for identifying the number of clusters possible on the Madrid dataset. The centroids of these clusters represent the thresholds for the EPL queries in the Flink. The chromosome in the proposed evolutionary approach represents the pair of centroids. The features are scaled using standard normalization procedures like min-max normalization. Figure 2 depicts the optimal number of individuals and generations taken by GA to converge to global optimum. Once the clustering is done, the obtained thresholds are denormalized to obtain actual thresholds. We use crossover and mutation to obtain fitter individuals. A set of individuals are selected from the generation, and then we select the chromosomes indexed by elements in the sample. We initialize an arbitrary number between 0 and 1 for gene. If random number is significantly less than mutation rate, the gene is replaced by another random number, and mutation is initiated. Through experiments, we chose 0.8 and 0.2 as crossover and mutation rates, respectively. The proposed fitness function in the GA clustering is represented as follows:

$$\text{Fitness Function} = \frac{1}{\text{DBI}} \tag{1}$$

The metric is defined as follows:

$$\text{DBI} = \frac{1}{\text{num}} \sum_{m=1}^{\text{num}} C_m \tag{2}$$

where

num = number of clusters,

C_m represents the individual clusters on the data.

Once the fitness function is computed, we can implement it to evaluate the fitness of chromosomes so that fitter individuals can participate in the clustering process. The GA clustering can be implemented using generic GA operations.

2.4 Complex Event Processing Engine

Complex event processing engines help in detection and matching of predefined event sequences in incoming and unbounded data streams. FlinkCEP is CEP library that is augmented with Apache Flink for easier identification of patterns in data streams. CEP engines run live data over stored queries. Data is acquired and preprocessed by Apache Kafka and subsequently sent to the CEP engine. We execute Kafka, a high-speed messaging pipeline and a CEP engine, Apache Flink, on the server on separate cores. Due to this, Kafka producer publishes the data on a specific topic, and a Kafka consumer can consume from the topic and send it to push it to Flink for event processing. As Kafka and Flink are on separate cores, the concurrent executions prevent starvation of the CEP engine. The distributed execution of Kafka and Flink on server allows each process to run freely without hampering the ML predictions.

The machine learning approaches presented in this work can work on the server with higher IoT workloads. However, we also integrated the feature of sliding window and multi-bandit optimization that allows the execution of the machine learning approaches in resource-constrained environments. We also integrate a fail-safe mechanism, in which a policy is created in case there is a failure in communication between server and IoT nodes. Mostly, the server takes the data and processes it and creates insights using machine learning approaches, and Flink uses the predictions to generate events. However, if there is a disruption of communication between client (Raspberry Pi) and server, then a recent copy of machine learning approaches in the node is used. In the edge node, we have created a CEP engine with same functionalities as Flink using Node-RED "CEP" modules. A checkpoint is done after five successful iterations of machine learning models execution on the server. The models are written in PMML format, and a copy of the models is created, thus deleting the older copies created by previous checkpoints. Once the connection is lost, the IoT node waits for a timeout of three minutes, and if it receives no message from server regarding its status, it functions as an independent module and runs the Node-RED CEP and machine learning approaches with windowing mechanisms as described in the succeeding section. This forms the fault-tolerant nature of CEP and also disallows any delays in congestion alerts in a real-time ITS scenario.

2.5 Sliding Window

In this work, we intend to predict congestion in the city of Madrid. For predicting congestion, we employ a CEP engine to infer patterns augmented with predictive analytics approaches. The machine learning approaches are not resilient to “data drift” which is a inherent characteristic of time series data. The dataset is also heterogeneous in nature as it contains traffic as well as weather data to predict congestion. To prevent the impact of data drift in IoT streams and also enhance the ability of the machine learning approaches to execute on IoT boards, we take the help of adaptive windowing mechanisms. The regression and clustering approaches considered in this work intake the data through a window containing 150 samples. We have developed a train window and a test window. The train window takes 150 data points, and the model is trained on these data instances. Once trained, the model executes till the statistical properties or data distributions in both windows are unchanged. For this purpose, we employed a Wilcoxon signed rank test on both windows. When the p-value falls below the threshold, the train window is emptied and the current test window becomes the new train window on which the models are retrained. The recent data points are ingested by a test window which are used as new test set.

2.6 Evaluation Metrics

In this paper, we assess the performance of the forecasting approach on the basis of SMAPE metric. SMAPE is given by

$$\text{SMAPE} = \sum_{t=1}^{T} \frac{|\hat{X} - x|}{|\hat{X}| + |x|} \tag{3}$$

where $\hat{X}$ and x represent the predicted and actual values of intensity or velocity at time instant t. A better performance in SMAPE is exhibited by a lesser value of the metric.

For evaluating the unsupervised learning approaches, we use the homogeneity score metric. Homogeneity score is an external cluster validation metric to assess whether the data instances in a particular class belong to a certain class or not. It is denoted by

$$hs = 1 - \frac{H(Y|\hat{Y})}{H(Y)} \tag{4}$$

The homogeneity score is bounded between 0 and 1, and low values represent lesser homogeneity and thus lesser resemblance with the comparative approaches. When the value of $\hat{Y}$ decreases the uncertainty associated with Y, the hs tends to 1 as $H(Y|\hat{Y})$ reduces significantly.

3 Experimental Results

Figure 3a depicts the execution time of gradient boosting tree on velocity attribute. The figures also exhibit the impact on time as we fuse different features together. We noticed that least time is taken when forecasting approach is trained on only velocity in execution times. Figure 3b depicts the performance comparison between gradient boosting trees and random forests. It is noticed that SMAPE value for gradient boosting trees is lower than random forests signifying better performance. The probable reason behind this is gradient boosting trees identify outcomes for each case based on the error gradient with respect to the prediction. The use of multi-bandit optimization not only helps in faster execution of the forecasting approaches of gradient boosting trees and RF but also helps in choosing the best model at any point of time.

Figure 4 depicts the behavior of the sliding window used in this approach. Each vertical line represents a retrain when a distribution change in the data was detected.

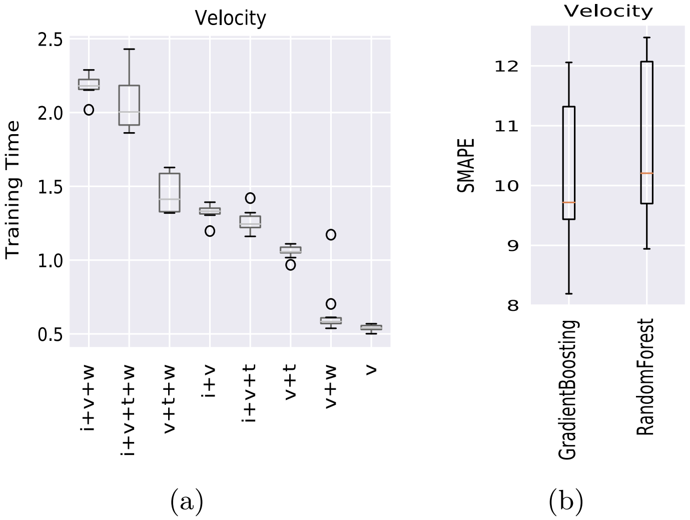


Fig. 3 **a** Execution time of gradient boosting tree on velocity. **b** Performance comparison between gradient boosting trees and random forests on velocity attribute

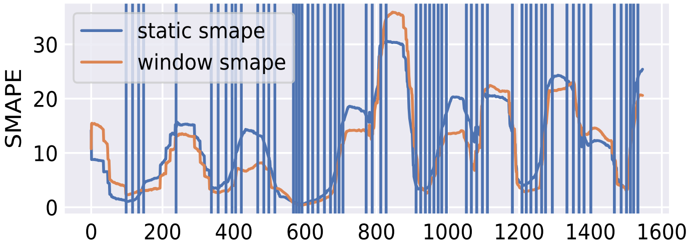


Fig. 4 Retraining gradient boosting trees on window of 132 samples using Wilcoxon test

HeuristX considers gradient boosting and random forest regression approaches for predicting future events on IoT data streams. As the prediction approaches execute on the IoT node, the best model should furnish predictions with minimum latency on less number of samples encompassed in a window. The length of the window is empirically set to contain 150 data instances both for the regression approach and clustering approach using least spectral square analysis method. The functionality of the sliding window is twofold. First, it speeds up the training and testing process. Secondly, data distributions in IoT are dynamic, and hence, taking snapshots of data prevents the model from getting trapped in concept drifts. To extract the data chunk for training the approach, we have used a rolling window approach.

Figure 5 depicts the correlation between DBI and accuracy of the clustering. We notice as DBI tends to 0.4, and we obtain a very high accuracy for the clustering. GA is extremely fast and can work on variable number of clusters. The adaptive thresholding allows us to update CEP rules or queries and hence leads to a significant reduction in the complexity of rules developed for event generation. This also supports that DBI is a good indicator of cluster health, and reduction in cluster health will definitely affect rule accuracy.

Table 2 represents the thresholds of intensity and velocity attribute for different times of the day and week (weekend/weekday). It is clearly noticed that there is a significant difference between the thresholds obtained for the weekdays and that

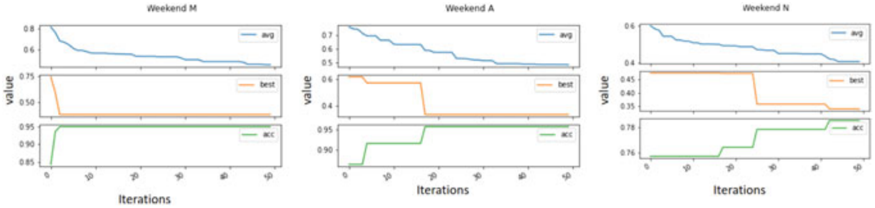


Fig. 5 Best and average performance of DBI as a fitness function in GA with accuracy in congestion predictions

Table 2 Computed thresholds for various times of the day and day of the week

		Weekday	Weekend
Features	Time of day	Thresholds (C1)	Thresholds (C2)
Intensity	Morning	4599	4687
Velocity	Morning	72.578	77
Intensity	Afternoon	4536	3465
Velocity	Afternoon	59	70
Intensity	Evening	4161	3607
Velocity	Evening	59.88	72
Intensity	Late evening	2854	2474
Velocity	Late evening	65	83

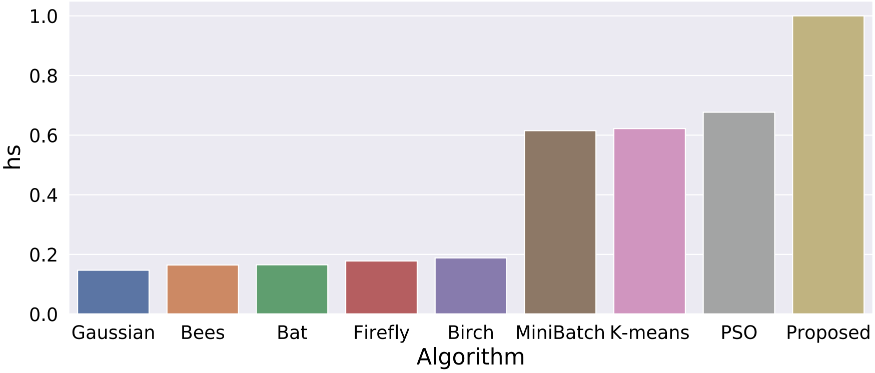


Fig. 6 Evaluation of proposed GA approach against other clustering approaches on the basis of homogeneity score

obtained for weekends. Morning time shows the highest thresholds indicating lighter traffic density on the road and significantly drops during peak hours of afternoon and evening and returns back to normalcy after late evening. This also strengthens our claim on the necessity to incorporate dynamic thresholds in CEP queries.

In this work, we analyze the cluster quality and the performance of the proposed approach against a multitude of state-of-the-art approaches on the basis of homogeneity score.[6] We evaluate the cluster quality and performance of GA clustering against state-of-the-art approaches on the basis of homogeneity score. Figure 6 depicts the results of homogeneity score of all approaches. It is observed the homogeneity is almost similar between firefly and BIRCH, artificial bee colony clustering and chaotic bat and minibatch k-means and midpoint-based k-means clustering approach [8]. GA performs much better than particle swarm optimization (PSO)-based approach by a huge margin and better than midpoint-based k-means approach too. We have compared our approach with the work [8], and it is noticed that our work performs better with lesser false positives. This also signifies that cluster quality for GA is better than baselines, and there is no overlapping in performance with other approaches. For validating the effectiveness of clustering approach, the validation metrics require labels (ground truth) which are fetched from TomTom API signifying congestion or no congestion.

In the proposed work, GA also takes the help of an adaptive window. GA is trained on data samples in train window and tested on recent data instances in the test window. We monitor the statistical distributions of train and test windows using a Wilcoxon test that oversees that the distributions do not change and on the basis of DBI that checks that clusters thus formed for computation of thresholds are dense and disjoint. Once the p-value of the statistical test drops below 0.05, the data instances in train window are discarded, and the current test window becomes the new train

[6] Similar outcomes were observed when the experiments were conducted using normalized mutual information, adjusted Rand index and Fowlkes–Mallows index that are not depicted in this work.

Table 3 Execution times of GA approach for threshold computation

RAM	No. of cores	Time
1 GB	4	38.8364
800 MB	3	42.3004
600 MB	3	45.54102
400 MB	2	55.90624
200 MB	1	71.54083

window. The model is now retrained on the new data points in the new train window and predicts on recent data points as they are ingested by the test window. In Madrid traffic scenario, data points are sampled at a five minute interval, thus giving adequate time for the retraining of the proposed approaches.

Table 3 exhibits the execution of the proposed GA clustering approach on varying cores and varying RAM capacities. The experiments were conducted on a Raspberry Pi model B, and the process of GA is executed on the quad cores of the IoT board. It is noticed that the time taken by GA is much less than the average time taken by the Madrid API, i.e., five minutes. Both the forecasting approaches and clustering approaches work under 5 min, thus making the models ideal candidates for execution on resource-constrained edge devices.

4 Conclusion

In this paper, we propose HeuristiX which is a fault-tolerant CEP framework for inferring patterns on IoT data streams. We use HeuristiX with gradient boosting trees to predict future congestion events and evolutionary approaches for creation of adaptive thresholds in CEP queries. We identify that predicting future events and adaptively changing the thresholds are limitations of current state-of-the-art CEP engines and have addressed these issues with our proposed approach. We depicted windowing approach for faster predictions and retraining of approach on encountering data drifts. We also exhibited the execution of HeuristiX on servers and its extended functionality on IoT nodes.

References

1. Ahmed N, De D, Hussain I (2018) Internet of things (IoT) for smart precision agriculture and farming in rural areas. IEEE Int of Things J 5(6):4890–4899
2. Akbar A, Khan A, Carrez F, Moessner K (2017) Predictive analytics for complex IoT data streams. IEEE Int Things J 4(5):1571–1582

3. Akbar A, Kousiouris G, Pervaiz H, Sancho J, Ta-Shma P, Carrez F, Moessner K (2018) Real-time probabilistic data fusion for large-scale IoT applications. IEEE Access 6:10015–10027
4. Candanedo LM, Feldheim V, Deramaix D (2017) Data driven prediction models of energy use of appliances in a low-energy house. Energy Build 140:81–97
5. Cugola G, Margara A (2012) Processing flows of information: from data stream to complex event processing. ACM Comput Surv (CSUR) 44(3):15
6. Kousiouris G, Akbar A, Sancho J, Ta-Shma P, Psychas A, Kyriazis D, Varvarigou T (2018) An integrated information lifecycle management framework for exploiting social network data to identify dynamic large crowd concentration events in smart cities applications. Future Gen Comput Syst 78:516–530
7. Puschmann D, Barnaghi P, Tafazolli R (2016) Adaptive clustering for dynamic IoT data streams. IEEE Int Things J 4(1):64–74
8. Ta-Shma P, Akbar A, Gerson-Golan G, Hadash G, Carrez F, Moessner K (2017) An ingestion and analytics architecture for IoT applied to smart city use cases. IEEE Int Things J 5(2):765–774

A Hybrid Clustering Approach for Faster Propagation of Emergency Messages in VANET

Puja Padiya, Amarsinh Vidhate, and Ramesh Vasappanavara

Abstract Road accidents can be considerably reduced by employing local warning systems across wireless vehicular communications. Vehicular Ad-hoc Networks (VANET) is one of the breakthroughs to enhance vehicle and road safety, traffic adeptness, and convenience to both drivers and passengers. Considering the high mobility model, dynamic topology changing at a rapid speed requires a real-time message propagation system that delivers a message in emergencies. The proposed approach provides reliability in terms of guaranteed delivery of messages to hidden vehicles, faster propagation of emergency messages during emergencies and reduce network load, which will further help to reduce accidents and future impact of accidents.

Keywords Accident · Vehicle · Emergency · Hidden · Guaranteed · Messages

1 Introduction

Road deaths on the Indian highways are the main reasons of death. In India, one person is killed every 4 min, according to media statistics, in a road accident. The causes of traffic injuries include unsafe road travel, unattended drivers and faulty bridge and building construction, inadequate regulation, and the lack of speedy trauma. As outlined in Fig. 1, in 2018, 4,69,418 road accidents caused 1,51,417 deaths, and injury to 4,67,044 persons were recorded by the Indian Minister for Road Transport in India [1].

Head-on collision, hit from the back, hit from the side are some of the collision types mentioned in the report. World Health Organization (WHO) states that approximately 1.3 million deaths worldwide are caused due to road accidents. Road death

P. Padiya (✉)
Ramrao Adik Institute of Technology, Navi Mumbai, India

A. Vidhate
Ramrao Adik Institute of Technology, Navi Mumbai, India
e-mail: amar.vidhate@rait.ac.in

R. Vasappanavara
Vasavi College of Engineering, Hyderabad, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_14

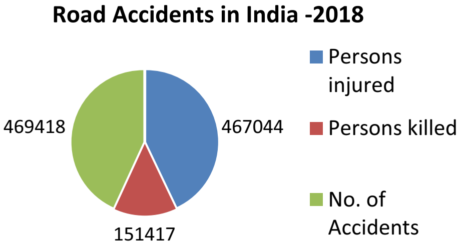


Fig. 1 Road accidents in India—2018

is expected to be the third-leading reason for death if precautionary measures are not taken. By implementing local warning systems over vehicular communications [2], road accidents can be considerably lowered. Example: Passing vehicles can notify other vehicles that they expect to proceed the highway and vehicles arriving at intersections can send caution messages to other vehicles crossing that intersection by means of vehicular ad-hoc communication.

VANET is a platform that is developing. VANET is one of the special types of ad hoc mobile networks established as required among moving vehicles. Like MANET, VANET vehicles can organize themselves, provide multi-hop communications, and operate without a fixed infrastructure. Contrary to MANET, vehicles have different requirements, have higher velocities (0–40 m/s), and require low latency rates. Vehicles do not have battery and storage constraints. Vehicular applications [3] as classified are (1) active road safety, (2) traffic efficiency and management, (3) infotainment applications. Vehicular network system components comprise the on-board unit (OBU), road side unit (RSU), and application unit (AU) for communication. The standardization used in VANET is DSRC, IEEE 802.11p, and WAVE protocol stack [4]. Propagation of message in VANET considers position, speed, direction, the association of vehicles.

The focus of the paper [5] relates to the quality of Service of VANET in event of failure in communication primarily with roadside unit (RSU).

For safety and traffic efficiency, message propagation is critical. It will also promulgate alerts, to avoid more confusion, with a minimum duration, if the vehicle senses a collision, lane shifts, etc. Vehicle communication involves sending information from one automobile to another for information about the incident. The decision on the way messages are communicated between vehicles according to specific criteria or techniques is called communication methods. Path work has measured ways of selecting vehicles (vehicles) along a multi-hop path [6].

Specific routing protocols [7] for the delivery of messages in VANET are categorized according to topology, initiator location, geocast, cluster-based, broadcast, and platoons. Delays must be as minimum as possible because communications are to be transmitted quicker. We agree that delays are virtually equal for topology, position-based, and geocast-based message propagation strategies and rely only on the number of intermediate vehicles from source to destination. These protocols often do not fix the hidden vehicles or a colliding inability to relay. The proposal addresses the problems mentioned and ensures that the message dissemination is reliable in terms of guaranteed message transmission to hidden vehicles and reduce network load.

The rest of the paper is structured as follows: a brief of related work is in Sect. 2. In Sect. 3, the proposed methodology is described in detail. Section 4 appraises the performance of our protocol and Sect. 5 concludes.

2 Related Work

As in [3, 8], the DSRC has been established to facilitate connectivity among vehicle to vehicle and vehicle-to-infrastructure. Within the 5 GHz frequency band, DSRC is defined with a minimum 75 MHz bandwidth (between 5.850 GHz and 5.925 GHz) [9–11]. The band consists of seven channels each of 10 MHz. The networks are split into six service channels and one control channel. CCH is the control channel that is utilized for the dissemination of network administration messages (resource allocation, topology administration) and high urgency/priority messages (crucial messages concerning to road safety). The other six channels, SCHs, are reserved for data transmission of different services. IEEE 802.11p is a VANETs IEEE Framework Standard [3, 4]. Enhanced Distributed Channel Access (EDCA) is used for enhancing service efficiency. Every single synchronization interval is subdivided into 50 ms CCHI and SCHI intervals of 50 ms as seen in repeated synchronization intervals of 100 ms. The CCHI is used to send/receive safety messages of high importance or to declare a service to be delivered on a certain communication channel. The space for the Inter Frame (IFS) is an idle medium after transmission and pre-requisite for 802.11 protocols [12]. An IFS aims to provide both an interference-preventable buffer, control, and frame transmission prioritization. Different forms of IFS exist (Fig. 2): SIFS, DIFS, PIFS, and AIFS.

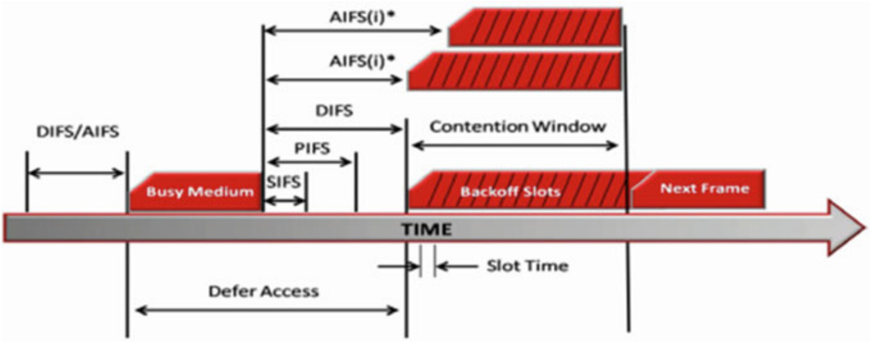


Fig. 2 InterFrame space comparison [4]

Inter-Frame Space (SIFS) Short space is utilized for RTS/CTS and high-priority transmission based on positive ack. The transmission will begin immediately after each SIFS. PCF Inter-Frame Space (PIFS) is used for transmitting containment-free data using the Point Coordination Function (PCF). During this time, all conflict-based operations will be stopped. Inter-Frame Space (DIFS) from DCF is the Distributed Containment Function (DCF) based on which the services/applications dependent on the contention are used. Network devices will automatically access the media if they are free to run longer than the DIFS value. The AIFS is used to transmit all data frames, management frames, and control frames, e.g. PS-Poll, RTS, CTS.

2.1 Cluster-Based Protocol

VANET clustering means grouping vehicles in groups based on certain guidelines, requirements, or common features. Clustering has many benefits, such as maximizing the use of bands, efficient distribution of resources; reducing overhead communication, fast data packet transmission, and low latency [13]. To boost the reliability of transmitters of safety messages, Yang and Tang [14] present VANET Cooperative Clustering-based MAC (CCB-MAC) for VANET. The average CCB-MAC packet delay is higher because of the extra time taken for ineffective vehicles to be forwarded by helpers. Cooperation increases the package arrival rate at the point where the package delay has risen. In the emergency broadcasting of a cluster-based alert by Ramakrishnan et al. [15], the source vehicle for the identification of dangerous incidents, the broadcasts request to transmit the RTB packet by the collision prevention factor. In a highway algorithm based on stability clusters on highway scenarios, [16] implemented Clustering Enhancement for VANET. This algorithm eliminates problems with disconnection. For the option of a cluster Head in VANET, Bhosale and Vidhate [17] suggested an agglomerative approach to improving VANET quality of service parameters. In applying the principle of graph theory and the K-median algorithm, Khan and Fan [18] proposed a Triple Cluster dependent Routing Protocol, new cluster creation, and requirements for head selection. To pick unified and stable VH, the Floyd-Warshall algorithm is used. DHCV, a D-hop clustering algorithm by Azizian et al. [19], organizes vehicles into uncomplicated clusters with flexible dimensions based on their mobility. A propensity to re-elect the existing cluster heads until the network structure changes is one of the features of this algorithm. ACO technology is used for cluster head selection for Abbas and Fan [20], efficient, low-latency routing-based clustering. The reliability requirements for the cluster head are considered using heuristic techniques. This plan aims to find the best possible way forward and eliminates end-to-end latency at high energy consumption costs. Some authors have suggested AIFS medium duration in cluster-based routing and others have proposed DIFS medium duration for transmission and channel access. The study of the output of a VANET-cluster, an analytical model to appraise the execution of a clustered VANET is suggested by Pal et al. [21].

2.2 Broadcast-Based Protocol

Routing protocols based on broadcasting adopt a basic streaming approach where vehicles retransmit the message to other vehicles. Maia et al.'s [22] Hybrid Data Diffusion Protocol is a protocol for the dissemination of data in highway scenarios that incorporate both the sender and the recipient approaches for addressing the question of the broadcast storm under various traffic scenarios and storage transport technology for disconnected networks.

QASA's methodology, which is mainly a vehicle selection transmission algorithm for message dissemination in opportunistic vehicular networks, was suggested in QoS-aware Node Selection Algorithm, by Mostafa et al. [12]. QASA permits vehicles on one side to select a vehicle on the other side of the highway that enhances QoS, a Novel Protocol Paradigm for improving vehicle network reliability ad hoc metrics. The downside to creating a new link is the overhead impact of the beaconing of messages and back-off algorithms. Wang et al.'s [23] Content-Centric Routing Protocol is a new Named Data Network (NDN) routing protocol for VANET. Incremental transmission and adaptive transmitting approaches according to vehicle density are used by the content-centric routing protocol and are requestor initiated. Not acceptable when sending safety messages. In the sense of effective polling broadcasting, Nguyen-Minh et al. [24] propose a polling scheme that will identify missing vehicles and ask them to broadcast them again. Each other vehicle node is a temporary transmitter and a lot of processing is needed inside the vehicle node. This method delays the transmission of the message by the sender. Rai et al. overcome this constraint [25]. Instead of the sender itself, the author proposed that forwarders in the spectrum of polled vehicles transmit the message. A simultaneous broadcast protocol based on transmission suggested Zhang et al. [26] using time sync to forward packets to minimize latency.

3 Research Proposal

Regarding the need for an integrated vehicle-to-vehicle safety and traffic system, we have been working on two entities to ensure effective road-safety message dissemination: message diffusion approach for loading and faster dissemination of network and assured emergency alert transmission to hidden vehicles, to avoid further chaos. We suggested that the transmission of messages based on a cluster and broadcast method be merged. The cluster-based message propagation strategy can be used in normal circumstances to lower the number of overlapping network messages. However, the vehicle would be able to broadcast messages to other vehicles for faster transmission during an emergency.

Vehicles follow a cluster-based strategy with the cluster head in charge of the cluster during normal operation. Cluster creation would only allow the head of the cluster to relay messages to all cluster members reducing network loads. But what would happen if the cluster head vehicle leaves the cluster unexpectedly? Whenever the cluster head leaves the network, no controller monitors the cluster. This disrupts the cluster and affects its performance because the steadiness of the entire cluster depends completely on the cluster head vehicle. In the conventional method, when a cluster head is not present, the cluster members start re-electing a new cluster head in these unexpected circumstances, which adds a lot of messages to the exchange, thus increasing network load and adding delay when the cluster head is being chosen. This also contributes to information loss with the former head of the cluster.

Our proposed method allows the cluster to continue without disruptions by constructive co-head appointment mechanisms that choose a vehicle as a co-head as the entire cluster's stability depends entirely on the cluster head vehicle. A pragmatic co-head approach for selecting all vehicles is based on a ranking. The ranking is calculated based on the location of a vehicle within the cluster, vehicle velocity, and distance between its cluster head and the vehicle's specific cluster position. One of the three variables above is independent; the stability of the cluster is determined by them. The calculation of a certain vehicle rank during the creation of the cluster is determined based on the priority of the above variables. In this strategy first, we need to calculate the average velocity V_{avg} [22] as in Eq. 1, of all the vehicles which can form a cluster.

$$V_{\text{avg}} = \frac{\sum_{i \in CM} V_{i\,(\text{current})}}{n} \tag{1}$$

where,

$V_{i(\text{current})}$ = Current velocity of i vehicle

CM = vehicle as cluster member

n = Number of vehicles in the cluster

V_{avg} = Average velocity.

Then each vehicle calculates the difference of its velocity from the average velocity, V_i as in Eq. 2.

$$V_i = \left|V_{i\,(\text{current})} - V_{\text{avg}}\right| \tag{2}$$

Average distance d_{avg} of each vehicle is calculated as in Eq. 3,

$$d_{\text{avg}} = \frac{\sum_{i \in CM} d_{i\,(\text{current})}}{n} \tag{3}$$

where, d_{avg} = average distance of all cluster members w.r.t cluster head.

$d_{i\,(\text{currenet})}$ = distance of vehicle i from cluster head.

Then we calculate the distance variance d_{i} from average distance d_{i} as in Eq. 4.

$$d_i = \left|d_{i(current)} - d_{avg}\right| \tag{4}$$

The smaller the value of V_{i}, the nearer is the speed of the vehicle *i* to its cluster head's average speed. The bigger the $\mathrm{d_i}$, the more forward or a backward vehicle is. Finally, based on the above-calculated value, each vehicle decides its priority value, *p*, as follows:

$$p_i = d_i * e^{-\alpha V_i} \text{where } 0 < \alpha <= 1 \tag{5}$$

The highest *p*-value vehicle has been granted top priority as rank 1 and is also being named as co-head of the cluster as it is closer to the current head and will be in a cluster for a longer period with a smaller velocity gap. The head of the cluster is updating co-leader. The co-leader immediately takes up control of the cluster if the cluster head travels outside the cluster and acts as a cluster manager. With the same formula, the new cluster leader will be pro-actively named and co-head changed. It brings down the number of messages needed to re-elect the head of cluster and permits the cluster unchanged in a stable situation.

In the event of an emergency, the identified vehicle would relay messages to others for quicker dissemination. The source transmits the Broadcast Confirmation (BC) after a safety message. The vehicles when receive both the safety message and the BC ignore re-broadcast messages. The polling method is used by the vehicles that receive the BC message but do not detect the safety message. The source vehicle re-broadcast the message to the hidden vehicle in the conventional method as indicated in Fig. 3a. The retransmission is performed by the forwarder vehicle in our approach instead of the source vehicle, as shown in Fig. 3b, which improves the chances of receiving a message by the requester. It also helps to disperse messages more easily to other cars.

The communication among the source and the recipient vehicles is given with the time slots. These time slots are split based on channel propagation delay. The time slots are further split into minislots, for collision-free communication. A BC is broadcasted at random mini slot within DIFS immediately after data transmission.

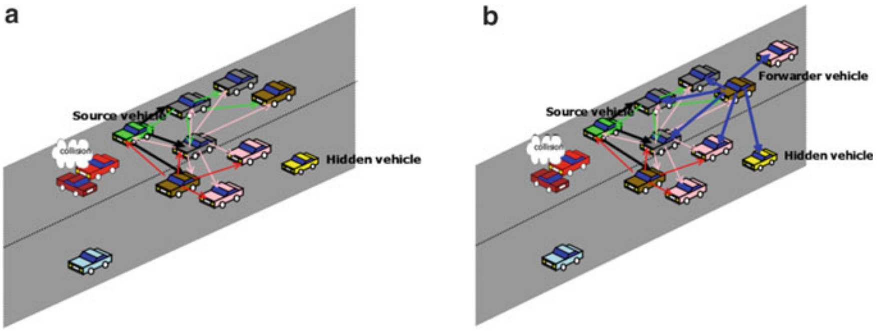


Fig. 3 **a** Re-broadcast by source vehicle. **b** Re-broadcast by forwarder vehicle

A non-data receiver vehicle, after hearing a BC, sends a Poll message requesting the source vehicle for retransmission of the previous data packet. It sends the poll response after the last busy medium in a DIFS after a random number of T minislots due to the BC transmission *R*. Other data recipients who correctly overhear a vote will freeze their polling and wait for future data. After one SIFS length, these vehicles can submit their polls for several attempts to request or collect their data packet if there is no data packet. Any vehicle receiving BC will postpone one DIFS plus one poll for the length of the BC Length. Forwarder after hears a poll message will wait for SIFS, and immediately start re-broadcast of a message. Message re-broadcast by forwarder helps faster propagation of message to the hidden vehicle as well as to the other vehicles on the road that are not in the range of the source vehicle. It thereby helps to reduce network load.

In the next section, the results of the proposed approach are evaluated.

4 Results and Analysis

Simulation experiments test the feasibility of the proposed procedure. A vehicle network simulation called VEINS is used to conduct these simulation experiments [22]. VEINS (Vehicles in the Network Simulation) is a network simulator construct that is built on the OMNeT++ environment. The OMNeT++ kernel is used for the event-based simulation of incidents, i.e. OMNeT++ does all the simulation management and data processing. Veins also integrate instant SUMO and have a flexible platform to simulate personalized applications driving cars. With 200 cars each has a high degree of mobility pattern, a VANET scenario is developed. Random deployment was selected for this ad hoc vehicle network, taking into account versatility of vehicles. Table 1 lists the simulation parameters for simulation.

During the cluster-based approach, in the absence of the cluster head, the co-head automatically takes the charge reducing the messages to elect a new cluster head. This reduces the network load by avoiding messages to re-elect a new cluster head. Also, the stability of the cluster is maintained for a longer duration of time. The stability of the cluster is increased by 33% as indicated in Fig. 4.

During an emergency such as an accident, the broadcast-based approach is initiated by source vehicle and it immediately broadcasts the message that normally

Table 1 Simulation parameters

Parameter	Values
Network simulator	OMNet++
Traffic simulator	SUMO
Transmission range	500
MAC protocol	802.11p
Simulation time	700 s

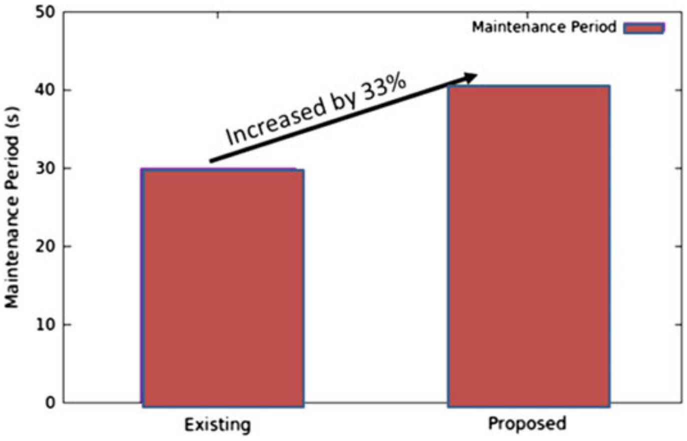


Fig. 4 The nomination of co-head increases the stability of the cluster

requires up to 150 microseconds. Then it sends BC and waits till a hidden vehicle sends a poll. As indicated in Fig. 5, in the traditional approach, the source vehicle waits for SIFS, which is equal to 32 microseconds and re-broadcasts the message.

Until this time, the forwarder vehicle is unable to broadcast the message and has to wait for one more slot of SIFS after the re-broadcast is complete. This incurs an additional delay of 182 microseconds for transmission by forwarder vehicle to other vehicles.

Figure 6b imitates the proposed model, where the forwarder vehicle re-transmits the safety message. This will help faster dissemination of emergency messages to hidden vehicles, and to other vehicles that are out of range of source vehicle. The messages are propagated faster in time in the proposed method and it also helps to reduce network load by avoiding repeated re-transmission by source vehicle.

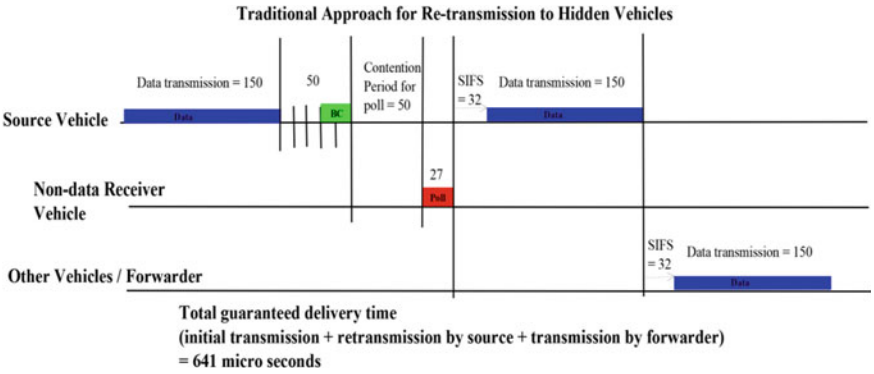

Traditional Approach for Re-transmission to Hidden Vehicles
Data transmission = 150
50
Contention Period for poll = 50
SIFS = 32
Data transmission = 150
Source Vehicle
Data
BC
Data
27
Poll
Non-data Receiver Vehicle
SIFS = 32
Data transmission = 150
Other Vehicles / Forwarder
Data
Total guaranteed delivery time
(initial transmission + retransmission by source + transmission by forwarder)
= 641 micro seconds

Fig. 5 Re-transmission by source vehicle

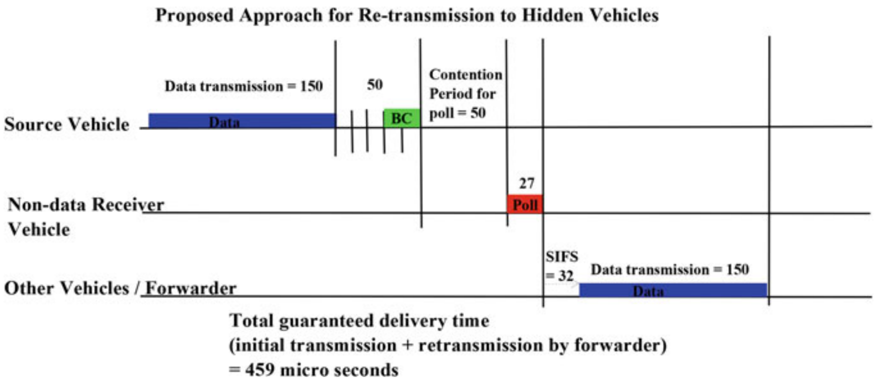

Proposed Approach for Re-transmission to Hidden Vehicles
Data transmission = 150
50
Contention Period for poll = 50
Source Vehicle
Data
BC
27
Poll
Non-data Receiver Vehicle
SIFS = 32
Data transmission = 150
Data
Other Vehicles / Forwarder
Total guaranteed delivery time (initial transmission + retransmission by forwarder) = 459 micro seconds

Fig. 6 Re-transmission by forwarder vehicle

With the proposed method, the transmission of an emergency message to other vehicles that are beyond the range of source vehicles is reduced by 28% as shown in Fig. 7, thus accomplishing guaranteed delivery of messages to hidden vehicles, with faster dissemination of emergency messages.

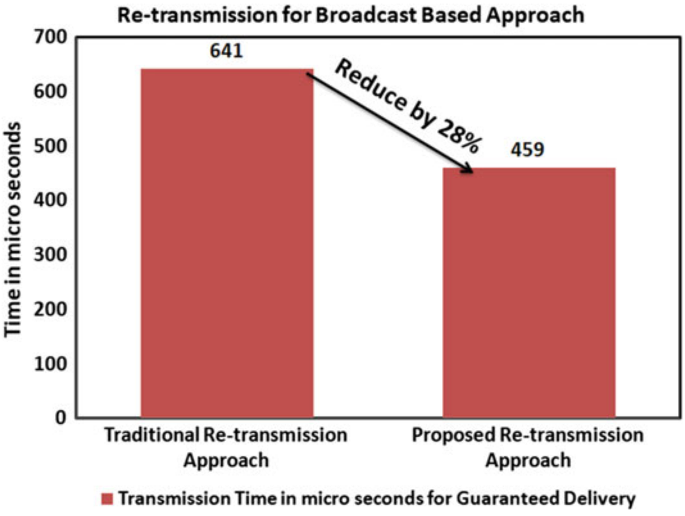


Fig. 7 Re-transmission for broadcast-based approach

5 Conclusion

A proposed hybrid approach contributes to reliable message propagation in terms of guaranteed delivery of messages to hidden vehicles, faster dissemination of message in emergencies, which reduces the time by 28% and improves cluster stability by 33% by pro-actively nominating co-head of cluster thus reducing network load, which, in turn, will help to reduce accidents and impact of accidents thus saving many lives.

References

1. Global status report on road safety 2018 (2018) World Health Organization. License: CC BYNC-SA 3.0 IGO. Geneva
2. Global status report on road safety 2018 (2018) Geneva: World Health Organization. License: CC BYNC-SA 3.0 IGO
3. Morgan YL (2010) Notes on DSRC and WAVE standards suite: its architecture, design, and characteristics. J IEEE Commun Surv Tutorials 12(4):504–518
4. Campolo C, Molinaro A, Scopigno R (2015) Vehicular ad hoc networks. New York Dordrecht London
5. Raut CM, Devane SR (2017) Intelligent transportation system for smartcity using VANET. In: 2017 International conference on communication and signal processing (ICCSP), Chennai, pp 1602–1605. https://doi.org/10.1109/ICCSP.2017.8286659
6. Karagiannis G, Altintas O, Ekici E, Heijenk G, Jarupan B, Lin K, Weil T (2011) Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards and solution. J IEEE Commun Surv Tutorials 13(4):584–616
7. Singha PK, Nandia SK, Nandi S (2019) A tutorial survey on a vehicular communication state of the art, and future research directions. Veh Commun 18(2019)100164
8. Zeadally S, Hunt R, Chen YS, Irwin A, Hassan A (2017) Vehicular ad hoc networks (VANETS): status, results, and challenges. J Telecommun Syst 50(4):217–241. https://doi.org/10.1007/s11235-010-9400-5 (Accessed 23 Mar 2017)
9. Hadded M, Muhlethaler P, Laouiti A, Zagrouba R, Saidane LA (2015) TDMA-based MAC protocols for vehicular ad hoc networks: a survey, qualitative analysis, and open research issues. J IEEE Commun Surv Tutorials 17(4): 2461–2492
10. Carpenter SE (2017) Inter-vehicle commu-nications (IVC): current standards and supporting organizations. Last accessed online 23 Mar 2017
11. IEEE (2013) IEEE Guide for wireless access in vehicular environments (WAVE) Architecture. In: Intelligent transportation systems committee of the IEEE vehicular technology society. IEEE-SA standards board
12. Mostafaa A, Vegnib AM, Bandaranayakea A, Agrawal DP (2014) QoS-aware Node Selection Algorithm for Routing Protocols in VANETs. In: Paper presented at fourth international conference on selected topics in mobile & wireless networking. Roma, Itália, Sept 2014. Published in J Procedia Comput Sci 40:66–73
13. Fan P, Haran JG, Dillenburg J, Nelson PC (2005) Cluster-based framework in vehicular ad-hoc networks. In: Syrotiuk VR, Chávez E (eds) Ad-Hoc, mobile, and wireless networks. Lecture Notes in Computer Science, vol 3738. Springer, Berlin, Heidelberg
14. Yang F, Tang Y (2014) Cooperative clustering-based medium access control for broadcasting in vehicular ad-hoc networks. J IET Commun 8(17):3136–3144
15. Ramakrishnan B, Bhagavath Nishanth R, Milton Joe M, Selvi M (2015) Cluster-based emergency message broadcasting technique for vehicular ad hoc network. J Wirel Networks 23(1):233–248

16. Daknou E, Thaalbi M, Tabbane N (2015) Clustering Enhancement for VANETs in Highway scenarios. In: Paper presented at the 5th international conference on communications and networking, Tunis, Tunisia, 4–7 Nov 2015, pp 1–5
17. Bhosale P, Vidhate A (2016) An agglomerative approach to elect the cluster head in VANET. In: Paper published at the 2016 international conference on signal processing, communication, power and embedded system, Paralakhemundi, India, 3–5 Oct 2016, pp 1340–1344
18. Khan Z, Fan P (2016) A novel triple cluster-based routing protocol (TCRP) for VANETs. In: Paper published at the 83rd vehicular technology conference, Nanjing, China, 15–18 May 2016, pp 1–5
19. Azizian M, Cherkaoui S, Hafid AS A Distributed D-hop Cluster Formation for VANET. In: Paper published at the 2016 IEEE wireless communications and networking conference, Doha, Qatar, 3–6 Apr 2016, pp 1–6
20. Abbas F, Fan P (2018) Clustering-based reliable low-latency routing scheme using ACO method for vehicular networks. J Veh Commun 12:66–74
21. Pal R, Prakash A, Tripathi R, Singh D (2018) Analytical model for clustered vehicular ad hoc network analysis. J Korean Inst Commun Inf Sci (KICS) 4(3):160–164
22. Maia G, Aquino ALL, Viana A, Boukerche A, Loureiro AAF (2012) HyDi: a hybrid data dissemination protocol for highway scenarios in vehicular ad hoc networks. In: Paper published in the proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications. New York, NY, USA, pp 115–122, 21–22 Oct 2012
23. Wang X, Liu W, Yang L, Zhang W, Peng C (2016) A new content-centric routing protocol for vehicular ad hoc networks. In: Paper presented at The 22nd Asia-Pacific conference on communications. Yogyakarta, Indonesia, pp 552–558, 25–27 Aug 2016
24. Nguyen-Minh H, Benslimane A, Deng D-J (2016) Reliable broadcasting using polling scheme based receiver for safety applications in vehicular networks. J Veh Commun 4:1–14
25. Rai Y, Vasappanavara R, Padiya P (2017) Re-transmission strategy for broadcasting in VANETs. In: Paper published at the 2017 International Conference on Circuits, Controls, and communications. Bangalore, India, 15–16 Dec 2017, pp 171–174
26. Zhang XM, Yan L, Zhang H, Sung DK (2019) A concurrent transmission based broadcast scheme for urban VANETs. J IEEE Trans Mob Comput 18(1):1–12

Machine Learning and IoT-Based Messaging Device for Blind, Deaf, and Dumb People

Vishnu Vardhan Nimmalapudi, Mohammad Farukh Hashmi, and Avinash G. Keskar

Abstract Recent advances in science and technology have improved the standard of living of human beings and made the life comfortable. But the life of people with visual, speech, and hearing impairments is still uncomfortable in terms of communication. They do not have a proper/simple messaging device that suits their standard of living. The devices that are currently available for them to use have certain limitations in aspects like the cost, simplicity, and limited to vicinal communication. According to reports from World Health Organization, there are about 300 million people who are deaf, 1million who are dumb and 285 million who are blind. This paper describes the device prototype that was designed through which people with visual, speech, and hearing impairments can send messages easily based on their hand gestures. An appropriate machine learning model is used for this task of decoding the meaning of hand gesture that the deaf or dumb person is using. The model was trained on the prepared dataset which consists of the hand gesture images and the corresponding labels. The message decoded by this machine learning model from the hand gestures is sent as mail to the designated person using Amazon web services (AWS), simple email service (SES). This device is mostly suitable for deaf and dumb people compared to blind people in terms of ease to use it.

Keywords Machine learning · IoT · AWS · Hand gestures · VGG16 · Raspberry Pi

V. V. Nimmalapudi (✉) · M. F. Hashmi
National Institute of Technology Warangal, Warangal 506004, India
e-mail: nvishnu@nitw.ac.in

M. F. Hashmi
e-mail: mdfarukh@nitw.ac.in

A. G. Keskar
Visvesvaraya National Institute of Technology, Nagpur 440010, India
e-mail: agkeskar@ece.vnit.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_15

1 Introduction

From the time humans evolved on this planet circa (6–2 million years ago), there has been a constant desire to express emotions and ideas among them. Human communication manifests in many forms. Speech, body language, facial expressions, hand gestures, and writing are some of the examples to quote. This form of communication, i.e., through speech, has marginalized the unfortunate ones who have lost the ability to speak or listen. Sign language based on hand gestures is the means by which the deaf and dumb people communicate with the people in front of them. But when these people want to communicate with a person who is far away from them, the mode of communication through hand gestures is also not possible. This paper presents an intelligent device that can understand the hand gestures and transfers the corresponding messages to the respected person who is far away. With the growing presence of science and technology in our lives, there lies a huge responsibility to empower these deprived people with the power of speech. A prefixed camera on the device records the input hand gesture and gives output to the person on the other end as a written message on the screen. Instead of using each alphabet to each hand gesture and building the sentences, the prototype has been with an emergency message for each hand gesture to decrease the time for building the sentence using every alphabet. The type of message to be assigned for a specific hand gesture can be designed according to the person's wish, and many number of messages can be assigned depending on the memorizing capacity of the person. The use of machine learning along with Internet of things (IoT) enables us to build this device on a Raspberry Pi module.

Machine learning [1] is a part of artificial intelligence (AI) that aims at understanding the structure of data and fits that data into models that can be utilized by the people. It comes in different forms, based on the algorithm and its objectives. Machine learning algorithms can be divided into three main categories based on the type of problems; it can solve as supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is a learning in which the machine is trained using data which is well-labeled. This is mainly divided into two categories based on the tasks it can solve. They are regression and classification tasks. In unsupervised learning, the machine is trained using information that is not classified or labeled and allows the algorithm to learn from that information without guidance. In reinforcement learning, the algorithm is presented with examples that lack labels, as in unsupervised learning. Classification mode of supervised learning is used to solve this task. A pre-trained Visual Geometry Group16 (VGG16) deep neural network is used and is fine-tuned for few of its last layers to make it work for this task.

On the other hand, IoT refers to a system of physical devices that can transfer and receive the data over wireless networks without human intervention. This is made possible by integrating simple computing devices like esp32 and Raspberry Pi with sensors like DHT11 and passive infrared (PIR) motion sensor. AWS is one of the cloud-based platform for performing IoT applications. AWS SES service is used for this project. Proposed system (Sect. 4) describes in detailed explanation of the functioning of the device.

2 Literature Review

In recent years, there have been several systems proposed for the communication of the people with visual, speech, and hearing impairments, through hand gestures which can be basically divided into two categories. The first is based on the use of hand data gloves [2–8], and the second is based on capturing the hand gesture image through a camera and then processing the image [9–12]. Chandana et al. [10] proposed a system that can detect the hand gestures from a laptop webcam and preprocess the image captured to get the corresponding hand gestures ID. Red and green tapes are wrapped around the fingers for hand gesture recognition. This ID is sent to microcontroller and is displayed on the ALCD and speaker outputs voice of the corresponding command. Abhishek et al. [9] present a sign to speech converter for dumb people. A webcam is placed in front of the differently abled person, and it will capture the hand gesture of that person and performs the image processing operations using principal component analysis algorithm (PCA) algorithm. The hand gesture coordinates are mapped with the picture from the database. Sunita et al. [2] used a data glove to capture hand gestures of a user. For this purpose, flex sensors are embedded on the data glove along the length of each finger. These flex sensors give the output data based on the change in resistance which in turn depends on the angle of bend. This resulting output signal is converted into digital signal using a microcontroller like ARM7TDMI and is then fed to gesture recognition section and the corresponding text information is identified. This text is converted into speech through a speaker.

The systems existing currently are used just for vicinal communication for blind or deaf/dumb people. As described early, for the purpose of recognition of hand gestures, mostly data gloves are used currently in market [2]. But it is not cost-effective. The approach followed by [10] for hand gesture recognition needs tapes to be wrapped around fingers every time message need to be sent. This is one of the limitations. Although if these limitations are neglected, these systems are only used for vicinal communication (when both subject and receiving person are nearby). They are not used for long-distance communication, in the case when receiver is far away from the sender. For example, if a differently abled person wants to send a message to his caretaker who went to the market, it is not possible by using the above-mentioned existing systems. Recently, there are some applications introduced in the smart phone for these differently abled people to send messages. But since only 10

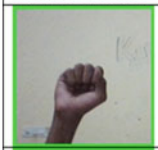
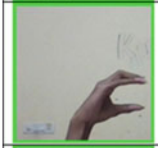
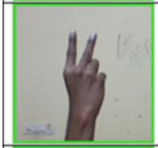
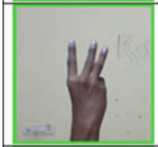
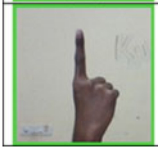
So existing systems have limitations like cost-ineffective, requires literacy, requires colors tapes wrapped around fingers for effective recognition of hand gestures, used only for vicinal communication. So in order to overcome the above limitations and make the life of these differently abled people comfortable in terms of communication, a device prototype was designed which is able to identify hand gestures with much better accuracy without using tapes and gloves and can be used for long-distance communication.

3 Methods and Materials

3.1 Dataset Preparation

Since the device uses 25 megapixel webcam for recognizing the hand gestures, to improve the recognition accuracy the model should also be trained with the similar kind of images instead of using other hand gesture datasets available online. So the dataset prepared for both training and testing is by using the same webcam. The dataset used for training the VGG16 consists of five different folders each for one class, containing 1000 images of each hand gesture. The number of classes can be increased to many more depending on the requirement of the user. The hand gestures used and their corresponding labels are listed in Fig. 1. The dataset is split into 800 train and 200 test images from each class. The type of hand gesture or the type of label can be changed before designing the product depending on the users interest, to which they are comfortable. These images are captured from the webcam of the laptop using opencv python.

Fig. 1 Hand gesture and its corresponding label

Hand Gesture	Message to be sent
	"I am sick .I need help"
	"Come home soon"
	"I reached safely"
	"I am going outside"
	"Some strangers are into our house"

3.2 *Raspberry Pi and Associated Hardwares*

A Raspberry Pi3 with OS installed in it and connected to the Wi-Fi or Ethernet is required for preparing this device. Raspberry Pi is a microcontroller that can be used as a single-board computer which is made by the Raspberry Pi foundation, in order to educate people toward the computing world. The Raspberry Pi is a very cheap microcontroller that has Linux and allows one to control electronic components for physical computing and to explore the Internet of things (IoT) using the general-purpose input/output (GPIO) pins. A Quantum webcam, 25 megapixel is connected to the Raspberry Pi in order to capture hand gesture images. A breadboard is used in order to provide the user with a switch-type push button.

4 Proposed System

4.1 *Architecture*

Figure 2 shows the basic architecture of the proposed device. This design can be divided into three parts. First is to capture the hand gesture image from a webcam connected to Raspberry Pi. Second is to decode the message hidden in the image captured using machine learning model. Finally sending the message decoded to the respective person using Amazon web services.

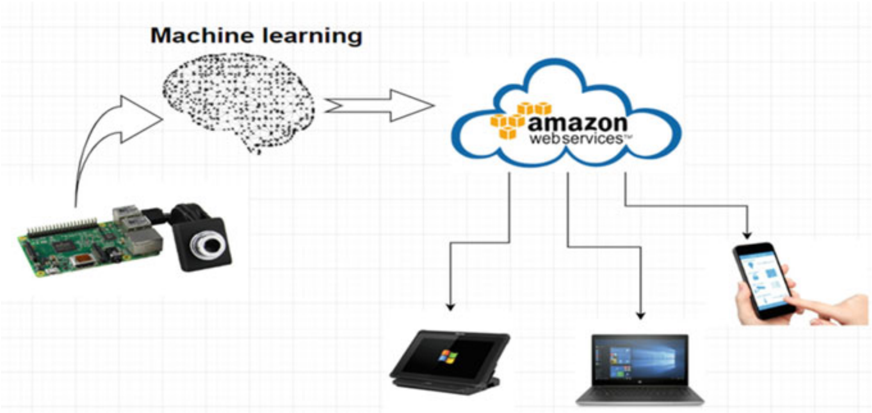


Fig. 2 Basic architecture of the proposed device

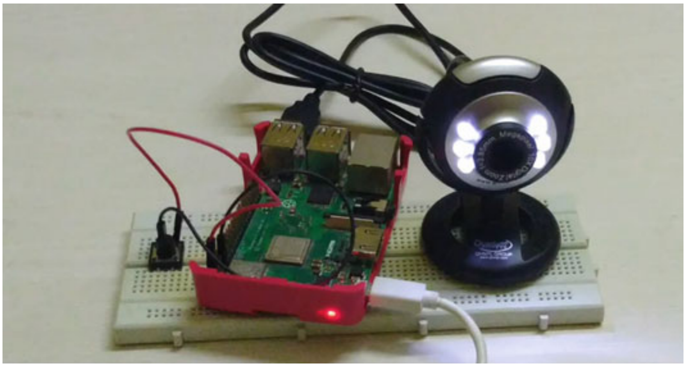


Fig. 3 Hardware setup

4.2 Hardware System

For capturing the hand gesture images Raspberry Pi microcontroller is used, to which an USB webcam is connected. A push button is connected to it using connecting wires via breadboard as shown in Fig. 3. The python scripts for capturing the image when push button is pressed, hand gesture recognition using machine learning model, sending messages via AWS, are dumped into Raspberry Pi. So when it is powered ON and the push button is pressed the image in front of the webcam is captured and fed to the machine learning model and then its output is sent to respective person via AWS SES. One precaution to be followed while using this device is to make sure that the users hand is at a specified distance from the webcam in order for the hand to be in the field view of the webcam. This distance depends on the zoom rate and the type of webcam being used. For this paper, the distance is in the range of 1–4 m. This distance should be generally provided by the designer while making the product. For deaf and dumb people, it is easy to use this device since they can see at what distance approximately they put their hand in front of webcam. But for blind people, this setup should be fixed on a table and the position of the chair on which the user sit and put their hand should also be fixed. The Raspberry Pi is connected to either WiFi or ethernet based on the availability. The webcam used is Quantum, and its resolution is 25 megapixel.

4.3 Software System

Machine Learning Model One of the rapidly growing field in machine learning is deep learning which is widely used in many computer vision, natural language processing applications. Deep learning excels in solving problems related to image classification. Image classification involves classifying a specific image to a most likely related class based on visual content. Transfer learning can be used for solving the image classification problem by making use of deep learning. The identification of hand gestures is also an image classification problem which can be solved using transfer learning. Using transfer learning, you start the training process from the previously learnt patterns obtained when solving similar kind of problem, instead of starting the learning process from beginning. In this way, previous learning can be used rather than starting from scratch. In this paper, pre-trained VGG16 deep neural network is used for which fine-tuning the last four layers is done and a few additional dense layers are added to make it useful for this application. The architecture of the VGG16 network is shown in Fig. 4. The image captured with the webcam is resized to 224 × 224 and fed to the VGG16 deep neural network.

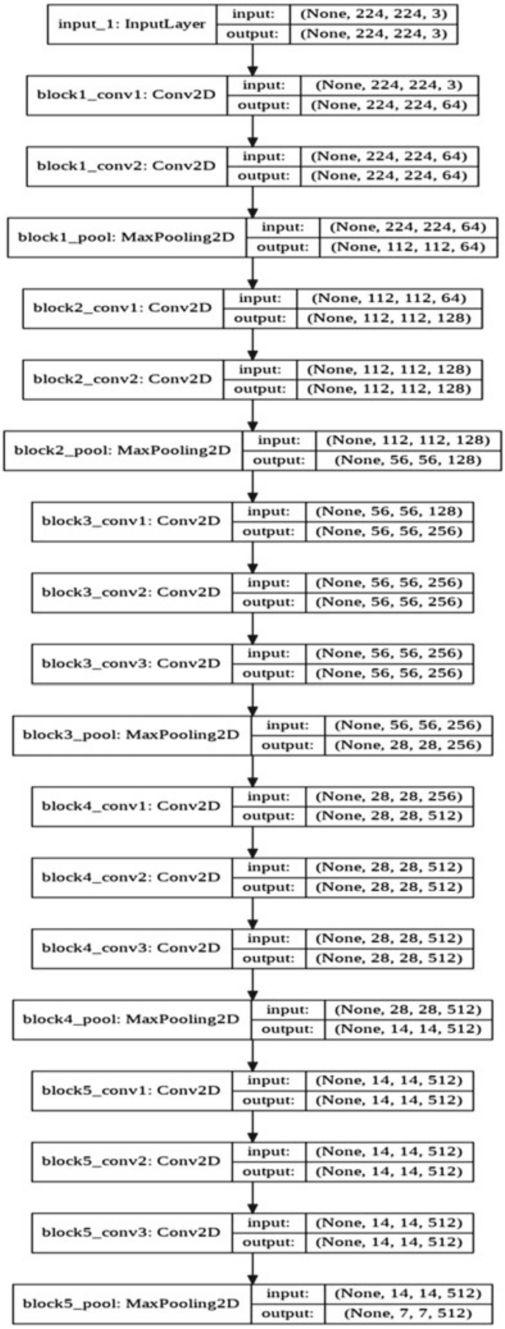
VGG16 is a convolutional neural network (CNN) model proposed in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" by K. Simonyan and A. Zisserman from the University of Oxford [13] . The model achieves 92.7% top-5 test accuracy on a dataset of about 14 million images belonging to 1000 classes. VGG16 was trained for several weeks using NVIDIA Titan Black GPU's.

A RGB image of fixed size 224 × 224 is given as input to cov1 layer. The image is passed through a stack of convolutional and max-pooling layers, where 3 × 3 convolutional filters were used. In one of the configurations, 1 × 1 convolution filters are used, which can be seen as a linear transformation of the input channels (followed by nonlinearity). The spatial padding of convolutional layer input is done in such a way that the spatial resolution is preserved after convolution. The convolution stride is fixed to 1 pixel. Spatial pooling is done by five max-pooling layers, which follow some of the convolutional layers. A 2 × 2 pixel window is used for performing max-pooling, with stride 2. Convolutional layers are followed by three fully connected (FC) layers of which the first two contain 4096 channels each, and the third contains 1000 channels (one for each class) which is removed and a five-channeled layer is added for this paper while doing fine-tuning. The final layer is the softmax layer. Rectification nonlinearity (ReLU) activation is used for all hidden layers.

The first layers of the network understand the low-level features of the image and its level of understanding high-level features goes on increasing as we go on to the last layers. Since in this hand gesture classification problem, also there is a necessity for identifying the similar low-level features, pre-trained weights of the VGG16 are used as it is without any training as it has already mastered in identifying the low-level features. Since the high-level features and the number of image classes are different for this hand gesture classification, fine-tuning is done on the network. Training is done for the last four layers of the VGG16, and the last softmax layer is replaced with a new one having five neurons.

AWS SES Amazon's Simple Email Service (SES) is a email sending service which is a cost-effective and reliable service for businesses of all sizes that use email in order to keep in contact with their customers. This is used for this paper in order to send the decoded message via mail to the desired person or caretaker. AWS is providing one year free subscription, and after that, charges will be applied based on the usage of the account. Since sending mails do not need much computation or memory, it is cost-effective.

Fig. 4 Architecture of VGG16 network


input_1: InputLayer | input: | (None, 224, 224, 3) | output: | (None, 224, 224, 3)
block1_conv1: Conv2D | input: | (None, 224, 224, 3) | output: | (None, 224, 224, 64)
block1_conv2: Conv2D | input: | (None, 224, 224, 64) | output: | (None, 224, 224, 64)
block1_pool: MaxPooling2D | input: | (None, 224, 224, 64) | output: | (None, 112, 112, 64)
block2_conv1: Conv2D | input: | (None, 112, 112, 64) | output: | (None, 112, 112, 128)
block2_conv2: Conv2D | input: | (None, 112, 112, 128) | output: | (None, 112, 112, 128)
block2_pool: MaxPooling2D | input: | (None, 112, 112, 128) | output: | (None, 56, 56, 128)
block3_conv1: Conv2D | input: | (None, 56, 56, 128) | output: | (None, 56, 56, 256)
block3_conv2: Conv2D | input: | (None, 56, 56, 256) | output: | (None, 56, 56, 256)
block3_conv3: Conv2D | input: | (None, 56, 56, 256) | output: | (None, 56, 56, 256)
block3_pool: MaxPooling2D | input: | (None, 56, 56, 256) | output: | (None, 28, 28, 256)
block4_conv1: Conv2D | input: | (None, 28, 28, 256) | output: | (None, 28, 28, 512)
block4_conv2: Conv2D | input: | (None, 28, 28, 512) | output: | (None, 28, 28, 512)
block4_conv3: Conv2D | input: | (None, 28, 28, 512) | output: | (None, 28, 28, 512)
block4_pool: MaxPooling2D | input: | (None, 28, 28, 512) | output: | (None, 14, 14, 512)
block5_conv1: Conv2D | input: | (None, 14, 14, 512) | output: | (None, 14, 14, 512)
block5_conv2: Conv2D | input: | (None, 14, 14, 512) | output: | (None, 14, 14, 512)
block5_conv3: Conv2D | input: | (None, 14, 14, 512) | output: | (None, 14, 14, 512)
block5_pool: MaxPooling2D | input: | (None, 14, 14, 512) | output: | (None, 7, 7, 512)

5 Experimental Results

This section presents the results of the experiments performed on the introduced dataset using different machine learning models. The resolution of the webcam is 25 megapixel, and the dataset contains images with different background complexity, lighting changes, and viewpoint. The hand gestures here are static which means it involves only one image to convey the information. The training of these models and testing is done on Google's colaboratory which provides a free GPU but on conditions. Once the training is done, the weights of the trained model are dumped into Raspberry Pi along with the model. The libraries required should be installed on the Raspberry Pi, and a push button is interfaced via breadboard. The testing is done for 5 times, each time splitting dataset randomly into 800 training samples and 200 testing samples, and the average of the five results is taken into consideration. The first model tried is a three-layered convolutional neural network which gave an accuracy of 92.4%. Later pre-trained VGG16 and ResNet50 [14] models are used excluding the top layer and by adding a five-layered softmax layer. ResNet-50 is a CNN that has been trained on more than a one million images from the ImageNet database. This network can classify objects such as pencil, keyboard, and many animals. Pre-trained VGG16 and Resnet50 resulted an accuracy of 86.3% and 85.9%, respectively. When these two models are fine-tuned, it gave an accuracy of 99.1 and 97.4%. The model used and the accuracy obtained are listed in Fig. 5. The VGG16 model resulted in higher accuracy when it is fine-tuned and trained for 10 epochs with train batchsize of 100, which gave an accuracy of 99.1% when tested with test dataset.

The accuracy of the VGG16 model when fine-tuned is compared with some of the existing methods by testing all of them on the above-mentioned prepared dataset, and the results are shown in Fig. 6. Nagashree et al. [12] proposed an approach for hand gesture recognition which involves several image processing techniques like canny's edge detection, histogram of gradients (HoG), and classification using support vector machine (SVM). This approach resulted in an accuracy of 91.8%.

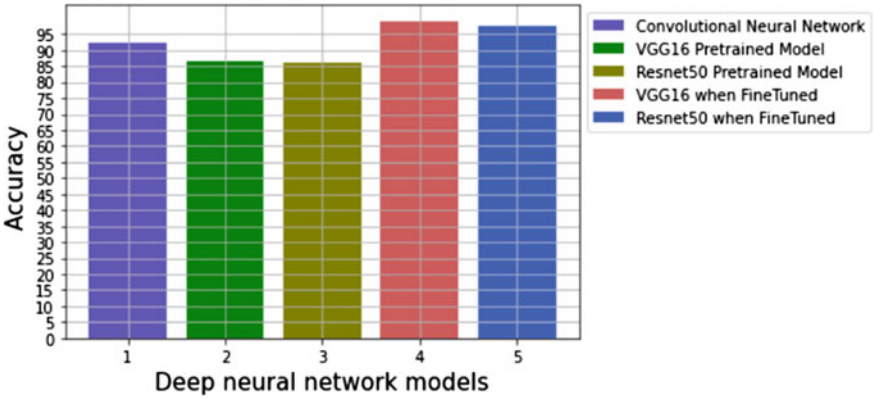


Fig. 5 Accuracy achieved by different deep learning models

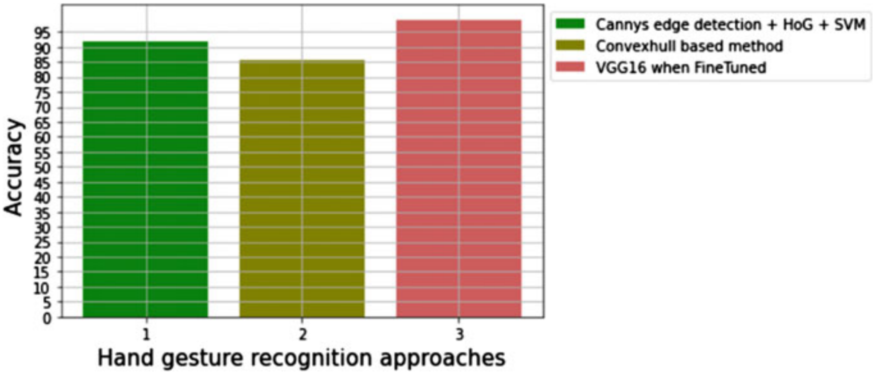


Fig. 6 Comparison of proposed method with existing methods

Lai et al. [15] proposed a method that calculates the finger angles and fingertip positions from convex defect character points of the hand contour to recognize the hand gestures. It resulted in an accuracy of 85.4% on the proposed dataset.

Hence, it can be claimed that the proposed VGG16 framework outperforms several existing alternative frameworks with an accuracy of 99.1% in hand gesture recognition tasks.

Configuring the AWS SES to Raspberry Pi involves the following steps:

- Install awscli and boto3 libraries on Raspberry Pi.
- Sign in to aws management console and search for simple email service.
- Verify the email address by clicking on the link sent to the respected mail by aws once email is entered in the "verify a new email address" box.
- Select an available region from the top right region panel.
- Go to security credentials and create access keys.
- Add these keys and the region name in the python script to set the connection to the AWS SES server.

Once the above steps are done, Wi-Fi or Ethernet is connected to the Raspberry Pi. The device is able to send the messages to the respected person according to the hand gestures shown in front of the device with an accuracy of 99.1%. In this way, the proposed device helps the visual, speech, and hearing-impaired people to communicate easily with others who are far away from them by sending a mail using hand gestures, in a user-friendly way.

6 Conclusion and Future Scope

Thus, a portable communication device has been designed for the deaf, dumb, and blind using the machine learning and IoT via Raspberry Pi which is able to detect hand gestures efficiently without using tapes or data gloves and can be used for long-distance communication. Future enhancement is to focus on developing a model using long short-term memory (LSTM)s to recognize video hand gestures. Our model focuses on recognizing hand gestures based on an image, but to identify a series of hand gesture actions recurrent neural networks and LSTMs can be used. This model can be configured or controlled using a push button on Raspberry Pi like when the button is pressed the pi cam records the video, and once the button is pressed again the recording stops, and the video is pushed to the model where recognition of hand gesture can happen and corresponding message is sent to the respected person.

References

1. Alpaydin E (2020) Introduction to machine learning. MIT Press
2. Matiwade SV, Dixit MR (2016) Electronic support system for deaf and dumb to interpret sign language of communication. Int J Innov Res Sci Eng Technol 5(5):8683–8689
3. Manikandan CT, Karthick Ramachandran D, Santhosh G, Sathyanarayanan B (2019) An innovative method for communication among differently abled people using an electronic gadget. Int Res J Eng Technol 6(3):1–5
4. Pryor TW (2019) Hand motion interpretation and communication apparatus. U.S. Patent 10,446,059, issued 15 Oct 2019
5. Kumar V, Shekhar V, Verma V (2019) An intelligent wearable to aid speech impaired people by detection of specific hand gestures using flex sensors. AIP Conf Proc 2142(1):210007
6. Jagadish R, Gayathri R, Mohanapriya R, Kalaivani R, Keerthana S (2018) Hand gesture recognition system for deaf and dumb persons. Indo-Iran J Sci Res 2(1):139–146
7. Sharma D, Vora K, Shukla S (2017) Hand assistive device for deaf and dumb people. Int J Adv Res (IJAR) 1042–1046
8. Sriram N, Nithiyanandham M (2013) A hand gesture recognition based communication system for silent speakers. In: 2013 international conference on human computer interactions (ICHCI). IEEE, pp 1–5
9. Jain A, Jain L, Sharma I, Chauhan A (2018) Image processing based speaking system for mute people using hand gesture. Int J Eng Sci Res Technol (IJESRT) 368–374
10. Channaiah C, Nikhita KK, Nikitha P, Bhavani NK, Sudeep J (2017) Hand gestures recognition system for deaf, dumb and blind people. IJIRCCE 5(5):10058–10062
11. Devi R, Hemalatha P, Anishya F (2018) Review on gesture recognition for visually challenged people using machine learning. Asian J Appl Sci Technol 2(1):287–292
12. Nagashree RN, Michahial S, Aishwarya GN, Azeez BH, Jayalakshmi MR, Rani RK (2005) Hand gesture recognition using support vector machine. Int J Eng Sci 4(6):42–46
13. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
14. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
15. Lai HY, Lai HJ (2014) Real-time dynamic hand gesture recognition. In: 2014 international symposium on computer, consumer and control. IEEE, pp 658–661

6G Communication: A Vision on the Potential Applications

Sabuzima Nayak and Ripon Patgiri

Abstract 6G communication technology is a revolutionary technology that will revolutionize many technologies and applications. It will be true artificial intelligence (AI)-driven and will be operated at intelligent space. 6G communication technology promises high quality of services (QoS) and high quality of experiences (QoE). 6G communication technology will prove as a game-changer communication technology in many fields and will be capable to influence many applications. 6G will enable Internet of everything (IoE) which will also impact many technologies and applications. With the combination of IoE and 6G communication technology, many applications will explode in the coming future, particularly, UAV, holographic communication, augmented/virtual reality, and so on. In this paper, we envision the potential applications of 6G communication technology in the near future.

Keywords 6G communications · Networking · Wireless communication · Health care · Vehicular technology · Virtual communications · Robotics communications · Internet of things · Internet of everything · Industrial Internet of things · Industrial Internet of everything

1 Introduction

6G mobile communication technology is one of the most prominent emerging research areas and radical technology, which will change our perception on lifestyle, society and business. With the advent of 5G communication technology, the researcher is focusing on the next generation mobile communication technology. Since we have already evidence that every decade there is a mobile generation, so it is expected that 5G will serve from 2020 to 2030; 6G will serve from 2030 to 2040, and 7G will serve from 2040 to 2050. Currently, 5G is deployed in few countries and it is able to impact the global economy. Hence, it is anticipated that the next generation of 5G will be able to create more impact on the global economy.

S. Nayak · R. Patgiri (✉)
National Institute of Technology Silchar, Silchar, India
e-mail: ripon@cse.nits.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_16

However, the requirements of 6G are extremely different from 5G [7]. Consequently, the parameters of 6G will enable tremendous new applications and technologies [33]. For instance, 6G will be a prominent player in vehicular technology, health care, and commercial enterprise. Also, 6G communication technology will be able to influence many researchers, practitioners, and industrialists. Hence, there are diverse research articles already discussed on parameters of 6G [6, 14, 15, 21]. Bi [4] highlighted ten trends of 6G communications. Kato et al. [20] exposed ten great challenges on integrating AI with 6G. Letaief et al. [23] envisioned the truly AI-driven 6G communication technology.

The promises of 6G communication technology create numerous issues and challenges [9, 46], albeit, worldwide deployment is expected from 2030. Moreover, rural deployment of such technology is a grand challenge [41]. Currently, 5G communication technology focuses on campus solution, and thereby, the mobility and coverage are the main issues. Therefore, 5G is unable to support many applications. Nonetheless, the 6G communication technology promises global coverage and mobility using satellite communication. Presently, we are lacking $\geq$1 Tbps data rates and extremely reliable and low latency, and this triggers compromise in QoS. As per the requirements and promises of 6G communications, there are numerous research possibilities and opportunities which will endure many new applications to be conceived. Thus, 6G will bloom the global economy, society, and lives.

In the 6G communication era, many existing technologies will be redefined and restructured. Reshaping existing technology will shift our lifestyle, society and business. Prominently, Internet of things (IoT) will be redefined as Internet of everything (IoE) which will be the dawn of many novel technologies. IoE will be a key player in the 6G communication era, and it will enable intelligent city, intelligent health care (intelligent wearable devices and intelligent Internet of medical things), intelligent industrial Internet of everything (IIIoE), intelligent power grid, and intelligent robots. We also foresee that the 6G communication technology will multiply in spawning many new applications. Therefore, in this article, we envision the impact of 6G communication technology in various fields. Thus, Sect. 2 briefs on 6G technology to highlight the key features. Besides, Sect. 3 explores the enabling technology of 6G communications. Sections 4, 5, 6, 7, and 8 expose the impact of 6G communications technology on vehicular technology, robotics, virtual reality, health care, and city, respectively. Likewise, Sect. 9 exposes on how 6G communication technology will be able to enable an industrial revolution. Furthermore, Sects. 10 and 11 expose that the nano-things and datacenters depend on the 6G communication technology. Finally, Sect. 12 draws a suitable conclusion.

2 6G Technology

6G communication is all about sixth sense communication [19]. Therefore, many countries have already begun project on 6G, particularly, Finland, USA, South Korea, China [9], and Japan [11]. Its prerequisites are very challenging, and numerous requirements are discussed in many articles [10, 21, 22, 27, 42]. Particularly, 6G will operate at terahertz (THz) frequency to gain 1 Tbps data rate [12, 20, 30, 38]. However, 1 Tbps data rate is not enough for holographic communications. It demands more data rate to provide support to full holographic communication. The wavelength frequency is expected to be 100–300 μm. Researchers are exploring to increase the efficiency of THz signal by spectrum reusing and sharing. Some techniques already exist for spectrum reuse such as cognitive radio (CR). It helps many wireless systems to access the same spectrum through a spectrum sensing and interference management mechanisms [6]. In case of spectrum sharing, temporally underutilized or unlicensed spectrum is utilized to maintain availability and reliability. Symbiotic radio (SR) is a new technique to support intelligent and heterogeneous wireless networks. It will aid in efficient spectrum sharing. Even so, deploying these techniques in the 6G wireless network are still big challenges. Also, 6G is expected to deliver truly AI-driven communication technology [20, 25, 27, 29]. In addition, 6G promises to provide high security, secrecy, and privacy [9]. Likewise, it will be three-dimensional communication technology, particularly, time, space, and frequency.

2.1 Transition from Smart to Intelligent

Due to 6G, all smart devices will be converted to intelligent devices, and these intelligent devices will be truly AI-driven devices [28]. Thus, the intelligent device (may be tiny device such as smart watch and nano-chip) will be able to predict, make a decision, and share their experience with other intelligent devices. So, there is a paradigm shift from smart to intelligent era using 6G communication technology and AI.

2.2 Quality of Services

6G technology promises to provide high quality of services (QoS) and the parameters include high data rate of at least 1 Tbps, extremely reliable, further-enhanced mobile broadband (FeMBB), low latency communication (ERLLC), long-distance and high-mobility communications (LDHMC), ultra-massive machine-type communications (umMTC), and extremely low-power communications (ELPC) [44]. Moreover, QoS includes massive broad bandwidth machine type (mBBMT), mobile broad bandwidth and low latency (MBBLL), and massive low latency machine type (mLLMT) [15]. These parameters will serve diverse applications to revolutionize. In addition, high QoS will enable many new applications, for instance, telesurgery. Also, 6G technology promises high QoE along with high QoS which defines user-centric communications. It will be achieved by holographic communications, augmented/virtual reality, five sense communications, and tactile Internet. Moreover, QoE will bring revolution in intelligent devices, intelligent cars, intelligent drones, intelligent ambulances, and many more [27]. Achieving high QoE depends on the implementation of all desired parameters by 6G technology. Similarly, quality of life (QoL) is determined to enhance the lifestyle with QoS and QoE in health care. Likewise, 6G technology will enable the high QoL. The key parameters of QoL are intelligent ambulance services, remote health monitoring of patients, intelligent accident detection, hospital-to-home (H2H) service, telesurgery, and precision medicine.

3 6G Enabling Technology

3.1 Internet of Everything (IoE)

6G follows 6Cs, namely capture, communicate, cache, cognition, compute, and control [15]. Cognition helps in formulating feasible determinations based on input digital data. These are intelligent determinations that make the computing easy. Then, computed data are transmitted to intelligent devices to control the action taken by the devices [15], for example, raising an alarm. IoE will use the core service of combined and enhanced eMBB and mMTC. Requirement of IoE from 6G is huge capacity to connect millions of intelligent devices and high data rates to support touch experiences in those devices. Industrial IoT (IIoT) uses the core service of combined and enhanced URLLC and mMTC. Huge capacity to connect the sensors, actuators, and staff for communication and low latency to maintain seamless communication among them [15]. When 6G will be commercial, it will be the era of big data 2.0 that requires a supercomputer to compute and analyze data [28].

3.2 Edge Intelligence

6G will rely on edge computing [34] to bring the cloud features closer to intelligent devices. Edge technology will offer uninterrupted and high-speed Internet services to the intelligent devices. It collects, computes, and analyzes the data in real time in edge nodes. Edge node also filters data and transmits only important information to the cloud for storage. Edge technology reduces communication and computation cost. Some other advantages are low latency, reliability, adaptability, scalability, and privacy. All the advantages of edge technology will greatly help 6G to meet its prerequisites to provide high QoS. Edge intelligence is combining edge computing and AI [45]. Edge analytics, implement AI algorithms for analysis in edge nodes [35]. With the help of AI, edge nodes are capable of embedding image, data, and video edge analytics. Execution of AI algorithms requires high computational resources and power consumption, which is limited in the edge nodes. One of the network nodes in 6G will be edge nodes. Moreover, 6G will have real-time intelligent edge that computes and analyze on live data [18].

3.3 Artificial Intelligence

6G will provide global coverage, i.e., space–air–water. 6G will achieve this goal by making different aspects of communication "intelligent," and, intelligence means AI. Truly AI-driven 6G helps the system in self-aware, self-compute, and self-decide on a situation [13, 27]. At the physical layer of the network, AI will help channel state estimation and prediction, automatic modulation classification, adaptive encoding and decoding, and intelligent beamforming. Likewise, physical layer security [16] is a key important feature of 6G communication technology. At the data link layer, AI helps in resource allocation by implementing deep reinforcement learning [20]. Likewise, at the transport layer, route computing and intelligent traffic prediction algorithms will be deployed [15]. 6G will explore dynamic spectrum access. Moreover, as discussed in the above section AI will also support edge technology. 6G will depend on AI for reducing the heavy computation responsibility in edge devices to provide next generation services.

4 Vehicular Technology

Many vehicles can be driverless and intelligent [39]. The intelligent vehicles will optimize fuel consumption, route, and work efficiency based on real-time services of 6G. Intelligent vehicles are more economical, because they can predict future and optimize the problem economically. The future vehicle will be truly AI-driven and high-speed wireless communications. The future vehicles will also be integrated with diverse sensors to enhance the intelligence.

4.1 Intelligent Cars

It is expected that 6G will be able to deliver self-driven vehicle, i.e., steering-less vehicles. The vehicles will be truly AI-driven, learning from practical experiences [37]. The cars will monitor passenger health, for example, blood pressures, heartbeat rate, body temperature, brain wave capturing, and emotion detection due to truly real-time communication capability of 6G. Also, the intelligent cars will turn into an intelligent ambulance in case the passenger need treatment during the ride. It is achieved by consulting the remote doctors using mix-virtual reality/holograph through 6G communication. Intelligent cars will avoid accidents by exchanging information with nearby vehicles. Mobile UAVs will be able to provide wireless battery charging [43].

4.2 Unmanned Aerial Vehicles

UAV will be used for many purposes [24], for instance, fronthaul and backhaul services in rural area. The intelligent drones will be capable of communication in drone-to-drone (D2D) and drone-to-infrastructures (D2I). Intelligent drones can share their knowledge. It will also be engaged for faster delivery of online products. Moreover, it will be utilized in critical operations where human cannot reach. Police will use drones to catch criminals or spray the tear gas to control the mobs. Also, the surveillance intelligent drone will continuously monitor the populated location. These types of intelligent drones will detect fire, accidents, traffics, etc., in real time to inform the police in real time.

Currently, CCTV is used for security and surveillance. However, CCTVs are not efficacious in securing many things and identifying the threats. For instance, CCTV is installed in a fixed pole or wall and they cannot chase any criminal. Therefore, the drone will be added to fortify the security and surveillance systems. For example, drone can be used to catch a criminal. The drone can stream the videos continuously to the monitoring rooms. However, it requires high-speed data link to provide high-definition videos in real time. Also, intelligent drone can alert possible threat to the security force. The drone will play a prominent role not only in the city, but also in border areas. It is very dangerous in line of control (LOC) to monitor the enemies for 24 × 7. Thus, the drone can provide security and surveillance in extremely dangerous border locations. To achieve these goals, intelligent drone requires the 6G communication technology.

4.3 Intelligent Transportation

Achieving a complete intelligent driving is very difficult because its prerequisite is a combination of many complex algorithms. Some example of the complex algorithms is automatic driving, path planning, obstacle detection, vehicle monitoring, and emergency rescue operations. They produce a huge amount of real-time data which needs to be computed quickly. However, they require high data rate, low latency, seamless connectivity, etc., which can be provided by 6G communication technology.

Domestic Transportation

Intelligent domestic transportation will remain connected to the 6G Internet during the whole journey. 6G with its high-speed Internet, low latency, and global coverage will provide services smoothly without interruption. It will provide reliable and real-time information to the passenger and reduce their waiting time. The real-time data will also help the driver to remain updated about the traffic or any blockage in the route. In case of heavy traffic, it will help in safe driving and blockage in the route due to an accident or natural disaster will be known earlier. Based on the information, the driver can reroute the path and reduce harmful gas emission. Another key point is maintaining security. In case of small public transportation, e.g., cab or taxi, the passenger can detect the route taken by the driver. In case of any threat, the passenger can inform the police without any direct call. Moreover, police can use the camera present in public transportation to track a suspected vehicle. The intelligent vehicles will continuously monitor its own components and parameters. Upon detection of any damage or abnormality, then immediate action is taken.

International Transportation

6G will help in keeping track of vehicles crossing the border of a country for the security personnel. It will help in reducing crimes such as human trafficking and smuggling. Airplanes and ships will also remain connected to 6G IoE; hence, Internet services will be provided during this journey. The 6G will provide global coverage, thus predicting any sudden changes in weather such as rain or thundering. This real-time information will help to take necessary actions. With intelligent transportation, the airplane and ships can be made fully automated. Moreover, global coverage will help tracking the airplane and ships in case of any accident and immediate medical and rescue services can be provided. In addition, 6G will provide services in underwater. Therefore, a ship or airplane can be tracked; in case, it has drowned and got displaced due to strong water current.

5 Intelligent Robotic Communication

Robotic communication refers to communication between robots or between humans and robots. Ad-hoc networks are always used for robotic communication because robots are independent nodes. Robots depend on the AI for communication. Robots will be greatly helpful in home services. Intelligent robot can learn to serve efficiently as personal assistance such as for physically challenged and elder people. Robots can be a great teacher as well as a learner, which will require mobile communication connection to the Internet. Intelligent robots can share their experience with other intelligent robots. Therefore, robot-to-robot (R2R) communications can improve intelligent robots in service. Moreover, intelligent robots can be used in the industry for economical, efficient, and faster manufacturing of products.

5.1 Aerospace Robotic Communication

Aerospace robotic communication requires an intelligence space. An intelligence space is an intelligence environment to monitor and control activities in space and establish a smooth communicate with ground station. Robots will handle the maintenance of intelligence space [2] by observing the environment and computing the data to take appropriate action in case of extreme situations. They will constantly transmit data to ground station. To achieve such communication, 6G will provide a smooth and high-speed communication. Moreover, in the intelligence space, the robots will be distributed and require a reliable and dynamic communication network. Robots will also transmit images and videos. Transmitting such high-sized data compared to files requires high speed and high data rate of 6G. Moreover, the features of 6G such as high data rate, speed, high QoS will help to provide a 3D image of the space for a better understanding of the space activities.

5.2 Underwater Robotic Communication

Robotics is also explored for underwater tasks such as search and rescue, imaging, and security. The terrorists also use waterways for entering the country. So, constant surveillance is essential. In such cases sometimes, robots will be more efficient. However, the underwater robots have to constantly interact with the ground station. 6G will be very helpful for such communication because 6G is planning to provide underwater communication services. A smooth and reliable communication channel needs to be established for real-time analysis and action planning. 6G high speed and low latency will help in obtaining images and video data. It will also help in constantly locating the robot on the map and providing navigation instructions from ground station.

6 Virtual Communication

The QoS provided by 6G communication technology will satisfy many requirements of virtual communication because it requires uninterrupted high speed, low latency to constantly maintain virtual presence, touch, experience, etc., which is provided by 6G.

6.1 Holographic Communication

Holographic communication requires high data rates to provide good QoS and streaming of high-definition videos. It requires very low latency for real-time voices and spontaneous control responses [5, 14]. Holographic communication will be a major breakthrough during 6G. 6G will be capable of satisfying all the requirements. Holographic communication will use the core service of combined and enhanced eMBB and URLLC. Due to high-speed and uninterrupted Internet services, communicating with a person through holographic communication will be similar to face-to-face interaction. The person will be able to move around without anyone's intervention. Hence, physical presence at a formal meeting or interview will not be compulsory.

6.2 Augmented Reality and Virtual Reality

The requirements of augmented reality (AR) and virtual reality (VR) are similar to holographic communication. The core services used are combined and enhanced eMBB and URLLC. High data rates to provide good QoS and high-definition videos. Very low latency is essential for real-time voices and immediate control responses [15]. Peak data rate requirement of AR and VR is 1 Tbps and user experience of >10 Gbps with >0.1 ms latency which can be provided by MBBLL [15]. AR provides visibility inside the object without separating the different components of the object. For example, various components of a machine are visible without removing the machine parts. Moreover, the physical depth of the visibility can be changed along with the size of viewed area. VR helps in creating one's own artificial environment. It helps in adding details related to games, movies, or animation. Moreover, using 6G the holographic communication, AR and VR will be combined for new services.

6.3 Tactile/Haptic Internet

Tactile Internet requires ERLLC and high-speed communication to grab the tactile in real time. Usually, the remote user sends the signal to another human or robot. This technology is applicable where a physical presence is compulsory to perform a task. Usually, this task requires precise work and robots cannot always be trained to perform such tasks. For example, surgery or defusing the bomb. During telegery, the doctor can use tactile technology to perform a surgery using a human or a robot. As per the movement of the remote doctor, the robot will move. Such detailed work required uninterrupted and high-speed Internet. Thus, 6G will provide great support in achieving these goals.

7 Intelligent Health Care

QoL will be the greatest player in ensuring intelligent health care. It is expected that 6G communication will revolutionize healthcare. Intelligent health care will implement hospital-to-home (H2H) service which is implemented upon intelligent vehicle. This mobile hospital will replace ambulance services in the near future. H2H will also implement real-time accident detection and automatic emergency detection. Diverse intelligent wearable device and sensors will help in the detection of accident automatically in real time. Moreover, needle-free blood sample reader (BSR) sensor will greatly help in the detection of diseases automatically [26]. BSR sensor will read the blood sample and send it to the pathology laboratory. Therefore, intelligent wearable devices (IWD) reduce the risk of medical staff in contacting with viruses.

7.1 Intelligent Internet of Medical Things (IIoMT)

Intelligent Internet of medical things (IIoMT) will help in avoiding the time and space barriers [26]. For instance, remote doctor can perform surgery using telesurgery which requires high-speed communication. With new research, the healthcare devices will become cheaper and affordable by all patients. 6G with its high capacity feature will be able to connect these devices to IIoMT. The network nodes will be intelligent to help the patient in case of any adversity, such as informing the hospital automatically without human intervention.

8 Intelligent City

The key requirement of intelligent city is a real-time communication system. Intelligent city manages the traffic, waste, home, grid, and environment intelligently in real time which requires high QoS from wireless communication technology.

8.1 Intelligent Traffic

Intelligent cities will enhance the QoL along with the protection of environment. The traffic signals will be connected to IoE. The intelligent traffic signal will analyze the data to determine the average traffic opening duration for smooth movement of vehicles. It will help in reducing emission of harmful gases by reducing the halt of vehicles for long duration in traffic [8]. In case an ambulance is traveling for an emergency, then it can send notification to intelligent traffic signals. Upon receiving the notification, the intelligent traffic signal will track the ambulance using GPS. At appropriate times, the traffic will open for uninterrupted ambulance movement. For such applications, high-speed and low latency 6G Internet will be essential. Similarly, the police or army vehicles will be provided an uninterrupted movement. Moreover, intelligent traffic signals can also be used to interrupt any suspected vehicle. In addition, they will monitor the vehicles for any traffic rule violation. Whenever any dispute happens among people, the intelligent traffic signal can immediately alert the police. Moreover, camera present in traffic signals can monitor for any accident that happens in an isolated and remote area. In such situations, the live data have to be analyzed quickly for informing appropriate authority and 6G edge nodes will be capable of performing those tasks.

8.2 Intelligent Waste Management

Intelligent waste management is also a part of an intelligent city to reduce pollution and health risk. The dustbins will have sensors to determine whether it is full or not. In case it is full, the waste collection vehicle will be notified and will be collected quickly. In some areas, the dustbins may get full quickly; hence, it requires waste pickup twice or more in a day. Whereas, in some localities the dustbin may be getting full after two days or more. Therefore, intelligent waste collection will help in optimizing utilization of resources. All the dustbin sensors will be connected to IoE. 6G global coverage will help to connect every corner of the city.

8.3 Intelligent Home

An intelligent home assistant will connect all intelligent home appliances such as television, home security, washing machine, music system to IoE. Human presence and some commands will activate the appliances. The people of the home can use intelligent phones or some wearable to physically control the intelligent home appliances. An intelligent light will switch off in the absence of any person in the room or maintain the light intensity based on sunlight intensity. This is very useful for small children and elderly people. The intelligent home will also have an intelligent environment monitoring system. It will keep track of the environmental condition of the home. For example, if the weather is hot, then the central air conditioner will start to keep the house cool. This helps in keeping the people of the house healthy. If a family member is ill, then that room's environmental condition will be different from the rest of the rooms. The data generated will be huge, and these need to be processed by 6G edge nodes quickly. During an emergency, an intelligent home assistant will detect the issue and inform the appropriate authority. For example, a fire at a dangerous level the water sprinkler is automatically opened or informs the fire station in case fire is not under control. However, 6G edge nodes have to compute the data and inform the appropriate authority quickly even without the permission of the owner.

8.4 Intelligent Power Grid

Intelligent power grid is important because 6G nodes will perform heavy computation to maintain continuous energy supply. The intelligent power grid is AI-based, fully automated, remote control enabled with self-healing features. With the exhaustion of natural resources, intelligent power grid will rely completely on renewable energy resources. It will intelligently integrate the distributed and renewable energy resources. 6G nodes will both depend and help in the power grid. The intelligent

6G nodes placed at different locations can determine the power consumption during different times of the year or season. These data will be used to determine the power production requirement for an intelligent city. After sunset, the 6G nodes will automatically ON the street light. Intelligent power grid will also store energy in case of emergencies. 6G nodes will intelligently switch to reserve energy in case of power cuts. Another concept is micro-grid. Using renewable energy resources, the consumers will also produce electricity. Such consumers are called prosumers. Prosumers will sell excess electricity to the grid. Such a small grid is called a micro-grid [3]. Moreover, in case of a power cut the intelligent nodes will use micro-grid to resume power supply. With 6G connectivity, the monitoring and maintenance of the micro-grid will be easy. Real-time electricity reserve in the micro-grid is recorded and saved. Any illegal sale of electricity, such as industries or illegal institutions doing illegal activity, is monitored. The electricity billing will also be intelligent, the intelligent home will send the electricity consumption details to the power grid datacenters, and a bill will be automatically forwarded to the consumer.

9 Industrial Revolution

The Industry 4.0 was revolutionized as digitalization [32]. Industry 5.0 is about personalization and human-centric manufacturing. Industry 5.0 will be able to integrate AI and 5G communication to increase productivity. Moreover, the industrial robots will become more smart due to the advent of AI-driven communication technology. umMTC takes place in the industry due to the huge number of sensors and robots. The robots and sensors require extremely reliable and low latency communication for precision and faster production. Therefore, it is expected that productivity will increase in Industry 5.0. Moreover, intelligent robots can optimize their works efficiently and economically. The industry focuses on all the processes from procuring the raw materials to delivery of finished product to customers. Therefore, an industry has to solve many complex problems to reduce overall profit. The communication system has to support industrial IoT (IIoT). IIoT application focuses on power system monitoring, demand-side energy management, integration of renewable energy generators and coordination of distributed power storage [17]. Moreover, the sensors and robots will produce huge volumes of data. These data will be processed in edge nodes [40]. However, Industry 5.0 will not be capable of supporting industries completely due to the presence of many issues. But, it is expected that Industry 6.0 will be truly AI-driven industry. 6G edge nodes will be able to handle heavy computation, hence providing spontaneous responses. 6G with high density will be able to manage a massive number of robots and sensors. Industry 6.0 will deliver industrial Internet of everything (IIoE) [27]. Thus, Industry 6.0 will be the intelligentization of industry.

10 Internet of Nano-Things

Nano-things have components of nano-size; hence, they are capable of performing simple tasks. It has small memory, and thus, data storage capacity is low. A communication network consisting of nano-things is called the Internet of nano-things (IoNT) [1]. Nano-things communicate within short range. Therefore, within a small range more nano-nodes need to be deployed. Communication of IoNT can be performed by using molecular or THz communication. THz communication will be secure, reliable, and faster compared to molecular communication [36]. For THz communication, 6G will be an excellent choice. Therefore, 6G will support IoNT and will make it possible for deployment. Nano-things are small and simple; hence, speed will be more. In this regard, 6G will >1 Tbps speed will perfectly complement IoNT. 6G technology will have high density; hence, controlling the IoNT network with a massive number of nano-things will be easy. Another important point is in some situations the nano-things have to continuously generate valuable information, but due to small memory these data need to be transmitted to the datacenter. Therefore, with high-speed 6G communication network nano-things will transmit the data smoothly.

11 Datacenter Connectivity

6G network nodes will have smaller power supplies and memory. The data generated by the 6G network nodes are high in both volume and quality. For example, underwater rescue operations stream a real-time high-quality video. This high-sized data are not possible to store in the 6G network nodes. Thus, the data are transmitted to the datacenter for storage. Datacenter requires high data rate and extremely low latency [31]. Due to the high speed of 6G, the data generation and data transmission to the datacenter can be done simultaneously. Moreover, the data can be saved in multiple locations in parallel. 6G also provides security during data transmission.

12 Conclusion

In this article, we have discussed all potential applications of 6G communication technology, and we envision and identify the potential of 6G communication technology in many fields, for instance, transportation and city. It has the potential to revolutionize many technologies and applications. We also explore the impact of 6G communication technology in diverse applications. There are tremendous applications that will fully depend on 6G communication technology, particularly, vehicular, health care, modern cities, and industries. Many new technologies and applications are yet to be conceived due to the lack of an efficient and powerful communication technology. We will enter into an intelligent era from smart era, and this transition will change our perception on lifestyle, society, and business. Furthermore, 6G communication technology will also impact the global economy. Therefore, we will evidence that 6G communication technology will be a game-changer technology.

References

1. Akyildiz IF, Jornet JM (2010) The internet of nano-things. IEEE Wirel Commun 17(6):58–63
2. Alsamhi S, Ma O, Ansari MS (2019) Survey on artificial intelligence based techniques for emerging robotic communication. Telecommun Syst 72(3):483–503
3. Atasoy T, Akinc HE, Ercin O (2015) An analysis on smart grid applications and grid integration of renewable energy systems in smart cities. In: 2015 international conference on renewable energy research and applications (ICRERA), pp 547–550
4. Bi Q (2019) Ten trends in the cellular industry and an outlook on 6G. IEEE Commun Mag 57(12):31–36. https://doi.org/10.1109/MCOM.001.1900315
5. Calvanese Strinati E, Barbarossa S, Gonzalez-Jimenez JL, Ktenas D, Cassiau N, Maret L, Dehos C (2019) 6G: the next frontier: from holographic messaging to artificial intelligence using subterahertz and visible light communication. IEEE Veh Technol Mag 14(3):42–50. https://doi.org/10.1109/MVT.2019.2921162
6. Chen S, Liang Y, Sun S, Kang S, Cheng W, Peng M (2020) Vision, requirements, and technology trend of 6G: how to tackle the challenges of system coverage, capacity, user data-rate and movement speed. IEEE Wirel Commun 1–11. https://doi.org/10.1109/MWC.001.1900333
7. Chettri L, Bera R (2020) A comprehensive survey on internet of things (IoT) toward 5G wireless systems. IEEE Int Things J 7(1):16–32
8. Cikhardtová K, Bělinová Z, Tichý T, Ružiča J (2016) Evaluation of traffic control impact on smart cities environment. In: 2016 smart cities symposium prague (SCSP), pp 1–4
9. Dang S, Amin O, Shihada B, Alouini MS (2020) What should 6G be? Nat Electron 3(1):1131–2520. https://doi.org/10.1038/s41928-019-0355-6
10. David K, Berndt H (2018) 6G vision and requirements: is there any need for beyond 5G? IEEE Veh Technol Mag 13(3):72–80. https://doi.org/10.1109/MVT.2018.2848498
11. DOCOMO N (2020) White paper 5G evolution and 6G. Accessed on 1 Mar 2020 from https://www.nttdocomo.co.jp/english/binary/pdf/corporate/technology/whitepaper_6g/DOCOMO_6G_White_PaperEN_20200124.pdf
12. Elmeadawy S, Shubair RM (2019) 6G wireless communications: future technologies and research challenges. In: 2019 international conference on electrical and computing technologies and applications (ICECTA). UAE. IEEE, pp 1–5. https://doi.org/10.1109/ICECTA48151.2019.8959607
13. Elsayed M, Erol-Kantarci M (2019) AI-enabled future wireless networks: challenges, opportunities, and open issues. IEEE Veh Technol Mag 14(3):70–77. https://doi.org/10.1109/MVT.2019.2919236
14. Giordani M, Polese M, Mezzavilla M, Rangan S, Zorzi M (2020) Toward 6G networks: use cases and technologies. IEEE Commun Mag 58(3):55–61. https://doi.org/10.1109/MCOM.001.1900411
15. Gui G, Liu M, Tang F, Kato N, Adachi F (2020) 6G: opening new horizons for integration of comfort, security and intelligence. IEEE Wirel Commun 1–7 (2020). https://doi.org/10.1109/MWC.001.1900516
16. Hamamreh JM, Furqan HM, Arslan H (2019) Classifications and applications of physical layer security techniques for confidentiality: a comprehensive survey. IEEE Commun Surv Tutor 21(2):1773–1828. https://doi.org/10.1109/COMST.2018.2878035
17. Hasan MZ, Al-Rizzo H (2019) Optimization of sensor deployment for industrial internet of things using a multiswarm algorithm. IEEE Int Things J 6(6):10344–10362

18. Huang T, Yang W, Wu J, Ma J, Zhang X, Zhang D (2019) A survey on green 6G network: architecture and technologies. IEEE Access 7:175758–175768. https://doi.org/10.1109/ACCESS.2019.2957648
19. Kantola R (2019) 6G network needs to support embedded trust. In: Proceedings of the 14th international conference on availability, reliability and security. In: ARES'19, association for computing machinery, New York, NY, USA. https://doi.org/10.1145/3339252.3341498
20. Kato N, Mao B, Tang F, Kawamoto Y, Liu J (2020) Ten challenges in advancing machine learning technologies toward 6G. IEEE Wirel Commun 1–8
21. Katz M, Matinmikko-Blue M, Latva-Aho M (2018) 6Genesis flagship program: building the bridges towards 6G-enabled wireless smart society and ecosystem. In: 2018 IEEE 10th Latin-American conference on communications (LATINCOM). IEEE, Guadalajara, Mexico, pp 1–9. https://doi.org/10.1109/LATINCOM.2018.8613209
22. Katz M, Pirinen P, Posti H (2019) Towards 6G: getting ready for the next decade. In: 2019 16th international symposium on wireless communication systems (ISWCS). IEEE, Oulu, Finland, pp 714–718. https://doi.org/10.1109/ISWCS.2019.8877155
23. Letaief KB, Chen W, Shi Y, Zhang J, Zhang YA (2019) The roadmap to 6G: AI empowered wireless networks. IEEE Commun Mag 57(8):84–90. https://doi.org/10.1109/MCOM.2019.1900271
24. Li B, Fei Z, Zhang Y (2019) UAV communications for 5G and beyond: recent advances and future trends. IEEE Int Things J 6(2):2241–2263
25. Nawaz SJ, Sharma SK, Wyne S, Patwary MN, Asaduzzaman M (2019) Quantum machine learning for 6G communication networks: state-of-the-art and vision for the future. IEEE Access 7:46317–46350. https://doi.org/10.1109/ACCESS.2019.2909490
26. Nayak S, Patgiri R (2020) 6G communications: a vision on intelligent healthcare. IEEE Int Things J (under communication)
27. Nayak S, Patgiri R (2020) 6G communications: envisioning the key issues and challenges. CoRR abs/2004.040244
28. Nayak S, Patgiri R, Singh TD (2020) Big computing: where are we heading? EAI Endorsed Trans Scalable Inf Syst. https://doi.org/10.4108/eai.13-7-2018.163972
29. Piran MJ, Suh DY (2019) Learning-driven wireless communications, towards 6G. In: 2019 international conference on computing, electronics communications engineering (iCCECE), pp 219–224. https://doi.org/10.1109/iCCECE46942.2019.8941882
30. Rappaport TS, Xing Y, Kanhere O, Ju S, Madanayake A, Mandal S, Alkhateeb A, Trichopoulos GC (2019) Wireless communications and applications above 100 GHz: opportunities and challenges for 6G and beyond. IEEE Access 7:78729–78757. https://doi.org/10.1109/ACCESS.2019.2921522
31. Rommel S, Raddo TR, Monroy IT (2018) Data center connectivity by 6G wireless systems. In: 2018 photonics in switching and computing (PSC), pp 1–3. https://doi.org/10.1109/PS.2018.8751363
32. Rossi B (2020) What will industry 5.0 mean for manufacturing? Accessed on Mar 2020 from https://www.raconteur.net/technology/manufacturing-gets-personal-industry-5-0 (2018)
33. Saad W, Bennis M, Chen M (2019) A vision of 6G wireless systems: applications, trends, technologies, and open research problems. IEEE Netw 1–9. https://doi.org/10.1109/MNET.001.1900287
34. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: vision and challenges. IEEE Int Things J 3(5):637–646
35. Shi Y, Yang K, Jiang T, Zhang J, Letaief KB (2020) Communication-efficient edge AI: algorithms and systems. In: IEEE Communications Surveys & Tutorials, vol 22, no 4, pp 2167–2191, Fourthquarter 2020. https://doi.org/10.1109/COMST.2020.3007787
36. Sicari S, Rizzardi A, Piro G, Coen-Porisini A, Grieco L (2019) Beyond the smart things: towards the definition and the performance assessment of a secure architecture for the internet of nano-things. Comput Netw 162:106–856. https://doi.org/10.1016/j.comnet.2019.07.012
37. Tang F, Kawamoto Y, Kato N, Liu J (2020) Future intelligent and secure vehicular network toward 6G: machine-learning approaches. Proc IEEE 108(2):292–307. https://doi.org/10.1109/JPROC.2019.2954595

38. Tomkos I, Klonidis D, Pikasis E, Theodoridis S (2020) Toward the 6G network era: opportunities and challenges. IT Prof 22(1):34–38. https://doi.org/10.1109/MITP.2019.2963491
39. Viswanathan H, Mogensen P (2020) Communications in the 6G era. IEEE Access 1. https://doi.org/10.1109/ACCESS.2020.2981745
40. Wu H, Lyu X, Tian H (2019) Online optimization of wireless powered mobile-edge computing for heterogeneous industrial internet of things. IEEE Int Things J 6(6):9880–9892
41. Yaacoub E, Alouini M (2020) A key 6G challenge and opportunity-connecting the base of the pyramid: a survey on rural connectivity. Proc IEEE 1–50. https://doi.org/10.1109/JPROC.2020.2976703
42. Yang P, Xiao Y, Xiao M, Li S (2019) 6G wireless communications: vision and potential techniques. IEEE Netw 33(4):70–75. https://doi.org/10.1109/MNET.2019.1800418
43. Zhang L, Liang Y, Niyato D (2019) 6G visions: mobile ultra-broadband, super internet-of-things, and artificial intelligence. China Commun 16(8):1–14. https://doi.org/10.23919/JCC.2019.08.001
44. Zhang Z, Xiao Y, Ma Z, Xiao M, Ding Z, Lei X, Karagiannidis GK, Fan P (2019) 6G wireless networks: vision, requirements, architecture, and key technologies. IEEE Veh Technol Mag 14(3):28–41. https://doi.org/10.1109/MVT.2019.2921208
45. Zhou Z, Chen X, Li E, Zeng L, Luo K, Zhang J (2019) Edge intelligence: paving the last mile of artificial intelligence with edge computing. Proc IEEE 107(8):1738–1762
46. Zong B, Fan C, Wang X, Duan X, Wang B, Wang J (2019) 6G technologies: key drivers, core requirements, system architectures, and enabling technologies. IEEE Veh Technol Mag 14(3):18–27. https://doi.org/10.1109/MVT.2019.2921398

Analysis and Comparison Between Reversible DS GATE and Irreversible NXOR Logic Circuit

Tanuja Pande and N. K. Shukla

Abstract In today's scenario, the system is configured with electronic simulated modules. To run this kind of system, a lot of functional digital electronic elements are required. Especially this specific configuration of holding some data in them for some period so that the system could perform some intelligent works. This field is termed as digital signal processing, used in various phases like smart doors, or mobile, and there are countless many folds in its merit list. Here, in the study, the function is to manage the circuit to limit elements while maintaining the efficiency on output calibrations. The main concern of this study is quantum computing in addition to make it more efficient, and optimize the leftover residues. The system here is confined to the functional ability and the concern reversibility in the logic gates. Not all of them but only few are studied under reversibility condition. The proposed system has been constructed to make the analysis clearer.

Keywords Reversible logic · Reversible gate · Power dissipation · Dynamic power · Power leakage

1 Introduction

The structural foundation of digital electronics is the basic function of a flow of electronics in the semiconductors. These electrons when flow they collide this each other this generates the energy. This energy changes its form to heat. This energy affects the performance of the electronic elements [1]. Here, the function of it is negative, resulting from the function of the device inefficient due to its heat dissipation. No doubt the heat can be managed by a heat sink but it is increasing the losses in the form of heat. This means some parts of the input are wasted in this. So, for this purpose, the reversible computing was introduced. So, to these aspects, the laws to create devices that do not have this problem in them are researched [2]. Some new phenomena were brought to performance. These methods were working functionally and the system is working on core level. This brings nearly lossless devices. This

T. Pande (✉) · N. K. Shukla
University of Allahabad, Allahabad, Uttar Pradesh, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_17

is working with quantum computational working. This brings reversibility in the circuit. This ideology is so efficient that it reduced the operating time and tediously reduced the size [3]. As now, the heat is not an issue to be concerned of.

So, if reversible is studied, question arises that what are reversible circuits. So, the answer to this question is the function designed in a digital module is said to be reversible if one output is linked to a particular one input point. This statement itself explains how the heating problem is eliminated to its own aspects [4]. So now input gate has equal quantity linked to output points in the system configuration.

After this, the designing comes to the picture. It is different. Not same as the irreversible logic which are conventionally used. In the structural manner [5]. Function is different in configuration. Especially, in the binary signals, the system is unconfined to shorter regions.

Due to the above explanation, the operation of irreversible circuits is different. It does not need reversible network in it. On the other hand, the reversible circuits are different they work with the help of a signal which is duplicate to the original of the original signal [6]. In addition to duplicate signal, reversible networks also have Boolean function to address, which is obviously a non-balanced one unlikely the irreversible circuits. Where the highly efficient devices are needed, a constant signal is passed through some of the inputs, but this is not compulsory in the devices placed in less efficient requirements [7]. The fundamental unit of reversible circuits is also a gate. So, discussing the gate, it is an input-output device with multiple inputs and multiple outputs. There are many types of gate available, but mostly universal gate is preferred in the reversible circuits. This gate has a connection condition that it should have at least three inputs [8]. Most efficient of all is 3-3 (input-output) circuit.

Here, the functional study is done regarding the reversible M-M (input-output) circuitry design with Boolean function applied [9]. The effectiveness of the designed gate is calculated by the quantity of relations a circuit can handle in a particular gate module. So, by this, it is important to understand that the termed meaning of 1-1 gate is that out of all only 2-2 will be trivial. This is much likely to be a partial outcome but this is the only way these gates work. Due to these phenomena, the effectiveness of the circuit is noted in the form of energy losses, which is a kind of crucial consideration in the point of signal analysis [10]. But as the advancement in the technology is discovered and the use of lossless material in the module in addition to the special, cooling techniques, the heat problem is nearly out of the box now. With this upgrade in last few years, reversible circuits came up with rising colors in field of low power transfer with minimal power dissipation, which is a useful beam for the sectional viewpoint of VLSI designing [11]. Still it has widened scope in may fold of application like a under low power CMOS and Optical information processing, quantum computation and nanotechnology. On the other hand, the old technique, i.e. irreversible hardware circuitry, is losing its grip for the section [12, 13]. A study done on the energy lost in the irreversible circuit working is nearly KT (ln2) joules.

Here $K = 1.38065 \times (10^{-23})$ m^2 kg^2 K^{-1} (joule/Kelvin-1), this K is termed a Boltzmann's constant and T is the temperature in kelvin.

A study also showed that energy will not loose from the module until it makes to operation of reproducing inputs for some particular outputs [14]. These loops can be eliminated if the computation is aligned to get the information regarding the system performance. In such case, the reversible circuitry enhances the movement of data in either direction. So, this gives a plus point to the reversible points that the function is campaigned for the point in last operation and also moves for advance commands for inputs [15, 16].

2 Proposed Methodology

All the research is done to increase the outcome of the conventionally accepted module [17]. One of such is the NOT gate. Technically it is 1-1 (input-output) gate [18]. The basic structural phenomenon of the NOT gate is to reverse the input and deliver the opposite function [19]. It just multiplies the input with the multiplier of (-1). This takes some time to perform the operation and deliver the response [20]. NOT gate can be utilized in two forms, one is controller and other is uncontrolled. The controlled NOT gate is also termed as the Feynman gate [21]. This is a technical name, in most cases, it is accepted as the NOT gate. As discussed above, the reversibility circuits have a perk, that they preserve the information. This help in regaining the data from backwards computed a set of data as a record of the system. This is very appreciated in the system and helped in the form of logical resigning of the circuit so can also be termed as logical reversibility [22]. One such kind is the physical reversibility; in this kind of the reversibility, power is not eliminated in the form of heat. But sky is the limit, no module can be the ideal, they all have some error/losses. So theoretically, it is possible but practically making a precise reversible is not possible [23]. As the functions are observed by the module, the system generates energy, which is dissipated in the form of heat. This energy is needed by getting changed in heat, which results in reduction in efficiency. Mostly, this occurs when input varies its value from positive to negative. Here, reversible circuits deal this situation with some different manners in spite of making difference in the voltage grade; it changes the absorbing input point for that particular part. This step is the only point where it gets the energy to be transformed to heat and also consume time. This type of circuitry changed the way of orientation in the field of function bases. Reversible logic elements are needed to recover the state of inputs from the outputs. Due to this point, the section changed the programming and the working procedure of the existing modules, but in the better way and in right direction [20]. The new system designed to make the work easy should be working on the principle of this reversible structure. The following characteristics are needed for this.

They are

- Fan-out is not permitted.
- Loops or feedback are not permitted.
- Garbage outputs must be minimum.
- Minimum delay.

- Minimum quantum cost.

3 Need of Reversible Computing

The above discussion stated some merits of reversible computing technique, which of them were, cheap, efficient, low power consumption and most importantly they hold the information in it and also this is confined to the fractional viewpoint [20]. The unconventional platform can still hold on to irreversible circuits, but for the development and profit point of view, only reversible networking is the option for us to handle the justification to all the question of digital processing. This can maintain the pace of work and speed. All the computational factors can only be triggered by this section of signal modeling [18]. This is the option for resource utilization in an optimum manner. This reduces the size as there is no need for heat sink anymore [17]. This cut half of the size resulting saving in raw material.

4 Fault Tolerant Reversible Logic Gates

The reversible networking is capable to handle any fault. As it endurance to it is quite large, it can perform if its partial part is affected due to malware. This shows its superiority. Inspire of this feature this can handle a fault which occurs on single pin and let the other pins function on there on this practice is termed as parity. This parity is so efficient, that if the system allows it to run throughout the time, there will be no requirement for any other kind of functionality for protection. Do its efficiency, it is termed to operate in Nanotechnology in the robotic part. The parity can be considered as the shield for the reversible circuits for future module. The networking is considered as protected for the system along the gate for parity. So, in short, the system will need to be preserving the system.

5 Proposed Reversible DS Gate

Various 3 * 3 opposite logics have been presented in journal by means of different implementation and effective rate. The research work presented 3 * 3 opposite logic known as DS opposite logic with its different implementation intended to series production. Figure 1, that is new logic, depicts the 3 * 3 DS opposite logic. The truth table is illustrated of opposite logic in Table 1. After investigating it be detected that achieved outcome plan which has a different picture of I/P plan that is some I/P plan differently produced by their subsequent O/P.

The presented opposite DS logic could be utilized in counterparts. It is a serial apparatus, which is amalgamated with FF. Counterparts as its term denotes that it is a

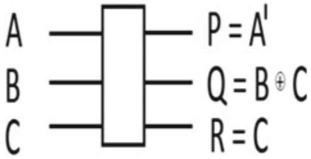


Fig. 1 Proposed reversible DS logic

Table 1 TRUTH table of 3 * 3 proposed reversible DS gate

A	B	C	P	Q	R
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1

mechanism that calculates moreover in growing, decreasing and in some particulars categorize planned through exclusive. At any time, some I/P chronometer rhythm prompt oppose subsequently yield achieves be a dual numeral of particular including series. At each growing, perimeter of chronometer rhythm O/P of counterparts be the after that numeral in series. It has various implementations of counterparts for example: in our routine livelihood viz. household apparatus that is utilized for time estimation. With all the other above-elaborated implementations, this serial apparatus might be utilized to achieve wave of dissimilar frequencies and prototype.

The EX-NOR logic utility is a digitized logic, which is opposite state of EX-OR task. The fundamental diagrammatic representation of EX-NOR logic is given as follows (Fig. 2):

Fundamentally, EX-NOR logic is an amalgamation of the EX-OR logic and NOT logic except it has a truth table, which is identical to the standardized NOR logic in which it has an outcome, which is generally unity and systematically becomes zero at the time, when any of the input has gate form “1”.

Though, a yield “1” be merely achieved when dual of its inputs are over the identical gate state, moreover dual “unity” or “zero”. For example: “double zero” or “eleven”. The input is the combined form and illustrates the Boolean appearance as follows:

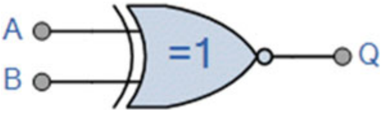


Fig. 2 Basic representation of EX-NOR gate

$$Q = (A' \oplus B') = A'.B' + A.B$$

Subsequently the outcome of the digitized EX-NOR logic merely “elevated” when dual inputs sides, A and B are at similar states which can be “unity” or “zero”. In addition, a flush numeral of gate “unities” on their inputs depicts the “unity” over the outcome, which is over the state “zero”. Consequently, this kind of logic depicts and outcome will be “unity” at the time of the logic are identical to every former that is the reason an EX-NOR logic is occasionally known as **correspondence logic**.

The sense sign in favor of an EX-NOR logic is merely an EX-OR logic by means of a round over their outcome to depict the NOT purpose. Subsequently, the gate EX-OR (A ⊕ B) logic is the opposite of the EX-OR logic we have observed earlier (Fig. 3).

The **EX-NOR logic**, too printed as: “Ex-NOR” or “XNOR”, purpose is obtained through combined standardized logic jointly towards shape extra composite logic purpose and an example: of a 2-input EX-NOR logic is depicted as follows:

The Digital Logic “Ex-NOR” Gate

See Table 2.

Giving the Boolean expression of: Q = A’B’ + AB.

The reason purpose replicated through a 2-input Exclusive-NOR logic is depicted as at the time, when dual A and B are identical, which illustrates an outcome at Q. In normal, an EX-NOR logic would depict an outcome parameter of logic “unity” merely at the time; it has an even numeral of unities on the inputs to the logic leaving when all their inputs are “short”. Consequently, as EX-NOR purpose with respect to greater than dual is known as “even purpose”. This elaboration could be prolonged to be applicable to some numeral of person inputs as depicted follows for 3 * 3 input EX-NOR logic (Table 3).

Fig. 3 Input EX-OR plus NOT gate

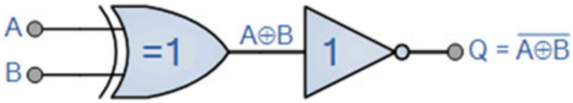


Table 2 2-input EX-NOR gate truth table

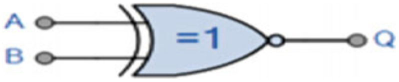
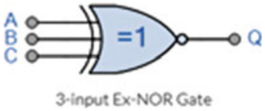
	Truth table		
 2-input Ex-NOR Gate	B	A	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	1
Boolean Expression $Q = \overline{A \oplus B}$	Read if A AND B the same gives Q		

Table 3 Truth table of 3 * 3 proposed 3-input EX-NOR gate

Symbol	Truth table			
	C	B	A	Q
	0	0	0	1
	0	0	1	0
	0	1	0	0
	0	1	1	1
	1	0	0	0
	1	0	1	1
	1	1	0	1
Boolean Expression $Q = \overline{A \oplus B \oplus C}$	1	1	1	0
	Read as any EVEN number of inputs gives Q			

$$Q = A'B'C' + ABC' + AB'C + A'BC$$

Here, it is depicted that former the EX-NOR purpose is a combined form of various fundamental gates EX-OR and a NOT logic, and through utilizing the dual-input truth table earlier, it can be elaborated the EX-NOR purpose as:

$$Q = A' \oplus B' = (A.B) + (A'.B').$$

This interprets that the logic could be realized by the specific logic by utilizing the novel equation (Fig. 4).

The main drawback of making the EX-NOR logic is that it consists of three various kinds of logic, i.e. AND, NOT and lastly OR logic in the very beginning construction. It is easier to make the EX-NOR logic by NAND logic as depicted below (Fig. 5):

Ex-NOR gates are used mainly in electronic circuits that perform arithmetic operations and data checking such as *Adders, subtractors* or *Parity Checkers*, etc. As the Ex-NOR gate gives an output of logic level "1" whenever its two inputs are equal, it is accustomed to compare the magnitude of two binary digits or numbers and then Ex-NOR gates are utilized in Digital Comparator circuits (Fig. 6).

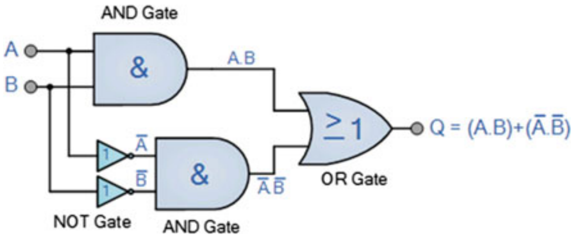


Fig. 4 Ex-NOR gate equivalent circuit

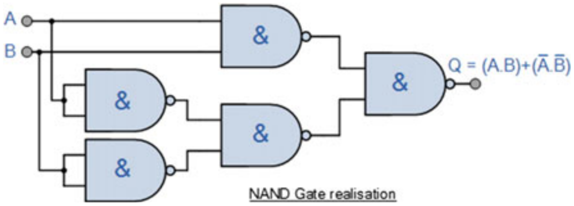


Fig. 5 Ex-NOR function realization using NAND gates

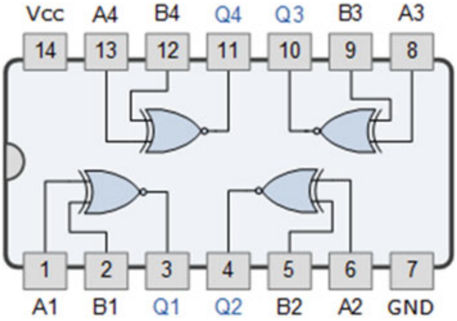


Fig. 6 74266 Quad 2-input Ex-NOR Gate

6 Comparison and Results

In favor of producing outcomes and have comparative analysis with former approaches that are opposite for that “Fredkin” logic be utilized. The following table illustrates the pursuance of the presented gate, i.e. DS gate as compared to former approached by application of MATLAB/SIMULINK replica, which is represented in Fig. 7. In opposite gate that is taken above in that opposite logic is utilized to achieve the expected task of counteract, due to its capability to utilize low energy. For any case, DS gate utilizes low control. The advanced gate that is presented takes only advance cost 2 whereas Fredkin logic takes 5. The presented logic can

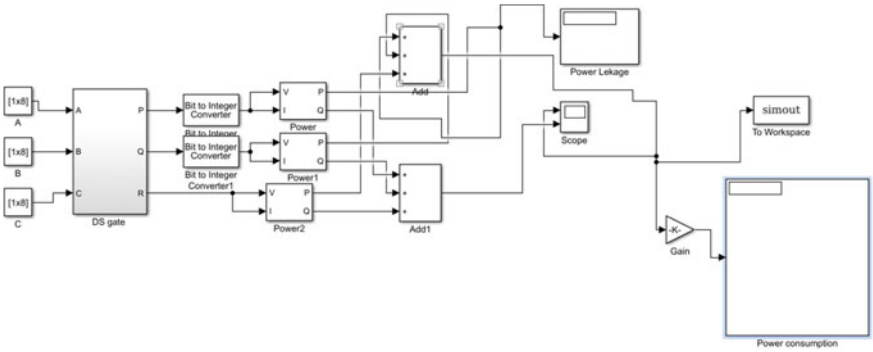


Fig. 7 Novel reversible gate model

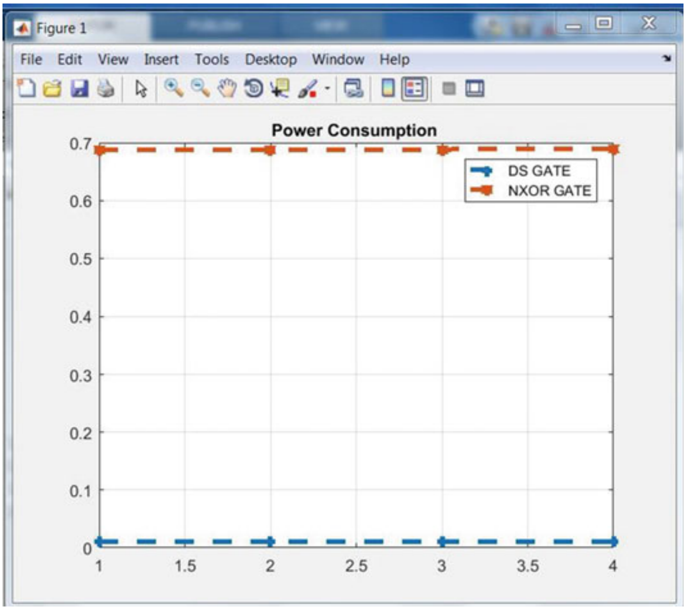


Fig. 8 Power consumption curve of both power

be formed by assuming one NOT and one CNOT logic whereas, Fredkin logic is formed assuming 2CNOT and 1 Toffoli gateway. It is clear from the following truth table and graphical representation that DS reversible gate consumes less power than EX-NOR irreversible gate. DS gate can efficiently solve complex operations, so this gate is far better than irreversible gate in all aspects (Figs. 8, 9, 10, 11, Table 4).

7 Conclusion and Future Scope

The main thought proposed in this research is to present 3 * 3 reversible DS gate. The presented DS gate is used to optimize the construction of counters. It is absolutely clear by the analysis that the proposed DS gate logic is far improved than the preexisting logic of counter with respect to power utilization and junk yield. The effective reversible sense is crucial and has applicable in different areas for example: less wattage CMOS, classical computing, microtechnology and visual computing. The utilized logic gate could also be utilized to construct huge opposite scheme. The research work also gives the comparative analysis among reversible and irreversible gates in which the reversible gate that is DS gate gives better energy performance and proved to be efficient.

It also consumes less energy. In irreversible gate, we have not used EX-NOR gate, which does not give those much efficient outcomes as DS gate logic. The

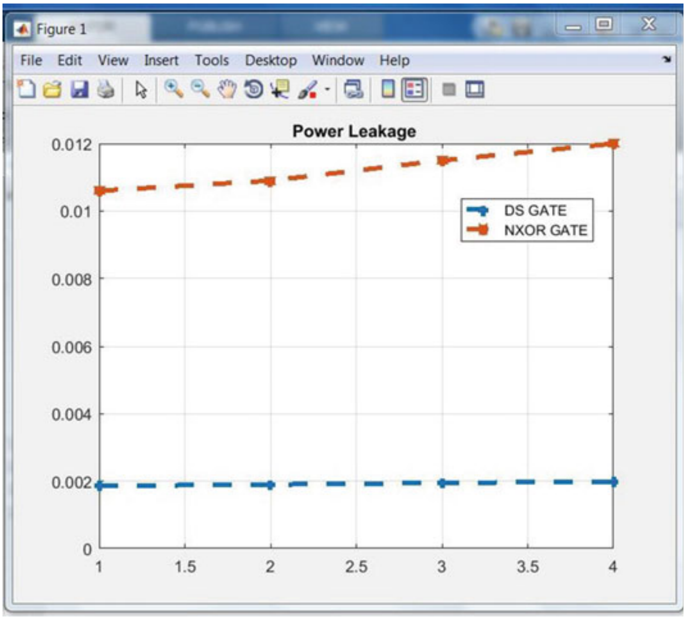


Fig. 9 Power leakage of both gates

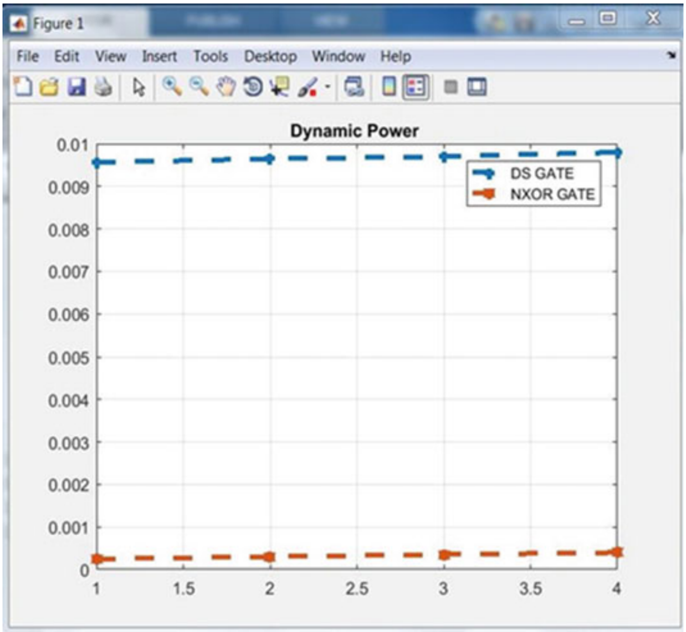


Fig. 10 Dynamic power of both gate

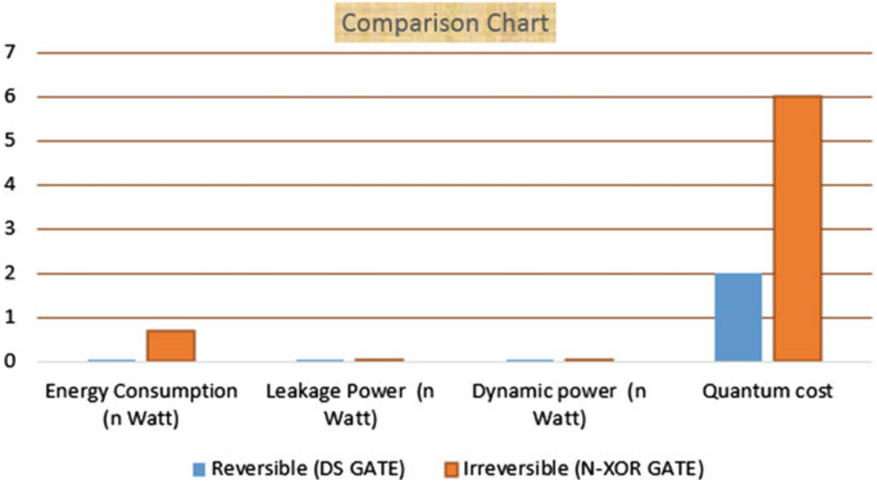


Fig. 11 Comparison graph to show the distinction between two gates

Table 4 Truth table of 3 * 3 proposed reversible DS gate

Parameters	Reversible (DS GATE)	Irreversible (N-XOR GATE)
Energy consumption (n Watt)	0.01016	0.6878
Leakage power (n Watt)	0.00186	0.01056
Dynamic power (n Watt)	0.00956	0.00025
Quantum cost	2	6

reversible gate is able to handle complicated circuitry. Lastly, despite the fact that the comparative analysis has been performed between former approach and novel technique which is applied in constructing the complicated scheme which is utilized in microtechnology. Additionally, paper goals are to achieve the specialized characteristics and to study the system if it is cost-effective. The beginner scientists would have gone through the roots of digital circuits and utilized them for optimizing purposes and for that it has used different approaches.

References

1. Lindauer R (1961) Irreversibility and heat generation in the computational process. IBM J Res Dev 5:183–191
2. Gaur HM, Singh AK, Ahankara U (2018) In-depth comparative analysis of reversible gates for designing logic circuits. In: 6th international conference on smart computing & communications 2017, vol. 125, pp 810–817
3. Bennett C (1973) Logical reversibility of computation. IBM J Res Dev 17:525–532
4. Toffoli T (1980) Reversible computing. Springer Berlin Heidelberg

5. Fredkin E, Toffoli T (1980) Conservative logic. Int J Theor Phys 21:219–253
6. Peres A (1985) Reversible logic and quantum computers. Phys Rev 32(6):3266–3276
7. Parham B (2006) Fault-tolerant reversible circuits. Signals Systems and Computers. ACSSC'06. Fortieth Asilomar Conference on. IEEE
8. Thapliyal H, Prasad Vinod A (2007) Design of reversible sequential elements with feasibility of transistor implementation. Circuits Syst ISCAS 2007. IEEE international symposium on. IEEE
9. Muallem P et al (2014) Optimized reversible arithmetic logic units. J Electr (China) 31(5):394–405
10. Sharman F, Mitra RK, Hasan R, Rahman A (2013) Low cost reversible signed comparator. Int J
11. Morrison M, Lewandowski M, Ranganathan N (2012) Design of a tree-based comparator and memory unit based on a novel reversible logic structure. VLSI (ISVLSI), 2012 IEEE computer society annual symposium on. IEEE
12. Thapliyal H, Ranganathan N (2009) Design of efficient reversible binary subtractors based on a new reversible gate. VLSI, 2009. ISVLSI'09. IEEE computer society annual symposium on. IEEE
13. Naaman AN, Jayashree HV, Bhagya Lakshmi HR (2011) Novel low power comparator design using reversible logic gates. Indian J Comput Sci Eng 2(4):566–574
14. Claudio M, Hadjam FZ (2012) On double gates for reversible computing circuits. Proc Intl Workshop on Boolean Problems
15. Majid H et al (2008) Design of a novel reversible multiplier circuit using HNG gate in nanotechnology. World Appl Sci J
16. Morrison, Matthew, and Nagarajan Ranganathan. "Design of a reversible ALU based on novel programmable reversible logic gate structures." VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on. IEEE, 2011.
17. Lindauer R (1961) Irreversibility and heat generation in the computing process. IBM 1 Res Develop 5(3):183–191
18. Bennett CH (1973) Logical reversibility of computation. IBM Res Develop 17:525–532; [3] Saiful Islam Md, Member, IACSIT, Rahman MM, Begum Z, Hafiz MZ (2011) Fault tolerant variable block carry skip logic (VBCSL) using parity preserving reversible gates. Int J Comput Electr Eng 3(1)
19. Moore GE (1965) Cramming more components onto integrated circuits. Electronics 38(8)
20. Islam MS, Rahman MM, Begum Z, Hafiz MZ (2009) Low cost quantum realization of reversible multiplier circuit. Inf Technol J 8(2):208–213
21. Joziah PL, Tishchenko A, Al-Rabadi A, Coppola A, Buller A, Song X, Khan M, Yanushkevich S, Shmerko VP, Chrzanowska-Jeske MA (2001) General decomposition for reversible logic. In: Proceedings of the international workshop on methods and representations (RM)
22. A comparative study of reversible logic gates (2012) Int J VLSI Signal Process Appl 2(1):51–55, ISSN 2231–3133
23. Praveen RS, Tilak G (2011) Design of optimized reversible BCD adder/subtractor. IACSIT Int J Eng Technol 3(3)

Deep Neural Models for Early Diagnosis of Knee Osteoarthritis and Severity Grade Prediction

Tilak N. Shenoy, Mathew Medayil, and Kamath S. Sowmya

Abstract Osteoarthritis (OA) is a type of arthritis that results in malfunction and eventual loss of the cartilage of joints. It occurs when the cartilage that cushions the ends of the bones wear out. OA is the most common joint disease which frequently occurs after the age of 45 in case of males and 55 in the case of females. Manual detection of OA is a tedious and labour-intensive task and is performed by trained specialists. We propose a fully automated computer-aided diagnosis system to detect and grade osteoarthritis severity as per the Kellgren-Lawrence (KL) classification. In this paper, we experiment with various approaches for automated OA detection from X-ray images. Image-level information such as content descriptors and image transforms are identified and assigned weights using Fisher scores. KL-grade is then projected using weighted nearest neighbours, and different stages of OA severity are classified. Pre-processing, segmentation, and classification of the X-ray images are achieved using data augmentation, deep neural network, and residual neural networks. We present experimental results and discussion with respect to the best-performing models in our experiments.

Keywords Osteoarthritis · Severity prediction · Deep neural networks · Precision medicine · Medical informatics

1 Introduction

Osteoarthritis (OA) is one of the most ordinary bone joint disorders, an extensively persistent condition that causes substantial discomfort and pain among afflicted ageing populations. Knee OA causes a set of heterogeneous symptoms that act on all joint tissues of the knees. According to a study conducted in the Indian subcontinent [1], there is a widespread occurrence of knee OA in rural areas, to the extent of 13.7%, due to the hardcore, stressful physical labour in the rural settings. In urban areas,

T. N. Shenoy (✉) · M. Medayil · K. S. Sowmya
Department of Information Technology, Healthcare Analytics and Language Engineering (HALE) Lab, National Institute of Technology Karnataka, Surathkal 575025, India
e-mail: sowmyakamath@nitk.edu.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_18

it is reported to be 6.9%, which can be attributed mostly to lifestyle choices, badly designed work environments, overwork etc. Community surveys conducted in India alone have reported that the occurrence of knee OA among the ageing population is quite prevalent, in the range of 17–60.6% [2].

The assessment of particular and articular structures of knees are available through diagnostic methods like magnetic resonance imaging (MRI) and X-rays. X-ray is the most widely used tool because it is inexpensive and is also easily accessible, when compared to other modalities [3]. From the X-rays, a trained radiologist/orthopaedician will manually note their observations by measuring factors like narrowing of the tibial–femoral joint space or presence of osteophytes. Based on their observations, the joints are classified into one of the five grades as per the standard Kellgren-Lawrence (KL) classification system [4]. On the KL scale, Grade 0 represents normal/absence of OA. Grade 1 records mild or early onset OA, while Grade 4 represents the most severe case of OA. This classification considers osteophyte features (growth of bones adjacent to the joint), narrowing of part or all of the tibial–femoral joint space and sclerosis of the subchondral bone. More information is extracted from KL classification as compared to the above three indicators considered separately.

Figure 1 illustrates the KL grading system with some sample X-ray images. A normal joint is classified as grade 0, doubtful OA is represented as grade 1, minimal OA is shown as grade 2, and grade 3 and grade 4 show moderate and severe OA, respectively. As we can see from the figure, useful information for OA detection is localized around the joint. The samples demonstrate that the X-ray scans are similar for the initial KL-grades to an untrained eye. Even the experts find it burdensome to classify a particular scan by assigning a KL-grade to it, especially in case of early onset OA. Due to its debilitating nature, detecting the onset of OA early can help in designing proper medical interventions and changing lifestyle choices for the afflicted person. Since there is no cure for OA, early detection is the only way by which the person's quality of life can be adequately managed, despite the progress of the disease. The need of automated techniques that can recognize the presence and severity of OA for computerized image analysis is critical. To improve chances

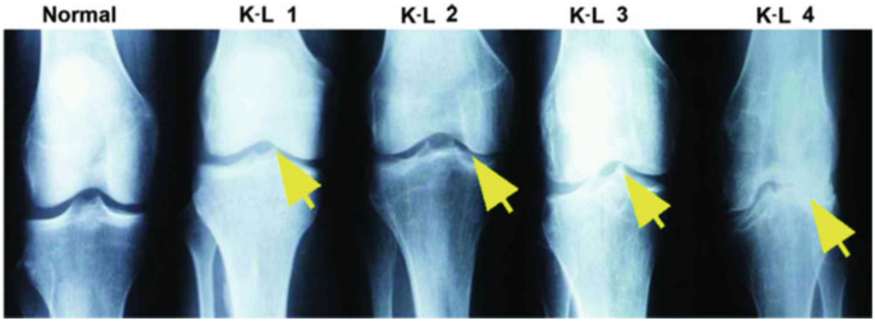


Fig. 1 Knee OA severity classified using KL grading system [5]

of early diagnosis as well as reduce the tedious work of overworked radiologists, such automated systems can function as clinical decision support tools, which can learn from historical data and improve predictions to reduce time of diagnosis and treatment too [6, 7].

In this paper, an automated knee OA detection and grading approach using X-ray images is proposed. The available data set is augmented and is used to train a modified VGG16 and modified ResNet model to perform feature extraction, segmentation, and finally, classify each scan. The remainder of this paper is structured as follows: Sect. 2 presents a discussion on the existing works in the area of osteoarthritis detection and prediction. In Sect. 3, the proposed methodology is explained in detail. Section 4 presents the experimental results and analysis followed by conclusion and future directions for research.

2 Related Work

Detection of the presence and severity of osteoarthritis has seen active interest from the research community. Mengko et al. [8] used the areas of interest from the visual characteristics in knee OA images and used an edge-based segmentation method to detect such regions of interest. Basically, the distance between femur and tibia bone that is used to grade the disease is calculated. This system helps doctors realize the region of interest and give the accurate measurement of unimpaired joint space width. After the edge detection and ROI detection, extraction of joint space width is found to determine if joint space narrowing is present. Classification process involves using a neural network consisting of 2 layers and 2 neurons while training using back propagation. Only 53 radiographs are used in the classification process; thus, a small data set was used for training and classification but did not focus on detection of subchondral sclerosis and osteophytes' formation. Shamir et al. [9] used a classifier to classify the radiograhic images into four KL-grades. Image analysis was done by extracting features and assigning weights to it, after which they are categorized using K-nearest neighbours (KNN). The images were first passed through a joint detection phase where using pixel intensity, the joint space between the bones was calculated. Using a large set of image feature extraction algorithms, a Fisher score was produced to remove all features with noise, followed by the application of simple weighted nearest neighbour rule. The proposed system worked well for KL-grade 2; however, for KL-grade 1, it was mediocre, and for KL-grade 4, it completely failed due to severe symptoms and feature extraction complexity.

Thomson et al. [10] proposed a method to detect OA that combines features from both bone image texture and appearance of tibia, for training two random forest classifiers, whose weighted sum was calculated. Their system uses 500 knee radiographs. Images were first used for shape model matching where they were passed to a RFCLM model trained and tested using halves for automated analysis. Using region selection, fractal signature and simple pixel features texture information was gathered. The classifiers were compared based on shape information and texture data, which ultimately showed that combining both increases the overall classification performance.

Stachowiak et al. [11] observed that in early stages of OA, Trabecular bone undergoes thinning and fenestration, whereas in the later stage, the bone thickens. They developed an automated system which chooses the areas with TB texture on knee and hand radiographs. Firstly, the ROI is checked for in the images, and then, variance orientation transform (VOT) is used to calculate fractal signatures for an accurate analysis of bone texture roughness. Using this, various changes in different types of knees based on different parameters were observed. A dissimilarity-based multiple classifier (DMC) is build to classify knee radiographs using features that are sensitive to OA.

Antony et al. [12] introduced a new approach that detects knee joints spontaneously using fully convolution neural network (FCN). Weighted ratio of categorical cross entropy and mean squared loss is optimized, and hence, OA severity is quantified by training convolution neural network from scratch. They used the OAI and MOST data sets. The dissimilarity in the distance between two knee bones and formation of osteophytes are detected with the help of knee localization using FCN. The model consists of four stages of convolutions with a max pooling layer after each convolutional stage. An upsampling and a fully convolutional layer follows the final stage of convolutions.

Brahim et al. [13] incorporated a pre-processing step based on a predictive model, in which multivariate linear regression is employed to reduce the inter-subject variability. Circular fourier filtering(CFF), histogram equalization and quantization, intensity normalization using multivariate linear regression (MLR), feature selection/extraction and classification are the various stages of the proposed CAD system. Independent component analysis was used to extract significant features from ROIs. The classification performances of the proposed CAD were tested using random forest and naive Bayes classifiers and the Leave-One-Out (LOO) cross validation strategy. Data set of 1024 knee radiograph images were used. However, the system only predicts 2 labels—KL-grade 0 (non-OA) and Kl-grade 2 (minimal), since it is harder to find the detection of OA at the early stages. Wahyuningrum et al. [14] classified knee X-rays using both CNN and LSTM. The input data for CNN is a sequential data using which LSTM is built. Stochastic gradient descent is used as a cost function which is optimized in order to train the LSTM network. The data set was taken from OAI with around 4796 participants. The high-level features are extracted best using VGG16 which helps LSTM to accurately classify between the two extremes of the KL-grades.

Based on the existing works discussed, we observed that various methods to detect OA have been proposed, but many of them do not focus on early OA detection. Some restrict the observations to addressing labelling whether the joint in the diagnostic scan shows signs of OA or not. For instance, Stachowiak et al. [11] uses only two labels to detect the presence of OA without assessing the severity. Since there is a very small difference between a normal joint and doubtful OA, the number of samples in training set should be high enough to get better accuracy. We intend to address these issues, thorough multi-label classification of OA with respect to all 5 grades of severity, as per KL Scale.

3 Proposed Methodology

In this section, we describe the proposed methodology in detail. For experimental validation of the proposed approaches, the KneeXRay data set [15] was used for training and testing. It contains 9810 knee X-ray images, out of which 3857 images are of KL-grade 0, 1770 images are of KL-grade 1, 2578 images are of KL-grade 2, while KL-grade 3 has 2162 images and KL-grade 4 contains 319 images. The data set is extracted from Osteoarthritis Initiative (OAI), which is a widely used data set in the context OA [16].

3.1 Pre-processing

Due to the limited number of images available in the original data set, it is inherently imbalanced, i.e. certain grade classes have very few images, while others had inordinately large number of training images. In order to improve the data distribution to enable detailed analysis of the model prediction, a much larger and well-balanced data set is essential [17]. To achieve this, we experimented with several data augmentation methods. Data augmentation is a pre-processing technique that is employed to significantly increase the diversity of the data available without actually adding new data that might introduce bias. Techniques like cropping, padding, horizontal or vertical flipping, shearing etc. are ways in which available data set images can be sufficiently altered to augment the data. We flipped the images along the vertical axis, then cropped them from 224 × 224 to 200 × 200, and finally, resized them back to 224 × 224 to balance it. These operations were performed using OpenCV. After the augmentation, the resultant training data set contained 2,304 images of KL-grade 0, 2,162 images of KL-grade 1, 2274 images of KL-grade 2, while KL-grade 3 and 4 contained 2295 and 1975 images, respectively.

3.2 Neural Classification Models

3.2.1 Convolution Neural Network (CNN)

Deep neural network architectures like CNNs have been successfully applied to classify 2D images [18]. CNN takes an image as input in the form of an array of pixel values and classifies it into a particular category. Every convolution layer consists

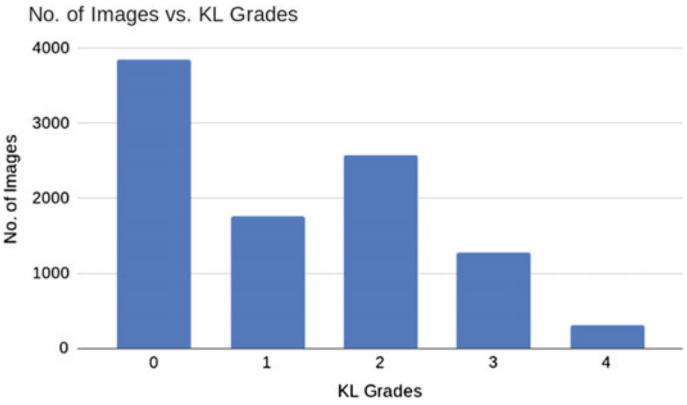


Fig. 2 Distribution graph before augmentation of the data

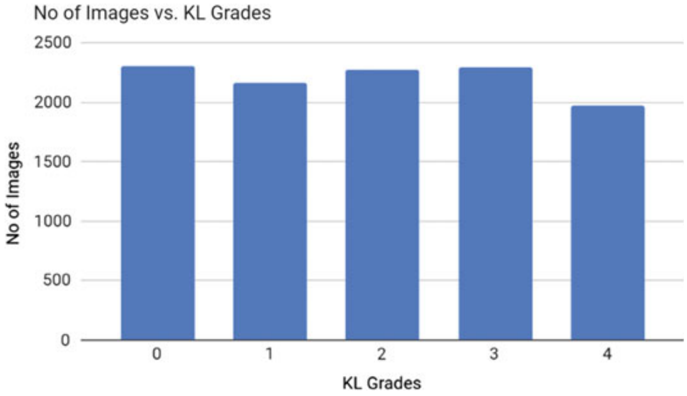


Fig. 3 Distribution graph after augmentation of the data

of filters, pooling, fully connected layers. Every image is passed through multiple convolution layers, and they are classified with a probabilistic value between 0 and 1 using softmax function. The convolution layer extracts the feature from the input image using small squares of input data. The size of this square is determined by the filter size which is an input to the convolution layer along with the image, while the pooling layer reduces the number of features in large images. We used max pooling layer that takes the largest element from the rectified feature map. We adapted the very well-known CNN architecture Visual Geometry Group-16 (VGG16), which has 16 layers—13 convolution layers, 3 dense layers, and 5 max pooling layers constituting a total of 21 layers, with 16 weight layers. Each layer in the architecture is depicted in Fig. 4. There are 64 filters in Conv 1, and this number of filters doubles as we go deep into the network. Hence, Conv 2 has 128, Conv 3 has 256 with an exception of Conv 4 and 5 which have 512 filters each. Our adapted custom network has the following combination of layers: 8 layers of Conv2D along with 4 max pooling

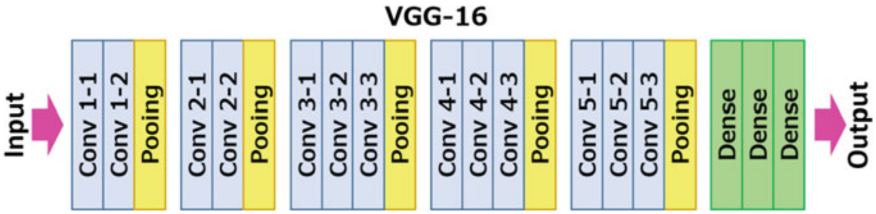


Fig. 4 VGG16 architecture

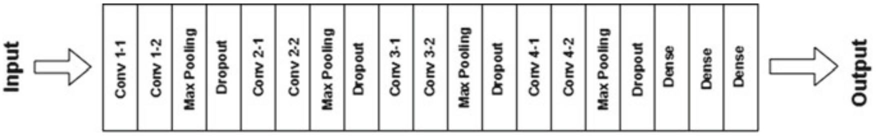


Fig. 5 Proposed custom model and its network layer definitions

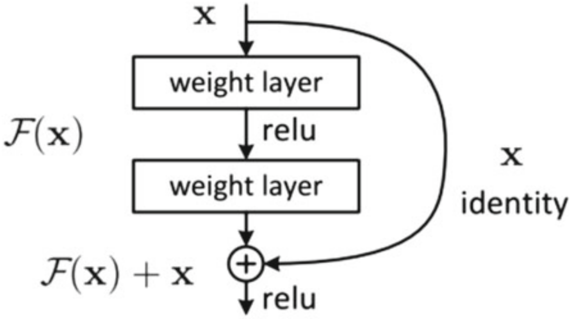
layers, 4 Dropout, and 3 dense layers. Conv 1 has 32 filters, Conv 2 has 64, and Conv 3 and 4 has 128 filters each. Figure 5 illustrates the architecture of the designed custom network.

3.2.2 Deep Residual Neural Network (ResNet)

Another CNN architecture that we have used is residual network (ResNet). The ResNet-50 comprises of four stages each with an identity and a convolution block, and each of these blocks has three corresponding layers. As a feed forward network with a single layer is enough to represent any function given enough capacity; however, the network might be vulnerable to overfit the data if the layer is not big enough. A deeper network can overcome this problem, but deeper the network, the harder it gets to train due to the vanishing gradient problem [19]. Hence, as network goes deeper, the performance gets saturated. Hence, ResNet was selected, as it tackles the vanishing gradient issue effectively.

Figure 6 shows the flow in a residual network in ResNet50 which consists of convolution pooling, activation, and fully connected layers stacked over each other. The core idea is to introduce an “identity shortcut connection” which omits one or more layers. From the different layered ResNet models, we chose ResNet50 as it worked best with the data set. There are four stages each with a convolution layer in ResNet-50. Every 2-layer block in the 34-layer network is replaced with a 3-layer bottleneck block, which results in 50-layer ResNet. All the residual network architecture executes the initial convolution of kernel size 7×7 and a 3×3 kernel-sized max pooling. Bottleneck design is used in deeper networks like ResNet50. For every residual function F, there are two 1 × 1 and a 3 × 3 layer stacked in alternate fashion. 1 × 1 convolution layers reduce and store the dimensions while 3 × 3 layer is left as bottleneck with lesser I/O dimensions.

Fig. 6 A residual block of deep residual network


x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

4 Experimental Results and Analysis

The proposed approach was validated by performing several benchmarking experiments with reference to state-of-the-art models on a standard data set, KneeXRay data set. Lilik et al. [20] used the hand-engineered features like GLCM, while Shamir et al. [9] employed Zernike features, mean, median etc. These features and many more features are automatically generated using deep neural networks. We train the VGG16 on the data set to classify the images with a batch size of 32 and 50 epochs, using TensorFlow.

It was observed that the performance of the VGG16 was very low, and hence, we created a custom model with 8 conv2D layers along with max pool and dropout layers between 2 conv2D layers each. The model performed very well in the later stages and decently in the initial stages of OA. The images are 224 × 224 greyscale images which are converted into numpy arrays and given as input to the network. Images of each grades in training set except for KL-grade 0 were augmented. The images with KL grade 1 has been flipped along the vertical axis. Images in the rest of the KL grades have been flipped along vertical axis and then cropped and then reshaped to the original dimensions. Figures 2 and 3 show the data distribution of the data set before and after data augmentation, respectively. As described in Fig. 4, there are two consecutive layers of Conv2D layers prior to max pooling and dropout layers, and the same set repeats for a total of four times after which there are three layers of dense layers. The individual layers were imported from Keras.

The same data set is used for training the deep residual neural network (ResNet). We used ResNet as it can resolve the vanishing gradient problem in deep neural networks. An untrained ResNet was loaded using Keras and trained with the preprocessed data set (we used CLAHE as well for contrast enhancement). The model was trained on the data set with different layered versions of ResNet, it was found ResNet50 worked best. To train the fine-tuned ResNet50, we used Keras with batch size of 64 and dropout (0.6). Also, the Adam optimizer with a categorical cross entropy for loss was used. The model was run in such a way to avoid overfitting in the best way possible. Next, data augmentation plus stratified K-fold was used to

Table 1 Benchmarking proposed models against state-of-the-art models

S. No.	Models	Accuracy (%)	Loss
1	Base VGG16	70.73	0.47
2	Base ResNet	43.50	0.50
3	Lilik et al. [20]	45.00	0.64
4	Wahyuningrum et al. [14]	75.28	0.09
5	Antony et al. [12]	63.40	0.66
6	Proposed CNN model	69.51	0.37
7	Proposed ResNet model	**92.02**	0.25

Bold value represents the result of the proposed model

train on generalized data with each fold (5 folds) using same ratio of class labels in training which significantly helped.

A multi-model approach was employed for each class separately to better understand individual accuracy for all five classes. Each model was trained with the same training set and then used to make predictions on the same validation/test set of images. On doing this, individual f1-score was recorded and parameters (epoch, batch size etc.) were tweaked to make sure there was no over-fitting and each model was actually making proper predictions to the same validation inputs. Initially, prediction probabilities were used to give out final prediction of KL-Grade, but each model was not weighted properly. The output of all five models (10 inputs) was then used for training the neural network, and predictions from this were used as final. Due to these optimization, an increases in performance over the base model was observed.

Table 1 compares the accuracy of the various approaches. The accuracy of VGG16 was 39.5% before data augmentation and 70.73% after the the augmentation, while Lilik et al. [20] reported an overall accuracy of 45%. The overall accuracy of the proposed customized convolution layered model was quite close to that of Wahyuningrum et al. [14] model, which achieved an accuracy of 75.28, while that of the proposed custom model was 69.51%. The ResNet model performed the best in class with respect to accuracy, with an overall accuracy of 92.02%. However, the overall loss parameter of both proposed models was slightly higher that than of Wahyuningrum et al. [14] model.

On implementing ResNet50 with the same data set, we found the accuracy to be 77.2%, while the accuracy for the validation set was only 43.5%. The poor accuracy was not due to overfitting in the model but rather due to the fact that the training data set was imbalanced with having "0" grade occurring the most. To avoid this, we compared the same model with one with data augmentation and stratified k-fold, due to which the overall accuracy improved by a huge margin to 92%. Also, another method using a 5-model ensemble was implemented to check individual accuracy of each model (label) for observing how to improve f1-score of model if required. The predictions of each model was pushed to a neural network to make sure the predictions of each model have appropriate weights due to augmented data set but previously imbalanced. This achieved an overall accuracy of 45

5 Conclusion and Future Work

In this paper, a fully automated computer-aided diagnosis system to detect and grade osteoarthritis severity as per the Kellgren-Lawrence (KL) classification was proposed. Image-level information such as content descriptors and image transforms were identified and assigned weights using Fisher scores. KL-grade was then projected using weighted nearest neighbours, and different stages of OA severity were classified. The base VGG16 and ResNet50 model performed poorly due to the imbalanced data set, but after the data augmentation, the performance of VGG16 has improved drastically. ResNet50 model also performs well but 5-model ensemble version requires further improvements. As part of future work, we aim to focus on improving the models so that more accuracy can be obtained in the task of early stage OA detection. This can be leveraged to treat patients at an opportune time and provide better quality of life.

Acknowledgements We gratefully acknowledge the use of the facilities at the Department of Information Technology, NITK Surathkal, funded by Govt. of India's DST-SERB Early Career Research Grant (ECR/2017/001056) to Sowmya Kamath S.

References

1. Fransen M, Bridgett L, March L, Hoy D, Penserga E, Brooks P et al (2011) The epidemiology of osteoarthritis in Asia. Int J Rheum Dis 14:113–21
2. Sharma MK, Swami HM, Bhatia V, Verma A, Bhatia S, Kaur G (2013) An epidemiological study of co-relates of osteoarthritis in geriatric population of Chandigarh. Indian J Community Med 32:77
3. Karthik K, Kamath SS (2020) A deep neural network model for content-based medical image retrieval with multi-view classification. Visual Computer 1–14
4. Kellgren JH, Lawrence JS (1957) Radiological assessment of osteo-arthrosis. Ann Rheum Diseases 16(4):494. https://doi.org/10.1136/ard.16.4.494
5. Liu Baolong, Zhang Maoquan, Zhao Jingming, Zheng Mei, Yang Hao (2018) Imbalance of M1/M2 macrophages is linked to severity level of knee osteoarthritis. Experim Therap Med 16(6):5009–5014
6. Vikram M, Anantharaman A, Suhas BS (2019) An approach for multimodal medical image retrieval using latent Dirichlet allocation. In: Proceedings of the ACM India joint international conference on data science and management of data
7. Vikram M et al (2018) Multimodal medical image retrieval based on latent topic modeling. In: 32nd Conference on Neural Information Processing Systems (NIPS), Montréal, Canada
8. Mengko TL, Wachjudi RG, Suksmono AB, Danudirdjo D (2005) Automated detection of unimpaired joint space for knee osteoarthritis assessment. In: Proceedings of 7th International Workshop on Enterprise networking and Computing in Healthcare Industry (HEALTHCOM), Busan, South Korea, pp 400–403
9. Shamir L et al (2009) Knee X-Ray image analysis method for automated detection of osteoarthritis. IEEE Transactions on Biomedical Engineering 56(2):407–415
10. Thomson J, O'Neill T, Felson D, Cootes T (2015) Automated shape and texture analysis for detection of osteoarthritis from radiographs of the knee. In: Navab N, Hornegger J, Wells W, Frangi A (eds) Medical image computing and computer-assisted intervention (MICCAI 2015)

11. Stachowiak GW, Wolski M, Woloszynski T, Podsiadlo P (2016) Detection and prediction of osteoarthritis in knee and hand joints based on the X-ray image analysis. Biosurf Biotribol 2(4):162–172. ISSN: 2405-4518
12. Antony J, McGuinness K, Moran K, O'Connor N (2017) Automatic detection of knee joints and quantification of knee osteoarthritis severity using convolutional neural networks. https://doi.org/10.1007/978-3-319-62416-7_27
13. Abdelbasset B, Rachid J, Rabiâ RIAD, Janvier T, Khedher L, Toumi H, Lespessailles E (2019) A decision support tool for early detection of knee osteoarthritis using X-ray imaging and machine learning: data from the osteoarthritis initiative. Computer Med Imag Graph 73. https://doi.org/10.1016/j.compmedimag.2019.01.007
14. Wahyuningrum RT, Anifah L, Eddy Purnama IK, Hery Purnomo M (2019) A new approach to classify knee osteoarthritis severity from radiographic images based on CNN-LSTM method. In: 2019 IEEE 10th International conference on Awareness Science and Technology (iCAST), Morioka, Japan, pp 1–6
15. Chen P (2018) Knee osteoarthritis severity grading dataset. Mendeley Data, v1 http://dx.doi.org/10.17632/56rmx5bjcr.1
16. Eckstein F, Wirth W, Nevitt MC (2012) Recent advances in osteoarthritis imaging-the osteoarthritis initiative. Nat Rev Rheumatol 8(10):622
17. Padmakumar V et al (2018) A robust approach to open vocabulary image retrieval with deep convolutional neural networks and transfer learning. In: 2018 Pacific Neighborhood Consortium Annual Conference and Joint Meetings (PNC). IEEE
18. Soundalgekar P et al (2018) Medical image retrieval using manifold ranking with relevance feedback. In: 2018 IEEE 12th International Conference on Semantic Computing (ICSC). IEEE
19. Karthik K, Sowmya Kamath S (2018) A hybrid feature modeling approach for content-based medical image retrieval. In: 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS). IEEE
20. Lilik A et al (2013) Osteoarthritis classification using self organizing map based on Gabor kernel and contrast-limited adaptive histogram equalization. Open Biomed Eng J 7:18–28
21. Spector TD, Cooper C (1993) Radiographic assessment of osteoarthritis in population studies: Whither Kellgren and Lawrence? Osteoarthritis Cartilage 1:203–206. https://doi.org/10.1016/S1063-4584(05)80325-5

The GNS1 Algorithm for Graph Isomorphism

Radu-Iulian Gheorghica

Abstract This paper presents a new algorithm that was developed for graph isomorphism, the main goal being obtaining the correct results with the best execution times. The things described are the utility and domains of usage for the algorithm, the nomenclature, the input data, the preconditions and graph definitions, the access of GNS1 to the query graphs and the data graph, the implementation details for the algorithm structure, methods and pruning techniques, then a series of test cases, the system specifications, the acknowledgment, the conclusions and the references. Motif graph finding has gathered increased popularity due to its vast domain of applicability. GNS1 is a backtracking algorithm with new pruning techniques for reducing the search space. It returns to the user all occurrences of a query graph found in a data graph while at the same time having much lower execution times than the STwig [1] and VF2 [2–4] algorithms. The algorithms can be used in a multitude of domains.

Keywords Graph · Subgraph query · Network · Motif

1 Introduction

The work consists of implementing [5] and testing the STwig and VF2 algorithms in an original way without using their authors' source code. Afterward, the GNS1 algorithm was created. Real-life examples will be presented and usage of a RI human protein to protein interaction data graph.

Supported by Dr. Pârv Bazil, Professor Emeritus, Dr. Niculescu Virginia, Associate Professor and Dr. Sterca Adrian, Lecturer Professor.

R.-I. Gheorghica (✉)
Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Mihail Kogălniceanu Street, nr. 1, Cluj County, 400084 Cluj-Napoca City, Romania
e-mail: radugheorghica@protonmail.com

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_19

2 Practical Applications of Motifs

The benefits of motif graphs searching according to [6] are the following:

1. Biochemistry, neurobiology, ecology and engineering.
2. Ecological food webs, genetic networks, for example, Escherichia coli and Saccharomyces cerevisiae.
3. World Wide Web.
4. Biomolecules within a cell and synaptic connections between neurons, for example, in Caenorhabditis elegans.

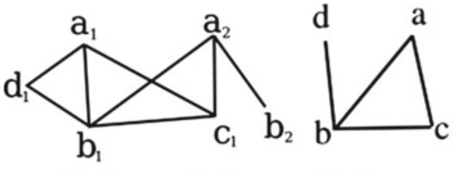
The authors of [6] specify that motifs could define universal classes of networks. If a problem can be modeled using graphs, then it can make use of motif graph finding.

3 Nomenclature

1. The terms "node" and "vertex" will be used interchangeably in the explanations.
2. Node label—There were used graphs representing human protein to protein interaction (PPI) [7]. The same label can be assigned to multiple nodes, or for the example in reality, a multitude of proteins can be associated with the same protein type.
3. Query graph—The input graph for the algorithm. The search in the data graph uses the label of each of its nodes and the adjacency of these nodes.
4. Data graph—In this graph, the algorithms execute the search.
5. Motif—In Fig. 1 there is an example of a data graph and a query graph. In Fig. 2, it can be seen that network motifs are patterns that recur much more frequently (A) in the real network than (B) in an ensemble of randomized networks [6].
6. Partial solution—A list which contains nodes. For each node of the query graph, the label is taken. For each position in the partial solution, there is a corresponding position in the query graph. A partial solution becomes a complete solution when the following conditions are met:
    (a) Each node has the label of the node on the same position in the query graph.
    (b) The number of nodes is equal to the one from the query graph.

4 Input Data for the Algorithms

The tests used first 10,000 edges from the RI human protein to protein interaction graph [7] with a total of 4652 nodes for use as the data graph. The query graphs used five graphs that have three nodes and then five graphs that have four nodes. These



(a) Data graph G (b) A query q

Fig. 1 Example of a data graph and a query graph [1]

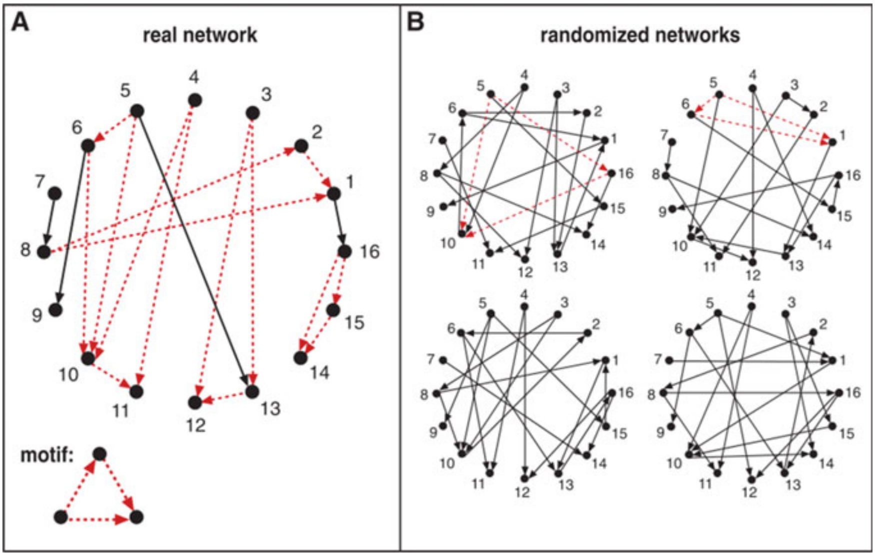
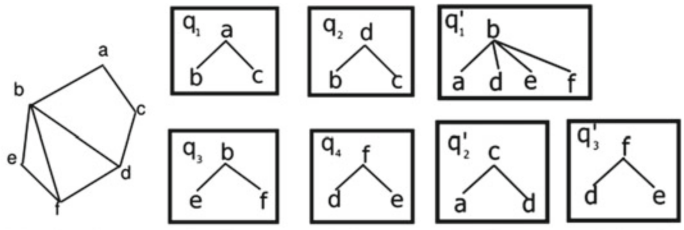


Fig. 2 Motif example [6]

queries have the adjacency of an STwig graph as shown in Fig. 3 where can be seen the decomposition of a query graph in STwigs by using the STwig-order selection algorithm [1]. In this example, the nodes have their ID values equal to lower-case letters and no labels. According to [6] in Fig. 4, examples of interactions represented by directed edges between nodes in some of the networks are used for their study. These networks go from the scale of biomolecules (transcription factor protein X binds regulatory DNA regions of a gene to regulate the production rate of protein Y), through cells (neuron X is synaptically connected to neuron Y), to organisms (X feeds on Y). Also, according to [6] in Fig. 5, all 13 types of three-node connected subgraphs are seen. According to [6], C is the number of appearances of the motif divided by the total number of appearances of all connected three-node subgraphs (Fig. 5). Subnetworks of size S were generated by choosing a node at random and adding to it nodes connected by an incoming or outgoing edge, until S nodes were obtained, and then including all of the edges between these S nodes present in the full network. Each of the subnetworks was randomized [8, 9] (shown are mean and SD



(a) The Query (b) Decomposition 1 (c) Decomposition 2

Fig. 3 Query graph decomposition in STwigs [1]

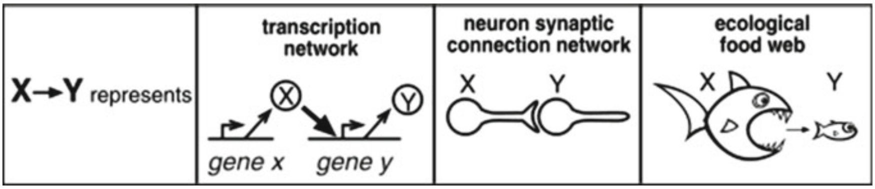


Fig. 4 Examples of interactions [6]

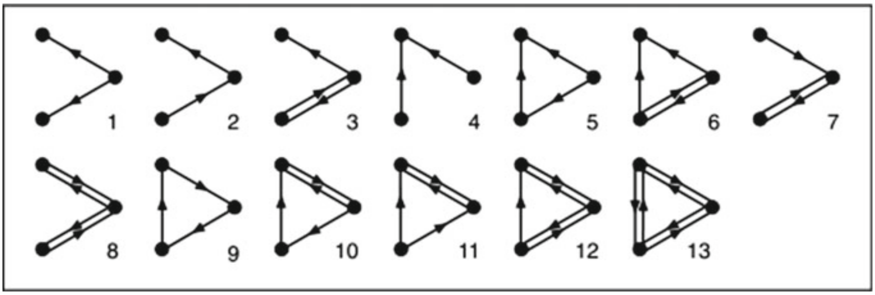


Fig. 5 All 13 types of three-node connected subgraphs [6]

of 400 subnetworks of each size). According to [1], the algorithm *MatchSTwig(q)* can be seen where $q = (r, L)$. An STwig query is denoted by q. The variable r represents the label of the root node, and L is the set of labels for the child nodes (Figs. 1, 2 and 6).

According to [1], the algorithm for query graph decomposition is called *STwig-order selection*(q): The input parameter q is the query graph from which the STwigs will be obtained. The function $f(v)$ is the *f-value of v* where v is a query node. The *f-value* is calculated with (1), where freq(v.label) is the number of vertices found in the data graph that have that label. For the deg(v) function according to [11] page 1169, the degree of a vertex in an undirected graph is the number of edges incident on it.

```
$S_r \leftarrow Index.getID(r)$
$R \leftarrow \emptyset$
**for** each $n$ in $S_r$ **do**
    $c \leftarrow Cloud.Load(n)$
    **for** each $l_i$ in $L$ **do**
        $S_{l_i} \leftarrow \{m | m \in c.children$ and $Index.hasLabel(m, l)\}$
    **end for**
    $R = R \cup \{\{n\} \times S_{l_1} \times S_{l_2} \times \cdots \times S_{l_k}\}$
**end for**
Return $R$
```

$$f = \frac{\deg(v)}{\text{freq(v.label)}} \tag{1}$$

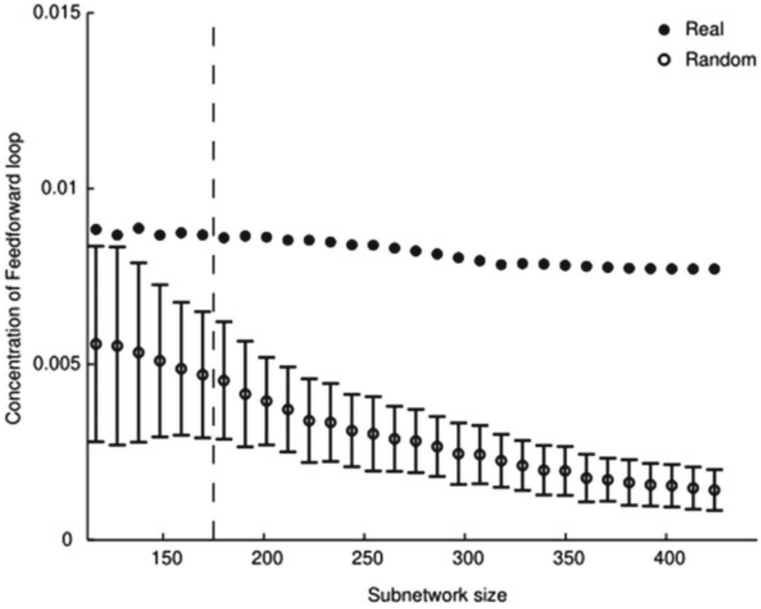


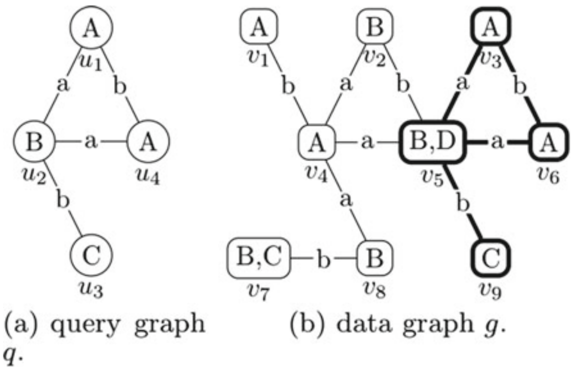
Fig. 6 Concentration C of the feedforward loop motif in real and randomized subnetworks of the *E. coli* transcription network [6, 10]

```
$S = \emptyset$
$\mathbb{T} = \emptyset$
while $q$ has more edges do
  if $S = \emptyset$ then
    pick an edge $(v, u)$ such that $f(u) + f(v)$ is the largest
  else
    pick an edge $(v, u)$ such that $v \in S$ and $f(u) + f(v)$ is the largest
  end if
  $T_v \leftarrow$ the STwig rooted at $v$
  add $T_v$ to $\mathbb{T}$
  $S \leftarrow S \cup$ neighbor($v$)
  remove edges in $T_v$ from $q$
  if $deg(u) > 0$ then
    $T_u \leftarrow$ the STwig rooted at $u$
    append $T_u$ to $\mathbb{T}$
    remove all edges in $T_u$ from $q$
    $S \leftarrow S \cup$ neighbor($u$)
  end if
  remove $u$, $v$ and all nodes with degree 0 from $S$
  return $\mathbb{T}$
end while
```

5 Preconditions and Graph Definitions

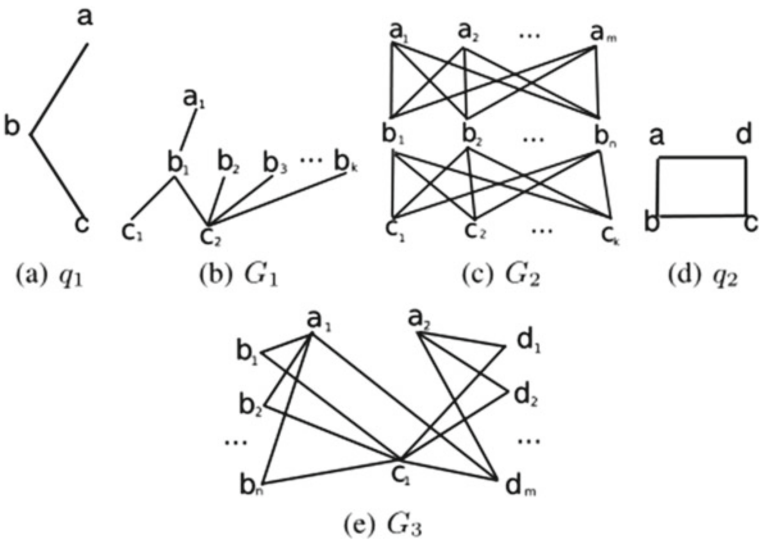
In the implementation and tests, there were used undirected graphs. In the example from Fig. 7, the emphasized occurrence of a query graph inside a data graph is seen. In Fig. 8, a few more examples can be seen of query graphs and data graphs. The subgraph matching is performed on a labeled graph. Let $G = (V, E, T)$ be a graph, where V is the set of vertices, E is the set of edges, and $T : V \rightarrow \Sigma^*$ is a labeling function that assigns a label to each vertex in V [1].

Fig. 7 Matching example [3]



(a) query graph q.

(b) data graph g.



(a) q_1 (b) G_1 (c) G_2 (d) q_2

(e) G_3

Fig. 8 Graph examples for discussion [1]

Theorem 1 *DEFINITION 1 (SUBGRAPH QUERY). The authors of [1] have denoted a subgraph query as* $q = (V_q, E_q, T_q)$*, where* $T_q : V \rightarrow \Sigma^*$ *represents the label constraint for each vertex in* V_q *[1].*

Theorem 2 *DEFINITION 2 (THE PROBLEM OF SUBGRAPH MATCHING). For a graph G and a subgraph query q, the goal of subgraph matching is to find every subgraph* $g = (V_g, E_g)$ *in G such that there exists a bijection* $f : V_q \rightarrow V_g$ *that satisfies* $\forall v \in V_q, T_q(v) = T_G(f(v))$ *and* $\forall e = (u, v) \in E_q, (f(u), f(v)) \in E_g$*, where* $T_G(f(v))$ *represents the label of the vertex* $f(v)$ *in G [1].*

6 Database

6.1 For the GNS1 Algorithm

The algorithm uses undirected graphs. For each query graph, a NetworkX [12] undirected graph object was created. For the data graph, he Neo4J Enterprise [13] graph engine was used. It can be configured to work as a single Neo4J instance database or set it up as a cluster using Docker [14], *docker compose* [15] and an *.yml* setup file. This file specified the use of Neo4J, the number of instances, the role of each one (core or read replica) [16] and the addresses with which these instances can be interconnected.

7 The New GNS1 Backtracking Algorithm

The structure of *GenericQueryProc* [2, 3] was followed.

7.1 *Pseudocode [2, 3]*

```
$M = \emptyset$
for each $u \in V(q)$ do
  $C(u) = FilterCandidates(q, g, u, \ldots)$
  $[[\forall v \in C(u)((v \in V(g)) \wedge (L(u) \subseteq L(v)))]]$
  if $C(u) = \emptyset$ then
    Return
  end if
end for
SubgraphSearch($q, g, M, \ldots$)
Subroutine SubgraphSearch($q, g, M, \ldots$)
if $|M| = |V(q)|$ then
  report $M$
else
  $u$ = NextQueryVertex($\ldots$)
  $[[u \in V(q) \wedge \forall(u', v) \in M(u' \neq u)]]$
  $C_R$ = RefineCandidates($M, u, C(u), \ldots$)
  $[[C_R \subseteq C(u)]]$
  for each $v \in C_R$ such that $v$ is not yet matched do
    if IsJoinable($q, g, M, u, v, \ldots$) then
      $[[\forall(u', v') \in M((u, u') \in E(q) \implies (v, v') \in E(q) \wedge L(u, u') = L(v, v')]]$
      UpdateState($M, u, v, \ldots$)
      $[[(u, v) \in M]]$
      SubgraphSearch($q, g, M, \ldots$)
      RestoreState($M, u, v, \ldots$)
      $[[(u, v) \notin M]]$
    end if
  end for
end if
```

7.2 *GNS1 Algorithm Structure and Methods*

GNS1 is a backtracking algorithm with pruning techniques that effectively reduce the search space for occurrences of the query graph. There were provided original pruning techniques for executing the search of complete solutions in the data graph. For each position of the partial solution, the algorithm is searching for a node in the data graph whose label is equal to the nodes in the corresponding position of the node in the query graph. Each candidate data graph node must be adjacent with the candidate root node that has the label equal to the root node of the query graph. For each position of the partial solution, the candidate data graph node placed in that position must have its label equal to the label of the node on the same position in the query graph list of nodes. According to *GenericQueryProc*, there are the following methods. Here are also described the distinct modifications that were brought:

1. *UpdateState(node, partial_solution)*
   A deep copy is made of the input node and of the partial solution. The algorithm appends the node to it and returns the updated partial solution. Appending the node to the partial solution is done in the following manner: A deep copy of the partial solution is made to which the algorithm then appends the data node. This is because simply initiating a new variable and the value of a list will just make a shallow copy of the said list. If a modification is made to the new list, it will also take place in the old list, so this has to be avoided.
2. *RestoreState(partial_solution)*
   A check is made first if the length of the partial solution is larger than zero. If it has any nodes, the last one will be removed, and a deep copy will be made of the new list and then return it.
3. *IsJoinable(data_node_to_be_joined, partial_solution, data_graph)*
   This is the most important method in the implementation. It is described in full detail in its own section below.
4. *NextQueryVertex()*
   It returns the next query graph vertex that will be processed.
5. *SubgraphSearch(partial_solution, query_graph_dict, current_node, data_graph)*
   This is a recursive method which brings together the methods for building the partial solutions and verifying if each of them can be declared a complete solution that can then be appended to the list that contains all such solutions.

7.3 Pruning Techniques and Input Data Cases. The *is_joinable()* Method

The *is_joinable()* method applies original pruning techniques. These are the following:

1. The algorithm works with vertices, which will be a list of nodes. For each position of the partial solution, the label is defined for that vertex (the label serves as the vertex type) that the data vertices will have to belong to. For the first position, the search is made in the data graph for the vertex label of the query vertex on the first position that has the mentioned first position label, then the search in the data graph stops, and the algorithm appends the node to the log corresponding to the first position and continues the search in this manner for the next position. Every position has its own log.

2. From the first position onward, each found valid vertex will be one of the vertices adjacent to the root vertex found in the previously described step.
3. Then, the algorithm declares the partial solution as being a complete solution by appending it to the list of complete solutions. It then starts a new partial solution and goes through the steps described.
4. After finding all the complete solutions, the algorithm stops, and the list containing them is displayed in the user console.

8 Tests

The query graphs used have the structure of a two-level tree, which is a root node and at least two other nodes that are adjacent to it. A graph such as this is called *STwig* [1]. Comparisons have been made for the average execution times of five runs for each of the following query graphs for the GNS1, VF2 and STwig algorithms (Table 1):

1. Query graph 1, four nodes:
   edges = [[“18”, “23”], [“18”, “11”], [“18”, “9”]]
   labels = [“18”, “23”, “11”, “9”]
2. Query graph 2, four nodes:
   edges = [[“6”, “4”], [“6”, “20”], [“6”, “18”]]
   labels = [“6”, “4”, “20”, “18”]
3. Query graph 3, four nodes: edges = [[“27”, “14”], [“27”, “26”], [“27”, “3”]]
   labels = [“27”, “14”, “26”, “3”]
4. Query graph 4, four nodes:
   edges = [[“25”, “28”], [“25”, “29”], [“25”, “27”]]
   labels = [“25”, “28”, “29”, “27”]
5. Query graph 5, four nodes: edges = [[“29”, “25”], [“29”, “19”], [“29”, “13”]]
   labels = [“29”, “25”, “19”, “13”]
6. Query graph 6, three nodes:
   edges = [[“18”, “5”], [“18”, “7”]]
   labels = [“18”, “5”, “7”]
7. Query graph 7, three nodes:
   edges = [[“24”, “26”], [“24”, “11”]]
   labels = [“24”, “26”, “11”]
8. Query graph 8, three nodes:
   edges = [[“32”, “18”], [“32”, “17”]]
   labels = [“32”, “18”, “17”]
9. Query graph 9, three nodes:
   edges = [[“14”, “22”], [“14”, “25”]]
   labels = [“14”, “22”, “25”]

Table 1 Comparison of average execution times using the RI human PPI data graph having 10,000 edges and 4652 nodes

Query graphs	GNS1	VF2	STwig
Query graph 1, four nodes	14.7315	183.9175	26.5569
Query graph 2, four nodes	8.5028	159.0373	9.8075
Query graph 3, four nodes	14.7247	219.1603	28.7548
Query graph 4, four nodes	14.8219	178.1951	28.2544
Query graph 5, four nodes	10.2924	179.5263	23.1392
Query graph 6, three nodes	9.3765	181.5807	16.6471
Query graph 7, three nodes	10.0080	188.0332	16.2089
Query graph 8, three nodes	12.5657	172.2554	34.2284
Query graph 9, three nodes	12.8323	167.2234	28.8154
Query graph 10, three nodes	8.9278	165.6417	16.5563

Five executions for each query graph for each algorithm

10. Query graph 10, three nodes:
    edges = [[“31”, “6”], [“31”, “20”]]
    labels = [“31”, “6”, “20”]

9 System Specifications

The hardware used has an Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz processor, 8.00 GB RAM, Windows 10 Pro 64-bit operating system, x64-based processor, a disk drive with 931.51 GB and 512 bytes/sector. The integrated development environment used is JetBrains Pycharm Professional Edition 2020.1.1 x64 [17] with Python 3.7.6 [18], NetworkX 2.4 [12], NumPy 1.18.4 [19], neobolt 1.7.17 [20], neotime 1.7.4 [21], PY2neo 4.3.0 [22]. For the data graph handling, the Neo4j Enterprise 3.5.6 graph engine with Neo4j Browser 3.2.20 [13] was used.

10 Conclusions

In this paper, GNS1 was presented, a new algorithm created and then compared with two algorithms for graph isomorphism: STwig and VF2. For the last two, an original implementation [5] was made without using their authors' source code. Also, the essential parts of the code were described and made a comparison of the execution times. The algorithms were tested using the RI human protein to protein interaction data graph, having 10,000 edges and 4652 nodes. The algorithms can be used in a multitude of domains such as biochemistry, neurobiology, ecology, engineering, ecological food webs, genetic networks, World Wide Web, biomolecules within a cell and synaptic connections between neurons.

Acknowledgements This research was supported by Babeş-Bolyai University Cluj Napoca, Faculty of Mathematics and Computer Science. Thanks to my Supervisor Dr. Pârv Bazil, Professor Emeritus, Dr. Niculescu Virginia, Associate Professor and Dr. Sterca Adrian, Lecturer Professor who provided insight and expertise in helping me learn graph theory and how to implement and understand graph isomorphism algorithms.

References

1. Sun Z, Wang H, Wang H, Shao B, Li J (2012) Efficient subgraph matching on billion node graphs. In: Proceedings of the VLDB endowment. PVLDB. https://doi.org/10.14778/2311906.2311907
2. Cordella LP, Foggia P, Sansone C, Vento M (2004) A (sub)graph isomorphism algorithm for matching large graphs. IEEE Trans Pattern Anal Mach Intell 26(10):1367–1372
3. Lee J, Han WS, Kasperovics R, Lee JH (2012) An in-depth comparison of subgraph isomorphism algorithms in graph databases. In: PVLDB
4. Gheorghica R-I (2018) Algorithms for graph isomorphism. A comparative study. In: 8th international multidisciplinary scientific symposium "challenges and opportunities for sustainable development through quality and innovation in engineering and research management", Universitaria SIMPRO 2018, Petroşani, 11–13 Oct 2018
5. Gheorghica R-I: Algorithms for graph isomorphism. A comparative study on STwig and VF2. In: Proceedings of the 26th workshop on information and communication technologies, 28th international conference on software, telecommunications and computer networks (SOFTCOM 2020), Hvar, Croatia Sept 2020, pp 17–19. ISSN 2623-7350
6. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U (2002) Network motifs: simple building blocks of complex networks. Science 298(5594):824–827
7. Ferro A, Pulvirenti A, Alaimo S, Micale G, Marceca GP, Sciacca E, Maria AD, Sciacca E, Ferlita AL, Martorana E (2020) University of Catania, Department of Clinical and Molecular Biomedicine (MEDBIO). https://www.researchgate.net/profile/Alfredo_Ferro, http://www.medclin.unict.it/faculty/alfredo.ferro. Accessed 14 Aug 2020
8. The randomized networks used for detecting three node motifs preserve the numbers of incoming, outgoing, and double edges with both incoming and outgoing arrows for each node. The randomized networks used for detecting four-node motifs preserve the above characteristics as well as the numbers of all 13 three-node subgraphs as in the real network. Algorithms for constructing these randomized network ensembles are described [8]. Additional information is available at www.weizmann.ac.il/mcb/UriAlon [6]
9. Methods are available as supporting material on Science Online [6]

10. Shen-Orr S, Milo R, Mangan S, Alon U (2002) Nat Genet 31:64
11. Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms, 3rd edn. MIT Press
12. Hagberg A, Shult D, Swart P (2020) Network analysis in Python. https://networkx.github.io/. Accessed 14 Aug 2020
13. Neo4j (2020) Neo4j documentation. https://neo4j.com/docs/. Accessed 14 Aug 2020
14. Docker Hub. https://hub.docker.com/. Accessed 28 Sep 2020
15. Spiegelberg E (2020) Neo4j causal cluster docker Quickstart. https://graphaware.com/neo4j/2018/01/03/casual-cluster-quickstart.html, 3 Jan 2018. Accessed 28 Sept 2020
16. Neo4j (2020) Neo4j operations manual, clustering introduction. Available at: https://neo4j.com/docs/operations-manual/3.5/clustering/introduction/. Accessed 28 Sept 2020
17. JetBrains s.r.o, PyCharm Professional Edition. https://www.jetbrains.com/pycharm/ Accessed 14 Aug 2020
18. van Rossum G (2020) Python Software Foundation. https://www.python.org/. Accessed 14 Aug 2020
19. Oliphant T et al (2020) The fundamental package for scientific computing with Python. https://numpy.org. Accessed 14 Aug 2020
20. Neo4j Sweden AB (2020) Neo4j Bolt connector for Python. https://github.com/neo4j-drivers/neobolt. Accessed 15 Aug 2020
21. Neo4j Drivers Team (2020) Nanosecond resolution temporal types. https://neotime.readthedocs.io. Accessed 15 Aug 2020
22. Small N (2020) Python client library and toolkit for Neo4j. https://py2neo.org/v4/. Accessed 15 Aug 2020

Deep Learning-Based Pothole Detection for Intelligent Transportation Systems

Ilaiah Kavati

Abstract The presence of potholes on the roads is one of the major causes of road accidents as well as wear and tear of vehicles. Various methods have been implemented to solve this problem ranging from manual reporting to authorities to the use of vibration-based sensors to 3D reconstruction using laser imaging. However, these methods have some limitations such as the high setup cost, risk while detection or no provision for night vision. In this work, we use the Mask R-CNN model to detect potholes, as it provides exceptional segmentation results. We synthetically generate a dataset for potholes, annotate it, do data augmentation and perform transfer learning on top of Mask R-CNN model which is pre-trained on MS COCO dataset. This support system was tested in varying lighting and weather conditions and was performed well in these situations as well.

Keywords Pothole detection · Transfer learning · Mask R-CNN

1 Introduction

Diverted driving, speeding or other driver mistakes are primary reasons of accidents around the world. Bad status of roads is also a major reason. The roads become dangerous because of many reasons, for example, flooding, harms caused by overloaded large vehicles, etc. The assessment of road condition includes recognizing and analysing different kinds of road surface distress, such as potholes and road breaks.

A pothole is a sort of street distress. It is normally characterized as an arbitrarily shaped structural imperfection of the street. Due to it having both an arbitrary shape as well as depth, it creates identifying potholes as a challenging object detection task. However, they do need to be dealt with because they are a grave danger to human life and transportation. There have been 10,846 street mishaps because of potholes in India in between 2013 and 2016 [1]. Potholes additionally require noteworthy

I. Kavati (✉)
National Institute of Technology Warangal, Warangal 506004, India
e-mail: ilaiahkavati@nitw.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_20

expenses to be worked with. England has assessed that the expense of fixing all streets with potholes in the nation would cost 12 billion [2]. A prior and precise recognition of potholes can possibly spare lives as well as diminish costs on reconstruction.

2 Related Work

The following are a few of the significant strategies for pothole recognition tried out till now:

Public Reporting: This sort of framework depends upon civil responsibility by the administration as well as participation by people. These frameworks use individuals as sensors [3]. The main advantageous aspect of this technique is no need for expensive equipment of specialized programming.

Vibration-Based Methods: These methods utilize techniques of gaining information about odd vibrations through accelerometers placed in a vehicle going over road potholes. Vibrations of the vehicle are assembled using an accelerometer. The obvious disadvantage of the method is that observing vehicle must pass over the pothole first to gain information about it.

To find road potholes, Seraj et al. [4] utilize a support vector machine to construct an artificial intelligence-based solution. The suggested framework utilizes an accelerometer, spinner and a Samsung Galaxy smartphone as data labeling sensors; knowledge marking is essentially done, and instead a high-pass channel is used to remove the low-frequency components triggered by acceleration and turns.

2D Vision-Based Methods: These utilize the captured 2D information in a picture or a video and compute on this information utilizing 2D pictures or video handling techniques [5]. The decision of these image processing techniques is exceptionally reliant on the use case for which 2D pictures are being used. One technique utilizing this method was proposed by Koch and Brilakis [6] where the deformed and non-deformed regions in an image were separated using histogram shape-based threshold. The authors here consider pothole as an approximate ellipse based on a perspective view.

Another similar framework proposed by Thekkethala et al. [7] utilized two stereoscopic cameras and applied sound system coordinating to gauge depth of a pothole on asphalt surfaces. Following this binarization and morphological tasks are performed, and a basic estimation of a pothole is evaluated. The framework is eventually tested on over 20 images while no information regarding depth of the pothole has been provided. The system can detect skeletons of potholes having great depressions.

3D Scene Reconstruction-Based Methods: 3D scene recreation is a technique for catching the shape, depth and presence of objects; it depends on 3D surface reproduction which ordinarily requires a greater number of calculations than 2D computer vision-based techniques. Rendering of surface heights assists with understanding accuracy during the planning of 3D vision frameworks. This method can be based on utilizing different kinds of sensors, for example, Kinect sensors [8]. Kinect sensors are for the most part utilized in fields of gaming or(indoor) robotics. For this, stereo vision cameras are viewed as being much more financially cheaper in comparison with other sensors. Stereo vision focuses on powerful and precise differences, determined from left-and-right picture sets, to be utilized for assessing depth or separation; see, for instance, [9].

Learning-Based Methods: Several convolutional neural networks(CNNs) have been developed for recognizing objects in images, for instance, Chen et al. [10], RefineNet [11]. Fully convolutional neural networks (FCNs) for recognizing stuff at the pixel level in image data have been proposed; for instance, see Long et al. [12], or SegNet by Badrinaraynan et al. [13]. Another example would be the technique proposed by Staniek [14], where they utilize stereo vision cameras for capturing data about street surface as 3D points clouds. The technique places importance on calculating stereo matching through utilizing a variation of recurrent ANN, a Hopfield neural network. The technique utilized a CoVar method [15] to compute matching pixels and secured 66% accuracy.

3 Methodology

Potholes are an important defect of a road found all over the world. They are especially common in poorly made Indian roads. Given their high prevalence, having a good way to find them forms an important support system for a driverless car. Since Mask R-CNN forms the basis of our work, we implemented pothole segmentation using Mask R-CNN model [16]. The Mask R-CNN model is shown in Fig. 1.

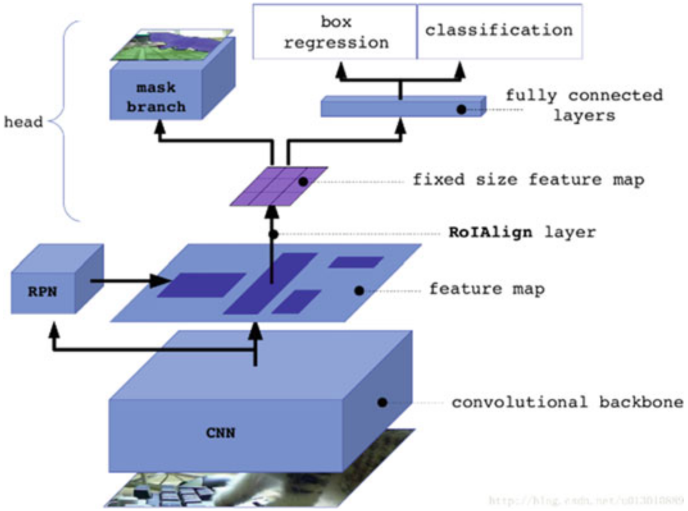


Fig. 1 Mask R-CNN architecture [17]

Fig. 2 Few dataset images

We first found a collection of images of potholes, annotated the dataset using tools, augmented the dataset using data transformation language (DTL), trained the model to learn them using transfer learning. We first found the potholes images combined as the dataset [18]. The dataset consisted of 431 images taken from the dashcam of a moving vehicle, thus making them perfect for our particular scenario. Some images contain zero potholes, while some seem to contain multiple ones. The potholes in the images are also present in all sizes, and in sort of lighting and road conditions (Fig. 2).

We uploaded the dataset on supervisely[19] and utilized the online tools for annotation. We segmented all the potholes in the 431 photos with this process. DTL is an extremely useful language used to make it easier to transform data, especially when working with datasets. We can use DTL to divide the dataset into training and validation sets as follows. First, we mention the annotated database directory as the source. Next, we mention the tag names and the percentage divide we want between the training and validation sets. We choose 80% of the images to be a member of the training set. The folders in the training set are tagged as 'train' while the images in the validation folder are tagged as 'val'. Both the folders are then combined as one final output folder. This transformation results with us in getting images tagged as either training or validation with exactly the divide needed (Fig. 3).

Since we have a small number of images (431) in our dataset, we perform data augmentation. Typical ways to enlarge the dataset are to flip, rotate, scale, crop and translate an image. We use the flip, multiply and crop transforms on all of our images using DTL. We first specify the input folder containing the initial dataset. We then specify the option to flip the images along the vertical axis and put them in a separate folder. We combine these two folders. Then all the images are multiplied and further cropped with width and height cropped to 70–90% of the original values, and after tagging them as mentioned above, put them in the augmented dataset folder. This results in a total of 9416 images. We remove 416 of these to use 9000 images (Fig. 4).

The images are then fed to the Mask R-CNN model for transfer learning. First, the weights corresponding to the model trained on MS COCO dataset [20] are loaded, following which learning is done in three stages, with the learning rate = 0.001. The

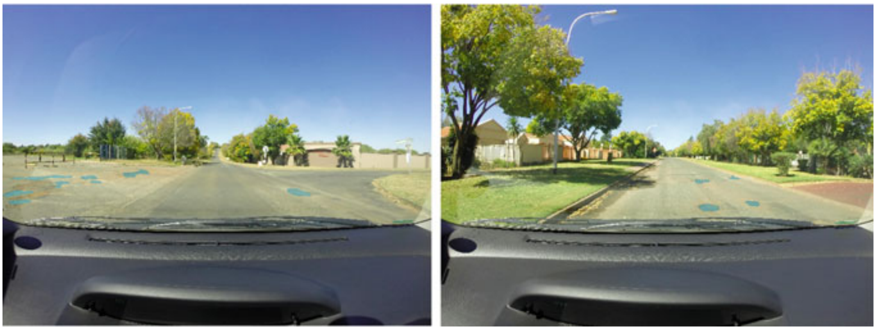
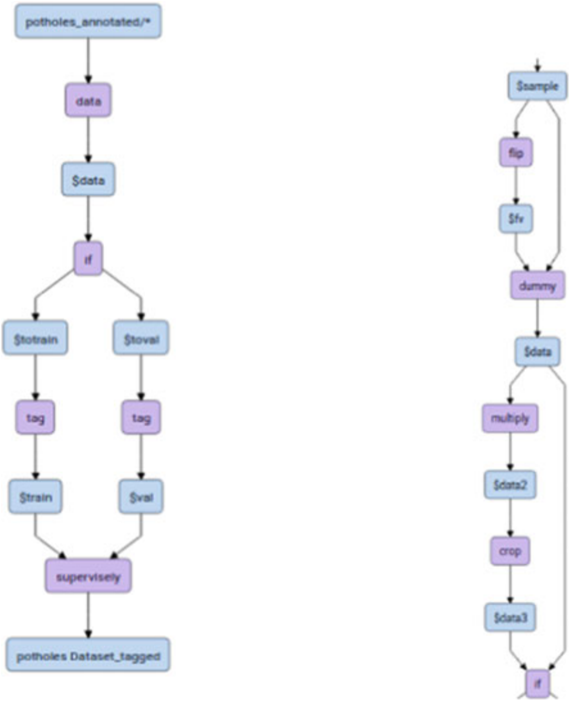


Fig. 3 Annotated images (blue parts are annotated potholes)



(a) DTL graph for dividing dataset images into training and validation sets

(b) DTL graph for data augmentation

Fig. 4 DTL code graphs for dataset manipulation

model is modified so that we only have two final classes in the model, ‘Pothole’ and ‘Background’. The images per GPU parameter are kept to 1, and the number of steps per epoch is chosen as 1000, with batch size as nine images since the total number of dataset instances is kept at 9000.

Due to the lower GPU performance that was available to us, we decided to perform transfer learning to the model in parts. First, the layers of the ResNet backbone of the model are trained for 20 epochs. Next, the head layers (head layers consist of the layers to which the FPN feature maps are fed, i.e. the layers in the parallel mask branch and the fully connected layers) are trained for 20 epochs keeping all the parameters same. This is done to fine-tune the model. Finally, all layers are trained together for 20 epochs, which is the most crucial part that makes all the learnable parameters attain satisfactory values. With the completion of the training, we end up with a robust support system which can detect unwanted potholes in the path of the vehicle and provide the necessary information to the autonomous driving system to take action beforehand to provide for a smoother and a more comforting ride for the passengers.

This support system was tested in varying lighting and weather conditions and was performed well in these situations as well. The experiments and necessary results of the model are performed and observed using TensorBoard [21].

4 Experimental Results

4.1 Datasets

MS COCO dataset: The Microsoft Common Objects in Context (MS COCO) dataset [20] is a well-known dataset that has annotations for instance segmentation and bounding boxes used to evaluate how well object detection and image segmentation models perform. It contains 91 common objects as classes. Out of the total classes, 82 have 5000 annotations or more. There are overall 328,000 images with more than 2,500,000 annotated objects. The MS COCO dataset has been used to train prior weights for Mask R-CNN model for the task of performing transfer learning on potholes dataset.

Pothole Dataset: Pothole dataset consists of 431 images taken from the dashcam of a moving vehicle, with 428 of them consisting of having at least one pothole. The potholes are not easy to spot, are at varying distances from the car and are of different sizes. The dataset offers all sorts of variety regarding potholes on the road. Training and test datasets are constructed from this dataset, with 9000 images being used for training and validation, and 400 images used for testing [18]

4.2 Operating Environment

We utilized Intel® Xeon® CPU @ 2.30GHz with NVIDIA Tesla K80 GPU, 12 RAM (accessible on the Google Colab platform) to perform the training and testing for the models involved. The programme modules were written in $Python - 3.6$ and made use of $TensorFlow - 1:14:0$ and $Keras - 2:2:4$ libraries. Video and image processing was done using $OpenCV - 4:1:2$ library.

4.3 Evaluation Metrics

The performance of various support systems is measured by certain common and certain task-specific metrics. The common metrics which may be used across various subparts include precision (Eq. 1) and recall (Eq. 2), which are calculated from the number of true positives (TP), false positives (FP) and false negatives (FN). A curve is generated between precision and recall with detection values sorted based on their confidence. As precision and recall values usually have a trade-off leading to the precision–recall curve intersecting for various models, an additional criterion, average precision (AP), is used to obtain the accuracy of the model. Average precision is the precision value averaged across all unique recall levels. In order to reduce the impact of the noise in the curve, we first interpolate the precision (Eq. 3) at multiple recall levels before actually calculating average precision (Eq. 4).

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{1}$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{2}$$

$$p_{\text{interp}}(r) = \max_{r' \geq r} p(r') \tag{3}$$

$$\text{AP} = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{\text{interp}}(r_{i+1}) \tag{4}$$

4.4 Results

Here are some images sent as part of test dataset to the Mask R-CNN model subsequent to performing transfer learning (Fig. 5).

The performance of the model is judged using the parameters precision and recall. As mentioned in Sect. 4.3, they are important attributes which help us to know how balanced the performance of our model is. If it segments the potholes correctly, we gain a high recall, and if it does not segment the flat road as potholes, we get

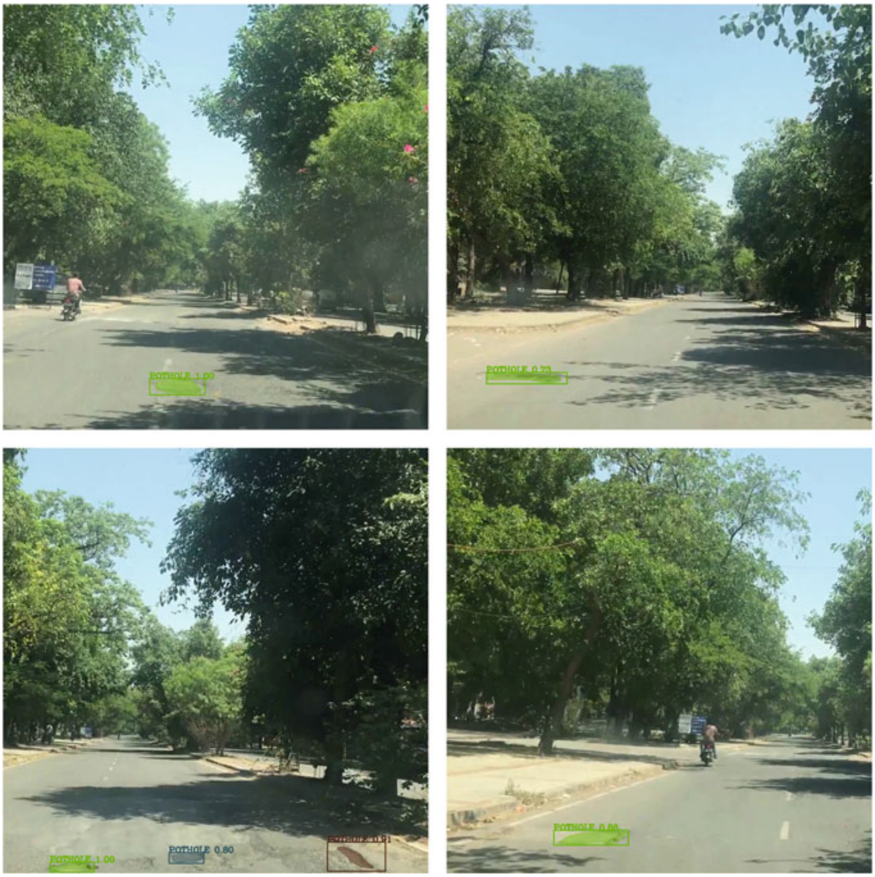


Fig. 5 Some output images from the trained Mask R-CNN model(coloured masks are the segmented potholes)

high precision. Usually as precision increases, recall decreases and vice versa. The interpolated precision, on the other hand, is calculated at each recall level, r, by taking the maximum precision measured for that r.

We compute the precision and recall by both checking if our model got the number of potholes in the image right, as well as the accuracy of the masks. This is done by having the predicted and the ground truth masks, and checking for the overlap between the same. We use the IoU parameter to check if the predicted mask corresponds well with the ground truth masks. The IoU is given by the ratio of the area of intersection and area of union of the predicted bounding box and ground truth bounding box. To find the best predicted mask matching ground truth box, we arrange the predicted masks by their IoU scores to check which ones match the best. If we do not find match with a IoU score over the threshold of 0.5, we move to the next ground

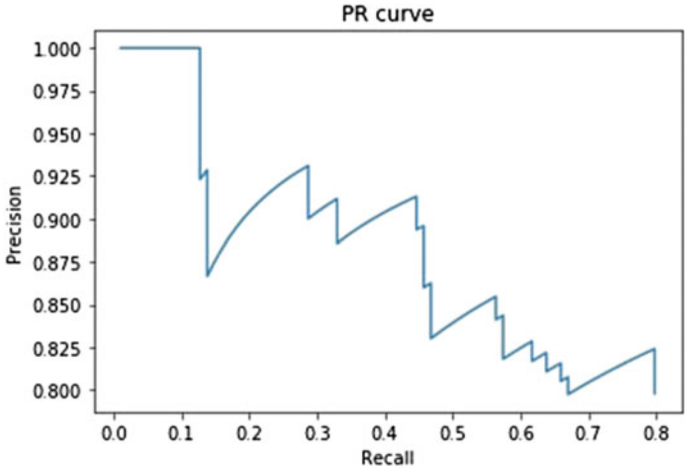


Fig. 6 Precision–recall curve plotted on output of test dataset

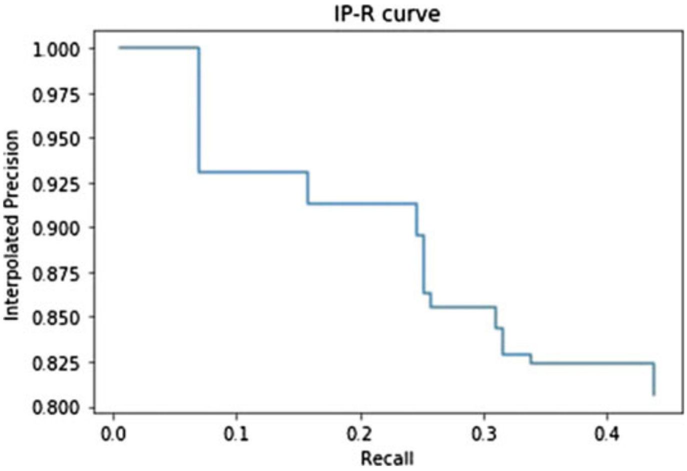


Fig. 7 Interpolated precision–recall curve plotted on output of test dataset

truth box and consider this one unmatched. This process is repeated (Figs. 6 and 7; Tables 1 and 2).

Thus, the details of its training are as follows: As the previous approaches were experimented on small datasets (Dhiman et al. with 1494 images used for transfer learning vs 9000 for our approach) or using of non-deep learning-based techniques (Dhiman et al. with multi-frame fusion-based method and Koch et al. with shape-based thresholding), our approach presents better or equal results than previous approaches.

Table 1 Evaluation result for pothole detection with Mask R-CNN tested on custom test set

Evaluation parameter	IoU = 0.5
Precision	0.8935
Recall	0.8568
Average precision (AP)	79.51%

Table 2 Evaluation of our approach versus other previous approaches

Approach	Precision (%)	Recall (%)
Dhiman et al. [22] (multi-frame fusion-based method)	67.4	51.2
Dhiman et al. [22] (Mask R-CNN)	88	84
Koch et al. [6] (shape-based thresholding)	81.6	86.1
Our approach	89.35	85.68

5 Conclusion

Pothole detection is performed using the cutting-edge Mask R-CNN machine learning model. The use of lane detection provided much more information regarding lane positions results in an improved method of providing data about potholes to a self-driving car. The cohesive but separate structuring of the subparts ensures the model brings results in an adequate amount of time and gives results with good accuracy regarding recognizing and segmented potholes in various sorts of lighting conditions. The proposed framework can segment potholes with an average precision of 89.35%, while having 85.68% recall.

References

1. Srivastava S, Sharma A, Balot H (2018) Analysis and improvements on current pothole detection techniques. In: 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE). IEEE, pp 1–4
2. Song H, Baek K, Byun Y (2018) Pothole detection using machine learning. Advanced Science and Technology, pp 151–155
3. Tedeschi A, Benedetto F (2017) A real-time automatic pavement crack and pothole recognition system for mobile android-based devices. Adv Eng Inform 32:11–25
4. Seraj F, van der Zwaag BJ, Dilo A, Luarasi T, Havinga P (2015) Roads: a road pavement monitoring system for anomaly detection using smart phones. In: Big data analytics in the social and ubiquitous context. Springer, Berlin, pp 128–146
5. Georgieva K, Koch C, König M (2015) Wavelet transform on multi-GPU for real-time pavement distress detection. In: Computing in civil engineering, pp 99–106
6. Koch C, Brilakis I (2011) Pothole detection in asphalt pavement images. Adv Eng Inform 25(3):507–515
7. Thekkethala MV, Reshma S et al (2016) Pothole detection and volume estimation using stereoscopic cameras. Int J Ind Electron Electr Eng 4(5):47–51

8. Rasheed A, Kamal K, Zafar T, Mathavan S, Rahman M (2015) Stabilization of 3d pavement images for pothole metrology using the Kalman filter. In: 2015 IEEE 18th International conference on intelligent transportation systems. IEEE, pp 2671–2676
9. Klette R (2014) Concise computer vision. Springer, Berlin
10. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
11. Lin G, Milan A, Shen C, Reid I (2017) Refinenet: multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1925–1934
12. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
13. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495
14. Staniek M (2015) Neural networks in stereo vision evaluation of road pavement condition. In: Proceedings of international symposium on non-destructive testing civil engineering, pp. 15–17
15. Cyganek B, Siebert JP (2011) An introduction to 3D computer vision techniques and algorithms. Wiley, London
16. He K, Gkioxari G, Dollaár P, Girshick R (2017) Mask r-cnn
17. Jiang R (2018) Understanding-mask rcnn. https://ronjian.github.io/blog/2018/05/16/Understand-Mask-RCNN. Accessed 30 May 2020
18. Pothole dataset. http://augmentedstartups.info/potholedataset. Accessed: 30 Jan 2020
19. Kolomeychenko M (2019) Supervisely platform. https://supervise.ly. Accessed 2 Dec 2019
20. Lin TY, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL, Dollár P (2014) Microsoft coco: common objects in context
21. Abadi M et al (2015) TensorFlow: large-scale machine learning on heterogeneous systems. http://tensorflow.org/. Software available from tensorflow.org
22. Dhiman A, Klette R (2019) Pothole detection using computer vision and learning. IEEE Trans Intell Transport Syst

Disaster Management Using Artificial Intelligence

K. Savio Rajan, Amith Abraham Rajan, Steve Maria Waltin, Tom Joseph, and C. Anjali

Abstract Natural calamities are known to inject a feeling of helplessness into the minds of the people involved with it, and that is exactly the situation for the personnel who are in charge of helping with the situation. Primarily focusing on the issue of rescue personnel not being able to locate the affected casualties, a solution to the same is proposed via two distinct technologies, social media analytics and aerial person detection. When any kind of disaster occurs, social media gets flooded with different kinds of data such as images, videos, texts, etc. These could be posted by people who are directly or indirectly affected by the occurrence of the disaster. In this paper, a social media analytics method is proposed to monitor social media for disaster-based posts and give detailed insights to take necessary action. Another major challenge experienced by search and rescue teams during a disaster situation is the search for survivors and victims and also to reach out to distant areas. Drones can be used for person detection in case of emergency situations. It can fly through a particular area and help in the detection of people stranded in emergency situations such as floods, isolated buildings, etc. Here, another method is also proposed for performing person detection from an aerial viewpoint, and all of these are integrated into a web app from which the user can select the features according to the need.

Keywords Disaster management · Person detection · Social media analytics · Data science · Deep learning

1 Introduction

The root of this paper is based on the past decade of natural calamities that have been ravaging the planet and the fact that no matter the technological advances, rescue operations always lag behind. One of the recent events that motivated the paper

K. Savio Rajan (✉) · A. A. Rajan · S. M. Waltin · T. Joseph · C. Anjali
Department of Computer Science and Engineering, Mar Baselios College of Engineering and Technology, Trivandrum, Kerala, India

C. Anjali
e-mail: anjali.c@mbcet.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_21

further is the Kerala floods of 2018 [1], where there were large numbers of people stranded paired with a large array of rescue personnel but no medium to connect them both. A requirement that arises in these kinds of situations is the need for a system that can efficiently allow real-time monitoring of a natural calamity-affected area for the purpose of rescue operations. The fundamental problem trying to be solved here is the call for help and the response for the same. The two instances that were noticed are when people who have no forms of communication at all to the world outside them and the people who are able to plead for help via social media. For the former problem, it was concluded that a robust pipeline of hardware and software and hardware can be used to solve this, namely, an aerial device, i.e. a drone, an edge computing device attached to the same, a pipeline of object an effective object detection model, RetinaNet, and communication network to transmit the same. This paper mainly focuses on the object detection part of that pipeline (for the format problem alone).

When it came down to the decision of deciding on a model that there were two main categories of object detection models to choose from, single-stage and two-stage detectors. Some of the main models coming within those categories were YOLO [2] and SSD [3]. On comparing these models to the one selected, RetinaNet [4], some interesting conclusion on our decision was derived. YOLO and SSD used 98, 1 k, 8–26 k boxes white RetinaNet at around 100 k towards data distribution imbalance for the data samples for the foreground and background examples. Although the vast amount of foreground examples could provide vast amounts of training examples, it could provide vast amounts of learning points in that sense too. But these points are easy to identify as compared to the objects required, that's a huge margin. As a defect, the model learns to detect the easier measure more than the harder one, therefore reducing the accuracy in the end. Even though a picture might have six data points or identifiable objects, there would be around 100 k boxes or reference boxes per se, most belonging to the foreground of the objects required. So, this would set the cross-entropy at an imbalance, which would mean that the unwanted sections will have very less loss because of the large number of samples that have less loss (because they are easy to learn but not important) over at the section where data that is not important. While the important section will have a high loss, so essentially, the solution that RetinaNet is proposing is introducing a new value called focal loss [4] rather than the loss value, which is essentially just an addition of a variable gamma. The idea is to increase the focus on the harder section of the classification. (Thereby imposing a balance between the two, that is the hard and the easy section). Hence by controlling the value of gamma, to control the focus on the harder section of the classification, i.e. the lowest value 0 would mean that the equation would go default which is the cross-entropy, and the higher the gamma value, the end equation focuses on the harder portion. Then on top of that, introduce another value alpha which goes onto balance the change. Essentially the same gradient descent problem it is trying to avoid, that is, for every small value change of alpha made to increase the focus, the value change would be rather high, hence it might never reach the balancing point. So, the change with alpha is balanced out. Hence, that increases the accuracy, but to increase the speed, one-time scanning (one stage) is introduced. But the lower accuracy might reduce the accuracy that is already available. Finally, to improve over this, the pyramidal network was introduced.

Now onto the second part of the problem, which is on the usage of social media to aid in disaster management. The proposed method considered Twitter as the social media platform because there is a greater possibility of someone using a microblogging social media platform instead of going for other platforms such as Instagram, Facebook, etc., and also the data were also made available to the developers via an API [5]. Twitter has turned out to be an important medium of communication during times of emergency. The smartphone enables people around the world to announce an emergency that they are observing in real time. Because of this, more organizations are interested in monitoring Twitter by using self-monitoring programs. In the proposed approach, the data are pre-processed initially, and then basic classification is performed on the tweets to determine which of the tweets were actually related to disasters. These data along with the location tagged in the tweet are then passed on to emergency responders to take necessary actions.

2 Literature Review

Nouar et al. [6] experimented with three different object detection models, CNN, S-CNN, and HELM model. The key aspect of the methods showcased here is the optical flow section of the pipeline, which essentially accounts for the main issue that hinders object detection, which is the camera movement. Meaning that, at any given point of focus, no matter the variation of the speed of the aerial image source, the motion will be fast. Hence, optical image flow is the stage that stabilizes this motion without making compromises in the quality of output of this stage as it is the input for the prediction models. So, the models therein, after the optical stabilization performed such that, the pre-trained CNN model outperformed S-CNN and HELM in terms of accuracy by 98.9% although the constraints being, time taken and amount of computing power used. On the other hand, HELM performed at a close rate at 95.9% in an efficient time print while also taking very less computing power. Hence, the experiment concluded that using an optical flow model paired with an object recognition model such as HELM would yield quicker, good quality imagery with less computing power used. Such are the requirements for aerial data sources that use edge computing. Zhang et al. in [3] researched with the realization of the advent of aerial surveillance as an intelligent and effective form of surveillance as opposed to the stationary ground options that are available in the current/past period. The purpose was to find an efficient aerial dataset format and structure to better the process of processing and detection. Here, importance was paid to the fact that in aerial imagery, unlike focused photography, the subject to environment size ratio is large, that is small objects, large environments. Here, person ReID focused on, so as to diminish the dependency on the quality of the image. Essentially, the process starts with using SVDNet to create an orthogonal matrix while keeping the entire system end-to-end trainable. This is followed by a subspace pooling layer using a

CNN network as a backbone, this is done in order to reduce the feature redundancy. The next stage is the loss function for which softmax cross-entropy and batch triplet loss are used. Hence, the output of the entire pipeline is a feature map from a set of data collected from different airborne sources, which can be effectively used by end-to-end models to perform the detection of various subjects of interest. Radovic et al. in [7] have one of the early adaptations of the idea of detecting objects from an aerial source more specifically UAVs. Here, the implementation in a fairly simple manner, that is, using a simple CNN model to detect and classify the objects. The model, unlike the prior implementations, was not pre-trained but rather trained using a set of UAV image dataset. To further make the implementation more robust one of the models in the frontline of object detection was used, which was YOLO. Here too, the model was trained specifically to the problem at hand. Although the base CNN model performed at a higher accuracy rate (97.8%) than the YOLO implementation (84%), the model suggested by this experiment to be used in a real-time application is the YOLO model since it proved to be more efficient in these cases. To be more specific, this model is capable of running the detection at 150 frames per second, which would mean that it is possible for the model to process the video with a latency less than 25 ms. This amount of response and efficiency is highly critical for these types of situations and hence the conclusion of this research.

3 Proposed Approach

This paper presents a complete platform that allows technology to be leveraged such that real-time monitoring, SOS calls, and rescue decisions can be effectively performed in a calamity-affected region.

This is achieved via an aerial monitoring device: a drone with an intelligent model running within to detect and alert rescue personnel in real time all bound to a platform that allows for proper monitoring and analytics for further clarity of the affected through other mediums of communication.

The main objectives of this project are:

1. To detect human beings and vehicles like cars, bikes, and buses from aerial video footage
2. To implement social media analytics to gather data to detect the presence of disasters.
3. To create a web app to integrate both the above features.

3.1 Person Detection

Dataset. The proposed Stanford Drone Dataset (SDD) contains 60 aerial footages with pedestrians, cars, bikes, buses taken over eight scenes and an annotated bounding box for each of the classes [8].

Model. The model was trained using RetinaNet [4]. RetinaNet is a single-stage dense object detector. Two main parts of this network are the featured image pyramid [9, 10] and the usage of focal loss [4].

This model, unlike most object detection models that consist of m**ultiple** stages of image processing, is a single-stage detector model. This model also has a keen feature extraction method called Feature Pyrami**d N**etwork (FPN) along with a method to formulate loss that allows these models to stand out among others in having to produce a time-efficient output with satisfactory accuracy levels, this method of loss calculation is called focal loss.

Feature Pyramid Network. Essentially, this feature extraction method consists of a keen image representation form that allows the images, in theory, to be represented in a pyramidal form or in practice, the feature map consists of images that would be represented such that the subsample taken apart from the initial sample would be taken of a smaller size and the resulting image thereafter would be of a lower resolution. This implementation is done in order to mimic the efficiency rich methodology of the convolutional neural network, which also has a structure that is pyramidal hierarchical. Hence, similar to the method of image representation in a convolutional neural network, the output image would have a size that would be of a lower resolution hence the size of the subsamples further on would keep decreasing thus forming a structure that is similar to that of a pyramid. The key aspect for this sort of image representation is that at each level, starting from the high-resolution subsample, moving on to the lower resolution subsamples, each consists of a significant feature that is required to be extracted, hence it would combine the various features extracted from the different levels of the sample. This type of feature extraction, in theory, is performed by first consisting of a bottom-up convolutional layer, the second section of the pathway with the initial section being a top-down pathway with connections to the previous section that are lateral. Hence, by combining this form of a feature extraction within each of the subnetworks from the classifiers to the regressors, it forms a robust object detection model [9–11].

Focal Loss. The key feature that differentiates this object detection model from various others is the focal loss, this feature alone is responsible for molding the time efficiency to power consumption to over accuracy to be at the right ratio that is obtaining an optimal accuracy, achieved by consuming less power and time. The reason this type of result was not available to be leveraged by other models that use other means is due to a large amount of variation of information found between the foreground and the background. So, in most cases, the background would have a higher level of accuracy in the overall percentage, but in most cases not containing

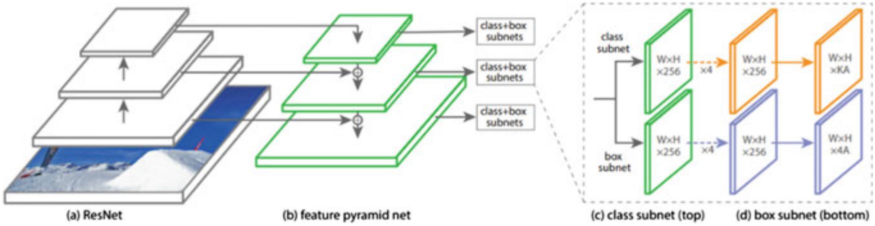


Fig. 1 Architecture of the proposed model [9]

the required information. This forces the overall accuracy to lean towards the background, without obtaining any useful information. Further on the foreground would indeed contain much-needed information that would be calculated at a much lower level of accuracy but indeed the overall accuracy would not reflect this as the majority of it is impacted by the background. The focal loss method brings a balance between the foreground and backgrounds by reducing the weights defined and increasing the weights defined respectively [4, 11].

RetinaNet architecture. The model consists of some key aspects. The ResNet network that is a bottom-up path will allow the feature maps to be scanned at different scales. This scanning will occur no matter the size of the image given as an input. Then the spatially rough feature maps that are from the upper levels of the pyramid, they are up-sampled, this is done in the top-down route. Alongside that, both the top-down and the bottom-up layers of the same spatial domain are combined using lateral connections. The classification subnetwork then proceeds to find or predict the probability of the presence of an object at the spatial location of each of the anchor boxes. Finally, the regression subnetwork will perform regression on the offset for the bounding boxes, this will be done for each ground truth object [4, 11] (Fig. 1).

Training. Initially, from the SDD [8], 2200 samples are taken for training, and each of the images consists of around 30,000 annotations. Of these images that were chosen, 1000 images were kept for the validation process. The parameters for the annotations are as follows:

$$\text{Directory\_of\_the\_image},\ x1,\ y1,\ x2,\ y2,\ \text{class\_label}$$

The anchor boxes have a default size, they are:

$$32 - 64 - 128 - 256 - 512$$

Due to the reason that the type of footage used here is that of aerial devices, this would require that the anchor box is less than size 32 as the object of interest will be very small in some cases. As a result, a small anchor box of size 16 is also added and had removed the large 512 size anchor box.

Evaluation. From the 1000 images that were used for validation, the model gave an output accuracy for each of the class indicated prior to the prediction, further to these values, mean value precision was applied to obtain the final result they are as follows:

Bike: 0.49
Car: 0.94
Bus: 0.79
Person: 0.71
Average: 0.64.

Results. The following figures are some of the outputs obtained from the model (Fig. 2).

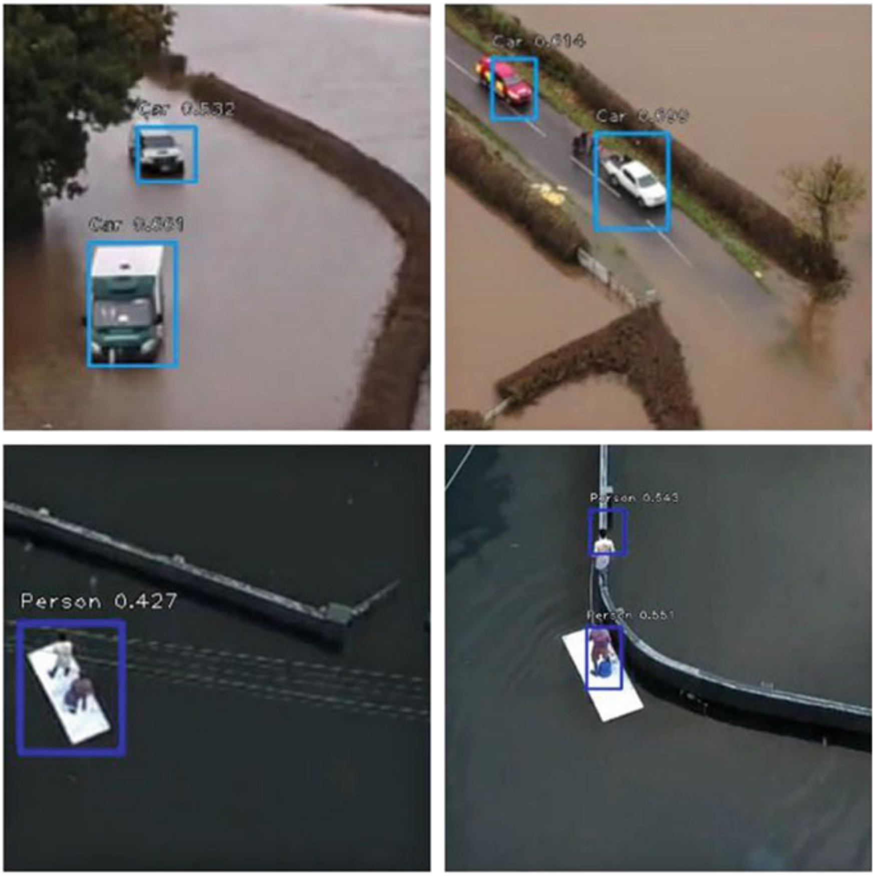


Fig. 2 Output generated using the trained model

3.2 Social Media Analytics

Social media analysis includes various methods of collecting and processing data and information from sites such as Facebook, Instagram, and Twitter. Users are often tracked about their conversations about product reviews and also about companies by market analysts. The initial steps in social media analytics include identification, analysis, and interpretation of data.

Social media platform. Here, Twitter is chosen as the social media platform because there is a greater possibility of someone using a microblogging social media platform instead of going for other platforms such as Instagram, Facebook, etc., and also the data were also made available to the developers via an API. Twitter has turned out to be an important medium of communication during the times of emergency [12]. The smartphone enables people around the world to announce an emergency that they are observing in real-time. Because of this, more organizations are interested in monitoring Twitter by using self-monitoring programs.

Extracting tweets related to particular hashtag (e.g.: #disaster #earthquake) using Twitter developer API [5]. The Application Programming Interface (API) allows us to access resources that are on the server. An API consists of a set of access levels for web software, which makes accessing data from the server easier.

Classifying extracted tweets as a real disaster or not. It is not always clear in all situations to determine whether a person's words are actually declaring a disaster or not. The proposed approach will be about performing text classification, by building predictive machine learning models. The primary steps include:

*Data analysis and exploration***.** Our primary objective is to analyze the data set then train and fit a machine learning model. Then the test tweets that are pulled live (using Twitter API) are classified to determine whether they are disaster-related or not. Disaster relief organizations and News Agencies would be the main users of this kind of application. The data for training are provided by Kaggle [13] and contain about 10,000 tweets that were hand-labeled. Given a dataset contains a training set and a test set that is created by pulling tweets from Twitter. The training file includes the text of a tweet, keyword from that tweet, location the tweet was sent from. Features in the train data file include:

- id—a unique identifier for each tweet
- text—the text of the tweet
- location—the location the tweet was sent from
- keyword—a particular keyword from the tweet
- target—It is available only in train.csv only, this denotes whether a tweet is a real disaster or not.

Pre-processing the data

- Data Cleaning is the first step in which the text data need to be pre-processed to convert it into a consistent format. The text will be cleaned, tokenized, and converted into a matrix.
- Make the text all lowercase or uppercase
- Removing Noise, i.e. everything in the text that isn't a standard number or letter, which includes punctuation, numerical values, etc. are needed to be removed.
- Tokenization is the process of converting the normal text into a list of tokens. Sentence tokenizer can be used to determine the list of sentences in the data and word tokenizer is used to find words in strings among the given data.
- Stopword Removal of some common words such as "a," "and," "but," "how," "or," and "what.", etc. contribute less to select documents that match the needs of a user. These are then excluded from the vocabulary. These are called stop words
- Stemming helps to reduce inflected words to their base form. It is generally a written word form.
- Lemmatization. The major difference between stemming and lemmatization is that stemming creates words that don't exist and whereas lemmas include actual words. So, lemma can be found in a dictionary, but this is not the case for the root stem.

The pre-processed text needs to be transformed into a vector matrix of numbers before a machine learning model can understand and learn from it. This can be done in many ways. A word embedding is a representation of text where words that have the same meaning also have a comparable representation. Word embedding is actually a category of strategies in which certain words are represented as real-time valves in a predefined space. Each word is embedded in one mode and the vector values are read in the same way as a neural network, which is why the process is often illuminated in the field of deep learning.

Global Vectors (GloVe) [12] is basically a word embedding method with a log-bilinear model and uses weighted least-squares. Here, the goal of training is to learn word vectors such that their dot product is equal to the logarithm of the probability of co-occurrence of terms. Here, the log of a ratio equals the logarithm difference, which compares the ratios of probabilities of co-occurrence with vector differences in word vector space.

Model performance metrics. After generating the output in the form of a probability or a categorical classification, the current model is needed to be improved. For that, the effectiveness of the model based on metrics has to be found out. These can be done using test data sets. Distinct performance metrics are used to evaluate the efficiency of each machine learning algorithm in different contexts. This problem is a classification problem so confusion matrix is used, F1-score, and accuracy as a performance metric to evaluate model performance.

Accuracy can be a good measure when the target class in the training data set is nearly balanced. The distribution of the target class in the provided training set is 57% for 0 class to 43% for 1 class, which is about equally distributed. Due to this reason, 'accuracy' is chosen to be a model performance evaluation metric for this problem.

Building a machine learning model. Here, a simple classification model using Adam optimizer is created, which combines the best properties of AdaGrad [14] and RMSProp [15]

Results. Based on the training and predictions made by various classifier models, the final results that are predictions on data pulled using Twitter API are obtained.

3.3 Web Application

Web application acts as a user interface with machine and deep learning models. The web application is implemented using a micro web framework written in Python-Flask [16], it is a microframework because no additional tools or libraries are required. The models trained are deployed with Flask API. Flask is very simple to use, built-in server and debugger development, integrated unit test support, RESTful request dispatch, and extensively recorded.

Workflow

On the homepage, the user has two choices for choosing person detection or social media analytics, depending on the user's choice it will be redirected to the relevant pages. The user has to upload a video as an input that needs to be evaluated in a deep learning model using Flask REST API, the model will detect and label the people and display the output. In the social media analytics page, the user needs to enter hashtags and the tweets with those hashtags will be grabbed from Twitter using the user authentication details. The data recorded for classification are cleaned and passed to the machine learning model. After classification, the disaster-related tweets will be displayed as the output along with the location (Fig. 3).

4 Conclusion and Future Works

The project deems to provide the appropriate results required for an optimal level of communication and assist for rescue operations at the time of the calamity. This conclusion is derived based on the following experimental observations:

1. The person detection feature of the project was tested with various input videos of stranded personnel during previous occurrences of calamities, and it showed to work accurately as per the accuracy calculation.
2. Although this might not be a live feed as in the case of an actual calamity, it is impossible to test this feature due to the fact that simulation of the environment of the calamity is impossible. Due to the fact that the results on the pre-recorded input were promising, it can be safely concluded that in the occurrence of a calamity, the results will also be the same.

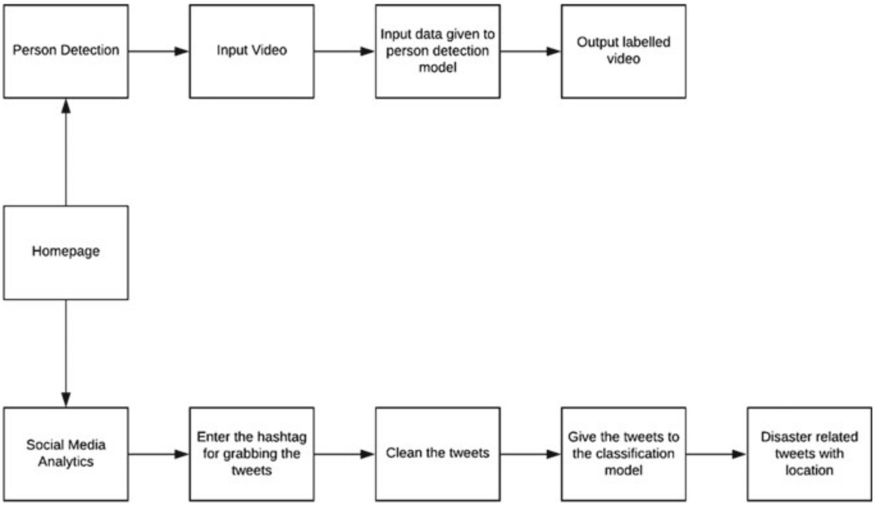

Person Detection
Input Video
Input data given to person detection model
Output labelled video
Homepage
Social Media Analytics
Enter the hashtag for grabbing the tweets
Clean the tweets
Give the tweets to the classification model
Disaster related tweets with location

Fig. 3 Workflow of web application

3. As for the analytics feature, when tested with the tweets during the period of a previous calamity, the system was successful in classifying and making available the content required for help that was asked via social media.
4. To put it all into perspective, the web application provided an easy and hassle-free method to make use of these features and be available during the time of a future calamity, if any.

Some of the problems faced in the early stages:

1. Unable to find out a model that was specific to our need, due to the fact that there was a requirement for one that was able to effectively perform classification at a high enough accuracy while also, taking less computational power and being able to take less time. This was crucial due to the fact that the person stranded had to be located and their location transmitted to the nearest rescue operation team, which would mean that neither can the model have a delay in detection due to the bottleneck of the hardware nor can there be a miscalculation due to less accuracy.
2. In the proposed method, it was unable to cluster down all posts in social media there were specific to help related to an ongoing calamity. More specifically, we did not have an anchor point to depend on to classify with surety that the cluster would include all posts related to the same.

Some of the measures taken to rectify the problems faced:

1. It was concluded from research that commonly used models such as CNN, YOLO, S-CNN, and ResNet could not be used due to the fact that it either needed a large amount of time to compute given the hardware conditions within

the aerial device or the efficiency was not at an adequate level, again given the conditions that it was analyzing and hardware conditions.

2. The task of converging into a cluster for the social media analytics was reduced to close-to-sufficient when there was a usage of #'s as an anchor point to discard off unwanted posts that were not relevant to the required condition. Apart from this, the accuracy of the model used to classify did play a major role, furthermore improving henceforth with technology such as context and sentiment analysis is expected to produce results that are more favorable.

4.1 Future Improvements

Further on from the current state of the project, it seems only feasible that the system is made more efficient. This can be achieved by implementing and or improving on the following features:

1. An intelligent predictor model that uses previous calamity data and the current updates of the weather from meteorological department to predict an upcoming calamity (those which can be predicted), in an attempt to allow the affiliated organizations to be as aware and ready for an upcoming calamity, wherein either case of the probability, the calamity occurs or not, the readiness for one will largely be in favor of the area under a future threat.
2. Allow the analytics engine to be more robust and efficient by implementing more intelligent ways to extract information from the social media platform via sentiment and context analysis systems, this will allow us to remove the dependencies from tagger such as #'s.
3. A more robust aerial source, like an autonomous drone. This will remove the biggest bottleneck when it comes to aerial detection, even more, specifically to our case, that is manual control of the drone which hinders the range of its travel with respect to the position of the controller. Allowing the drone to fly autonomously, while being monitored by a person on the ground will enable the range of search to be completely sufficient.
4. A better image or video dataset based on different environments from various calamities or scenarios and more annotated objects would be required to create a better model that can locate objects more precisely than the current model.

References

1. Hunt KMR, Menon A (2020) The 2018 Kerala floods: a climate change perspective. Clim Dyn 54:2433–2446. https://doi.org/10.1007/s00382-020-05123-7
2. Joseph R, Santosh D, Ross G, Ali G (2016) You only look once: unified, real-time object detection

3. Liu W et al (2016) SSD: Single Shot MultiBox Detector. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture notes in computer science, vol 9905. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2
4. Tsung-Yi L, Priya G, Ross G, Kaiming H, Piotr D (2020) Focal loss for dense object detection. Trans Pattern Anal Mach Intell 42(2):318–327
5. Twitter API docs: https://developer.twitter.com/en/docs/twitter-api
6. AlDahoul N, Md Sabri AQ, Mansoor AM (2018) Real-time human detection for aerial captured video sequences vis deep models. In: 2018 Computational intelligence and neuroscience, Hindawi
7. Matija R, Offei A, Qiaosong W (2017) Object recognition in aerial images using convolutional neural networks. J Imaging Delaware Newark, USA
8. Robicquet A, Sadeghian A, Alahi A, Savarese S (2016) Learning social etiquette: human trajectory prediction in crowded scenes. In: European conference on computer vision (ECCV)
9. Tsung-Yi L, Piotr D, Ross G, Kaiming H, Bharath H, Serge B (2017) Feature pyramid networks for object detection. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 936–944
10. Yanwei P, Tiancai W, Rao MA, Fahad SK, Ling S (2019) Efficient featurized image pyramid network for single shot detector. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)
11. Weng L (2018) Object detection part 4: fast detection models, lilianweng.github.io/lil-log
12. Sharma Y, Agrawal G, Jain P, Kumar T (2017) Vector representation of words for sentiment analysis using GloVe. In: 2017 international conference on intelligent communication and computational techniques (ICCT), Jaipur, pp 279–284
13. Training data: https://www.kaggle.com
14. Lydia A, Francis S (2019) Adagrad—an optimizer for stochastic gradient descent. 6:566–568
15. RMSProp: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
16. Python Flask Documentation: https://flask.palletsprojects.com/en/1.1.x

Classification of Tea Leaf Diseases Using Convolutional Neural Network

Subham Chakraborty, R. Murugan, and Tripti Goel

Abstract Assam is the highest tea-producing state in India. The economy of this state is greatly dependent on the cultivation and productivity of tea. The biggest challenge to tea growers is to produce tea without microbial or pesticide damage. Since leaves are the harvested product in tea, so leaf diseases play an important role. Lack of awareness and care to leaves causes unfavorable impacts on plants, product quality, and quantity get reduced. The symptoms of the disease can be observed on the leaves. The leaf shows symptoms by changing color or showing spots on it. The identification of these diseases is made manual, which can consume more time or may be costly. The idea is to identify and classify the diseases accurately from leaf images automatically. The Convolutional Neural Network is being proposed in this study, which has classified the diseases with an accuracy of 92.59% and is more accurate than the prevailing classifiers like Support Vector Machine and K-Nearest Neighbors for this specific purpose.

Keywords Tea leaf diseases · Computer vision · Image classification · Machine learning · Deep learning · Convolutional neural network

1 Introduction

Tea is considered one of the most consumed drinks in the world. Tea was just one of those weeds until it was discovered to give a great flavor and eventually introduced for cultivation. Tea is cultivated across various settings from the backyard or small fields to vast estates covering thousands of acres of land. Higher elevations and steep slopes are best for tea cultivation. From the perspective of plant protection, the demand for tea without pesticide residues is directly related to the demand for

S. Chakraborty
Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam 788010, India

R. Murugan (✉) · T. Goel
Department of Electronics and Communication Engineering, National Institute of Technology Silchar, Silchar, Assam 788010, India
e-mail: murugan.rmn@ece.nits.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_22

high quality and demand from the consumer. The majority of the tea-producing countries economy depends upon its production and of several constraints that affect production to a great extent. Insect and mite pests (arthropods) are some of the most damaging causes when it's come to tea, causing, on average, a 5–55% yield loss (68, 83, 90). The loss costs the approximately U.S. $500 million to $1 billion [1]. In some cases, loss of yield can be 100% [2]. Several attempts have been developed to reduce yield loss since the olden days. There has been a practice of widespread application of pesticides in the past few decades. The usage of chemicals such as bactericides, fungicides, and pesticides to check plant diseases can cause long-term resistance of the pathogens, undesirable reduction in the soil's ability to fight back, and produce adverse effects in the agro-ecosystem. Therefore, proper diagnosing a disease in its initial state and appropriate remedial action is vital for organized disease management [3]. Naked eye observation by experts had been the primary practice for plant disease detection and identification so far. A large team of experts and constant track of plant is needed, which has a high cost for large farms. Simultaneously, in some countries, lack of proper facilities and ideas to contact experts are some of the disadvantages of naked-eye observation. Consulting experts is inefficient both in time and cost. Moreover, some diseases that are not prevalent in those areas and are foreign diseases cannot be easily observed by simple observation. In this study, we are more concerned about two specific tea leaf diseases that are most destructive of the prevalent diseases in our vicinity Barak Valley, especially Silcoorie Tea garden, Silchar, Assam, Northeast, India, Red-Spider Mite and Mosquito Bug as described in Fig. 1.

It will be much helpful if an automated disease detector is made, which can detect disease through leaf images by image processing techniques. This could provide an immense favor to the novice in the gardening process. The trained professionals, too, can be helpful in the process of confirmation of disease diagnostics. Computer vision techniques, a modern approach, provide more accuracy than other preceding methods. Nowadays, disease detection is done with various data analysis technologies such as Machine Learning, which provide more accuracy in the field. Because of its performance in a wide range of applications, machine learning has covered almost

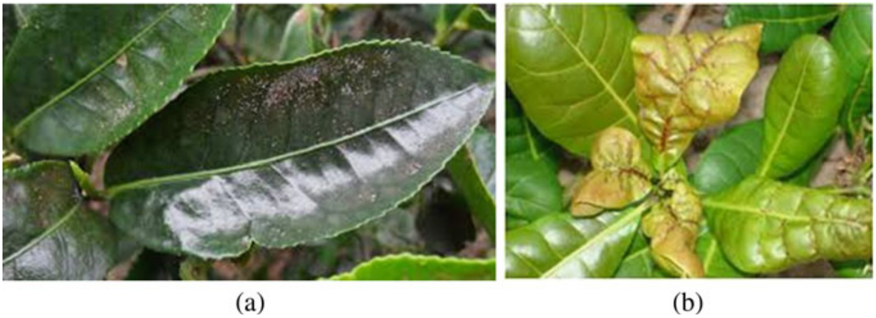


Fig. 1 **a** Red spider mite, **b** mosquito bug

every scientific domain, such as cognitive science, statistics, optimization theory, and various other engineering and mathematics works.

Neural Network (NN) is a computational approach operating in various computer science and other research disciplines. Biological nervous systems of the brain have been the design approach of NN to process the information and recognize patterns. The NN approach excels even in the area where the feature detection is troublesome. This technique is based on self-learning and training from the observational data rather than explicitly programmed and provides better data analysis accuracy. A deep neural network possessing many processing layers is an emerging technique, and it is used in Deep Learning. High-level abstractions in data can be modeled using multiple processing layers with complex structures consisting of multiple non-linear transformations. In this technique, an observation can be represented as a set of edges, regions of particular shapes, and many other forms. These representations ease the task of face recognition or facial expression recognition, weed detection crop quality, yield prediction, agricultural production systems, yield prediction, disease detection, etc. Deep learning is a handy tool in the analysis of a large amount of data.

Convolution Neural Network (CNN) is a deep learning-based tool that works proficiently in image processing. It provides excellent modeling for most of the complex processes and patterns or features recognition required in image processing. A CNN composed of alternating layers of convolutional layers, locally connected where every layer has the same number of filters. Down sampling (sub-sample) layers, and the fully connected layers working as a classifier. Figure 2 shows the overall architecture of a CNN. The blue-colored squares depict the feature maps, and the small grey-colored squares refer to the convolution filters.

Our main aim is to build a deep convolution neural network and check its performance versus two other Machine Learning classifiers viz., Support Vector Machine (SVM), and K-Nearest Neighbors (k-NN) to observe the healthiness of a tea plant

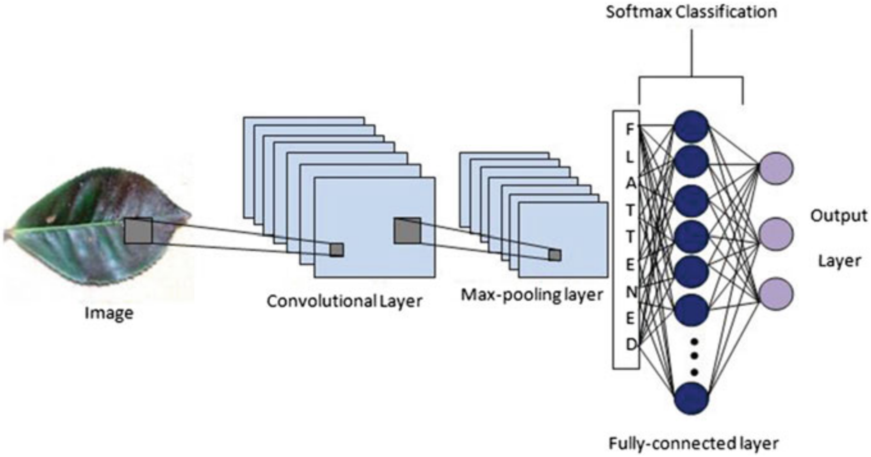


Fig. 2 Convolutional neural networks architecture

by analyzing its leaves. The proposed method will extract and identify the diseased leaves from the given set of images.

The contribution of this paper includes the following.

- The state-of-the-art methods of tea leaf detection have been presented.
- The Convolutional Neural Network-based deep learning architecture has been proposed to classify the tea leaf diseases.
- A detailed comparative analysis has been made against the current tea leaf detection methods.

The remaining part of the paper is organized as follows. In Sect. 2, the literature survey and related works are presented—the materials and proposed method to be found in Sect. 3. The results and discussions are presented in Sect. 4. The conclusions and future works are found in Sect. 5.

2 Related Works

Hazarika et al. [3] have reported that insects are affected tea production by an average of 5–55% loss in yield. Oligonychus coffee (red spider mite) is one of North East India's most devastating pests [4]. Tea mosquito bug is also one of the destructive pests that have caused losses of 55% of the crop loss in Africa and 11–100% loss in Asia [5]. The tea mosquito bug also attacks other non-crop host plants, which support their populations in the scarcity of the primary host. This behavior enables them to breed throughout the year [6].

Ghaiwat et al. [7] described various classification techniques for plant leaf diseases. Their classification technique involved the classification of distinct class's pattern based on morphological features. Various techniques such as Artificial neural network, Genetic Algorithm, Probabilistic Neural Network, Fuzzy logic, Principal Component Analysis, and k-Nearest Neighbor. It is a tough practice to select a classification method since every classification methods have their own disadvantage and advantage. This paper helps to conclude which classification method is suitable for a particular application. One of the simplest algorithms to test classes is k-NN, but it is very time complex for making predictions. While classifying high-dimensional data set, Support Vector Machine (SVM) is the best available machine learning algorithm. In SVM, quadratic optimization aids in limiting the complexity of frequency of error and decision rule but has a disadvantage as it is hard to find the optimal parameter for training non-linear data making. NN has the ability to tolerate noisy input but has a hard understanding of algorithm structure.

To overcome the time complexity of k-NN and SVM-like classifiers, an edge detection technique can be applied, which promises to reduce the amount of data to be processed and to filter out less relevant information while preserving the important structural properties of an image. However, sometimes it becomes hard to obtain such edges in real-life images. Ansari et al. [8] compare the various edge detection techniques and found that the Sobel operator has better noise suppression than other detection techniques and gives a moderate result compared to others. Kaura et al. [9] have presented an expert system survey to detect and diagnose the leaf diseases by pixel by pixel comparison algorithm for comparing two images of leaf disease of cereal. This algorithm is time complex and many times due to irregular background conditions. The effectiveness of this algorithm is considerably reduced.

A Deep Learning (DL)-based survey has been presented by Kamilaris et al. [10]. This survey suggested that DL beat traditional approaches such as SVM and ANN. DL methods' automatic feature extraction seems to be more effective than the manual feature extraction process through conventional approaches such as histograms, Scale Invariant Feature Transform (SIFT), area-based techniques (ABT) GLCM, statistics, texture, color, shape-based algorithms, and visual texture features.

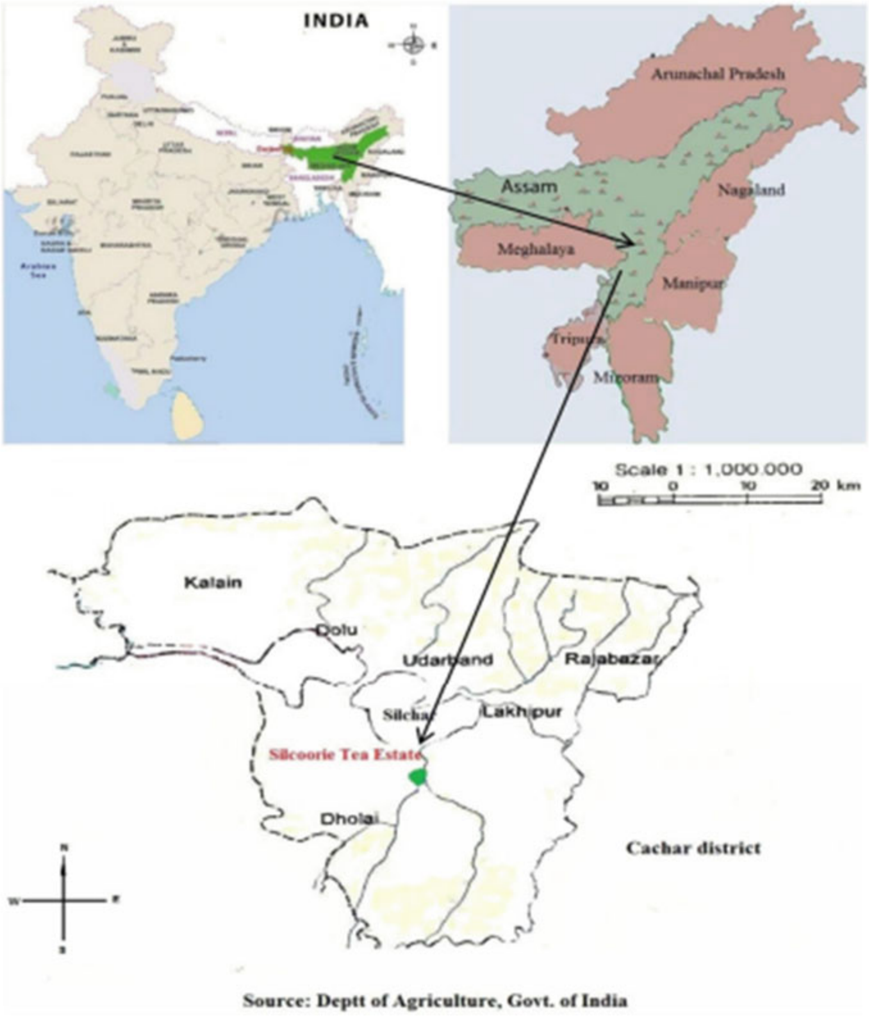
In recent times, CNN has produced benchmark results in various image classifications. Jaswal et al. [11] proved that CNN is the best classifier than others. Abdullahi et al. [12] have applied CNN for plant image recognition and classification. They have applied different image processing techniques like support vector machine, fuzzy logic approach, neural networks, etc. and discovered that the most effective approach and excellent results are obtained using the DL approach of image classification.

Most of the works have been done on some predefined databases. Not much work has been found in the field of tea disease detection using image processing techniques. This study proposes an improved CNN, which is the modification of pre-trained CNN such as AlexNet and ImageNet.

3 Materials and Methodology

3.1 Sample Collection and Preparing the Dataset

Collecting an unbiased and appropriate image dataset is the first step for any image classification analysis. Images were captured using a mobile camera of 8MP from the Silcoorie teas estate, Silchar, Assam. Figure 3 shows the area of the site of sample collection. A total of 117 images are found, out of which 32 are healthy leaves, and the others belong to either of the two previously mentioned diseases (Fig. 4). The images are then compressed and cropped to decrease the training time and ensure that features were learned only from the region of interest. Two classes of diseased leaves are labeled, and an extra class of healthy leaves is also added to differentiate the healthy leaves from the diseased ones. The images taken using the camera are normally of variable length and pixel size. These images are then resized to 100 × 100 by the OpenCV framework. To improve the classifier's generalization ability, which is more advantageous to the training of the network. Four different flips horizontal, flip vertical, left rotate 90 degrees, and right rotated 90 degrees were used to alter the image input and improve the classification. The data augmentation of the normal, Mosquito Bug and Red spider mite tea leaf images is shown in Figs. 5, and



Source: Deptt of Agriculture, Govt. of India

Fig. 3 Area of sample collection

6, respectively. Figures 7, 8, and 9 show some of the training images of the normal Mosquito Bug and Red spider mite tea leaf images.

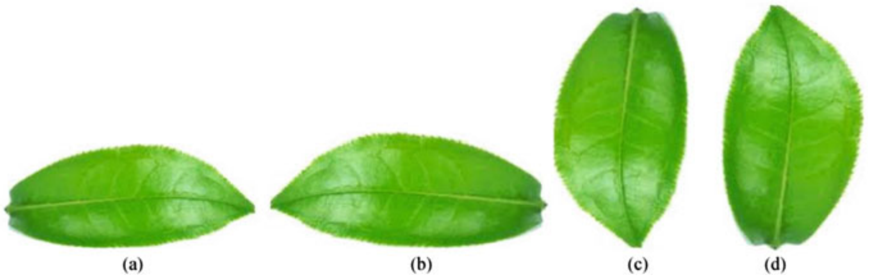


Fig. 4 Normal tea leaf data augmentation examples **a** flip horizontal **b** flip vertical **c** right-rotate 90° **d** left-rotate 90°

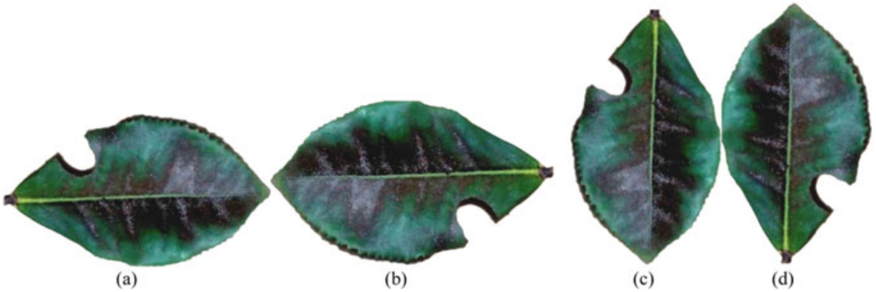


Fig. 5 Mosquito bug tea leaf data augmentation examples **a** flip horizontal **b** flip vertical **c** right-rotate 90° **d** left-rotate 90°

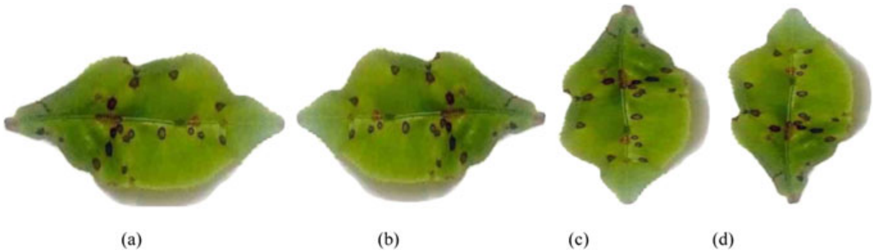


Fig. 6 Red spider mite tea leaf data augmentation examples **a** flip horizontal **b** flip vertical **c** right-rotate 90° **d** left-rotate 90°

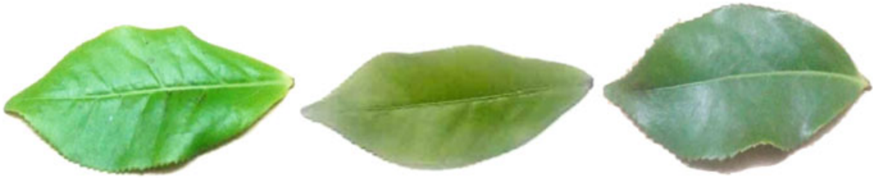


Fig. 7 Sample normal tea leaf training images

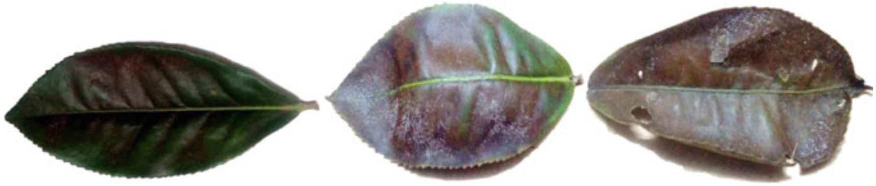


Fig. 8 Sample mosquito bug tea leaf training images

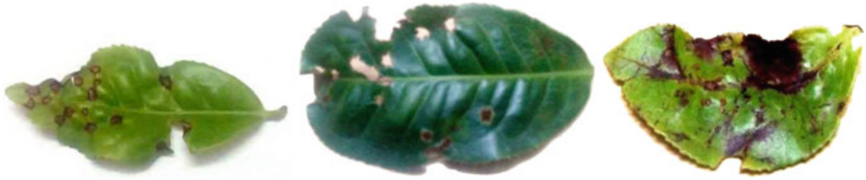


Fig. 9 Sample red spider mite tea leaf training images

3.2 Convolutional Neural Network (CNN)

CNN is a deep learning tool best suited for processing data with a grid-like pattern present in digital images. It is designed to divide hierarchies of features in a bottom-up manner automatically. CNN is composed of three layers—the convolution layer, the pooling layer, and the fully-connected layers.

3.2.1 Convolution Layer

A convolution layer is a basic constituent of the CNN architecture. It performs feature extraction through numerous linear and non-linear functions like convolution operation and activation function. The tensor's output at a certain point is obtained by summing up the elementary products between the kernel and the input tensor at each location of the tensor. This output is called a feature map. This procedure is repeated, and multiple kernels are applied to form an arbitrary number of feature maps. These feature maps depict different characteristics of the input tensors and are taken to be different feature extractors.

3.2.2 Pooling Layer

The pooling layer is the second layer of CNN, which provides a typical downsampling operation that decreases the in-plane feature maps' dimensionality and reduces the number of subsequent learnable parameters. The pooling layers do not contain any learnable parameter; it contains hyperparameters for pooling operations like filter size, padding, and stride similar to convolution operations.

3.2.3 Fully Connected Layer

After the extraction of features by the convolution layers and pooling layer down-sampling, the features like probabilities for each class of the classification tasks are mapped by a subset of fully connected layers to the network's final outputs. The final fully connected layer normally has an equal number of output nodes as the number of classes. A nonlinear function-Rectified Linear Unit follows each fully connected layer.

3.3 Training Parameters

3.3.1 Rectified Linear Unit (ReLu)

The Rectified Linear Unit is the most frequently used activation function in deep learning models. The function returns '0' for negative input and returns the value if the input is positive; it is presented in Eq. (1).

$$f(x) = \max(0, x) \tag{1}$$

3.3.2 Soft-Max Layer

It is an activation function that normalizes the input vector of 'N' real numbers, into a probability distribution consisting of 'N' probabilities that sum up to '1'. It is frequently used in NN, for mapping the non-normalized output of a network to a probability distribution over predicted output classes. The soft-max function utilized for this architecture is represented in Eq. 2.

$$\sigma(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{N} e^{y_i}} \tag{2}$$

For $i = 1, 2, 3\ldots N$ and $y = (y_1, y_2\ldots.y_k) \in N$.

3.3.3 Adaptive Moment Estimation (Adam)

The adaptive learning rates for each parameter are computed by a method called Adam. It stores an exponentially decaying average of past squared gradients 'v_t' like a delta and RMS prop, and also keeps an account of exponentially decaying average of past gradients 'm_t', similar to momentum. Adam acts like a heavy ball with friction, hence favors flat minima in the error surface in comparison to momentum which can be visualized as a ball running down a slope. The decaying averages of past and past squared gradients 'm_t' and 'v_t' are computed respectively found in Eqs. (3) and (4).

$$m_t = \beta_1 m_t - 1 + (1 - \beta_1) g_t \tag{3}$$

$$v_t = \beta_2 v_t - 1 + (1 - \beta_2) g_t^2 \tag{4}$$

'm_t' and 'v_t' are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the stochastic gradients 'g_t' respectively and β_1, β_2 are the hyperparameters function.

There are several popular frameworks for deep learning evolving recently, like Pytorch, Tensorflow, Sonnet, Keras, etc. Out of these, Google's Tensorflow comes out to be the most promising framework nowadays. Its core is developed in C++, and it uses Python to provide a convenient front-end. It is suitable for both easy model building and powerful experimentation for research. For our work, we used Tensorflow as the building block of the CNN architecture. The network architecture is an improvement of the AlexNet and ImageNet structure. The time complexity of this architecture is much better compared to the other structures. It consists of five convolution layers, two fully connected layers. The layered architecture is described in Table 1.

4 Results and Discussions

All the experiments were performed in Python version 3.6.5 in the Spyder Notebook of Anaconda prompt. The machine used was 64 bit, 8 GB RAM, Windows 10 Lenovo Ideapad 330 supporting Nvidia MX 150 2 GB graphics card. As shown in Fig. 10, the validation accuracy increases with the increase in the number of epochs. The accuracy was initially low when found for 50 epochs validation accuracy was 55.56%, but it increased to 92.59% when found at 200 epochs. But after that increase in the number of epochs does not have much effect on the accuracy. Records were taken for 50,100,150, and 200 number of epochs, and the respective validation accuracies are 55.56, 77.78, 88.89, and 92.59%.

This study has samples of two types of diseased leaves, mosquito bug and red spider mite, and a collection of healthy leaves. So the classification is based on these three types. The accuracy of the CNN, SVM, and k-NN classifiers in determining

Table 1 Proposed architecture of CNN

Layers	Parameters	Activation function
Input	100 × 100x3	–
Convolutional (layer1)	Filters = 18, Kernel size = 3 × 3	ReLu
Pooling (layer 1)	Max pooling (3 × 3)	–
Convolutional (layer 2)	Filters = 27, Kernel size = 3 × 3	ReLu
Pooling (layer 2)	Max pooling (3 × 3	–
Convolutional (layer 3)	Filters = 81, Kernel size = 3 × 3	ReLu
Pooling (layer 3)	Max pooling (3 × 3)	–
Convolutional (layer 4)	Filters = 9, Kernel size = 3 × 3	ReLu
Pooling (layer 4)	Max pooling (3 × 3)	–
Convolutional (layer 5)	Filters = 27, Kernel size = 3 × 3	ReLu
Pooling (layer 5)	Max pooling (3 × 3)	–
Fully Connected (layer 1)	243 Nodes	ReLu
Fully Connected (layer 2)	3 Neurons	Softmax

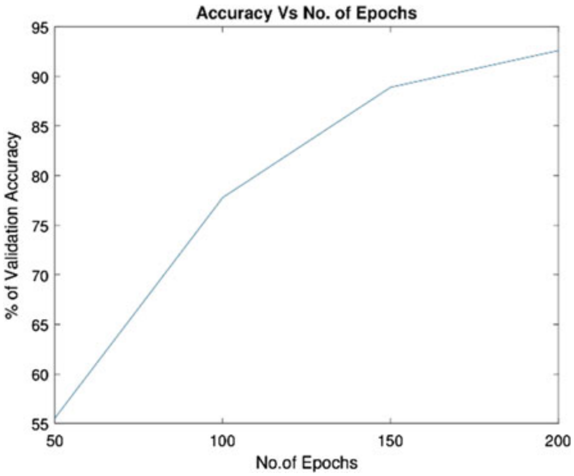


Fig. 10 Validation accuracy versus number of epochs

the tea leaves diseases is evaluated. The corresponding Error matrices were used to calculate the classifiers' accuracy, as described in Tables 2, 3, and 4. Tables show that CNN works better than the classical machine learning algorithms like SVM and k-NN in classifying the diseases. The absence of artificial feature detection in these

Table 2 Error matrix of k-NN

	Mosquito bug	Red spider mite	Healthy leaves
Mosquito bug	11	0	0
Red spider mite	0	7	0
Healthy leaves	2	0	7

Table 3 Error matrix of SVM

	Mosquito bug	Red spider mite	Healthy leaves
Mosquito bug	10	2	0
Red spider mite	1	7	1
Healthy leaves	1	2	3

Table 4 Error matrix of CNN

	Mosquito bug	Red spider mite	Healthy leaves
Mosquito bug	7	1	4
Red spider mite	0	5	4
Healthy leaves	0	2	4

machine learning classifiers may be a reason for low accuracy in image detection techniques. It is a good practice to compare the achieved results with some other works. Hence, in addition to the above algorithms, the proposed method is compared against two previously presented algorithms, such as NN-based ensemble classifier [10], SVM [13], presented in Table 5. From the comparison, it can be found that CNN provides better results in tea leaf disease detection.

The classification accuracy of the proposed CNN was contrasted and recently published LeafNet CNN [14], which is not high (Shown in Fig. 11), because of

Table 5 Comparison of the proposed CNN architecture results with other classical machine learning algorithms

Methods	Number of diseases classified	Accuracy (%)
Proposed CNN	2	92.59
k-NN	1	91
SVM	2	91

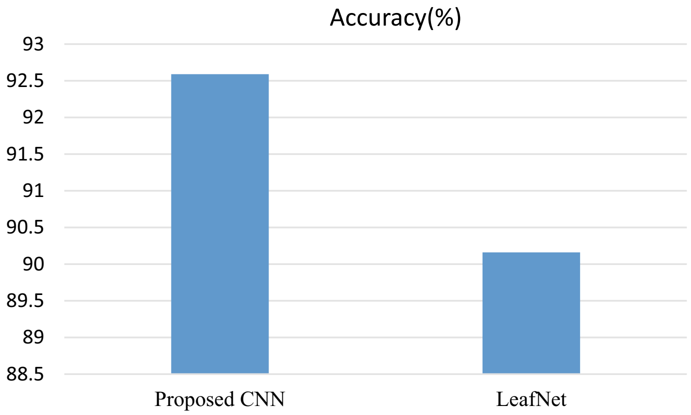


Fig. 11 Comparison of the proposed CNN with previously published LeafNet

the need for the artificial selection of features. To a huge degree, the presentation of these strategies relies upon whether examiners' attributes are sensible. At the same time, agents, for the most part, depend on understanding and show noteworthy naivety when choosing features. Albeit better outcomes are gotten by utilizing artificial feature classification, these features are explicit for datasets. Results may vary extensively if similar features are utilized to break down various data sets.

5 Conclusion

This paper presents an automatic tea leaf disease detection using image classification techniques and found CNN one of the best in this category. The identification of these diseases is made manual, which can consume more time or may be costly. This automatic method is to identify and classify the diseases accurately from leaf images. The unsupervised learning technique used in CNN captures even those patterns that are not detected by supervised learning techniques like SVM or k-NN. In the future, our idea is to build a mobile and web application for the farmers for early detection of tea diseases based on this idea. A GUI application based on CNN can be developed, as CNN is found to be the best of the others. Pathogens like red-spider mite are present on tea all year round, cause heavy damage, and require a good management process. So building an app where the farmers can enter the affected plant leaf image can know its category and process of control, and in this way, it will be much helpful to the farmers. We will try to cover more endemic diseases, which will further help people from all regions. We will try to increase the accuracy of disease detection by possible modifications in the algorithm and increase the number of images. Hopefully, this will be a complete tea disease control solution by the above-mentioned means and will reduce the chances of the epidemic in any region and thus raise the yield of tea and, therefore, farmers' prosperity.

References

1. Agnihothrudu V (1999) Potential of using biocontrol agents in tea. Global advances in tea science. Aravali Books, New Delhi, pp 675–692
2. Muraleedharan N, Chen ZM (1997) Pests and diseases of tea and their management. J Plant Crop 25:15–43
3. Hazarika LK, Bhuyan M, Hazarika BN (2009) Insect pests of tea and their management. Annu Rev Entomol 54:267–284
4. Das GM (1959) Bionomics of the tea red spider, *Oligonychus coffeae* (Nietner). Bull Entomol Res 50(2):265–274
5. Muraleedharan PK, Seethalakshmi KK (1993) Rattan plantation and its profitability. In: Rattan management and utilisation. Proceedings of the Rattan (Cane) Seminar India, Trichur. KFRI, Peechi and IDRC, Canada, pp 86–103
6. Roy S, Muraleedharan N, Mukhapadhyay A, Handique G (2015) The tea mosquito bug, *Helopeltis theivora* Waterhouse (Heteroptera: Miridae): its status, biology, ecology and management in tea plantations. Int J Pest Manage 61(3):179–197
7. Ghaiwat SN, Arora P (2014) Detection and classification of plant leaf diseases using image processing techniques: a review. Int J Recent Adv Eng Technol 2(3):1–7
8. Ansari MA, Kurchaniya D, Dixit M (2018) A comprehensive analysis of image edge detection techniques. Int J Multimedia Ubiquitous Eng 12:1–12
9. Kaura R, Dina S, Pannub PPS (2013) Expert system to detect and diagnose the leaf diseases of cereals. Int J Curr Eng Technol 3(4):1480–1483
10. Kamilaris A, Prenafeta-Boldú FX (2018) Deep learning in agriculture: a survey. Comput Electron Agric 147:70–90
11. Jaswal D, Vishvanathan S, Kp S (2014) Image classification using convolutional neural networks. Int J Sci Eng Res 5(6):1661–1668
12. Abdullahi HS, Sheriff R, Mahieddine F (2017) Convolution neural network in precision agriculture for plant image recognition and classification. In: 2017 Seventh international conference on innovative computing technology (Intech), IEEE, Londrés, pp 1–3
13. Hossain MS, Mou RM, Hasan MM, Chakraborty S, Razzak MA (2018) March. recognition and detection of tea leaf's diseases using support vector machine. In: 2018 IEEE 14th International colloquium on signal processing & its applications, pp 150–154
14. Chen J, Liu Q, Gao L (2019) Visual tea leaf disease recognition using a Convolutional Neural Network model. Symmetry 11(3):343

Epileptic Seizure Classification Using Spiking Neural Network from EEG Signals

Irshed Hussain and Dalton Meitei Thounaojam

Abstract Epilepsy is a life-threatening and challenging neurological disorder which results in a seizure. The seizure makes the functioning of the central nervous system abnormal and the patient ends up with convulsions and a loss of consciousness. Therefore, there is a need for in-depth diagnosis with high accuracy of the disease where electroencephalogram (EEG) signals play a vital role. Spiking neural network (SNN) is used as the classifier to classify EEG-based epileptic seizures. Due to its computational efficiency and biological plausibility, SNN is getting more attention to the classification of time-series data such as EEG signals. Although there are several other classification methods that exist, few of them is able to work with the EEG signals because EEG signals are generally highly variable and noisy. The SpiFoG algorithm is used to train SNN where inputs are encoded in temporal spikes from the feature extracted EEG signals. The dataset used in this research is the Bonn University EEG dataset for the epileptic seizure. From this dataset, three types of epileptic EEG signals are considered. The first is from healthy patients with open eyes, second is from healthy patients with closed eyes, and the third is from patients suffering from a seizure. Our proposed model outperforms state-of-the-art models.

Keywords Epileptic seizure · EEG · SpiFoG · Spiking neurons

1 Introduction

An unexpected electrical disturbance in the central processing part of the human brain results loss of consciousness and convulsion which is termed as the epileptic seizure [1–3]. It is a kind of abnormality in the firing mechanism of the neurons [4], those carries the most essential information and the tragic part is the lack of lucid information about the root cause of the disease. According to the World Health Organisation (WHO) data, to date, more than 50 million people are suffering worldwide from this life-threatening disease [5]. Therefore, an automated in-depth diagnosis with high accuracy is very essential and it is challenging also to the medical science commu-

I. Hussain (✉) · D. M. Thounaojam
National Institute of Technology Silchar, Silchar, Assam 788010, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_23

nity. In this scenario, electroencephalogram (EEG) comes into the picture. With the use of EEG, it is very effective to record and analyse the brains electrical activity [6]. EEG uses electrodes that can be non-invasive and can be set up on the scalp of the patient to retrieve the activity of electrical pulses which play a vital role in the recognition of epileptic seizure.

Manually, it is very difficult to detect seizures just by observing raw EEG signals because of the highly variable, highly nonlinear and highly non-stationary property of EEG signals [7]. Therefore, an automated detection system using an intelligent approach such as machine learning (ML) is important. Spiking neural network (SNN) is a very promising and attractive approach to ML nowadays. SNN is getting more and more attention in the field of neuroscience as well as in the field of classification of highly nonlinear patterns. SNN is considered as the third generation of neural network [8], which has the interesting properties such as computational efficiency [8, 9], biological plausibility [10] and energy efficiency [10, 11]. SNN imitates the human brain more closely and works in a different manner than that of its ancestor, the second-generation artificial neural network (ANN). In SNN, all input and output information is temporal rather than real-valued numbers, and thus, its characteristics match more closely with the characteristics of brains information processing since neuroscience proves to some extent that the brain uses temporal information rather than average firing rate as information [12]. Although brains information encoding is not very clear to date [11, 13], akin to most of the researchers, population coding is used in this research as used in [14]. Since in order to work with SNN in case of the classification task, there is a need for a proper selection of spiking neuron model, and a supervised learning algorithm. In this paper, leaky-integrate-and-fire (LIF) [15–17] spiking neuron model and SpiFoG learning algorithm [14] are used. The information sending neuron pre-synaptic neuron sends the information to its corresponding information receiver neuron post-synaptic neuron and upon receiving the stimuli when the post-synaptic potential (PSP) of the post-synaptic neuron reaches a threshold, the post-synaptic neurons fire spike.

Feature extraction is the most crucial and challenging part since it plays a vital role in classifying patterns into their respective classes. In our proposed approach, features are extracted from the raw EEG signal using Petrosian fractal dimension (PFD) [18], Higuchi fractal dimension (HFD) [19], SVD entropy [20], Fisher information [21] and detrended fluctuation analysis (DFA) [22]. The definition of all the aforementioned features is also explained in detail in [23]. The hyper-parameters used in our proposed model as used in SpiFoG [14] except for the range of synaptic weights and the range of random synaptic delays. Here in this research, a mixture of 50% excitatory synapse and 50% inhibitory synapse within the range $[-1, 1]$, and the range for random synaptic delays is set to $[1, 10]$. Moreover, we have analysed a variation in the population size. The experimental results for the proposed model are found better than the state-of-the-art algorithms.

The contributions of this research are:

1. Classification of EEG-based epileptic seizure using SNN and SpiFoG learning algorithm.
2. Range of synaptic weights is changed to [−1, 1] from [−0.25, 1]. Therefore, in this case, 50% excitatory, and 50% inhibitory neurons are present.
3. Range of random delays in the synapse is changed from [1, 6] to [1, 10] allowing more variation in the delay time.
4. Variation of population size using SpiFoG algorithm to analyse the exploration of the search space.

This paper is organised in sections and subsections for better clarity. Section 1 discusses the background concept and novelty, and Sect. 2 summarises the state-of-the-art followed by Sect. 3 which explains the development process of the proposed approach, feature extraction and classification. The experimental results and a detailed discussion is presented in Sect. 4 followed by the summary of the research given in Sect. 5.

2 Related Work

From the past few years, with the advancement of cognitive science, EEG remains one of the most attractive and popular techniques. In this section, some of the previous work on EEG-based signals is discussed. In [24, 25], Fourier transform is successfully applied to the task of frontal EEG asymmetry and emotion. However, the demerit of these methods is the ability to analyse the raw EEG signal in the frequency-domain only, neither time-domain nor time–frequency-domain information was used. In [26], EEG-based epileptic seizures were recognised with the use of time–frequency-domain feature information discrete wavelet transform (DWT) and classification method second-generation ANN. As DWT can extract both the time-domain and frequency-domain information from an EEG signal, many researchers have used this method to carry out the desired task [27, 28]. Ghosh Dastidar et al. [29] used the SNN classification method to classify epileptic seizures from the EEG signals. They have considered three-class classification approach one is healthy class, the second is transition class, and remaining is the seizure class. The DWT is used as the feature extraction method to extract wave information such as alpha, beta, gamma, delta and theta. From these wave frequencies, pre-synaptic initial input spike times are generated and then SNN is applied to classify the patterns to their respective classes.

Although Fourier transform is not capable to provide the time-domain information for the EEG signals, short-time Fourier transform (STFT) can do that by applying the windowing concept. This approach is also used by many researchers to analyse EEG signals. However, this method lacks behind when the selection of window size comes into the picture because it uses the same size window which can be incompatible in case of highly variable frequencies. Tzallas et al. [30] used the concept of STFT for the recognition of epileptic seizure. In [31], to analyse EEG background activity in

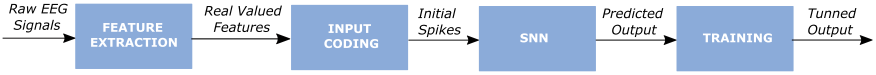


Fig. 1 Block diagram of the entire procedure followed for the development of the proposed model

Autism disease, STFT was used. In [32], Gupta et al. analysed the raw EEG-based epileptic seizure dataset using DWT and they have used convolutional neural network (CNN) as the classifier.

3 Materials and Methods

In this section, the procedure followed to develop the proposed model is discussed. Figure 1 shows the block diagram of entire methodology where it is observed that firstly real-valued features are obtained from the raw EEG signals by feature extraction method, and then those real-valued features are converted into temporal information such as spikes. Now, upon receiving spikes, SNN produces output called predicted output those are trained using SpiFoG [14] algorithm to produce better output so that total loss is minimised.

3.1 Features Extraction

We have extracted five features from the raw EEG signals, namely DFA [22], HFD [19], SVD entropy [20], Fisher information [21] and PFD [18] to produce feature vectors f_1, f_2, f_3, f_4 and f_5, respectively. Figure 2a shows the behaviour of raw EEG signals in case of a seizure patient and a healthy one where it is observed that the individual suffering from seizure is having higher value of frequencies than the healthy one for each of the 100 channels.

In Figs. 2b and 3a, b frequency of all 200 channels are shown for each of the five features f_1, f_2, f_3, f_4 and f_5. In Fig. 2b (top), feature vector f_1 is representing the DFA values where it is observed that the frequencies are varying much after 100 channels, those are frequencies of the seizure. Here, in Fig. 2b (bottom) feature vector f_2 represents HFD features.

In Fig. 3a (top), it is observed that SVD entropy (f_3) for the first 100 channels have almost similar frequencies, and for the remaining 100 channels, the frequency values of SVD entropy are also almost similar. The feature vector f_4 in Fig. 3a (bottom) which represents Fisher information is having higher frequencies for the first 100 channels. In Fig. 3b (top), it has been observed that PFD features (f_5) for the last 100 channels are having higher frequencies than that of the first 100 channels. Moreover, labelled desired output data for the two classes healthy and seizure is shown in Fig. 3b (bottom).

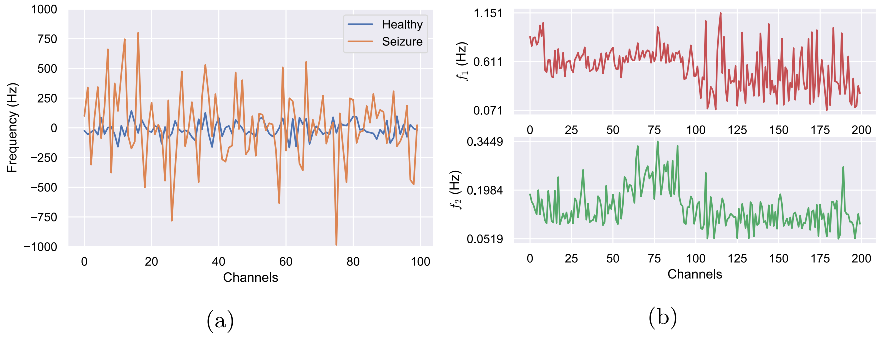


Fig. 2 **a** Behaviour of raw EEG signals in case of a healthy person and a patient having a seizure (set B and E). **b** Frequency of extracted feature DFA (feature f_1) and HFD (feature f_2) from the raw EEG signals for all 200 channels (set B and E)

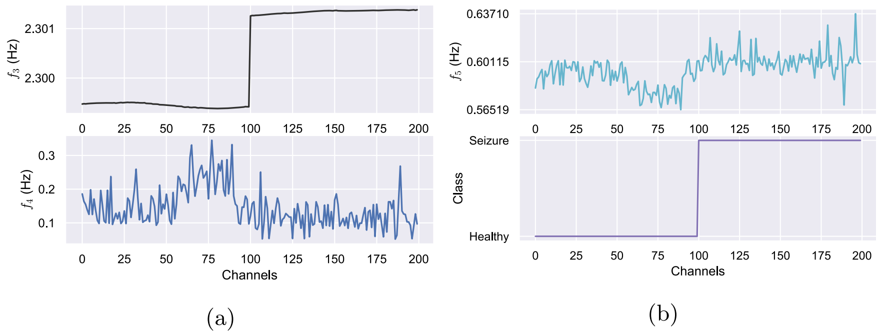


Fig. 3 (a) Frequency of extracted feature SVD entropy (feature f_3) and Fisher information (feature f_4) from the raw EEG signals for all 200 channels (set B and E) (b) Frequency of extracted feature PFD (feature f_5) from the raw EEG signals for all 200 channels, and labelled desired output data for the two classes Healthy and Seizure (set B and E)

3.2 Classification

SNN is used for classification where SpiFoG [14] algorithm is used to train the proposed model. The neuron model LIF is used to produce spikes, and the dynamics for a neuron j (post-synaptic) is given in Eq. 1.

$$\tau_{\text{mem}} \frac{\mathrm{d}U_j(t)}{\mathrm{d}t} = -U_j(t-1) + \sum_{j=1}^{N} \sum_{i=1}^{M} W_{ij} \times \psi(t - t_i - K_{ij}) \times R_{\text{mem}} \tag{1}$$

where $\mathrm{d}U_j(t)$ = small change in the PSP of post-synaptic neuron j at time t, $\mathrm{d}t$ = time step, R_{mem} = membrane resistance, τ_{mem} = membrane time constant. ψ is the double decaying kernel function where K_{ij} is the random delay in the synapse having values within the interval [1, 10]. W_{ij} is the synaptic strengths of synaptic weights between neuron i and neuron j having values within the interval $[-1, 1]$ and thus allows a mixture of 50% excitatory and 50% inhibitory neurons in the architecture that adds biological plausibility to the system. Our model has the topological structure of 16:8:1 that means there are $5 \times 3 + 1 = 16$ (5 real-valued features, 3 encoding neurons and 1 bias neuron) pre-synaptic input neurons, 8 hidden neurons and 1 output neuron. Single synapse model with the use of double decaying kernel is used.

4 Results and Discussion

In this section, the experimental results along with the description of dataset set A, set B and set E are discussed.

4.1 Dataset Description

The EEG-based epilepsy seizure dataset used in this research is the popular Bonn University dataset [33]. In the dataset, there are five different directories each having data for 100 channels namely A, B, C, D, E. Directory A holds the EEG data for normal person recorded with the eyes open, B holds the EEG data for normal person recorded with the eyes closed, C holds the EEG data from the healthy brain area although at the recording time tumour was identified, EEG data recorded from the tumour area is placed in directory D, and directory E has the EEG data of patients suffering from an epileptic seizure. Although there are five different types of classes, we can easily convert this dataset into binary classification where the status of the disease can be predicted (i.e. whether a person is suffering from an epileptic seizure or healthy). For this, we have arranged the dataset in two different ways and performed two binary classification task. First binary classification is between healthy with eyes closed vs seizure (i.e. B and E), and the second binary classification is between healthy eyes open vs seizure (i.e. A and E). Each data channel (total 100 channels) has 4097 data points sampled at 173.62 Hz yielding raw EEG signals up to the duration of 23.59 sec.

4.2 Performance Evaluation of the Proposed Model

The performance is analysed in terms of classification accuracies namely training accuracy and testing accuracy. In Table 1, the results with the variation of population are presented where it is observed that, the testing accuracy for population size 40 is found better in case of dataset B and E. For population size 40, 92.24% training accuracy and 92.54% testing accuracy are achieved.

Table 1 Performance for binary dataset B and E

Population size	Classification accuracy	
	Training (%)	Testing (%)
10	91.62	91.60
20	91.85	91.78
30	91.62	91.60
40	92.24	92.54
50	92.29	92.52
60	92.28	92.42
70	92.29	92.52
80	92.29	92.52
90	92.29	92.52
100	91.85	91.78

Table 2 Performance for binary dataset A and E

Population size	**Classification accuracy**	
	Training (%)	Testing (%)
40	94.36	95.77

Table 3 Performance comparison for binary dataset B and E

State-of-the-art	Approach	Test accuracy (%)
Ahmedt et al. [34]	EEG signals (raw) + LSTM	92.50
Acharya et al. [35]	EEG signals (raw) + CNN	88.70
Our proposed model	SNN + SpiFoG	92.54

For set A and set E, we have experimented with our model with population size 40. In Table 2, the results are presented where it shows 94.36% training accuracy and 95.77% testing accuracy. In Table 3, a comparison of our proposed model is shown with the state-of-the-art algorithms in case of set B and set E. It is clearly observed that the proposed model outperforms state-of-the-art algorithms.

In Table 4, a comparison of our proposed model is shown with the state-of-the-art algorithms in case of set A and set E. In this case, also, the proposed model shows very competitive results compared to the state-of-the-art algorithms.

The average training accuracies of all the population vs the number of generations are shown in Fig. 4a for the set B and E. We have trained the model till 1000 generations but the graphs are shown only for 100 generations where the changes in training accuracy occur to visualise more clearly. After 100 generations, the proposed model converges towards the optimal value. Figure 4b shows the behaviour of training in case of dataset A and E when population size 40 is used.

Table 4 Performance comparison for binary dataset A and E

State-of-the-art	Approach	Test accuracy (%)
Lasefr et al. [36]	SVM + Energy thresholding	96.20
Ahammad et al. [37]	DWT + Linear classifier	84.20
Ahmedt et al. [34]	EEG signals (raw) + LSTM	97.00
Our proposed model	SNN + SpiFoG	95.77

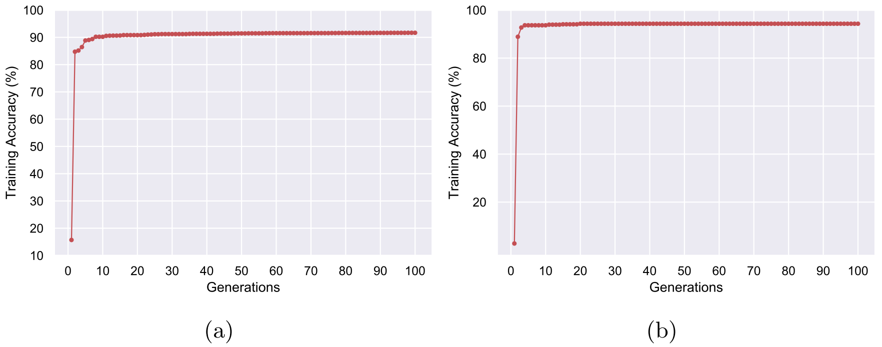


Fig. 4 **a** Average training accuracy for all variations in the population size using set B and E. **b** Training accuracy for the population size 40 using set A and E

For the set A and set E, we have used the population size 40 since using set B and set E it had been found that varying the size of the population does not really improve the testing as well as training accuracy drastically.

5 Conclusion

We have explored the computational power and biological plausibility of LIF spiking neuron and SpiFoG algorithm with the help of highly nonlinear EEG-based epileptic seizure data. Our proposed model successfully classifies the patients suffering from epileptic seizure from the healthy ones. The topology followed by our model is akin to SpiFoG except for the range of synaptic weights and the range of delays in the synapse. The SpiFoG uses random weight initialisation range as $[-0.25, 1]$ and range of random delay values as $[1, 6]$ but we have used $[-1, 1]$ and $[1, 10]$. The usage of 50% excitatory and 50% inhibitory synapse improves biological plausibility to some extent. Moreover, a variation in the population size is carried out to analyse the exploration of the search space.

This work can be extended to multiclass classification considering each five sets of data individually where there will be five classes.

References

1. Bezobrazova S, Golovko V (2007) Comparative analysis of forecasting neural networks in the application for epilepsy detection. In: 2007 4th IEEE workshop on intelligent data acquisition and advanced computing systems: technology and applications. IEEE, pp 202–206
2. Orhan U, Hekim M, Ozer M, Provaznik I (2011) Epilepsy diagnosis using probability density functions of EEG signals. In: 2011 international symposium on innovations in intelligent systems and applications. IEEE, pp 626–630
3. Veisi I, Pariz N, Karimpour A (2007) Fast and robust detection of epilepsy in noisy EEG signals using permutation entropy. In: 2007 IEEE 7th international symposium on bioinformatics and bioengineering. IEEE, pp 200–203
4. Scharfman HE (2007) The neurobiology of epilepsy. Curr Neurol Neurosci Rep 7(4):348–354
5. Epilepsy. https://www.who.int/news-room/fact-sheets/detail/epilepsy (20 June 2019)
6. Ocak H (2009) Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy. Expert Syst Appl 36(2):2027–2036
7. Klonowski W (2009) Everything you wanted to ask about EEG but were afraid to get the right answer. Nonlinear Biomed Phys 3(1):1–5
8. Maass W (1997) Networks of spiking neurons: the third generation of neural network models. Neural Networks 10(9):1659–1671
9. Maas W (1997) Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons. Adv Neural Inform Process Syst 9:211–217
10. Gerstner W, Kistler WM (2002) Spiking neuron models: Single neurons, populations, plasticity. Cambridge University Press, Cambridge
11. Natschläger T, Ruf B (1998) Spatial and temporal pattern analysis via spiking neurons. Network: Comput Neural Syst 9(3):319–332
12. Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol 117(4):500–544
13. Bialek W, Rieke F, Van Steveninck RDR, Warland D (1991) Reading a neural code. Science 252(5014):1854–1857
14. Hussain I, Thounaojam DM (2020) Spifog: an efficient supervised learning algorithm for the network of spiking neurons. Sci Rep 10(1):1–11
15. Stein RB (1965) A theoretical analysis of neuronal variability. Biophys J 5(2):173–194
16. Stein RB (1967) Some models of neuronal variability. Biophys J 7(1):37–68
17. Vazquez RA, Cachón A (2010) Integrate and fire neurons and their application in pattern recognition. In: 2010 7th International conference on electrical engineering computing science and automatic control. IEEE, pp 424–428
18. Petrosian A (1995) Kolmogorov complexity of finite sequences and recognition of different preictal eeg patterns. In: Proceedings eighth IEEE symposium on computer-based medical systems. IEEE, pp 212–217
19. Higuchi T (1988) Approach to an irregular time series on the basis of the fractal theory. Physica D: Nonlinear Phenomena 31(2):277–283
20. Roberts SJ, Penny W, Rezek I (1999) Temporal and spatial complexity measures for electroencephalogram based brain-computer interfacing. Medical Biol Eng Comput 37(1):93–98
21. James CJ, Lowe D (2003) Extracting multisource brain activity from a single electromagnetic channel. Artif Intell Med 28(1):89–104
22. Peng CK, Havlin S, Stanley HE, Goldberger AL (19951) Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. Chaos: An Interdiscip J Nonlinear Sci 5(1):82–87
23. Bao FS, Liu X, Zhang C (2011) Pyeeg: an open source python module for EEG/MEG feature extraction. Comput Intell Neurosci 2011
24. Allen JJ, Coan JA, Nazarian M (2004) Issues and assumptions on the road from raw signals to metrics of frontal EEG asymmetry in emotion. Biol Psychol 67(1–2):183–218
25. Sałabun W (2014) Processing and spectral analysis of the raw EEG signal from the mindwave. Przeglad Elektrotechniczny 90(2):169–174

26. Kumar Y, Dewal M, Anand R (2014) Epileptic seizures detection in EEG using DWT-based apen and artificial neural network. Signal Image Video Process 8(7):1323–1334
27. Omerhodzic I, Avdakovic S, Nuhanovic A, Dizdarevic K (2013) Energy distribution of EEG signals: EEG signal wavelet-neural network classifier. arXiv preprint arXiv:1307.7897
28. Sharmila A, Geethanjali P (2016) Dwt based detection of epileptic seizure from EEG signals using Naive Bayes and k-nn classifiers. IEEE Access 4:7716–7727
29. Ghosh-Dastidar S, Adeli H (2007) Improved spiking neural networks for EEG classification and epilepsy and seizure detection. Integr Computer-Aided Eng 14(3):187–212
30. Tzallas AT, Tsipouras MG, Fotiadis DI (2009) Epileptic seizure detection in EEGs using time-frequency analysis. IEEE Trans Inform Technol Biomed 13(5):703–710
31. Behnam H, Sheikhani A, Mohammadi MR, Noroozian M, Golabi P (2007) Analyses of EEG background activity in autism disorders with fast fourier transform and short time fourier measure. In: 2007 International conference on intelligent and advanced systems. IEEE, pp 1240–1244
32. Gupta S, Bagga S, Maheshkar V, Bhatia M (2020) Detection of epileptic seizures using EEG signals. In: 2020 International Conference on Artificial Intelligence and Signal Processing (AISP). IEEE, pp 1–5
33. Andrzejak RG, Lehnertz K, Mormann F, Rieke C, David P, Elger CE (2001) Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state. Physical Review E 64(6):061907
34. Ahmedt-Aristizabal D, Fookes C, Nguyen K, Sridharan S (2018) Deep classification of epileptic signals. In: 2018 40th Annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp 332–335. IEEE
35. Acharya UR, Oh SL, Hagiwara Y, Tan JH, Adeli H (2018) Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. Computers Biol Med 100:270–278
36. Lasefr Z, Ayyalasomayajula SSV, Elleithy K (2017) Epilepsy seizure detection using EEG signals. In: 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON). IEEE, pp 162–167
37. Ahammad N, Fathima T, Joseph P (2014) Detection of epileptic seizure event and onset using EEG. BioMed Res Int 2014

Policing Android Malware Using Object-Oriented Metrics and Machine Learning Techniques

Anand Tirkey, Ramesh Kumar Mohapatra, and Lov Kumar

Abstract The primary motive of a malware is to compromise and exfiltrate sensitive user data from a system generally designed to uphold the fundamental principles of information security, i.e., confidentiality, integrity and availability. Android being the most widely used mobile operating system is a lucrative ground for malware designers in leveraging system flaws to gain unauthorized user information access. In order to attenuate these issues, it is imperative to design and build robust automated tools for effective android malware prediction. In this paper, we bring forward a novel method for android malware detection using object-oriented software metrics and machine learning techniques. Five thousand and seven hundred and seventy-four android apps are collected from AndroZoo repository, then its software metrics are extracted and aggregated using sixteen aggregation measures which forms the basis of our metrics-based dataset. A total of three hundred and four different machine-learned models are built using various data sampling techniques, feature selection methods and machine learning algorithms. Finally, a machine-learned model built using SVMSMOTE data sampling technique applying significant predictor metrics (SPM) feature selection methods over GDCG2H (conjugate gradient with Powell/Beale Restarts and two hidden layers) machine learning algorithm, yields a better malware predictor with area under ROC curve (AUC) value of 0.86.

Keywords Android malware detection · Machine learning · Object-oriented metrics

1 Introduction

Android leads the mobile OS market with a share of 86.1% and is expected to increase to 86.5% by 2021 according to International Data Corporation (IDC, USA).

A. Tirkey (✉) · R. K. Mohapatra
National Institute of Technology Rourkela, Rourkela, Odisha, India
e-mail: rkmohapatra@ieee.org

L. Kumar
BITS Pilani, Hyderabad Campus, Hyderabad, Telangana, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_24

Meanwhile, in 2019, Google reported that 42.1% of android devices run unsupported versions of the OS. Internet Security Threat Report 2019 (ISTR), published by Symantec, shows that it blocked an average of 10,573 malicious mobile apps per day. Malware is more prevalent in apps categorized under tools (39%), lifestyle (15%) and entertainment (7%). It also reported that one in thirty-six mobile devices had high-risk apps installed. Nohl and Lell [11] show that security in android ecosystem is further compromised by handset vendors , since they fail to provide timely updates published monthly by Google, to their supported devices. This has created a problem known as android OS fragmentation for Google, where majority of the mobile devices are devoid of any OS support, which potentially exposes majority of the end users to malware attacks. Android OS has built-in permission management system that keeps tab of apps using different permissions, unfortunately the intricacies of this system are too cumbersome for majority of the end users. Hence, for a malware to gain unwanted access, the end user just has to ignorantly grant the requested permissions as a matter of habit, without understanding the consequences.

1.1 Objective and Research Questions

The principal objective of this study is to build automated tools toward an effective android malware detection. Another aspect of this study is also to assess the importance of object-oriented software metrics in evaluating android application packages for malware discovery and mitigation of mobile security risks. The following research questions (RQ) have been put forward in order to identify, analyze and summarize the findings of the experiment proceedings:

- RQ1: Is there an interesting and significant distinction in the performances manifested by the three data sampling techniques?
- RQ2: Is there a major difference in performance manifested by the three feature selection techniques?
- RQ3: How do the nineteen classifiers fare in their discriminatory power as adjudged by accuracy and AUC metrics? Do these classifiers vary greatly in their malware predictive performances?

2 Related Work

The malware detection methods can broadly be grouped into two categories such as static analysis and dynamic analysis. Many authors have used static analysis such as Ma et al. [7] use API flows as features for malware detection. They obtain application program interface (API) information from a control flow graph retrieved from an apk, and they use this API information to build three API datasets, capturing different aspects of API flows. Chavan et al. [2] use android permissions requests as their features in malware detection using SVM and ANN. Garg and Baliyan [4] use API and permissions requests as static features and dynamic features such as battery temperature and network traffic. Finally, they validate the effectiveness of their model using machine learning techniques such as SVM, RIDOR, PART, MLP and their ensembles. Yen and Sun [13] use apk source code visualization technique. They compute term frequency-inverse document frequency (TF-IDF) of the decompiled source code and transform that information into images, which is then fed as input to CNN for malware analysis. Martín et al. [8] have used Markov chains and dynamic analysis for malware classification. They have deployed DroidBox tool to gather run-time information, and this information is transformed into first-order Markov model. From this model, transition probabilities and state frequencies are used as input data in deep learning algorithm, for malware classification. Saif et al. [12] use hybrid set of features retrieved from static analysis, dynamic analysis and set of system calls from both malware and benignware android apps. This hybrid set of features is then used as input for malware classification in deep belief networks. Martín et al. [9] have collected android apk malware information through the usage of 61 antivirus softwares. They have then used this information to group android apks into malware classes using graph community algorithms and hierarchical clustering. Finally, using these groups as their dataset, they have performed malware classification using logistic regression and random forest machine learning algorithms.

3 Research Methodology

3.1 Metrics Extraction and Aggregation Measures

As shown in Fig. 1, initially 5774 android apps are collected from AndroZoo [1]. These android packages are decompiled into java archive format (JAR), for object-oriented software metrics extraction using CKJM extended tool [6]. An android app encapsulates multiple classes, and each class is represented by a tuple consisting

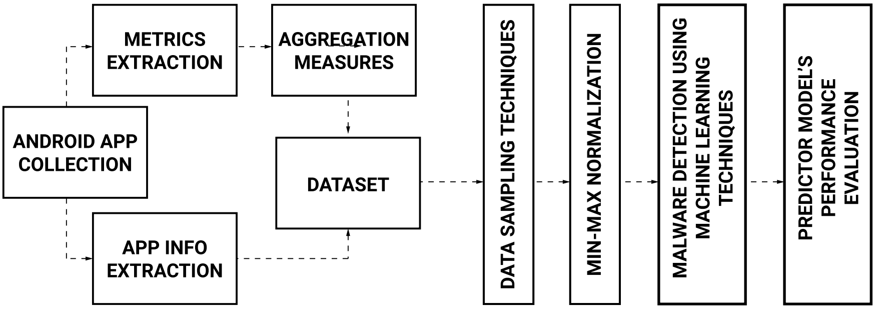


Fig. 1 Malware detection model

Table 1 List of object-oriented software metrics and aggregation measures

Efferent Coupling (Ce)	Afferent Coupling (Ca)
Response for Class (RFC)	Weighted Methods per Class (WMC)
Coupling Between Methods (CBM)	Inheritance Coupling (IC)
Number of Children in Tree (NOC)	Average Method Complexity(AMC)
Depth of Inheritance (DIT)	Data Access Metric (DAM)
LCOM3	Lack of Cohesion in Methods (LCOM)
Number of Public Methods (NPM)	Cohesion Among Methods of Class (CAM)
Measure of Functional Abstraction (MFA)	Measure of Aggregation (MOA)
Coupling Between Objects (CBO)	Lines of Code (LOC)

(a) Software Metrics

Hoover index	Variance
Standard deviation	Atkinson index
Maximum	Minimum
Gini index	Kurtosis
First quartile(Q1)	Skewness
Third quartile(Q3)	Median
Theil index	Generalized entropy
Mean	Shannon entropy

(b) Aggregation Measures

of eighteen object-oriented software metrics as described in Table 1a. Therefore, an app is represented as ($n \times 18$) intermediate matrix, where n represents the total no. of classes in an android app. Finally, sixteen aggregation methods as shown in Table 1b are applied over this intermediate matrix. Hence, for each aggregation method, an aggregated tuple of eighteen metrics is obtained. Consequently, all the sixteen aggregated tuples are conjoined serially to form a final tuple consisting of ($16 \times 18 = 288$) metric values. As a result, every android app is finally represented by a tuple with two hundred and eighty-eight features that forms a tuple of the final metrics-based dataset.

3.2 Data Sampling Techniques

Out of 5774 android samples, there are 1582 malwares. It is evident that there is a disparity in samples for benignware and malware. This class imbalance in a biased dataset affects the overall real-world performance in predicting classes. Therefore, in order to mitigate the class imbalance, three different data sampling techniques are employed such as [3] synthetic minority oversampling technique (SMOTE), [5] borderline SMOTE (BLSMOTE) and [10] support vector machine SMOTE (SVMSMOTE). SMOTE uses existing minority classes and interpolates them together to form new minority samples. BLSMOTE uses borderline minority samples in order to generate new synthetic samples, whereas SVMSMOTE employs SVM to detect minority samples which is then used to synthesize new minority samples. Performance of datasets obtained using these data sampling techniques is compared against that of the unsampled original dataset (OD).

3.3 Feature Selection Techniques

As the feature space increases, so does the incurred cost and complexity of building effective machine-learned models. It remains a challenge, to prune irrelevant features without affecting the loss of important information, that ultimately helps in achieving a trade-off between the number of selected features and overall effectiveness of dataset. In this experiment, we use three different feature selection techniques other than all metrics (AM) such as significant predictor metrics (SPM), univariate logistic regression (ULR) and principal component analysis (PCA). SPM is a set of source code metrics that are significant predictors of android malware. Initially, t-test is applied over each source code metric, and the metrics with p-values less than 0.05 are considered that has great discriminatory potential. ULR chooses the best scoring metrics based on various univariate statistical tests, and PCA reduces the feature space using singular value decomposition of the data and projects it into a lower dimensional space.

3.4 Classification Techniques

In this experiment, nineteen different classification techniques have been used to create various machine-learned models for malware prediction, such as logistic regression (LOGR), decision tree (DT), gradient descent [GD-(1H/2H/3H)] with 1/2/3 hidden layers, gradient descent with momentum—1/2/3 hidden layers [GDM-(1H/2H/3H)], variable learning rate gradient descent—1/2/3 hidden layers [GDLR-(1H/2H/3H)], scaled conjugate gradient—1/2/3 hidden layers [GDSG-(1H/2H/3H)], conjugate gradient with Powell/Beale Restarts—1/2/3 hidden layers [GDCG-(1H/2H/3H)], best training ensemble (BTE) and majority voting ensemble methods (MVE). These techniques are used over various sampled datasets applying different feature selection techniques. The predictive performance of these datasets is identified and evaluated in order to select a better yielding classifier.

3.5 Performance Evaluation Metrics

The malware prediction potential of a machine-learned model is measured using different standardized evaluation metrics such as accuracy and error rate. In this experiment, accuracy values and error rate may not reflect the true predictive potential of a machine-learned model, since these evaluation metrics fail to encompass imbalance and disparity of classes, existing in a dataset. Such imbalanced datasets tend to favor the effective prediction of majority class as compared to the minority class. Therefore, AUC evaluation metric (area under the ROC curve) is selected as this metric is immune to changes in class distribution in a given dataset. In order to prune any bias between the classes, a tenfold cross-validation technique is applied while building machine-learned models.

4 Experimental Results and Findings

Based on discussions in Sect. 3, we formulate and evaluate a null hypothesis H_0: "***Machine-learned models developed using various data sampling techniques, feature selection methods, classification algorithms and evaluated using AUC evaluation metric, for predicting android malware. It indicates no significant performance difference when compared against machine-learned models built using original dataset (OD)***".

4.1 Analyzing Data Sampling Techniques

In this experiment, three data sampling techniques are examined as discussed in Sect. 3.2. Boxplots for OD, SMOTE, BLSMOTE and SVMSMOTE-based datasets, depicting accuracy and AUC, along with its descriptive statistics are shown in Fig. 2 and Table 2a, b, respectively. It is observed from Table 2b that SMOTE and SVMSMOTE yield a higher median AUC of 0.75. Now considering these four principal metrics-based datasets, a total of ${}^4C_2 = 6$ unique pairs are possible. Analyzing the p-value of these unique pairs at 0.05 significance level, we can reject a null hypothesis if and only if the p-value is less than $0.05/6 = 0.00833$. In Table 2c, the p-values less than 0.00833 are denoted by the symbol "•". It can be inferred from Table 2c that datasets based on SMOTE and SVMSMOTE are similar between themselves and significantly different from datasets based on OD and BLSMOTE. Table 2b shows that SMOTE and SVMSMOTE-based datasets yield better AUC median values as compared to OD and BLSMOTE-based datasets. Therefore, SMOTE and SVMSMOTE-based datasets are expected to outperform the rest of the datasets.

4.2 Analyzing Feature Selection Techniques

Boxplots for datasets using AM, SPM, ULR and PCA feature selection techniques, depicting accuracy and AUC, along with its descriptive statistics are shown in Fig. 3 and Table 3a, b, respectively. It is observed from Table 3b that AM, SPM and ULR-

Table 2 Boxplot descriptive statistics and p-value for data sampling techniques

	MIN	MAX	MEAN	MEDIAN	Q1	Q3
OD	26.95	79.81	74.17	75.39	73.27	76.6
SMOTE	49.05	81.56	68.91	68.96	66.05	75.8
BLSMOTE	49.82	83.59	67.68	67.48	62.31	74.13
SVMSMOTE	49.76	82.1	68.21	67.56	65.1	74.58

(a) Accuracy

	MIN	MAX	MEAN	MEDIAN	Q1	Q3
OD	0.46	0.8	0.69	0.71	0.65	0.75
SMOTE	0.53	0.86	0.74	**0.75**	0.71	0.81
BLSMOTE	0.52	0.85	0.73	0.74	0.66	0.81
SVMSMOTE	0.57	0.86	0.74	**0.75**	0.69	0.81

(b) AUC

	OD	SMOTE	BLSMOTE	SVMSMOTE
OD		•		•
SMOTE				
BLSMOTE				
SVMSMOTE				

(c) p-values

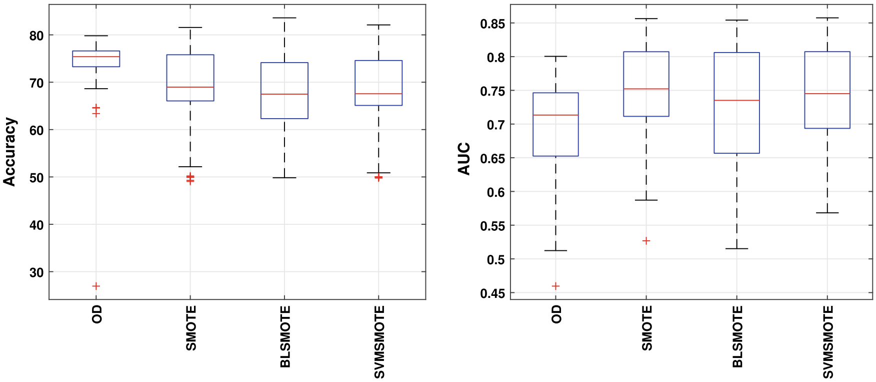


Fig. 2 Boxplots for data sampling techniques

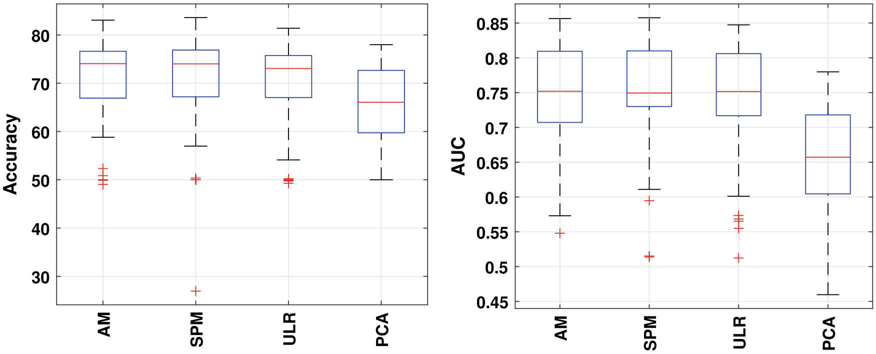


Fig. 3 Boxplots for feature selection techniques

based models yield a higher median AUC of 0.75. Now considering these four feature selection techniques, a total of ${}^4C_2 = 6$ unique pairs are possible. Analyzing the p-value of these unique pairs at 0.05 significance level, we can reject a null hypothesis if and only if the p-value is less than $0.05/6 = 0.00833$. In Table 3c, the p-values less than 0.00833 are denoted by the symbol "•". It can be inferred from Table 3c that datasets using AM, SPM and ULR are similar among themselves and are significantly different from datasets using PCA. Table 3b shows that datasets using AM, SPM and ULR yield better AUC median values as compared to datasets using PCA. Therefore, datasets applying AM, SPM and ULR are expected to outperform the datasets using PCA.

Table 3 Boxplot descriptive statistics and p-value for feature selection techniques

	MIN	MAX	MEAN	MEDIAN	Q1	Q3
AM	49.05	83.05	71.08	74.05	66.92	76.6
SPM	26.95	83.59	71.45	74.01	67.18	76.87
ULR	49.28	81.38	70.72	73.06	67.03	75.74
PCA	50	77.98	65.71	66.05	59.73	72.66

(a) Accuracy

	MIN	MAX	MEAN	MEDIAN	Q1	Q3
AM	0.55	0.86	0.75	**0.75**	0.71	0.81
SPM	0.51	0.86	0.75	**0.75**	0.73	0.81
ULR	0.51	0.85	0.74	**0.75**	0.72	0.81
PCA	0.46	0.78	0.66	0.66	0.6	0.72

(b) AUC

	AM	SPM	ULR	PCA
AM				●
SPM				●
ULR				●
PCA				

(c) p-values

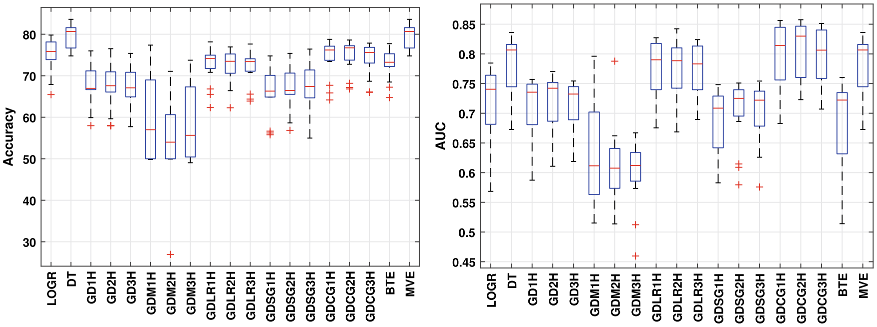


Fig. 4 Boxplots for classifiers

4.3 Analyzing Machine Learning Algorithms

Boxplots for datasets using various machine learning algorithms as described in Sect. 3.4, depicting accuracy and AUC, along with its descriptive statistics are shown in Fig. 4 and Table 4a, b, respectively. It is observed from Table 4b that GDCG2H yields a better median accuracy and AUC of 76.73% and 0.83, respectively. Now considering these nineteen machine learning techniques, a total of ${}^{19}C_2 = 171$ unique pairs are possible. Analyzing the p-value of these unique pairs at 0.05 significance level, we can reject a null hypothesis if and only if the p-value is less than $0.05/171 = 0.00029$. In Table 4c, the p-values less than 0.00029 are denoted by the symbol "•". It can be inferred from Table 4c that machine-learned models based on GDCG2H are significantly different from other machine-learned models. Table 4b shows that machine-learned models using GDCG2H yield better AUC median values as compared to other machine learning techniques. Therefore, datasets applied over GDCG2H are expected to outperform other classification techniques.

Table 4 Boxplot descriptive statistics and *p*-value for classifiers

	MIN	MAX	MEAN	MEDIAN	Q1	Q3
LOGR	65.45	79.81	74.95	75.83	73.89	78.15
DT	74.78	83.59	79.49	80.66	76.69	81.59
GD1H	57.97	76	67.8	66.96	66.66	71.18
GD2H	57.94	76.52	67.6	67.6	66.08	70.95
GD3H	57.73	75.39	67.68	67.09	64.92	70.83
GDM1H	49.82	77.38	59.97	57	50	68.99
GDM2H	26.95	71.06	54.81	54	50	60.63
GDM3H	49.05	73.74	58.73	55.64	50.43	67.32
GDLR1H	62.33	78.16	72.7	74.13	71.75	74.96
GDLR2H	62.29	76.95	72.47	73.48	70.6	75.25
GDLR3H	63.9	77.64	72.1	73.41	71.09	74.1
GDSG1H	55.79	74.78	66.15	66.3	64.87	70.09
GDSG2H	56.84	75.39	67.27	66.44	65.51	70.64
GDSG3H	54.99	76.43	67.54	67.41	64.68	71.4
GDCG1H	64.18	78.77	74.39	76.21	73.71	77.12
GDCG2H	66.83	78.61	74.78	76.73	73.76	77.24
GDCG3H	65.99	77.86	74.15	75.61	73.06	76.84
BTE	64.74	77.73	73.01	73.25	72.23	75.3
MVE	74.78	83.59	79.49	80.66	76.69	81.59

(a) Accuracy

	MIN	MAX	MEAN	MEDIAN	Q1	Q3
LOGR	0.57	0.78	0.72	0.74	0.68	0.76
DT	0.67	0.84	0.78	0.81	0.74	0.82
GD1H	0.59	0.76	0.71	0.74	0.68	0.75
GD2H	0.61	0.77	0.72	0.74	0.69	0.75
GD3H	0.62	0.75	0.71	0.73	0.69	0.74
GDM1H	0.52	0.8	0.63	0.61	0.56	0.7
GDM2H	0.51	0.79	0.61	0.61	0.57	0.64
GDM3H	0.46	0.67	0.6	0.61	0.59	0.63
GDLR1H	0.68	0.83	0.78	0.79	0.74	0.82
GDLR2H	0.67	0.84	0.78	0.79	0.74	0.81
GDLR3H	0.69	0.82	0.77	0.78	0.74	0.81
GDSG1H	0.58	0.75	0.69	0.71	0.64	0.73
GDSG2H	0.58	0.75	0.7	0.73	0.7	0.74
GDSG3H	0.58	0.75	0.7	0.72	0.68	0.74
GDCG1H	0.68	0.86	0.8	0.81	0.76	0.84
GDCG2H	0.72	0.86	0.8	0.83	0.76	0.85
GDCG3H	0.71	0.85	0.8	0.81	0.76	0.84
BTE	0.51	0.76	0.68	0.72	0.63	0.73
MVE	0.67	0.84	0.78	0.81	0.74	0.82

(b) AUC

	LOGR	DT	GD1H	GD2H	GD3H	GDM1H	GDM2H	GDM3H	GDLR1H	GDLR2H	GDLR3H	GDSG1H	GDSG2H	GDSG3H	GDCG1H	GDCG2H	GDCG3H	BTE	MVE
LOGR		●					●	●	●	●	●				●	●	●		●
DT			●	●	●	●	●	●				●	●	●				●	
GD1H							●	●	●	●	●				●	●	●		●
GD2H							●	●	●	●	●	●			●	●	●		●
GD3H							●	●	●	●	●				●	●	●		●
GDM1H									●	●	●				●	●	●		●
GDM2H									●	●	●	●	●	●	●	●	●		●
GDM3H									●	●	●	●	●	●	●	●	●	●	●
GDLR1H												●	●	●				●	
GDLR2H												●	●	●				●	
GDLR3H												●	●	●				●	
GDSG1H															●	●	●		●
GDSG2H															●	●	●		●
GDSG3H															●	●	●		●
GDCG1H																		●	
GDCG2H																		●	
GDCG3H																		●	
BTE																			●
MVE																			

(c) p-values

4.4 Analyzing Machine-Learned Models

Upon examining Sects. 4.1, 4.2 and 4.3, a total of three hundred and four different machine-learned model's discriminating power are compared using its accuracy and AUC values. Primarily, a better classifier is characterized by an AUC value closer to 1.0. Consequently, it is expected that a dataset based on either SMOTE or SVMSMOTE applying any of the feature selection techniques like AM, SPM or ULR over GDCG2H classification technique will yield a better machine-learned model. This expectation is observed and confirmed from Table 5a, b, where SVMSMOTE-based dataset applying SPM feature selection technique over GDCG2H machine

Table 5 Classifier accuracy and AUC against different datasets applying various feature selection techniques

		LOGR	DT	GD1H	GD2H	GD3H	GDM1H	GDM2H	GDM3H	GDLR1H	GDLR2H	GDLR3H	GDSG1H	GDSG2H	GDSG3H	GDCG1H	GDCG2H	GDCG3H	BTE	MVE
OD	AM	79.46	76.6	74.78	76.52	75.15	64.56	63.38	72.36	78.16	75.11	75.8	72.1	75.39	75.56	77.21	75.72	77.73	77.64	76.6
	SPM	79.81	76.6	76	75.13	75.04	77.38	26.95	73.74	76	76.95	75.82	74.09	73.31	76.43	76.95	76.86	76.6	77.21	76.6
	ULR	79.12	76.6	73.92	74.7	75.39	64.64	68.63	71.69	77.3	75.91	77.64	74.78	75.39	75.48	78.77	78.34	76.17	77.73	76.6
	PCA	74.26	74.78	72.96	72.7	72.44	72.44	69.56	72.62	73.4	75.39	73.31	72.42	72.51	72.7	73.98	72.77	74.61	73.22	74.78
SMOTE	AM	76.79	81.56	66.77	67.88	68.24	73.15	49.91	49.05	75	74.28	74.15	67.1	68.78	67.4	78.16	76.85	77.86	76.01	81.56
	SPM	76.25	80.73	69.39	67.24	66.23	67.84	50	62.95	74.7	76.88	73.03	68.08	66.61	67.6	76.61	76.91	77.48	73.27	80.73
	ULR	75.34	80.72	69.15	69.19	69.21	70.15	50.21	49.28	72.67	75.1	72.43	64.84	65.81	67	75.58	77.49	75.06	73.21	80.72
	PCA	67.9	77.15	59.9	57.94	61.58	52.15	53.88	50	66.83	66.41	64.38	56.21	58.65	60.68	65.87	66.83	68.68	68.5	77.15
BLSMOTE	AM	76.55	83.05	66.75	68.26	63.6	58.81	50	52.33	74.22	71.18	74.05	64.9	66.35	70.09	77.03	74.76	72.67	73.69	83.05
	SPM	75.4	83.59	66.57	66.17	67.12	50	50.36	56.98	74.34	72.9	73.93	67.84	65.75	67.94	77.31	77.33	76.79	73.87	83.59
	ULR	73.51	80.61	69.15	67.78	66.89	49.82	54.93	54.3	70.82	73.33	73.87	65.87	65.27	66.75	75.89	74.94	73.45	72.08	80.61
	PCA	65.45	76.78	57.97	58	62.15	53.37	56.12	59.82	62.33	62.29	63.9	55.79	56.84	54.99	64.18	67.18	65.99	64.74	76.78
SVMSMOTE	AM	78.46	81.61	66.77	67.42	67.06	49.97	50	50.87	74.05	73.63	71.42	65.51	65.99	64.62	73.45	77.15	76.31	74.58	81.61
	SPM	77.84	82.1	66.85	67	66.65	50	71.06	61.28	74.91	73.27	73.51	66.73	66.53	67.42	76.52	**78.61**	76.88	72.85	82.1
	ULR	74.58	81.38	67.06	65.99	68.44	50	54.12	49.76	72.91	70.01	70.76	65.45	68.62	64.74	75.06	76.61	73.93	72.37	81.38
	PCA	68.44	77.98	60.84	59.64	57.73	55.19	57.88	52.74	65.51	66.81	65.57	56.66	64.5	61.22	67.7	68.18	66.11	67.24	77.98

(a) Accuracy

		LOGR	DT	GD1H	GD2H	GD3H	GDM1H	GDM2H	GDM3H	GDLR1H	GDLR2H	GDLR3H	GDSG1H	GDSG2H	GDSG3H	GDCG1H	GDCG2H	GDCG3H	BTE	MVE
OD	AM	0.68	0.72	0.72	0.73	0.72	0.57	0.55	0.67	0.77	0.75	0.76	0.65	0.7	0.7	0.78	0.78	0.79	0.61	0.72
	SPM	0.68	0.71	0.73	0.75	0.74	0.74	0.51	0.65	0.76	0.78	0.76	0.71	0.74	0.73	0.8	0.78	0.79	0.59	0.71
	ULR	0.69	0.71	0.74	0.73	0.74	0.57	0.66	0.51	0.77	0.77	0.77	0.71	0.74	0.72	0.8	0.8	0.77	0.62	0.71
	PCA	0.57	0.67	0.59	0.64	0.65	0.52	0.58	0.46	0.7	0.73	0.71	0.63	0.61	0.65	0.72	0.73	0.71	0.51	0.67
SMOTE	AM	0.77	0.82	0.74	0.74	0.75	0.8	0.66	0.63	0.82	0.81	0.82	0.73	0.75	0.72	0.86	0.85	0.85	0.76	0.82
	SPM	0.76	0.81	0.75	0.77	0.75	0.76	0.61	0.66	0.82	0.84	0.79	0.73	0.73	0.74	0.85	0.85	0.84	0.73	0.81
	ULR	0.75	0.81	0.76	0.76	0.75	0.77	0.63	0.64	0.82	0.83	0.79	0.71	0.73	0.74	0.84	0.85	0.84	0.73	0.81
	PCA	0.68	0.77	0.63	0.61	0.64	0.53	0.59	0.59	0.72	0.71	0.69	0.6	0.61	0.63	0.73	0.72	0.74	0.68	0.77
BLSMOTE	AM	0.77	0.83	0.73	0.75	0.72	0.64	0.65	0.6	0.83	0.79	0.82	0.7	0.74	0.75	0.85	0.83	0.81	0.74	0.83
	SPM	0.75	0.84	0.73	0.74	0.73	0.52	0.63	0.61	0.81	0.81	0.82	0.75	0.72	0.74	0.85	0.85	0.85	0.74	0.84
	ULR	0.74	0.81	0.75	0.75	0.73	0.62	0.55	0.62	0.77	0.81	0.81	0.72	0.71	0.73	0.83	0.83	0.82	0.72	0.81
	PCA	0.65	0.77	0.6	0.62	0.66	0.56	0.58	0.61	0.68	0.67	0.69	0.58	0.58	0.58	0.68	0.72	0.73	0.65	0.77
SVMSMOTE	AM	0.78	0.82	0.74	0.75	0.73	0.66	0.63	0.61	0.81	0.81	0.79	0.7	0.71	0.7	0.8	0.84	0.84	0.75	0.82
	SPM	0.78	0.82	0.75	0.75	0.74	0.62	0.79	0.62	0.82	0.81	0.81	0.74	0.74	0.74	0.84	**0.86**	0.84	0.73	0.82
	ULR	0.75	0.81	0.75	0.75	0.75	0.6	0.57	0.57	0.81	0.78	0.78	0.7	0.75	0.72	0.84	0.84	0.81	0.72	0.81
	PCA	0.68	0.78	0.64	0.61	0.62	0.57	0.6	0.58	0.72	0.73	0.72	0.6	0.69	0.66	0.74	0.74	0.73	0.67	0.78

(b) AUC

learning algorithm yields a better accuracy and AUC value of 76.73% and 0.83, respectively. The corresponding ROC curve is illustrated in Fig. 5, marked as 2HL.

5 Comparison of Results

RQ1: Is there an interesting and significant distinction in the performances manifested by the three data sampling techniques? Considering the null hypothesis H_0 and analyzing Sect. 4.1, it is evident that out of six unique pairs, only two pairs reject the null hypothesis and is marked by the symbol "•" in Table 3c. In case, a null hypothesis is rejected, it implies that the distinction identified between samples is not by chance, and the observation is statistically significant. The machine-learned models based on SMOTE and SVMSMOTE yield better AUC as compared to OD and BLSMOTE. Therefore, SMOTE and SVMSMOTE-based models are interesting

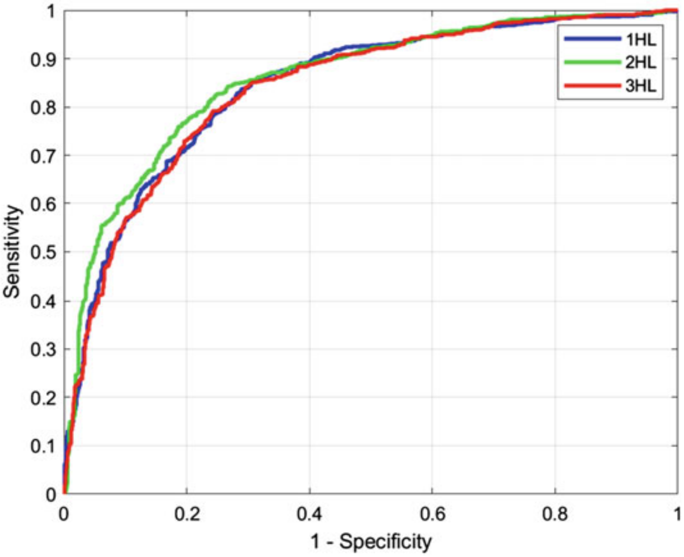


Fig. 5 SVMSMOTE-SPM-GDCG2H classification model's ROC curve

and manifest significant increase in malware prediction performance as compared against OD and BLSMOTE.

RQ2: Is there a major difference in performance manifested by the three feature selection techniques? Considering the null hypothesis H_0 and analyzing Sect. 4.2, it is evident that out of six unique pairs, only three pairs reject the null hypothesis and are marked by the symbol "•" in Table 4c. In case a null hypothesis is rejected, it implies that the distinction identified between samples is not by chance, and the observation is statistically significant. The machine-learned models based on AM, SPM and ULR are similar and yield better AUC as compared to PCA. Therefore, AM, SPM and ULR-based models are interesting and manifest significant increase in malware prediction performance as compared against PCA.

RQ3: How do the nineteen classifiers fare in their discriminatory power as adjudged by accuracy and AUC metrics? Do these classifiers vary greatly in their malware predictive performances? Considering the null hypothesis H_0 and analyzing Sect. 4.3, it is evident that out of 171 unique pairs, only one hundred and four pairs reject the null hypothesis and are marked by the symbol "•" in Table 5c. In case a null hypothesis is rejected, it implies that the distinction identified between samples is not by chance, and the observation is statistically significant. Therefore, the machine learning techniques are significantly different among themselves, and GDCG2H algorithm yields a better AUC of 0.86. Hence, machine-learned models based on GDCD2H are interesting and manifest significant increase in malware prediction performance.

6 Threats to Validity

In this work, one possible threat to internal validity identified is that the machine-learned models built with android apps from a certain point in time may be effective for malwares released within a fixed time frame (e.g., six months or so). And the model may loose its effectiveness against new strain of malwares released beyond this time frame. The information regarding this new strain of malwares must trickle down from antivirus companies through Google's VirusTotal service and finally to AndroZoo repository. Until, this new information is not available with AndroZoo, and it is difficult to build yet another effective machine-learned malware predictor model.

7 Conclusion

Initially, android samples are collected over AndroZoo, which is then decompiled, and software metrics are extracted using CKJM extended tool. For every android app, sixteen different aggregation measures are applied over the extracted eighteen software metrics, which becomes the metrics-based dataset to be used for malware prediction. In order to mitigate benignware and malware sample imbalance in the dataset, three data sampling techniques are used such as SMOTE, BLSMOTE and SVMSMOTE. In order to reduce the feature space, three feature selection methods are employed such as SPM, ULR and PCA. Finally, nineteen different classification algorithms are used to build various machine-learned models. A total of three hundred and four machine-learned models are evaluated using their AUC values. Consequently, machine-learned model built using SVMSMOTE data sampling applying SPM feature selection methods over GDCG2H classification algorithm yields a better AUC of 0.86, which exceeds the malware prediction potential against other models.

References

1. Allix K, Bissyandé TF, Klein J, LeTraon Y (2016) Androzoo: collecting millions of android apps for the research community. In: 2016 IEEE/ACM 13th working conference on mining software repositories (MSR). IEEE, pp 468–471
2. Chavan N, DiTroia F, Stamp M (2019) A comparative analysis of android malware. arXiv preprint arXiv:1904.00735
3. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
4. Garg S, Baliyan N (2019) A novel parallel classifier scheme for vulnerability detection in android. Comput Electr Eng 77:12–26
5. Han H, Wang WY, Mao BH (2005) Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: International conference on intelligent computing. Springer, Berlin, pp 878–887

6. Jureczko M, Spinellis D (2010) Using object-oriented design metrics to predict software defects. In: Monographs of system dependability. Models and methodology of system dependability. Oficyna Wydawnicza Politechniki Wroclawskiej, Wroclaw, Poland, pp 69–81
7. Ma Z, Ge H, Liu Y, Zhao M, Ma J (2019) A combination method for android malware detection based on control flow graphs and machine learning algorithms. IEEE Access 7:21235–21245
8. Martín A, Rodríguez-Fernández V, Camacho D (2018) Candyman: classifying android malware families by modelling dynamic traces with Markov chains. Eng Appl Artif Intell 74:121–133
9. Martín I, Hernández JA, delos Santos S (2019) Machine-learning based analysis and classification of android malware signatures. Future Gener Comput Syst 97:295–305
10. Nguyen HM, Cooper EW, Kamei K (2011) Borderline over-sampling for imbalanced data classification. Int J Knowl Eng Soft Data Paradigms 3(1):4–21
11. Nohl K, Lell K (2018) Mind the gap: uncovering the android patch gap through binary-only patch level analysis. In: HITB security conference
12. Saif D, El-Gokhy S, Sallam E (2018) Deep belief networks-based framework for malware detection in android systems. Alexandria Eng J 57(4):4049–4057
13. Yen YS, Sun HM (2019) An android mutation malware detection based on deep learning using visualization of importance from codes. Microelectron Reliab 93:109–114

Vascular Clog Loss Classification: An Advanced Alzheimer's Research Using ConvNets

Mansimran Singh Anand, Chirag Kedia, Ambarish Moharil, and Nikhil Sonavane

Abstract As of today, no cure has been found that can fully treat Alzheimer's disease. One in three people above the age of 60 die from Alzheimer's disease or either dementia. Alzheimer's disease (AD) kills more people than Breast Cancer and Prostate Cancer combined. It is predicted that, by the year 2050, the total number of people being affected by Alzheimer's will be around 14 million worldwide. Diagnosis of Alzheimer's disease at early stage would help in facilitating family planning and cost control and further reduce additional cost that might be involved in long-term care. The purpose of this study is to classify restricted and normal flowing cranial vessels in the brain using convolution neural networks (AlexNet, ResNet 50, 101, 152). Convolution neural networks have proven to have a robust architecture that can be used for extracting high-level features and further assist in image recognition and analysis. Using the AlexNet technique, we have been able to successfully classify a video (of blood flowing in a blood vessel) as stalled or flowing with an astonishing accuracy of 98.59%.

Keywords Alzheimer's disease · Convolution neural network · Deep learning · AlexNet · ResNet

1 Introduction

Alzheimer's disease is a neurodegenerative, irreversible, chronic brain disease that starts slowly and progressively worsens over time, destroying the brain cells leading to a loss of cognitive and thinking skills, which then ultimately lead to a state where

M. S. Anand · C. Kedia (✉)
Department of Computer Science, Vellore Institute of Technology, Vellore, India

A. Moharil
Department of Instrumentation and Control, Vishwakarma Institute of Technology, Pune, India

N. Sonavane
Department of Electronics and Telecommunication, Vishwakarma Institute of Technology, Pune, India
e-mail: nikhil.sonanvane16@vit.edu

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_25

the patient can't even perform simple diurnal tasks [1]. The decline in understanding, comprehension, effective communication, thinking attributes, behavioral, personality changes is caused by this disorder [2]. As of today, there has been no cure to this disease. Alzheimer's disease (AD) is the most (60–80%) common form of dementia. In 2019, the studies show that there are a total of 5.8 million people diagnosed with AD. It is the sixth leading cause of death in the United States, and it is predicted that by 2050, there would be approximately 14 million cases of dementia worldwide. It has also been found that about 1 in 10 people with the age more than 65 have been diagnosed with AD with females more likely to be a victim of this disease as compared to males.

Dementia is not identical to the typical deterioration of cognitive abilities that comes with aging to elderly [1]. The Normal "forgetful" elders can be abetted to remember, but in case of AD or dementia, a person is often unaware of time or place associated with an activity, and in severe cases, they may forget to eat also or maintain a good hygiene. This fact is rather prescient because of the way the AD damages the brain cells. AD is associated with cerebral cortex, which is the outer covering of the cerebrum of the brain. The irreversible deterioration of the brain cells leads from the formation of amyloid plague and neurofibrillary tangles. The Amyloid plagues binds om the nerve synapses ultimately destroying them which results in the memory loss which is the early symptom of AD. The Neurofibrillary tangles further damage neurons and nerve synapses.

Early diagnosis of AD would facilitate an early treatment intervention and family planning, as the studies have found out that if early diagnosis and treatment of AD would ultimately lead to saving $7.9 trillion dollars annually. But the main issue here is that diagnosis of AD is a very intense, vigorous, time-consuming, arduous task and significantly a very costly process involving a lot of mental and physical exams, followed by some laboratory and neurology tests, along with some neurological imaging (using MRI or CT scans). The cost of a brain MRI could very easily range from $500 to $1500. This cost is very unfeasible for families that are uninsured or underinsured.

Those days are old when the data related to health care are small [3]. Due to large growth and advancement of image acquisition devices available at the disposal of health care experts, the data size available is quite large, which further makes it challenging for image analysis [4]. The steady growth in the medical image and the various modalities associated with it requires extensive efforts from medical experts to analyze the images or videos, which could also sometime lead to human error. An alternative approach to analyze the data available is by using machine learning and deep learning techniques that could be used to automate and analyze the available data [5]. The data that are available are generally in the form of images and videos. The data in these forms are omnipresent and widespread that too in a humungous amount. Hence, the conventional algorithms cannot be used to analyze this information as they would take a great amount of time to process the data from head to tail and with no guarantee that they would conclude to a result with high accuracy.

Machine Learning and Artificial Intelligence techniques have gained importance swiftly in recent years. In various algorithms and models, these technologies have played a significant role in medical image processing, image interpretation, image registration, image fusion, computer-aided diagnosis and image-guided therapy. Image retrieval and analysis techniques of machine learning aid us in extracting information from the images and then represent information effectively and efficiently, which could further assist and facilitate doctors in diagnosing the disease quicker and with higher accuracy and present a solution to overcome them. These techniques also help researchers, practitioners, doctors and medical experts to understand and detect various generic variations, which could ultimately lead to disease [6]. Some of the techniques being referred here are Neural Networks (NN), KNN, and Support Vector Machines (SVM). Some of the deep learning methods can also be used such as Convolution Neural Networks (CNN), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Generative Adversarial Networks (GANs) and Extreme learning etc. [7]. Hierarchical or structured deep learning approach is the modern branch of machine learning, which has been developed based on intricate and complicated algorithms, which model high-level features. These methods help extract the abstractions from the data by using architecture similar to neural networks but with a little more complication and convolution.

Convolution neural networks are considered to be the futuristic model when it comes to tasks of image recognition and retrieval of information from the image [8]. The use of CNN has been employed in myriad application like facial recognition, classification of objects and generating scene description [4]. This success of using CNN for image analyzes can be attributed to the use of techniques that have scaled and improved the network from a mere hundred parameters to even up to tens of millions of parameters along with the evolution, advance and progress for deep architectures and availability of largely annotated datasets such as ImageNet, which can support the learning process. ImageNet is a huge group of hierarchical; annotated and labeled data. Under these conditions, it has been found that CNN has the ability to learn powerful features that can be used to interpret images. Encouraged by the positive results that have been obtained it is safe to say that the use of CNN for large scale video classification can be employed, where the model built using CNN can not only be used to access the appearance information from a single static image but also at the same time understand intricate temporal evolution.

The detection and classification of multiple classes from an image can be done using different architectures available with CNN. For example ResNet [2], CaffeNet [5], GoogLeNet [8, 9], VGGNet [10], AlexNet [6, 8, 11] used in conjugation with GoogLeNet and NIN (Networks in Network). ResNet and AlexNet have often been used for the purpose of image classification. It has been found that these two approaches attain a comparatively low error and at the same time be trained on millions of images producing a desired output quicker compared to other machine learning algorithms.

In this work, the convolutional neural network (CNN), which is one of the Deep Learning Network design, is used to characterize the blood flow of the cranial vessels in an Alzheimer's brain and creating a trained and prescient model. This work has been done with the primary objective of building learning models specifically using ResNet and AlexNet to effectively and efficiently detect blockage of blood flowing in the brain of a person and hence classifying that with having AD with a high accuracy of about 98.5% using models which are still under extensive research.

The inspiration of this work is to give the survey of profound deep learning-based methods in clinical and medical image examinations. Referring to the past work done in this field and the future directions stated in them, this paper gives the fundamental information and a cutting-edge approach about deep learning and specifically using CNN architectures in the field of medical imaging for the analysis of blood flow through the cranial vessels.

2 Approach

The objective of this paper is to propose a methodology to predict whether the marked blood vessel in the video is flowing or stalled. In order to achieve this objective, this paper discusses an astute methodology using transfer learning. The proposed approach is unique and anything but banal [1]. Videos are nothing, but a collection of frames and frames are nothing but images. Taking this fundamental concept into account, a robust architecture was constructed. The dataset consisted of 4600 videos of cranial vessels classified as flowing or stalled. After extracting the frames and manual filtering of the images, a dataset of 78,675 images in total was created. If we observe the dataset in a scrupulous manner, we find out that the difference between the images as a whole is quite inconsequential. The difference between both the classes viz. flowing and stalled can be determined only by the region outlined in these images. So, unless and until we consider just the outlined region, the difference in both the classes seems irrelevant. Hence, extraction of the outlined region in a particular image becomes extremely important. As the scope of this paper is to classify the blood flow through the marked region in an image, punctilious analysis of just the outlined region makes perfect sense. The extraction methodology of the red outlined region is discussed in Sect. 5 (Fig. 1).

Furthermore, once the frames were extracted from the videos, these images were cropped to train the classifier [3]. Regions of blood flow and stalled vessels were randomly cropped from the extracted frames. This exercise was performed to train the classifier in a more efficient way by omitting the unnecessary parts and extracting the necessary data from the images. This paper makes use of transfer learning and discusses results of classification by four distinct architectures viz. [3, 7] Resnet50, Resnet101, Resnet152 and AlexNet. Each of these architectures has been discussed in detail in Sect. 4. The approach can be divided as follows.

2.1 Extraction of Frames

The original dataset obtained from data driven consisted of 4600 videos of stalled and flowing cranial vessels. To analyze these videos and to classify them into these

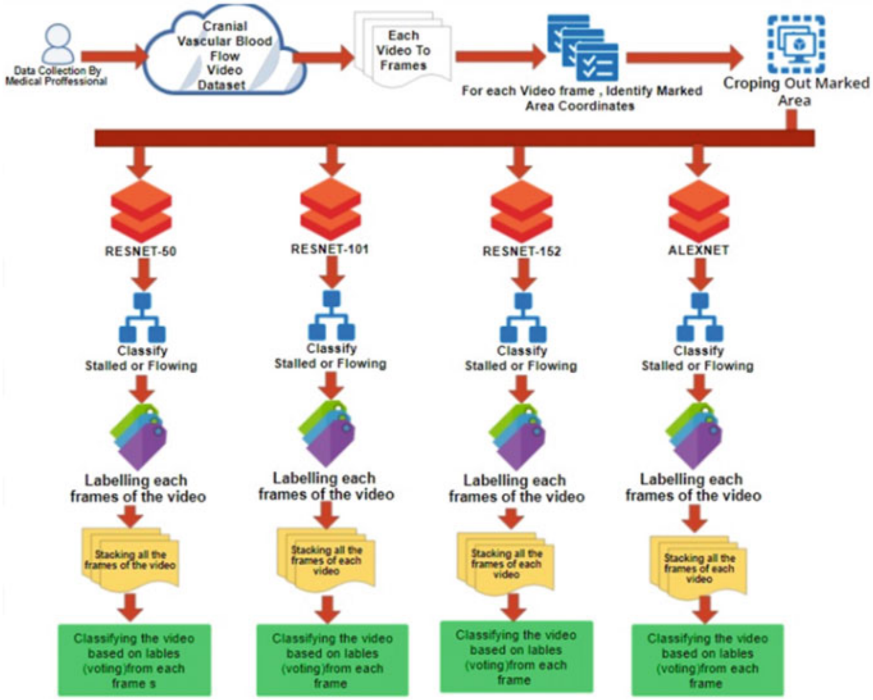

Data Collection By Medical Proffessional
Cranial Vascular Blood Flow Video Dataset
Each Video To Frames
For each Video frame , Identify Marked Area Coordinates
Croping Out Marked Area
RESNET-50
RESNET-101
RESNET-152
ALEXNET
Classify Stalled or Flowing
Labelling each frames of the video
Stacking all the frames of the video
Stacking all the frames of each video
Classifying the video based on lables (voting)from each frame s
Classifying the video based on lables (voting)from each frame

Fig. 1 Overview of approach

binary classes, these videos were treated as images. Frames at the rate of 20 frames per video were extracted and stored in a directory [7]. This process was performed using the Open Cv library available in python. Each frame encapsulates flowing and stalled cranial vessels. The pith of the intuition behind this process was, whenever the blood flows through a vessel, it flows in discrete packets or quanta with a minute respite between the current flowing quanta and the incoming quanta [1]. If we extract significant number of frames from a video, the number of frames consisting of a luminescent quantum in case of a flowing cranial vessel would be more than the bereft black spots present in the vascular region, which indicate a stalled flow (Fig. 2).

2.2 Detecting Marked Region in the Frames

As the objective of the paper is to classify, predict and determine the clog loss in a cranial vessel, it makes total sense to sequester the marked region from the entire image [1, 7]. A single extracted frame from a video consists of multiple cranial vessels. Some of which are stalled, and some are flowing. As we intend to detect the clog loss in the outlined region, feeding or even training the image to the classifier

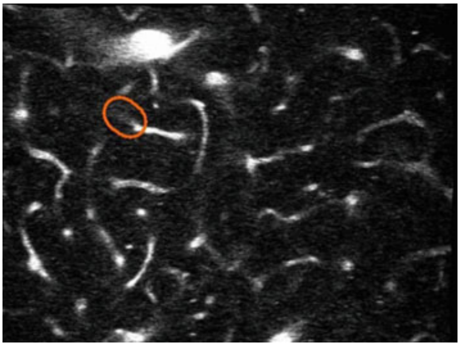


Fig. 2 Extracted Frame from the video

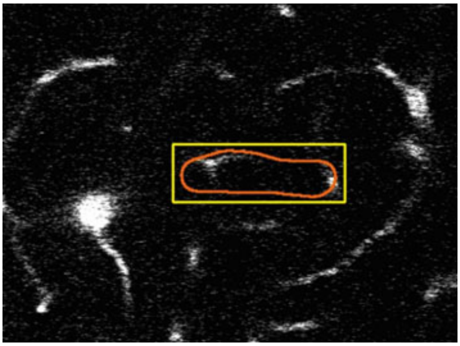


Fig. 3 Detecting the marked region

as a whole would be extremely erroneous. Hence, it is important for the algorithm to detect the outlined region from the frames and feed it to the classifier [3]. A dedicated approach to identify the marked red regions in the video frames has been explained in Sect. 5 using techniques like, multiscale template matching, haar cascading and pixel intensity detection. Using these approaches, the outlined region from the input frame is extracted and cropped around the detected bounding box. This cropped image is then fed to the classifier for classification and prediction of clog loss (Fig. 3).

2.3 Classification and Prediction from the Outlined

The classification of the clog loss is done on the images cropped around the marked red region. As explained previously, an image as a whole contains a lot of data and training a classifier on the entire image would result in false predictions. As the images contain numerous flowing and stalled cranial vessels, it becomes arduous for the classifier to focus and extract dedicated information from the outlined region. So, instead of creating directories of images extracted from videos of stalled and flowing cranial vessels, it becomes momentous to train the classifier on images of

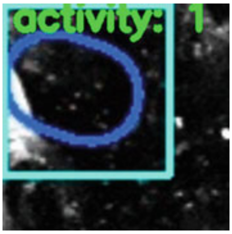


Fig. 4 Classified output

flowing and stalled vessels in particular. To achieve this task, manual extraction of flowing and stalled cranial vessels was performed. The flowing cranial vessels consist of a luminescent quantum and the stalled vessels are just isolated black spots. Cropping out these two distinct regions and training the classifier on them gives conspicuous results. In order to classify the input images into respective classes, four architectures were evaluated viz. Resnet50, Resnet101, Resnet152 and AlexNet. These architectures and their use in the approach of this paper have been scrupulously explained in Sect. 4. This paper studies the impact of transfer learning in solving the intricate problem of blood flow classification from videos. The results have been tabulated in Sect. 7 for these architectures. The logic behind classification is that, when a flowing cranial vessel is being analyzed, the maximum number of frames of the marked region extracted from the video contain a flowing bright quantum and some of them contain a black spot. A luminescent quantum in the marked region indicates the presence of blood in the respective vessel and a black spot portrays a stalled vessel. So, in the frames extracted from a flowing vessel, maximum frames consist of the bright quantum and some consist of a black spot (captured during the interval between the passing of current quanta of blood and the incoming quanta). The frequency of the frames consisting a luminescent flow would be more in flowing vessels. We pass the extracted group of frames to the classifier and classify the video as flowing or stalled based on voting of these frames by the classifier. If maximum frequency of prediction of frames is 1 then the outlined region is classified as a stalled vessel and if it is 0 then it is classified as a flowing vessel. The result of each frame is then written on the respective frame. These frames are then combined to form a video file. The changing output printed on each frame accurately depicts when the blood is flowing through the marked vascular region and when it is getting stalled (Fig. 4).

3 Architecture

Image Classification consists of two major tasks viz. feature extraction and feature classification [12]. Feature extraction has been a traditional method in computer vision. The motive behind feature extraction is to extract high-level information from raw pixels. This extracted information can be further tweaked, and distinction can be made amongst the involved categories. GIST, HOG, LBP etc. are the most widely used features from an image. The limitation that feature extraction faces is that the extracted information cannot be modified according to the classes present in the problem statement. It is considered as plodding method as it lacks the versatility to tweak and manipulate the available information. The predictions or the classification results suffer a lot due to this limitation and a proven solution to such pre-defined hardcoded methods is using feature classifier or ConvNets. Convolutional Neural Networks take inspiration from the human brain and mimic the entire operation a human brain performs while distinguishing and processing visual data [12]. The perceived data (input image) are manipulated to a great extent and a varied amount of data is extracted by a classical convolutional neural network. Putting forward a terse explanation, convolutional neural networks consist of four major stages viz. Convolution, Max Pooling, Flattening and Fully Connected Layer. The input image matrix is convolved with pre-defined image kernels in the convolutional layer producing several feature maps. Each kernel operation (convolution) with the input image produces a feature map. An operation called pooling is then performed on these feature maps [12]. The significance of pooling is to extract the important and impactful features from the feature matrix. There can be numerous ways to achieve the pooling operation like, sum pooling, mean pooling, max pooling etc. Max pooling extracts the numerically maximum features from the feature maps into pre-defined lower dimensional matrix size (for example 2×2). This helps in dimensionality reduction while retaining the vital information from the image. Once the pooling operation is performed, the features in the pool matrix are converted into a 1-D array (Flattening) and are then fed as input vectors to the fully connected layer.

To evaluate better performances, various ConvNet architectures have been designed [12]. Stacking convolutional layers, on one another, can be thought of as a way to create deep networks but a generic problem of vanishing gradient arises when a large number of layers are just stacked on one another. Several architectures have been constructed and are widely used in a lot of problems. This paper discusses the performances of these architectures in detecting the clog loss or classifying cranial vessels as flowing or stalled.

3.1 Residual Neural Networks

Residual Networks have proven their performances and are one of the most accurate architectures amongst ConvNets. Increasing depth of a convolutional neural network should increase the accuracy or performance (if over-fitting is taken care of) is an astute belief [12]. The problem with increased depth is that the signal that directs the network to change the weights, arises from the end of the layers by comparing the actual (true) data with the predicted data, becomes significantly small or “gets lost” in the initial layers due to the increased depth. This means that only the top layers (end) of the network undergo training, and the weights of the initial layers become

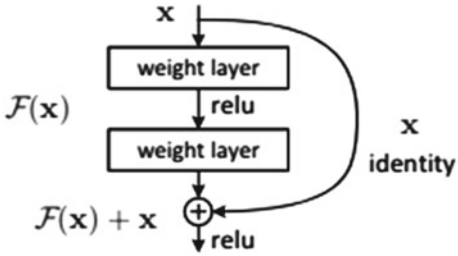

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 5 Building block of residual learning. *Source* https://arxiv.org/pdf/1512.03385.pdf

recalcitrant. This problem is classically referred as the vanishing gradient problem. The optimization of parameters becomes complex with increasing layers and this leads to misclassification of the data. This is referred as the degradation problem. Residual networks work around a solution to this problem. Residual networks consist of non-linear stacked layers like conventional nets. The advantage of these networks is that they define a residual mapping function $F(x)$. These networks do not directly map x to y with a mapping function $H(x)$, rather a residual hypothesis is defined as

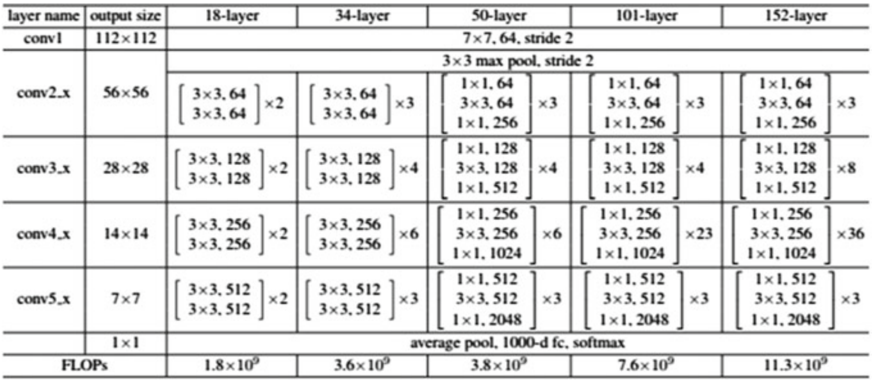
$$F(x) = H(x) - x \tag{1}$$

$$H(x) = F(x) + x \tag{2}$$

It is found that the residual layer $F(x)$ is easier and faster to converge than the direct mapping function of the non-linear stacked layers $H(x)$ [12]. The residual mapping function $F(x)$ optimizes faster than the unreferenced mapping $H(x)$ (Fig. 5).

If an optimal identity mapping is considered, the residuals can be easily pushed to zero, i.e. $F(x) = 0$ instead of fitting an identity mapping (input = output) using a stack of non-linear layers. The architecture of the residual networks is inspired by the structure of VGGNet and has 3 × 3 convolutional layers. In these networks, downsampling is performed directly by the convolutional layers consisting a stride of 2. These residual nets consist of a 1000 way fully connected layer with SoftMax as an output function and make use of the global average pooling layers in between them.

The cropped regions from the extracted frames were fit into these residual networks in the following way. The cropped image was sampled into a 224 × 224 random crop with per pixel mean subtracted from it. Then this image was converted into a 2D image by converting into grayscale and taking the mean of all the pixels. After the convolution is performed, batch normalization is carried out before activation. The weights were randomly initialized at the beginning of the training process [12]. Then, stochastic gradient descent (SGD) with a minibatch of 32. The model is then trained for 25 epochs with an epoch cycle of 52 per epoch. This architecture remains same for Resnet50, Resnet101 and Resnet152 used for training in this paper for the described problem statement. Resnet 50, Resnet 101 and Resnet 152 are the deep bottleneck architectures. The results with these bottleneck architectures



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

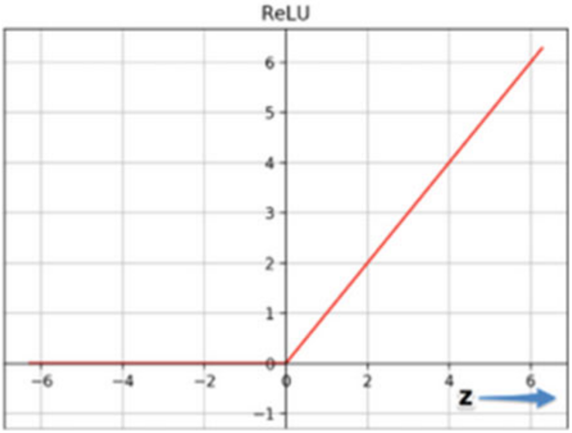
Fig. 6 Architecture of different Resnets (18, 34, 101, 152) used for ImageNet. *Source* https://arxiv.org/pdf/1512.03385.pdf

are not convincing and perform poorly on the dataset. This can be estimated due the non-linearity present in the images, lack of variety, limited or same kind of features, lesser distinctive nature and mainly due to the degradation problem of plain nets as the same architecture yields better results in less deep AlexNet. In Resnet 50, 101 and 152 instead of the two-layered stack, a three-layered stack is used. These three stacks are 1 × 1, 3 × 3 and 1 × 1 convolutions in which the 3 × 3 layers are used in reducing the dimensions and 1 × 1 layers are used to restore them. This leaves the 3 × 3 layer with a bottleneck of smaller input or output dimension. In Resnet 50, each two-layer stride present in Resnet 34 is replaced by a three-layered stack or a bottleneck block resulting in a 50-layered residual network. Similarly, in Resnet 101 and 152, three more-layer blocks are used than Resnet 50 and Resnet 101 respectively, to form a significantly deeper architecture [12] (Fig. 6).

3.2 AlexNet

AlexNet is less intricate in terms of architecture as compared with ResNet [13]. It can be said to be a less-deeper or a shallow architecture as compared to ResNet's. AlexNet consists of just five convolution layers and three fully connected layers in total eight learned layers. The first convolution layer consists of 96 kernels of size 11 × 11 × 3. The depth is same as that of the channels and remains constant. The convolution layers are followed by the pooling layers, to be specific max-pooling layers. Pooling layers as explained earlier in the paper are used to reduce the dimensionality or downsample the width and height of the tensors. AlexNet uses a pooling window of 3 × 3 with a double stride between the adjacent windows. The activation function used in AlexNet's fully connected layer is the Rectified Layered Unit or the ReLU function (Fig. 7).

Fig. 7 ReLU function



$$F(x) = \max(0, x)$$

By making use of the ReLU function, AlexNet [13] was trained much faster than the conventional saturating sigmoid function. ReLU makes the ConvNet train 25% faster than the conventional networks. Another advantage that helped in increasing the accuracy observed in Sect. 7 apart from the less complex architecture is the use of dropout. On a dataset of nearly 80,000 images, AlexNet helps in reducing the overfitting by applying a technique called dropout. The dropout method drops a neuron with a probability of 0.5 from the network. When a neuron is considered as dropped, its contribution to the feedforward neural network becomes inconsequential. As a result of this exercise, the training of the weights becomes more robust and overfitting can be easily avoided. This randomized dropping of neurons makes the architecture less intricate and contributes to achieving a higher accuracy (Fig. 8).

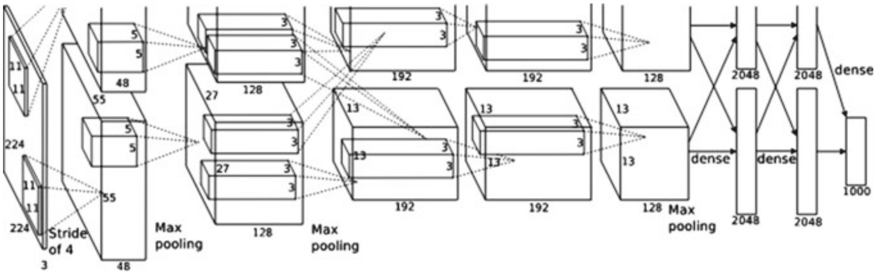


Fig. 8 Architecture of AlexNet. *Source* https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

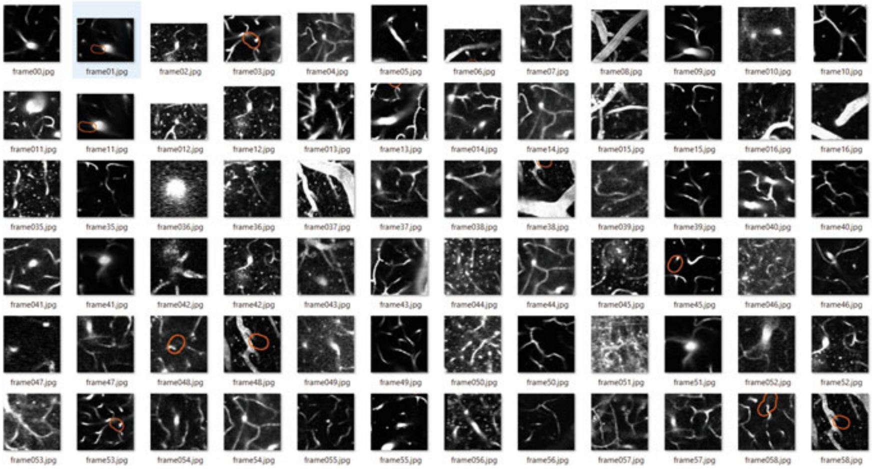


Fig. 9 Frames captured from videos

4 Dataset

The dataset has been obtained from https://www.drivendata.org/competitions/65/clog-loss-Alzheimer'ss-research/. These videos have showcased live flow of blood within the blood vessels in the brain. Each video has been recognized and spotted with its "filename" followed by ".mp4", for example 100000.mp4. These videos were made available by Stall Catcher's a project by the Human Computation Institute, based in Ithaca, New York.

The target vessel for each video has been identified by red band in each frame was the main area of focus we had, to grab the image region out and feed it to our model. The entire dataset of images is approximately of 6.4 GB size. Along with this, there has been an excel provided labeling next to each video whether a particular video is classified as flowing or stalled. The next task after downloading the dataset was to convert the videos to images, which was done by making several frames for a video (Fig. 9).

5 Marked Region Extraction

For each category flowing and stalled, we have made several frames respectively for training and testing. Initially, while using the complete image for analyses, we found that our accuracy was coming very low (approximately 50% using ResNet and AlexNet). So, then we used various approaches to crop the images into just the orange outline, which is the primary portion of the analysis. The most naïve approaches involve template matching and Haar cascade. Both the methods do not

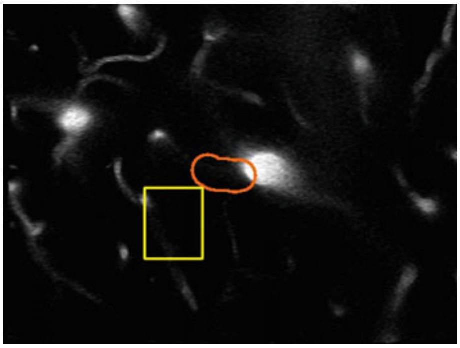


Fig. 10 Boundaries generated from template matching technique

give the accurate area in the red region so we developed our own method to do so (Mathematical Approach).

5.1 Template Matching

This is the method in which we try to match the feature in the experimental image with the cropped sample. The cropped sample crossed the experimental image. The area with the highest number of hits is chosen, and the result is shown with the rectangle box. The main difficulties due to which this method was giving inaccurate result because there was greater number of hits in the region outside the red area with the area which was inside the cropped image.hence the all the method of template matching 'cv2.TM_CCOEFF', 'cv2.TM_CCOEFF_NORMED', 'cv2.TM_CCORR', 'cv2.TM_CCORR_NORMED', 'cv2.TM_SQDIFF', 'cv2.TM_SQDIFF_NORMED' with 'cv2.TM_CCOEFF_NORMED' giving the closest result (Fig. 10).

5.2 Haar Cascade

Haar cascade is the most famous and among the most accurate method that can be used for detecting shape-based features. The steps involving in this method are clearly defined a document from Auckland university. The first method involves collection of positive samples, which involves the red bounded area and the negative background. Then we have to create a vector of positive image followed by training the classifier. The results we get from the haar classifier involved many false regions being classified in an image. The main reason of failure is matching of similar region inside the bounded region with the rest of area (Fig. 11).

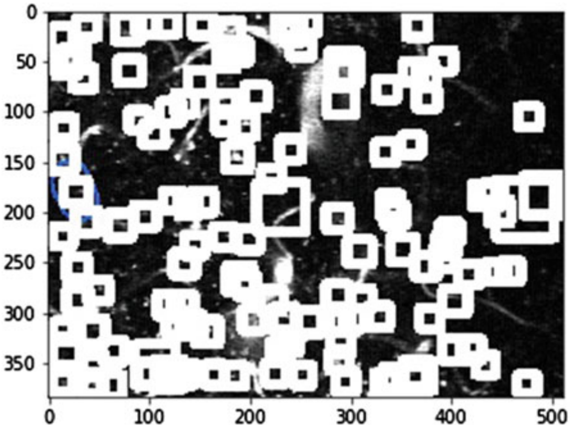


Fig. 11 Boundaries generated using Haar Cascade

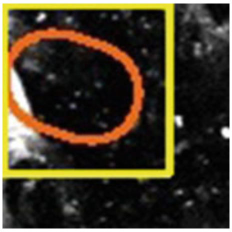


Fig. 12 Boundaries generated from mathematical approach (pixel intensity extraction)

5.3 Mathematical Approach (Pixel Intensity Extraction)

This approach involves finding the red color in the image. A range of color that falls in the bandwidth of the red region is checked fastidiously through the image and the coordinates are recorded. The lowest x and y coordinates and highest x and y are noted. Using these coordinated, we draw the diagonal of rectangle and so we get the region bounded by the rectangle box. Accuracy of this result is 100% as the image has only red color in the boundary. These coordinates are further used and marked in the original image. This region is further cropped out using the coordinates of top left and bottom right points in the rectangle and it is ready to be fed the model (Fig. 12).

6 Results

As discussed above, this paper proposes a methodology to predict whether the marked blood vessel in the video is flowing or stalled, which in turn helps us in predicting whether a person is suffering from Alzheimer’s or not. The dataset consisted of 4600 videos of cranial vessels classified as flowing or stalled. After extracting the frames and manual filtering of the images, a dataset of 78,675 images in total was created. This paper makes use of transfer learning and discusses the results of classification by four distinct architectures viz. Resnet50, Resnet101, Resnet152 and AlexNet. But out of these architectures, AlexNet architecture's performance was found to be the best. The following figures will give you an idea about its performance (Figs. 13, 14, 15, 16 and 17; Table 1).

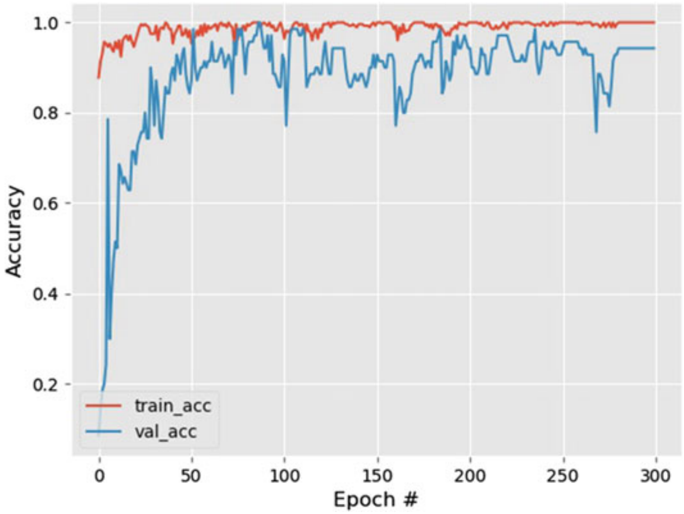


Fig. 13 Accuracy on the dataset

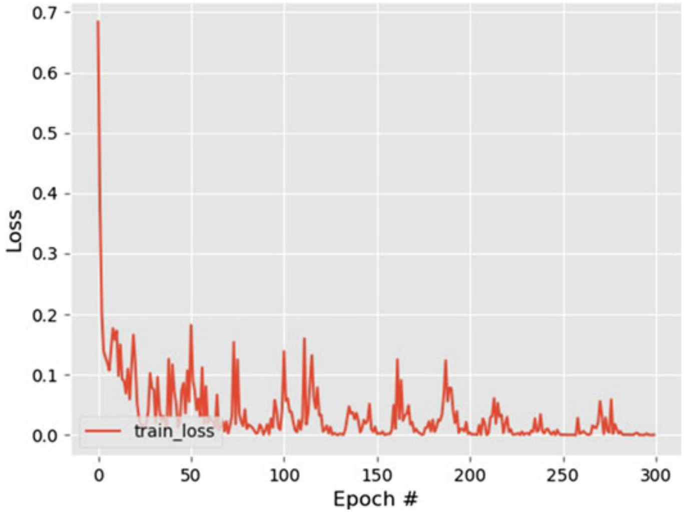


Fig. 14 Training loss on the dataset

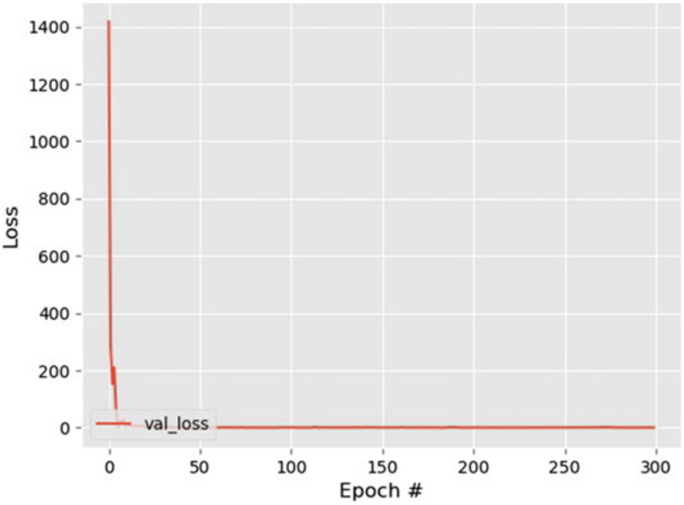


Fig. 15 Testing loss on the dataset

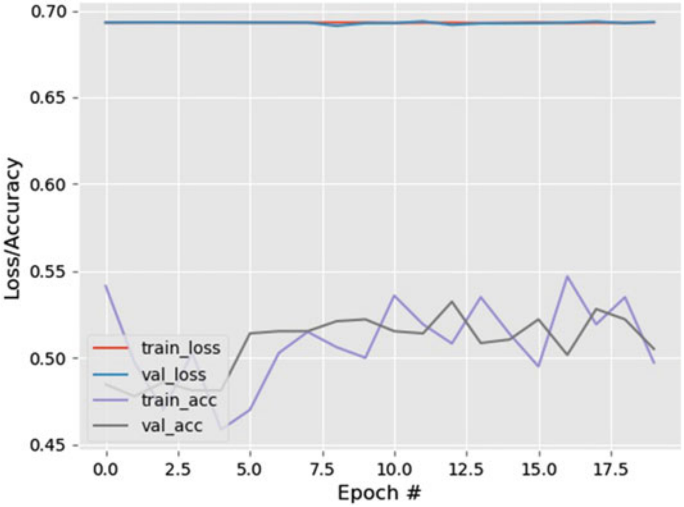


Fig. 16 Training loss and accuracy on dataset: Resnet 50

So, the dataset contained 4600 videos of cranial vessels classified as flowing or stalled. After extracting the frames and manual filtering of the images, a dataset of 78,675 images in total were created. The following is the Confusion Matrix for AlexNet (Table 2).

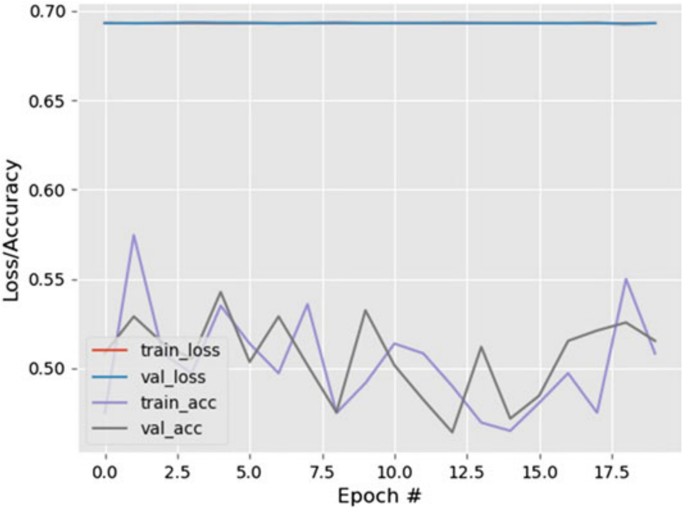


Fig. 17 Training loss and accuracy on dataset: Resnet 101

Table 1 Architecture accuracies and losses comparison

Architecture	Training accuracy (%)	Training loss (%)	Validation accuracy (%)	Validation loss (%)
ResNet 50	89.50	69.32	52.22	69.28
ResNet 101	89.72	69.34	50.51	69.35
ResNet 152	87.46	69.32	72.90	69.31
AlexNet	99.48	10.42	**98.57**	4.41

Table 2 AlexNet confusion matrix

	Predicted positive	Predicted negative
Actual positive	50,471	631
Actual negative	949	27,174

7 Conclusion

As discussed above, the convolutional neural network (CNN), which is one of the Deep Learning Network designs, is used to characterize the blood blow in cranial vessels in an Alzheimer's brain and to create a trained and prescient model. This work has been done with the prime objective of building learning models specifically using ResNet and AlexNet to effectively and efficiently detect blockage of blood flowing in the brain of a person. The dataset included 4600 videos of cranial vessels classified as flowing or stalled. After extracting the frames and manual filtering of the images, a dataset of 78,675 images in total was created. After getting our frames, which comprise to form the video itself, we extracted only out area of interest which gave us maximum information about the blood stream flow. Four different architectures, namely, were tested on the same dataset. Resnet50, Resnet101, Resnet152 and AlexNet. After trying out the same dataset on these architectures, AlexNet was seen to give the best performance with a huge accuracy of 98.57%. Not only the accuracy is high, but the losses are low as well. The AlexNet model helped hugely in predicting accurately a person affected by Alzheimer's.

References

1. Tofighi G (2016) Alzheimer's disease neuroimaging initiative. DeepAD: Alzheimer's disease classification via deep convolutional neural networks using MRI and fMRI. BioRxiv, 070441
2. Fulton LV, Dolezel D, Harrop J, Yan Y, Fulton CP (2019) Classification of Alzheimer's disease with and without imagery using gradient boosted machines and ResNet-50. Brain Sci 9(9):212
3. Razzaq MI, Naz S, Zaib A (2018) Deep learning for medical image processing: overview, challenges and the future. In: Classification in BioApps. Springer, Cham, pp 323–350
4. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks
5. Xiao Y, Pan D (2019) Robust visual tracking via multilayer CaffeNet features and improved correlation filtering. IEEE Access 7:174495–174506
6. Zahangir Alom Md, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS et al (2018) The history began from alexnet: a comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164
7. Sarraf S, Tofighi G (2016) Classification of Alzheimer's disease using fmri data and deep learning convolutional neural networks. arXiv preprint arXiv:1603.08631
8. Ballester P, Araujo RM (2016) On the performance of GoogLeNet and AlexNet applied to sketches. In: Thirtieth AAAI conference on artificial intelligence
9. Zhong Z, Jin L, Xie Z (2015) High performance offline handwritten Chinese character recognition using GoogLeNet and directional feature maps. In: 2015 13th International conference on document analysis and recognition (ICDAR). IEEE, pp 846–850
10. Ke H, Chen D, Li X, Tang Y, Shah T, Ranjan R (2018) Towards brain big data classification: epileptic EEG identification with a lightweight VGGNet on global MIC. IEEE Access 6:14722–14733
11. Espinosa JE, Velastin SA, Branch JW (2017) Vehicle detection using Alex net and faster R-CNN deep learning models: a comparative study. In: International visual informatics conference. Springer, Cham, pp 3–15
12. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition
13. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks
14. Tofighi G (2016) Deep learning-based pipeline to recognize Alzheimer's disease using fMRI data. In: 2016 Future technologies conference (FTC). IEEE, pp 816–820

Challenges and Risks Associated with Public Key Infrastructure

Muskaan, Sarvesh Tanwar, and Sunny Singh

Abstract Computer networks have been sufferers of the attacker even when it was considered as an inception. First, it was firewalls, later point interruption location frameworks, after that VPNs, and now certification authorities (CAs) and Public Key Infrastructure (PKI). Public Key Cryptography (PKC) is a system that enables gatherings to convey safely using public and private key sets. PKC-based correspondences can be both credible and mystery, despite the fact that the public keys are made generally known and accessible. This paper gives a short layout of the fundamental ideas and principles associated with the activity of a PKI including issues, for example, how a PKI works, its qualities and what issues should be tended to before the utilization of PKI turns out to be increasing across the board and likewise quickly takes a glance at the diverse zones to which PKI can be connected to take care of the existing issues and inspects a scope of current reactions and difficulties. It attempts to contemplate a portion of those inquiries after examined and a present writing survey regarding the matter. We demonstrate the challenges and risks related with PKI, contrary to a basic conviction that public key authentications and accreditation administrations can be showcased freely from applications and application conditions.

Keywords Certificate practice statement (CPS) · Certificate revocation list (CRL) · Certificate signing request (CSR) · Certification authority (CA) · Public key cryptography (PKC) · Public key infrastructure (PKI) · Symmetric encryption (SE)

1 Introduction

The Internet is rapidly turning into the biggest commercial center, permitting trade and business between gatherings who are physically removed and do not have any

Muskaan · S. Singh
Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India
e-mail: sunny.singh@chitkara.edu.in

S. Tanwar (✉)
Amity Institute of Information Technology, Amity University, Uttar Pradesh, Noida, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_26

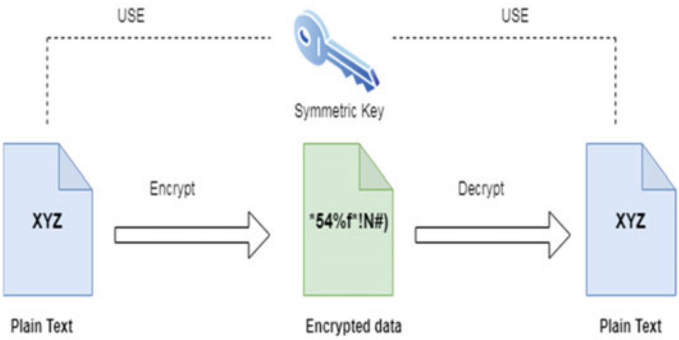


Fig. 1 Working of symmetric encryption

acquaintance with one another. Security must be usable by people extending from non-specialized clients to specialists and framework overseers. Besides, frameworks must be usable while looking after security. Without usable security, there is eventually no powerful security [1]. In many business associations, the parties need to build up some trust on one another [2], by accepting references from confided in delegates.

Cryptography is tied with encoding and unscrambling information. With encryption, you convert a plain content into an arbitrary exhibit of bytes. Decoding is the contrary procedure; you convert the arbitrary cluster of bytes into a plain text. Encrypting any bit of plain content needs a key to do the activity and furthermore, the unscrambling procedure needs a key to change over encoded information into a plain content. A key is the controller of the encryption procedure that is utilized by a calculation. Asymmetric encryption utilizes one key to encode and another key to unscramble your information, and these two diverse keys are scientifically identified with one another. One of these keys is public and can be utilized by everybody. The other key is private and it ought to be utilized distinctly by you and never imparted to anybody (Fig. 1).

Below depicted Fig. 2 shows the asymmetric encryption using different key pair sets.

2 Public Key Infrastructure

A Public Key Infrastructure (PKI) is an arrangement of advances that help the foundation of electronic trust between depending parties. In light of the idea of Public Key Cryptography (PKC), the innovation has so far turned out to be a powerful stage for secure interchanges including secure electronic business on the Internet and somewhere else, including banking systems. Each customer is most likely going to have different keys that require lifecycle the board. For example, customers usually have something like one key pair for each ensured application (for instance email, work

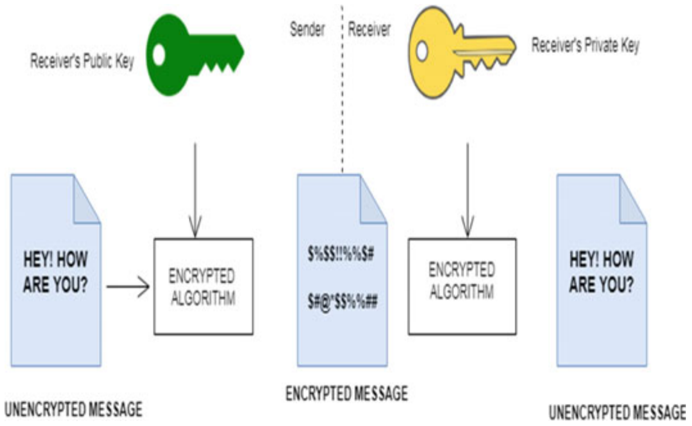


Fig. 2 Working of asymmetric encryption

territory record encryption, VPN). For PKC to work effectively on an expansive scale more often than not requires the foundation of a PKI. At a center of PKI is a confided in specialist, ordinarily alluded to a CA, which authenticates the legitimacy of the public keys of supporters. It does as such by marking these keys to frame an information structure known as a public key authentication. The public keys would then be able to be traded electronically and checked for uprightness utilizing the public key of the affirming CA, which should initially be gotten by a dependable strategy. Figure 3 depicts an architecture of the PKI.

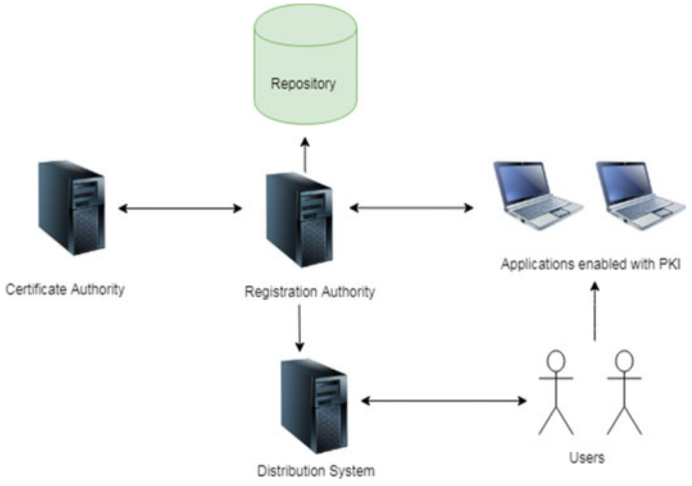


Fig. 3 Architecture of the PKI

3 Components of PKI

To develop trust in the PKI one can, form a CPS, which is a report one makes, which delineates the PKI setup, how you work it and the essential and approach for issuing confirmations. Figure 4 represents the components of PKI.

i. **Certification Authorities**

The Certification Authority or CA is the administration, which is in charge of issuing and renouncing declarations. This could simply be a straightforward setup with a couple (yet incredible) contents utilizing an opensource authentication toolbox, OpenSSL and a bundled arrangement, for example, Microsoft Certificate Services [3].

ii. **Private and public keys**

A key pair, i.e., a private and public key are associated with each driven assistance. A public key is added to an approval. As the name induces, this information is open to the globe to see, or possibly to the general population that will use the statement. The private key is the accessory to the open key and is private to the substance (individual or a PC gadget) that will use the validation. Only the private key can unscramble it when the data are mixed with the open key. Once again, when data are mixed with the private key, the open key can unwind the information or data.

iii. **Enrollment of certificates**

The public key can be embedded in a validation request that is sent to the certification specialist at the time when the general public and private key pair has been produced. For example, the name that will be consolidated into the validation could be used close

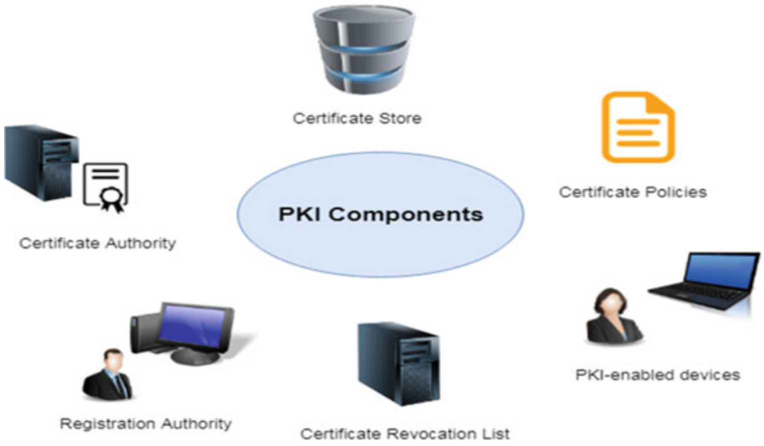


Fig. 4 Components of the PKI

to the confirmation request. This would be the site's name for an SSL confirmation, e.g. secure.networklore.com. Regardless of how the CSR can be incorporated, it is necessary for the CA to make sense of which information is kept and which information will be incorporated by the CA paying little attention to what has been fused into the CSR. A few CAs will neglect all in the request and mainly maintain the public key, while others will require certain areas to organize the CA's demands.

iv. **Digital certificates**

It will integrate the public key along with other statement information when the CA has given an approval. Together with the private key, a client could now use the authentication to decode the information sent to the client, or scramble information that can be unscrambled by others and thus confirmed with the approval itself.

4 Related Work

The key knowledge that builds secure frameworks expects consideration regarding ease of use returns about four decades. Saltzer and Schroeder [4, 5] distinguished "mental agreeableness" as a key standard essential for keeping up a safe processing condition. Morris and Thompson [6] noticed that PDP-11 [7] Unix clients regularly picked passwords that were effectively speculated, leaving their records public to settle. Karat [8] demonstrated that adjustments to an inward IBM security application dependent on customary ease of use inquire about brought about significantly expanded client execution and security.

Today, most of the total populace are clients of email and Internet-based life [9]. Early expectations that individuals who grew up submerged with data and interchanges innovation (ICT) would by one way or another comprehend it as "computerized locals" [10] have been appeared to be to a great extent unwarranted. For instance, Kurkovsky and Syta [11] overviewed more than 330 youngsters matured 18–25 who had grown up with the innovation and found that, despite the fact that they knew about protection and security dangers that they looked because of utilizing cell phones, the larger part did not take specialized measures to ensure themselves. Albeit 80% of the overview respondents knew that telephones could be bolted with a PIN, just 29% utilized PIN locks. Information security is the difficult issue of today that contacts numerous territories including PCs and communication. Present-day digital security assaults have sincerely performed with the effects of the clients. Cryptography is one such system to make sure that the validation, confidentiality, accessibility, secrecy and recognizable proof of client information can be kept up just as security and protection of information can be given to the client. The cryptography methods and different calculations are utilized to give the required security to the applications [12].

The cryptographic methods portrayed to this point depend on a common key between two gatherings, otherwise called Symmetric encryption (SE). SE is the most seasoned and understood method. It utilizes a mystery key that can be a number, word, or arbitrary letters. Every one of the gatherings, the sender, and collector need the key in their ownership. There is an issue with the idea of a mystery key [13]. Having the information of this mystery key can decode the message. As messages can start from any of the sources in the Internet, having the capacity to build up the uprightness of these messages through components, for example, message confirmation codes and advanced marks is vital, yet dependent on the foundation of keys between number of hubs a dealing with these pair-wise keys between rapidly ends up unmanageable. Key administration on a worldwide scale requires public key cryptography. The PKI [14] handles the demands for public keys starting from different nodes. PKI is discussed in more detail in the next section.

5 Risks Associated with PKI

Since security programming cannot generally settle on the right choice for the benefit of clients, clients are routinely compelled to know about security and protection issues and to settle on choices for themselves. Furthermore, they have to deal with the protection and security of their own data, shared on a consistently developing arrangement of internet, portable, and distributed computing stages, against dangers that 10 years back were less obvious.

i. Risks Associated with CA

There is a hazard from an uncertain utilization of "trust." A CA is regularly characterized as "believed." Who gave the specialist the right to permit such approvals to CA? Who trusted it? CA can complete an eminent activity of composing a nitty gritty Certificate Practice Statement, or CPS—each of those we have perused disavow all risk and meaning to the testament—and after that work superbly following that CPS, but that does not mean you can rely on your request for approval [15].

ii. Risks of PKI User

In any CA-based scheme, one of the most severe hazards is with your own one-of-a-kind private control key. How would one can ensure the key? You in all likelihood do not claim a safe figuring framework with physical access controls, protecting, air divider, organize security, and different insurances; client stores the private key on an ordinary PC. There, it is liable to assault by infections and different noxious projects. Regardless of whether your private key is sheltered on clients PC, is his/her PC in a bolted room, with video reconnaissance, If it is secured by a secret phrase, how hard is it to figure that secret key? On the off chance that your key is put away on a brilliant card, how assault safe is the card? In the event that it is put away in a genuinely assault safe gadget, can a contaminated driving PC get the dependable gadget to sign something you did not expect to sign?

iii. Risks of Verifying Agents

There is no privileged insight to secure in this way. Nevertheless, it utilizes at least one government “root” keys. On the off chance that the assailant can add his own public key to that rundown, he can issue his own one-of-a-kind validations at that point, which will be handled unambiguously like the authentic revelations.

iv. Risk of Certificate Practices

How do you figure important lifetime? Does the vendor use a year because that’s normal? A key has a lifetime of cryptography. It also has a lifetime of burglary as an aspect of the subsystem’s helplessness putting it back, the rate of implementation of physical and scheme, the allure of the manner to an aggressor, and so on. From these, the likelihood of key loss can be processed as an element of time and use. Is that calculation done by the vendor? What is the probability edge used to think about an invalid key?

v. Whom to trust?

Security is a chain; it is as strong as the weakest link. Security is based on countless links and not all of them are cryptographic. It includes individuals. Does the framework help those individuals, confound them or simply overlook them? Does it depend improperly on the genuine quality of individuals? PC frameworks.

vi. How to differential Identities?

Certificates usually associate a public key with a name, but few people speak about how important that affiliation is. Imagine receiving the testament from Robinson [16]. You may only understand by and by John Robinson, but what amount do you understand about the CA? How would you see if the authentication of your companion is the testament you received from John Robinson?

vii. Who secures the Keys?

With your own unique personal stamping key, one of the most severe hazards in any CA-based scheme. How could you secure it? You almost definitely do not ensure a sheltered figuring structure with physical access controls, TEMPEST ensuring safety for the "air divider" orchestra, and various protections; your private key is stored on a normal PC. There, illnesses and unique threatening ventures are forced to ambush [17, 18].

Viii. Does CA Cares about end User?

Does a validation request believe about the client or is it just about cryptography [19]? A prevalent client, for instance, settles on a choice to shop with a specified SSL-verified web page topic to what appears on that page. The affirmation has not emerged and has no connection whatsoever with what appears. SSL [20] safety is not capable of controlling or even reacting to the Web page’s substance, only its DNS address. In regards to anything the client sees, the corporate name does not appear differently and there are some web pages whose underwriting is for an organization that encourages networks, not for the organization whose logo appears on the website shown. Customers can not sort this company and cannot be dependent on it.

6 Reasons of PKI Failure

In this section, we will recognize various reasons that may prompt late PKI failures.

Technical reasons: The specialized purposes behind the PKI disappointment all have to do with the way that a PKI is more required than one might suspect at first sight in building up and working it.

Complexity: The subsequent information structures utilized by calculations, for example, X.509[x] [21] are non-instinctive and for human per users not extremely important. They are additionally relatively difficult to examine (via robotized preparing of information). For instance, this is as opposed to PGP [22] authentications. Later on, declarations dependent on XML/JSON might be an intriguing option.

Certificate management: The executive of certificate is a mind-boggling and testing task, and numerous things can turn out badly [23]. From a progressively specialized perspective, the most testing errand of testament to the board is likely the disavowal of authentications. Public key sets, for instance, should be produced productively and safely. This should be possible halfway or in a decentralized way.

Cross-certification: Once in a while people have contended in the past that CAs can cross-certify each other to frame multi-CA PKIs [24]. In any case, cross-accreditation requires fairness (or if nothing else entirely equivalent) of the relating CPSs. Tragically, practically speaking, cross-certification isn't working. Normally, CSPs contend that the confirmation administrations they give are superior to contender's benefits and are along these lines unfit to cross-certify them [25].

Economic reasons: The financial reasons for PKI dissatisfaction [26] are inextricably linked to the fact that establishing and running a PKI is an expensive undertaking, and it is difficult to charge customers for services not just when the customer receives his first confirmation.

7 Conclusion

While assaulting a product framework is just as troublesome all things considered to acquire powerlessness to abuse, the security quality of that framework is proportionate to the market cost of such a defenselessness. This paper has inspected a scope of specialized, infrastructural, operational and the board issues related with the utilization of PKI. PKI is still in its earliest stages but then numerous associations have just started sending authentication empowered applications and foundations. PKI is a promising security innovation and whenever utilized appropriately, organizations and associations can profit by it. However, given the complex nature of the framework required to execute and support a public PKI framework, setting up PKI-enabled apps for certain industry groups is to be done by the industries themselves. Additionally, public key cryptography as a rule, and advanced marks and public key-based key foundation systems are essentially excessively important than not to be utilized practically speaking. Indeed, there is not really any option in contrast to the utilization of computerized marks to give non-disavowal benefits on a substantial scale. By tending to the issues portrayed above, associations ought to have the capacity to exploit this new and noteworthy innovation. Another expansion could be the plan of another authority calculation to choose the request to look in changed stores and to improve the check if an endorsement is disavowed.

References

1. Department of Homeland Security's (DHS). Guarding against terrorism and threats to cyber networks and critical infrastructure. [online Available]: https://www.dhs.gov/xlibrary/assets/NIPP_Plan.pdf
2. Tanwar S, Prema KV (2018) Design and implementation of a secure hierarchical trust model for PKI. In: In cyber security. Springer, Singapore, pp 415–425
3. Tanwar S, Kumar A (2017) A proposed scheme for remedy of man-in-the-middle attack on certificate authority. Int J Inform Secur Privacy (IJISP) 11(3):1–14
4. Smith RE (2012) A contemporary look at Saltzer and Schroeder's 1975 design principles. IEEE Secur Priv 10(6):20–25
5. Siponen MT (2000) Critical analysis of different approaches to minimizing user-related faults in information systems security: implications for research and practice. Inf Manag Comput Secur 8(5):197–209
6. Morris R, Thompson K (1979) Password security: a case history. Commun ACM 22(11):594–597
7. Eckhouse RH, Sloan ME (1976) Minicomputer systems: organization and programming (PDP-11). IEEE Trans Syst Man Cybern 10:722–722
8. Karat CM (1989) Iterative usability testing of a security application. In: Proceedings of the Human Factors Society annual meeting, vol 33, No. 5. SAGE Publications, Los Angeles
9. Reaney P (2012). Most of world interconnected through email and social media
10. Prensky M (2001) Digital natives, digital immigrants part 1. On Horizon 9(5):1–6
11. Kurkovsky, S., &Syta, E. (2010, June). Digital natives and mobile phones: a survey of practices and attitudes about privacy and security. In: 2010 IEEE International symposium on Technology and Society (ISTAS). IEEE, pp 441–449
12. Zissis D, Lekkas D (2012) Addressing cloud computing security issues. Futur Gener Comput Syst 28(3):583–592
13. Blaze M, Diffie W, Rivest RL, Schneier B, Shimomura T (1996) Minimal key lengths for symmetric ciphers to provide adequate commercial security. A report by an ad hoc group of cryptographers and computer scientists. Information Assurance Technology Analysis Center Falls Church VA
14. Housley R (2004) Public key infrastructure (PKI). The internet encyclopedia
15. http://www.cse.psu.edu/~trj1/cse543-f06/presents/Schiffman_Risk.pdf

16. Whitten A, Tygar JD (1999) Why Johnny can't encrypt: a usability evaluation of PGP 5.0. In: usenix security symposium, vol 348
17. Ellison C, Schneier B (2000) Ten risks of PKI: what you're not being told about public key infrastructure. Computer Secur J 16(1):1–7
18. https://www.sans.org/reading-room/whitepapers/authentication/paper/1198. Accessed on 30 May 2020
19. Mozaffar A (2017) Implement symmetric and asymmetric cryptography algorithms With C# (2017) C-sharpcorner.com, from https://www.c-sharpcorner.com/article/implement-symmetric-and-asymmetric-cryptography-algorithms-with-c-sharp/
20. Sotomayor B (2005) The globus toolkit 3 programmer's tutorial. BorjaSotomayor
21. Fluhrer SR, McGrew DA (2000) Statistical analysis of the alleged RC4 keystream generator. In: International workshop on fast software encryption. Springer, Berlin, pp 19–30
22. Lu CC, Tseng SY (2002) Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In: The IEEE international conference on application-specific systems, architectures and processors. IEEE, pp 277–285
23. Yun-Peng Z., Wei, L., Shui-ping, C., Zheng-jun, Z., Xuan, N, Wei-di D (2009) Digital image encryption algorithm based on chaos and improved DES. In: IEEE International conference on systems, man and cybernetics (SMC 2009). IEEE, pp 474–479
24. Singh G (2013) A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. Int J Computer Appl 67(19)
25. ElGamal T (1985) A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans Inf Theory 31(4):469–472
26. Serrano N, Hadan H, Jean Camp L (2019) A complete study of PKI (PKI's known incidents). Available at SSRN 3425554

Mathematical Information Retrieval Using Formula2Vec Approach

Pankaj Dadure, Partha Pakray, and Sivaji Bandyopadhyay

Abstract Searching for the needed information involves finding relevant information with respect to the user's entered query which is varying from the basic database search space to the sophisticated web search levels. The theoretical and practical research in mathematical information retrieval (MIR) is ultimately based on formula representation and indexing which have been elaborated over a period of time. However, the main aim of the retrieval system is not to retrieve the more number of search, rather to retrieve the highly useful and relevant search which holds the potential to satisfy the user's need. The word2vec approach has achieved the remarkable performance in the textual knowledge representation. Motivated from this, we have proposed the formula2vec approach which learns the vector representations of mathematical formulas. The proposed formula2vec approach has been tested on the math stack exchange corpus of ARQMath-2020, and the obtained search results depict their proficiency.

Keywords Formula searching · Formula2Vec embedding · Indexing · Cosine similarity

1 Introduction

Information retrieval (IR) is a subfield of natural language processing (NLP) which aims to retrieve the needed information from the collection of documents. The general IR system takes the user's query as an input, works on the similarity estimation, and based on that returns the rank of relevant documents. For student or researcher, it is quite difficult to understand the unfamiliar mathematical equation while reading the scientific documents. In such cases, a math-based search engine is helpful for them to search the related information and to understand the unfamiliar mathematical equation. In IR system, the user query depicts the information need, and system should retrieve the relevant articles for the same. The retrieval of needed informa-

P. Dadure · P. Pakray (✉) · S. Bandyopadhyay
Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_27

tion is mostly depended on the terms present in user query. The more appropriate term leads to more relevant search. For students who have very little knowledge of mathematical concept, it is very difficult for them to form the math/text-based query which holds the complete version of needed information. Even after students take efforts to form the text/math query, they might not probably get the absolute results from the conventional or the math search engines, while retrieval of mathematical information is beneficial for numerous applications such as plagiarism detection, math recommendation system, and math notation analysis.

In the early stage of research, most of the formula retrieval approaches are based on tf-idf model; however, the tf-idf approach is not able to consider the syntactic as well as semantic structure of formula. For instance, the formulas x^2 and 2^x are syntactically and semantically diverse. Therefore, the designing of MIR approaches is still an open problem. In the process of MIR, the user query plays a vital role which when subjected to guide the searcher module to enhance the quality of result. The well-formed query assists the system in formulating a better search, by appending additional weights to the initial search query. In this paper, we have implemented the formula2vec approach which learns the vector representations of mathematical formulas. For similarity estimation, cosine similarity has used. The proposed formula2vec approach has been tested on the math stack exchange corpus of ARQMath-2020.

The paper is structured as follows: Sect. 2 describes prior works in domain of MIR. Section 3 provides detailed description about the system architecture. Section 4 given the detailed account of dataset. Section 5 describes the experimental results of proposed approach and their comparative analysis. Section 6 concludes with summary and directions of further research and developments.

2 Related Work

The prior research works have aimed to design the formula representation and indexing technique to deal with the obstacles of MIR. The enlarging research in science, engineering, and technology produces the number of documents which contains mathematical formulas filled with text. To perform the semantic search of formula, the formula retrieval system requires the more meaningful information to conduct the semantic search operation. For instance, AnnoMathTeX—a recommender system [12]—performed the formula annotation by appointing the text to identifiers. In variable typing approach [13], the mathematical text sentence assigned a specific type to each variable. Over the past few decades, a growing number of significant mathematical findings have needed human effort and computer support. A information model for "doing mathematics" suggests that the people need to integrate four aspects very effectively: inferentiality, computation, tabulation, and narration around a well-organized core of mathematical understanding are a central contribution [2]. The supremacy of deep learning neural network in task of information retrieval motivates the researchers to incorporate it in mathematical information retrieval. For example, the LSTM neural network-based approach [10] for the retrieval of math information formulates the entailment between the formula-based user query and formulas contained in scientific documents. The similarity estimation between the formula-based user query and information contained scientific documents is a significant milestone to perform the effective searching. For example, the evaluation of cosine similarity between the formulas leads to considerable search results [3]. To analyze the characteristic of natural language and mathematical language, the approach of "symbol2vec" method [6] is to learn the vector representations of formula symbols. For formula2vec transformation, the distributed memory model of paragraph vectors (PV-DM) has used, which can learn distributed representations of formulae. For similarity calculation, cosine similarity has employed. The systematic frequency distributions analysis of formulas from the scientific corpus furnish the fitness function of variety of applications [8]. For instance, the extended zbMATH's search engine [8] uses the mBM25 score for relevant expression extraction and the numerous filter for faceted search.

The cross-aspect retrieval approach [1] aims to store information in aspect-specific database. Consequently, it incorporates the binary queries from developed aspect-specific query representation and is spread by a query compiler into the respective indices. The query language allows the variables to be shared among aspects-specific sub-queries and non-trivial joints. Motivated from the concept of formula embedding and term-document matrix, the variable-size formula embedding is an approach [4] where the formula is converted into the variable-size vector, in which each bit of the vector represents their occurrence and corresponds to their position in BPIT. In order to convey ideas of scientific articles, both mathematics and text information are important content. Motivated from the contextual relationship of scientific text and mathematical equations, TopicEq approach [14] uses the mixture of latent topic and RNN for the creation of correlated model. The natural premises selection approach [5] acquired the supporting meanings and hypotheses which are suitable for producing informal mathematical evidence for a particular argument. To evaluate the performance of this approach, NL-PS data has been created which consists 20,401 different articles of definitions, lemmas, corollaries, and theorems.

3 Corpus Description

In order to comprehensively understand the essence and source of the data such as what the data is, the structure in which the data is displayed, the license to which data is obtained, how it was created, and who created it. The dataset used in this work is built by the ARQMath CLEF-2020[1] organizer where the posts are extracted from the knowledge sharing platform, i.e., math stack exchange (MSE). The math stack exchange of ARQMath-2020 dataset contains 28,320,920 formulas derived from the question, answer, and comment posts of MSE. The formulas in the dataset

[1] https://www.cs.rit.edu/dprl/ARQMath/.

Table 1 Math stack exchange of ARQMath-2020 corpus [15]

Corpus	Math stack exchange of ARQMath-2020
Type	Formula
Formats	L^AT_EX
Size	1.5 GB
No. of formulas	28,320,920
No. of test queries	10

are represented in L^AT_EX, presentation MathML, and content MathML format. For our work, we have used only the L^AT_EX format, and to test the proficiency of the proposed approach, ten formula-based query is used (Table 1).

4 Methodology

4.1 Prepocessing

4.1.1 Lower Casing

Formulas like $A^2 + B^2$ and $a^2 + b^2$ have same syntactic and semantic meaning, but when not converted to the lower case, those two are represented as two different formulas in the vector space model.[2] To handle this, we have converted all formulas into the lower case.

4.2 Formula2Vec Model

The neural network model for textual information retrieval uses the various deep neural networks techniques to recogniz the entailment between the words or sequence of words. Motivated from the existing word2vec model, we proposed the “formula2vec”-based MIR approach. Word2vec was developed by Tomas Mikolov [7], which uses the framework of the deep learning to get a distributed representation of the word in semantic space [9]. The term similarity can be determined by cosine similarity, Euclidean distance, with a given vector in a multiple dimension scale. Word2vec encodes word with the higher frequency terms in the Huffman tree and triggers fewer hidden layers, thereby efficiently reducing machine complexity. Word2vec benefits from high training performance and rich semanticization that can be used for clumping and for synonymous searching. Word2vec renders a distributed representation of the word using either the continuous skip-gram or the continuous bag-of-words

[2] https://thehelloworldprogram.com/python/python-string-methods/.

Table 2 Search results with similarity score

Formula query	Similar formula	Similarity score
$bf(c) - af(a)$	$f(b) - f(a)$	0.899768629
$\Phi(n) = 40$	$\phi(n) = 0$	0.900011919
$e^{3\Pi i/2}$	$e^{i\pi\frac{2}{3}}$	0.807504931
$f : B \to \mathbb{R}^m$	$x \in \mathbb{R}^m$	0.531128185
10^{-10}	10^{-350}	0.904112597

(CBOW) model. The network is trained to rebuild the meaning of the words so that the words in the corpus share the same meaning and make the words semantically closer to each other. The current term is forecasted in the CBOW architecture with its surrounding term, and the surrounding words are expected in skip-grams with the current word. CBOW is quicker than skip-gram, but skip-gram performs better for uncommon terms better. The word2vec model is used with hierarchical softmax or negative samples where hierarchical softmax is best for unique phrases, and for frequent terms, negative sampling is easier. In this work, we take the skip-gram design. The sample of retrieval search results with similarity score is shown in Table 2.

4.3 Similarity

To estimate the similarity between the index formulas and user's entered formula, the cosine similarity is taken into the consideration. The cosine similarity is used to compare the similarity between the documents/text data, and based on that, it provides the ranking to the documents with respect to user's entered query [3]. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space [11]. In this work, the two vectors containing the embeddings of the formulas and formula-based user's entered query are compared. Let x and y be two vectors for comparison. Using the cosine measure as a similarity function,

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} = \frac{\sum_{i=0}^{n-1} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n-1}(x_i)^2} \times \sqrt{\sum_{i=1}^{n-1}(y_i)^2}} \tag{1}$$

where $\|x\|$ is the Euclidean norm of vector $x = (x_1, x_2, \ldots, x_n)$, defined as $x_1{}^2 + x_2{}^2 \,||\!+ \cdots + \mathrm{x}_n{}^2$. Conceptually, it is the length of the vector. Similarly, $\|y\|$ is the Euclidean norm of vector y. The measure computes the cosine of the angle between vectors x and y. A cosine value of 0 indicates that the two vectors have really no correlation with each other and hold the 90 °C angle. The nearer the cosine function to 1, the lower the angle and the higher the vector match. The cosine similarity is useful, because even though the two identical formulas are very distant due to their size from the Euclidean, they still have a closer angle. The angle is lower, the resemblance is higher.

Table 3 Retrieved search results for ten formula-based query

Formula query	Retrieved formula
$5^{133} \bmod 8$	$5^{5^{33}} \bmod 100, 5^{288} \bmod 577, \bmod 10^4, 5^{12} \not\equiv 1 \bmod 25, 11 \equiv 1 \bmod 5,$ $49^{307} \bmod 713, 3^{3^{27}} \bmod 40, 5^{350} \bmod 701,$ and $13^k \equiv 1 \pmod{10}$, $7^\ell \bmod 5$
$(1+\sqrt[i]{3})^{\frac{1}{2}}$	$-2(1+i\sqrt{3}), z_4 = -1/2(1+i\sqrt{3}), (1+i\sqrt{3})^n, (1+i\sqrt{3})/2,$ $\frac{1}{2}(1+i\sqrt{3}), (1+i\sqrt{3})^2 = -2+\sqrt{3}, \frac{1}{2}(-\sqrt{3}+i), (i+\sqrt{2})^4,$ $(i+\sqrt{2})^2, (1+i\sqrt{3})^8$
$n{=}n_1 n_2 \ldots n_{k+1}$	$X_1 + X_2 + \cdots + X_n = 1, Y_n = X_1 + X_2 + \cdots + X_n,$ $S_n = X_1 + \cdots + X_n, F_{n+1} = \ldots, X = X_1 + X_2 + \cdots + X_n,$ $Z = Y_1 + \cdots + Y_n, Y_n = X_1 + \cdots + X_n, Z = Y_1 + Y_2 + \cdots + Y_n,$ $1 + 2 + 3 + \cdots + n = F_n, a = 1 \Longrightarrow b = 2$
$\frac{1}{\sqrt{-1}} = \sqrt{-1}$	$= \frac{2}{4-\sqrt{12}}, = \frac{\sqrt{2-\sqrt{3}}}{2}, \psi = \frac{1-\sqrt{5}}{2}, = -\frac{1}{2}\frac{1}{\sqrt{100}}, = \frac{9-4\sqrt{3}}{3}, -\frac{1}{2\sqrt{2-2}} = -\frac{1}{0},$ $\frac{2\sqrt{2}-1}{4} = \frac{\sqrt{2}}{2} - \frac{1}{4}, \psi = \frac{1-\sqrt{5}}{2}, R = \frac{1}{\sqrt{-K}}, \cos 108 = \frac{1-\sqrt{5}}{4}$
$\lim_{x\to\infty} a_n$	$\lim_{n\to\infty} a_n \neq 0, \lim_{n\to\infty}(-1)^n, \lim_{n\to\infty} a_n \leq x, \lim\limits_{n\to\infty} x_n = a,$ $\lim_{n\to\infty} b_n = 1, \lim_{x\to\infty} a_n, \lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = 0, \lim\limits_{x\to\infty} a_n,$ $\lim_{x\to\infty} b_n = b, lim_{n\to\infty} a_n = x$
$f([a,b]){=}c(a,b)$	$f(A) \subset f([a,b]), [a,b] \subset [f(a), f(b)], [a,b] \cap [f(a), f(b)] = \varnothing,$ $f(X) \subset [a,b], [f(b), f(a)] \subset f([a,b]), f([a,b]), f(c) \in f([a,b]),$ $f : [a,b] \to [a,b]$
$f(x) = x + \frac{1}{x}$	$f(x) = x + \frac{16}{x}, f(x) = \frac{x}{1+x}, f(x) = \frac{1}{1+\cos(x)}, f(x) = 1 + \frac{1}{x},$ $f(x) = \frac{x+2}{x}, g(x) = \frac{2}{x-1}, f(x) = 1 + \frac{1}{x}, f(x) = \frac{2}{3}, f(x) = \frac{\sec\ x}{1+\tan\ x},$ $f(x+1) = \frac{x+1}{x+3}$
$(-1)(-1){=}1$	$(4-1)(3-1) = 2, 18 \cdot (69-18) = 3 \mod 61,$ $6! - (6! - 4!3!) = 4!3! = 144, = 1 - (1 - \Phi(0.15416023)),$ $\ln(2) = \ln(1-(-1)), 11(85-84) = 11, 15 = 3 \cdot 5 \cdot (-1) \cdot (-1),$ $P(-1) = 0, 25 - 2(8) = 9, f(x) = -x + 2, [-1, 1]$
$p_n{=}\frac{1}{2}P_{n+1}$	$\frac{n-n_0}{p} = \left\lfloor \frac{n}{p} \right\rfloor, \left\lfloor \frac{n}{p} \right\rfloor = \frac{n-n_0}{p}, p_n = \frac{2}{n} - 1, p_n = \frac{500-n}{500} = 1 - \frac{n}{500},$ $p_2 = \frac{1}{n-1}, p_0 = 1 - \frac{\alpha p}{1-p}, p_2 = 1 - p_1 = \dfrac{n-1}{n}, p_{2n-1} = \frac{1}{2n-1},$ $p_n = \frac{5}{6}p_n - \frac{1}{6}p_{n-2}, p_3 = \frac{n-2}{n-1}\frac{n-3}{n-1}\frac{1}{n-1}$
$a^2 + b^2 = c^2 + d^2$	$a^3 + b^3 + c^3 = d^3, a^2 + ab + b^2 = c^2 + cd + d^2, a^2 + b^2 = c^2,$ $a^2 + b^2 + c^2 + d^2 = 3, a^2 + b^2 + c^2 + d^2 = 12, a^2 + b^2 + c^2 = d^2,$ $a^2 + b^2 = c^2 + d^2, ab^2 + 1 = c^2 + d^2, a^2 + b^2 + c^2 + d^2 = 1,$ $c^2 + 7d^2 = a^2 + 7b^2$

5 Experimental Results

We applied the formula2vec approach on the complete MSE corpus of ARQMath-2020 dataset. For experiment, we have set following parameters[3]: sg = 0 (using the continuous bag-of-words model), hs = 0 (no hierarchical softmax), negative = 10

[3] https://radimrehurek.com/gensim/models/word2vec.html#gensim.models.word2vec.Word2Vec.

(using negative sampling), size = 128 (the dimensionality of the vectors), min-count = 1 (ignore formula or entity tokens with total frequency lower than 10), window = 3 (the maximum distance between the current and predicted formula).

To measure the embedding accuracy based on query such as which formula is the most similar to $\sqrt{x}$ the same sense (permutation of k objects chosen from n distinct objects) as P_k^n is similar to ${}_nP_k$, i.e., the semantic-syntactic word relationship testor, simply, formula analogy test. The preliminary experiment results of "formula2vec" revealed the several characteristics of mathematical formulae, which indicated that the natural language embedding technologies are potentially useful for formula embedding task. The results for the ten formula-based query is shown in Table 3. The natural language embedding approach has ability to handle the mathematical representation and preserve their syntactic meaning.

6 Conclusion and Future Scope

In this paper, we have investigated the formula2vec model which learned the distributed representation of the formula. The neural network representation techniques have the ability to provide promising representational structure for mathematical formulas. The obtained search results have shown that the formula2vec model is able to preserve the meaning of the mathematical formulas.

The several semantic representations of the mathematical formulas contribute to the frailty of most of the existing systems. So in the future, the recursive neural network will adopt to effectively capture ambiguity of several semantic representations of the formula.

Acknowledgements The authors would like to express their gratitude to the Department of Computer Science and Engineering and Center for Natural Language Processing, National Institute of Technology Silchar, India, for providing the infrastructural facilities and support.

References

1. Berčič K, Kohlhase M, Rabe F (2020) Towards a heterogeneous query language for mathematical knowledge. In: International conference on intelligent computer mathematics. Springer, Berlin, pp 39–54
2. Carette J, Farmer WM, Kohlhase M, Rabe F (2019) Big math and the one-brain barrier a position paper and architecture proposal. arXiv preprint arXiv:1904.10405, pp 1–17
3. Dadure P, Pakray P, Bandyopadhyay S (2019) An empirical analysis on retrieval of math information from the scientific documents. In: International conference on communication and intelligent systems. Springer, Berlin, pp 301–308
4. Dadure P, Pakray P, Bandyopadhyay S (2020) An analysis of variable-size vector based approach for formula searching. In: Working notes of CLEF 2020—conference and labs of the evaluation forum, pp 1–13

5. Ferreira D, Freitas A (2020) Natural language premise selection: finding supporting statements for mathematical text. In: Proceedings of the 12th language resources and evaluation conference, pp 2175–2182
6. Gao L, Jiang Z, Yin Y, Yuan K, Yan Z, Tang Z (2017) Preliminary exploration of formula embedding for mathematical information retrieval: can mathematical formulae be embedded like a natural language? arXiv preprint arXiv:1707.05154, pp 1–4
7. Goldberg Y, Levy O (2014) Word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722
8. Greiner-Petter A, Schubotz M, Müller F, Breitinger C, Cohl H, Aizawa A, Gipp B (2020) Discovering mathematical objects of interest—a study of mathematical notations. In: Proceedings of the web conference 2020, pp 1445–1456
9. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
10. Pathak A, Pakray P, Das R (2019) Lstm neural network based math information retrieval. In: 2019 second international conference on advanced computational and communication paradigms (ICACCP). IEEE, pp 1–6
11. Rahutomo F, Kitasuka T, Aritsugi M (2012) Semantic cosine similarity. In: The 7th international student conference on advanced science and technology ICAST, vol 4
12. Scharpf P, Mackerracher I, Schubotz M, Beel J, Breitinger C, Gipp B (2019) Annomathtex-a formula identifier annotation recommender system for stem documents. In: Proceedings of the 13th ACM conference on recommender systems, pp 532–533
13. Stathopoulos Y, Baker S, Rei M, Teufel S (2018) Variable typing: assigning meaning to variables in mathematical text. In: Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies, vol 1 (Long Papers), pp 303–312
14. Yasunaga M, Lafferty JD (2019) Topiceq: a joint topic and mathematical equation model for scientific texts. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 7394–7401
15. Zanibbi R, Oard DW, Agarwal A, Mansouri B (2020) Overview of arqmath 2020: clef lab on answer retrieval for questions on math. In: International conference of the cross-language evaluation forum for European languages. Springer, Berlin, pp 169–193

Analysis of Cryptocurrency Mining in Gaming Consoles

E. Shanmuga Skandh Vinayak, N. Bhalaji, and Xiao-Zhi Gao

Abstract Ever since the invention of the bitcoin cryptocurrency in the year 2009, many blockchain ledgers have been initiated by several parties, offering over 1600 different types of cryptocurrencies all around the world by them. This increase in popularity has not only moved the masses toward performing trade using these currencies but also start earning them through mining with sophisticated computing hardware. In this article, an analysis is performed to estimate the performance and the profitability of mainstream gaming consoles, developed by the Sony Entertainment Company and Microsoft Corporation when subjected to mine five different types of cryptocurrencies (Bitcoin, Bitcoin Cash, Dogecoin, Litecoin, Dash). The article also provides real-time test results on the mining performance and profitability of the consoles under study. Although the Microsoft Corporation consoles can provide a monthly profit of 0.00096 and 0.00319% of the initial investment, an overall net loss of over 80% of the initial investment is observed in all the consoles.

Keywords Analysis · Cryptocurrency · Mining · Gaming consoles · Performance · Profitability

1 Introduction

The Bitcoin cryptocurrency has been increasingly gaining popularity since its initial release on January 3, 2009. Due to the immense hike in face value from ₹ 43,247.83 to ₹ 360,055.02 of Bitcoin by the year 2016 [1], the popularity peaked subsequently

E. Shanmuga Skandh Vinayak · N. Bhalaji (✉)
Department of Information Technology, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Chennai, Tamil Nadu, India
e-mail: bhalajin@ssn.edu.in

E. Shanmuga Skandh Vinayak
e-mail: shanmugaskandhvinayak16095@it.ssn.edu.in

X.-Z. Gao
School of Computing, University of Eastern Finland, Kuopio, Finland
e-mail: xiao-zhi.gao@uef.fi

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_28

and has been growing ever since. According to Statista, Bitcoin is the most valued cryptocurrency with a face value of ₹ 731,424 in the year 2020. This gain in popularity ignited trade to be carried out using this newly accepted currency as a secure source of payment between two organizations over the Internet, without any third-party liability. The population that performs trades using the cryptocurrency extended the use of the same by also starting to earn them. This is done using cryptocurrency mining. Due to the immense success of Bitcoin in its trade and mining pools, many independent and organizational parties have started to introduce their cryptocurrency by maintaining a public ledger, with over 1600 different types of cryptocurrencies being used worldwide. With such an increase in the number of cryptocurrencies exchange available, the miners have started to mine them all over the world by combining their computing capacity to form mining pools. Out of these mined currencies, the Litecoin and the Bitcoin are mined in a majority in the Asia-Pacific subcontinents totalling up to 52 and 44% of the total mining pool population by the year 2018, according to Statista [2]. By the year 2020, the size of the Bitcoin blockchain has grown to a size of 280 gigabytes [3] and a length of 535 million [4]. These observations show that mining cryptocurrency can be highly profitable with the right investment in suitable hardware. Out of the hardware considered for mining, the least sought out by the users is the current mainstream gaming consoles due to their closed systems.

The gaming sector has been an ever-expanding and everimproving industry since the early 1960s. With exceptional significance in the improvement of hardware used in the gaming consoles, the gaming systems have improved in performance and efficiency. With the global gaming console market generating a revenue of ₹ 596,430 crores in the year 2020 [5], the popularity in the use of gaming consoles is at its peak. Hence, in this article, an analysis is performed to examine the viability of gaming consoles in mining operations and to examine whether they are capable of producing a profit, as the major use of these consoles by the users is only gaming and home entertainment.

2 Cryptocurrency Mining

Cryptocurrency mining is the process of obtaining cryptocurrency incentives for the successful validation of any transaction occurring in the target network. The network consists of a digital ledger known as the blockchain, that stores these transactions once they are successfully validated by the peers present in them. The validation operation refers to the calculation of each transaction's fingerprint (hash) and checking if it matches with the target fingerprint. The digital transaction records and the hash of the previous block are hashed along with a number only used once (nonce) and are propagated in the network to the users to be validated. The operation of determining the nonce and hashing them redundantly to obtain the hash that matches the target hash is known as mining. If a user is successful in determining the nonce before any other participant, then that user is incentivized with a certain amount of cryptocurrency based on the amount of work done by the peer. Based on the difficulty, i.e., the difficulty to solve for the hash, the incentive amount of the currency varies. Because the solving of the hash algorithm is a tedious and repetitive process, computing power is utilized. Based on the user's hardware capability, the speed with which the hashes are calculated varies. The speed with which the hardware component can calculate these hashes is called the hash rate; i.e., higher the hash rate, better is the performance and profit. Since a stand-alone system may not be able to provide the hashing power to solve a single block, peers in a network form a combined unit of computation to solve each block. This is known as a mining pool. The incentive to each user is assigned based on the amount of computational power contributed to solving the block.

3 Related Works

James Clay et al. in their work "A Power Analysis of Cryptocurrency Mining: A Mobile Device Perspective" [8] analyze the impact of cryptocurrency mining systems on the power consumption of mobile devices. In their work, the authors analyze the ill effects of JavaScript-based web browser mining of websites such as CoinHive on mobile devices. The authors analyze the power consumption and the network usage of mobile devices, that is utilized by CoinHive to mine their network as an alternative for targeted advertisement. The authors show the impact of these mining operations on a low-powered device such as a mobile phone and how it can ultimately decrease its battery life. Similar to the approach of the author's work, this proposed experiment focuses on measuring the profitability of the mining operation in low-powered devices (locked gaming consoles that do not use extensive hashing peripherals to mine) and observing the impact of the performances of the native computation unit in solving block hashes. In the article "Goldstrike™1: CoinTerra's First Generation Cryptocurrency Mining Processor for Bitcoin" [9], the authors Javed Barkatullah et al. analyze the performance and efficiency of the newly designed Goldstrike™1 processor for Bitcoin mining. The processor is analyzed based on the power consumption efficiency during the mining process and its hash rate for a unit of power consumed. The architecture of the processor is studied to analyze and compare the hashing power and the power efficiency of the processor to application-specific integrated circuit (ASIC) miners. Similar to the mentioned work, this article analyzes the architecture of the low-powered processor in mainstream gaming consoles for performance and profitability. Authors S. G. Iyer et al. in their study "GPU and CPU Accelerated Mining of Cryptocurrencies and their Financial Analysis" [10] analyze and compare the performance and the profitability of central processing unit (CPU)-based and graphics processing unit (GPU)-based mining. In their study, it is evident that the hash rate of the CPU or the GPU is majorly dependent on the hashing algorithm of the cryptocurrency mined. Their work also reveals that GPU the is dominant option in a mining system, as it is able to produce 103 times the hash rate of the CPU for the Ethereum cryptocurrency. Since their study was performed on Intel® CPU processor, this proposing study is aimed at analyzing the performance and the profitability of the Advanced Micro Devices, Inc. (AMD®) CPUs.

4 Experimental Analysis

In this article, we assess the usability of four mainstream gaming consoles, in mining cryptocurrency. The four consoles are, the PlayStation 4™, released on November 15, 2013 (USA) and the PlayStation 4 Pro™, released on November 10, 2016, developed by the Sony Computer Entertainment and the Xbox One™, released on November 22, 2013 (USA) and the Xbox One X™, released on November 7, 2017, developed by the Microsoft Corporation.

These consoles are selected based on the popularity of use in the overall gaming industry and their possession of the most sophisticated hardware in the console industry so far. According to Statista, over 112 million PlayStation 4 units and over 46 million Xbox One units have been installed and used all over the world, in the year 2019 [6].

The ultimate goal of estimating whether a gaming console can be used to mine cryptocurrencies and produce a profit (return on investment) is based on the following factors.

1. Price and Specifications.
2. Cryptocurrency to be mined.
3. Hash Rate.
4. Power Consumption.
5. Lifespan.

4.1 *Price and Specifications*

The price and the specifications of the console are one of the most preliminary aspects to be considered before selecting a console, as it plays an important role in determining the profit that the console produces on the invested capital. Although selecting the cheapest console appears to be the most prominent option, selecting the console based on the specifications that would comparatively produce the highest profit in a short time duration is essential. Tables 1, 2 and 3 describe the price (at the time of writing this article) in rupees (₹) and the specification of the consoles under study.

Table 1 Console price

Console	Price (₹)
PlayStation 4	25,555
PlayStation 4 Pro	31,500
Xbox One	25,075
Xbox One X	41,999

Table 2 Console specifications

Console	CPU	Clock rate (GHz)	Memory (RAM)
PlayStation 4	x86-64 AMD Jaguar (8 cores)	1.6	8 (DDR5)
PlayStation 4 Pro	x86-64 AMD Jaguar (8 cores)	2.13	8 (DDR5)
Xbox One	x86-64 Graphics Core Next (GCN) 2 AMD Jaguar (8 cores)	1.75	8 (DDR3)
Xbox One X	x86-64 GCN 4 AMD Jaguar (8 cores)	2.3	12 (DDR5)

Table 3 Console properties

Console	Power consumption (W)	Max. temperature (°C)
PlayStation 4	165	5–35
PlayStation 4 Pro	310	5–35
Xbox One	120	15.5–48.9
Xbox One X	180	15.5–62.0

The specifications of the console that only aid in the cryptocurrency mining process are analyzed and specified in the following tables and sections; i.e., the internal memory, DVD drive, and I/O port specifications of the consoles are not analyzed.

One of the most significant components that enable profitable and worthwhile mining operations in a non-ASIC mining rig is the GPU. Although the GPU in the gaming consoles in this study has significantly improved when compared to its predecessor consoles (PlayStation 3 and Xbox 360), such as the use of DirectX-12 graphics in the Xbox One X consoles, unlike the predecessor consoles, these gaming consoles are locked by their respective manufacturing companies and cannot be hacked or modified to utilize the GPU directly. This prevents the GPU from being used directly by a non-licensed third-party application. Although the PlayStation consoles possess the ability to install homebrew applications in its platform, no third-party mining application that can harness the GPU is available at the time of writing this article. This not only forces the console to an application less mining environment but also performs CPU mining. CPU mining is proven to be significantly lower in efficiency and profit when compared to GPU mining. The CPU core is capable of executing 4 32-bit instructions per clock (using a 128-bit SSE instruction) or 8 via Advanced Vector Extensions (AVX) (256-Bit), whereas a GPU like Radeon HD 5970 is capable of executing 3200 32-bit instructions per clock (using 3200 ALUs). The difference in their architectures allows us to understand that, although a multi-core CPU has a higher clock-rate allocated for each core, the GPU is superior in efficiency when performing redundant operations such as mining. Although the PlayStation 4 and PlayStation 4 Pro possess 1.84 and 4.20 Tera Floating Points (TFLOPS) AMD Radeon™GPUs, respectively, and the Xbox One and the Xbox One X possess 1.31 and 6.0 TFLOPS, respectively; they are unavailable for the utilization in these non-licensed mining operations.

Out of the configurations considered, the RAM specification can be considered as the least significant parameter in the analysis of mining performance. This is because all the consoles under study possess 8 times (12 for Xbox One X) the RAM requirement for any cryptocurrency to be mined. But this is also analyzed in this article, as the Operating System (OS) of the console depends on the RAM to optimize kernel operations that are performed along with the multi-threaded mining operations. The Orbis OS v7.51 (at the time of writing this article) of PlayStation 4 and the PlayStation 4 Pro is a Unix-based FreeBSD operating system that is highly capable of internal RAM optimization. This shows that the console may not freeze or crash due to a shortage of memory if the mining operation succeeds. But unlike the PlayStation consoles, the Xbox One and Xbox One X consoles possess a Windows 10, initially Windows 8 core as its operating system along with a modified version of the Hyper-Viridian hypervisor. This utilization of the hypervisor in the Xbox One consoles allows the consoles to be superior to the PlayStation consoles during the mining operation. Hyper-V implements and isolates virtual machine partition (logical unit of isolation, supported by the hypervisor, in which each guest operating system executes) that it creates in x86-64 Windows machines. The Hyper-V is capable of hardware accelerating the address translation of Guest Virtual Address-spaces, by using second-level address translation provided by the CPU (EPT), on AMD. The mining interface utilizes the child partition to perform a hypervisor-based hashing operation. The Hyper-V utilizes the simultaneous multithreading (SMT) technique that allows the processor's resources to be shared by separate, independent threads. SMT generally offers a modest performance boost to most workloads by parallelizing computations when possible, thereby increasing instruction throughput. This allows the Xbox One and Xbox One X consoles to perform multi-threaded mining, bottlenecked by the hashing capacity of the CPU.

4.2 Cryptocurrency

The cryptocurrency is the reward incentivized to the miner for verifying the transaction. Although an exchange face value is available for each cryptocurrency, with which government-regulated money can be obtained, the difficulty to obtain a substantial amount of cryptocurrency varies within each cryptocurrency. For this experiment, the console is subjected to mine five cryptocurrencies. The difficulty and the price of each cryptocurrency (at the time of writing this article) are given in Tables 4 and 5.

The difficulty rates are dependent on the consensus protocol the cryptocurrencies follow and the volume of participants of the network. Depending on the consensus rules and the size of the network, the difficulty range of the nonce varies, which is directly proportional to the amount and the value of the incentive obtained from the mining operation. The miners are rewarded with the cryptocurrency obtained from the fee the transaction initiators pay to the network (Table 6).

Table 4 Cryptocurrency difficulties

Cryptocurrency	Difficulty
Bitcoin core	13,732,352,106,018.00
Bitcoin cash	363,666,388,375.33
Doge coin	2,644,822.07
Lite coin	8,671,505.36
Dash	156,499,539.48

Table 5 Cryptocurrency prices

Cryptocurrency	Price (₹)
Bitcoin core	731,424
Bitcoin cash	19,377.89
Doge coin	0.19
Lite coin	3547.17
Dash	5930.85

Table 6 Cryptocurrency consensus protocol

Cryptocurrency	Consensus protocol
Bitcoin core	Proof-of-work
Bitcoin cash	Proof-of-work
Doge coin	Proof-of-work
Lite coin	Proof-of-work
Dash	Proof-of-work

All the cryptocurrency mined for the purpose of this experiment follows the proof-of-work consensus protocol, that awards the participants of the network, based on the amount of computational resources contributed in solving the block hash. The architecture that follows describes the proof-of-work consensus and the pipeline of block verification in the network (Fig. 1).

4.3 Hash Rate

The hash rate of a mining rig is the most important aspect to be considered in calculating the profitability of the mining operation using the system under study. Any mining hardware is classified based on the rate at which hashes are calculated by them. The unit with which the hash rate is measured in hashes/second (H/s). The higher order of the unit is extended as kilo (10^3) hashes/second (kH/s), mega (10^6) hashes/second (MH/s), giga (10^9) hashes/second (GH/s), tera (10^{12}) hashes/second (TH/s), etc.

In this section, the expected and the actual hash rate of the consoles based on their CPU architecture is analyzed. Since the hash rate of the AMD x86-64 Jaguar processors is unavailable at the time of writing this article, the architecture of the CPU can be taken as a base consideration to be compared with the architecture of similar AMD CPUs to bring about an approximate hash rate for the consoles. But, unlike the other AMD CPUs available, the sole purpose of the Jaguar CPU is to provide low

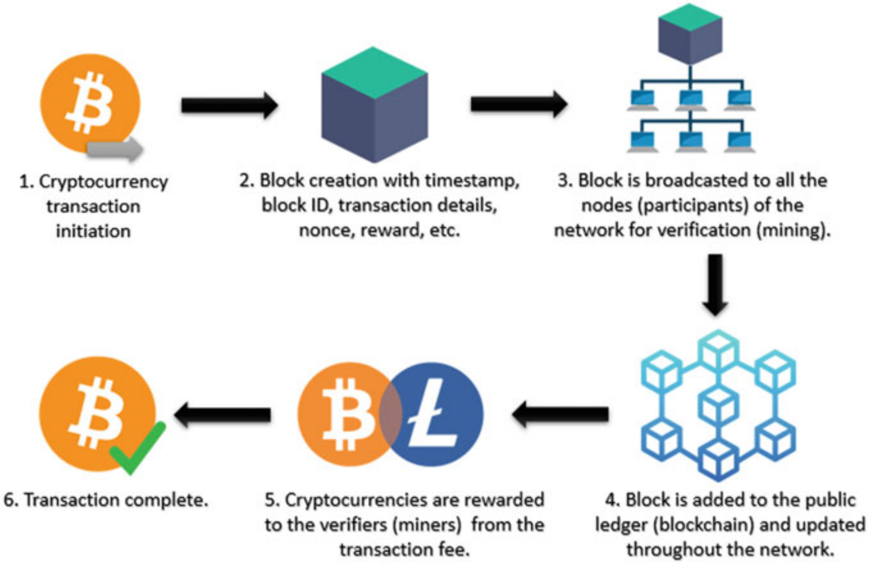

1. Cryptocurrency transaction initiation
2. Block creation with timestamp, block ID, transaction details, nonce, reward, etc.
3. Block is broadcasted to all the nodes (participants) of the network for verification (mining).
4. Block is added to the public ledger (blockchain) and updated throughout the network.
5. Cryptocurrencies are rewarded to the verifiers (miners) from the transaction fee.
6. Transaction complete.

Fig. 1 Proof-of-work architecture

power accelerated processing unit (APU) for the I/O devices. The cat-line of the AMD processor that has AMD's Bobcat APU as the predecessor of the Jaguar APU, which exhibits 22% fewer instructions per cycle (IPC), like the Jaguar APU, it also does not possess a hash rate benchmark due to low power operation expected by it. One of the processors closest in comparison is the x86-64 Atom processor developed by Intel Corporation. The benchmark hash rate of Intel's Atom is 146.72 H/s for the XMR - RandomX (XMRig) algorithm. Although a direct comparison can be considered, Intel's Atom is superior in terms of performance due to the CPU configured to perform various tasks in a fully-fledged computer, whereas the AMD Jaguar in the gaming consoles is required to only render video and synchronize I/O operations. Hence, the AMD Jaguar of the PlayStation 4 and Xbox One is expected to perform significantly lower, with a much lower hash rate for the same algorithm. However, with an increase in performance of the AMD Jaguar APUs in the PlayStation 4 Pro and the Xbox One X consoles, the computing capacity of the video render is compared to the Core i5-3570K, developed by Intel corporation which has a base clock rate of 3.40 GHz boosted to 3.80 GHz. Although they are benchmarked to have similar performances in video rendering capability, they can have highly different performances in mining with the AMD Jaguar being much inferior. The hash rate of the Core i5-3570K processor is benchmarked as 1319.65 H/s for the XMRig algorithm. Similarly, the expected hash rate and performance of the PlayStation 4 Pro and the Xbox One X are expected to be significantly lower. For this experiment, the hash rate of the gaming consoles is observed by mining each of the five cryptocurrencies for a duration of 24 h. Unfortunately, the PlayStation 4 was unable to start the mining operation for any of the cryptocurrency. Hence, the average/net hash and income of the PlayStation 4 are considered as 0. Similarly, the PlayStation 4 Pro was unable to mine any cryptocurrency blockchain other than Dash. Figures 2, 3, and 4 show the observed hash rates for a 24 h duration in each console.

4.4 Power Consumption

The only factor that determines whether the net mining operation would result in a profit or a loss is the power consumption cost of the consoles. The mining operation would result in a profit if the power consumption cost is less than the amount of equity earned in cryptocurrency, else it would result in a loss. Referring to Table 3, the power consumption values for each console are considered and the net consumption cost is calculated in this section. Table 7 shows the power consumption cost for each console, considering the user is under a domestic bimonthly tariff electricity billing system.

This cost comparison is further analyzed in the results section by determining whether a return on investment is possible with these gaming console with a consideration of the calculated life span.

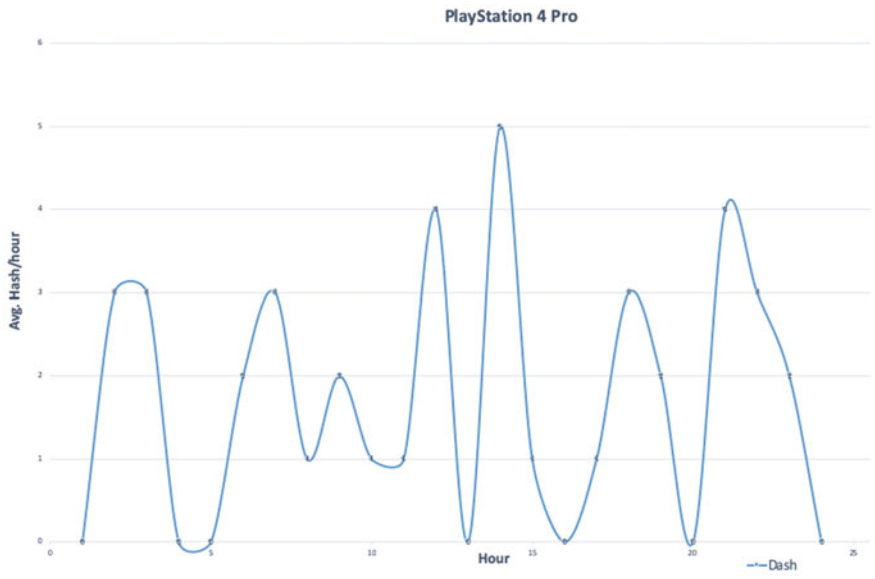


Fig. 2 PlayStation 4 Pro hash rate

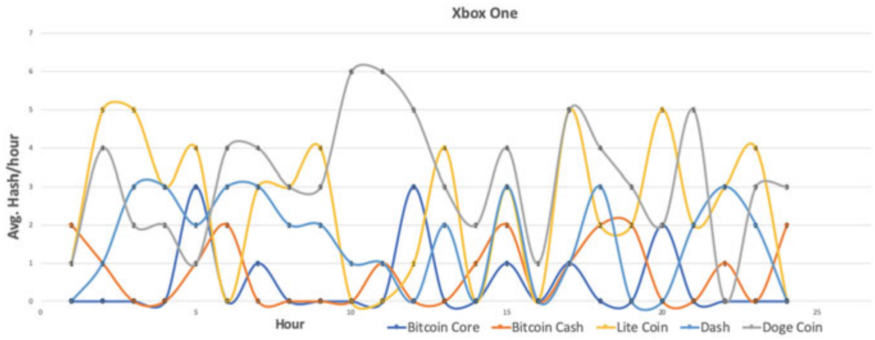


Fig. 3 Xbox One hash rate

4.5 *Life Span*

In this section, the functional working life span of the gaming consoles under study is analyzed and estimated based on the user reviews. This data is analyzed to estimate if the consoles can turn a profit before becoming faulty. Considering the average working temperature of the consoles, Table 8 shows the life span (prime working condition) of each console.

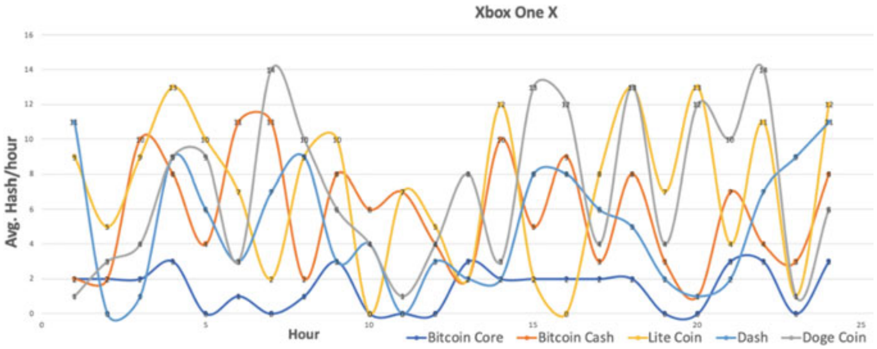


Fig. 4 Xbox One X hash rate

Table 7 Power consumption cost

Console	Power consumed/month (kWh)	Cost/month (₹)	Cost/year (₹)
PlayStation 4	118.8	51.0	18,615
PlayStation 4 Pro	223.2	291.0	1,06,215
Xbox One	86.4	0.0	0.0
Xbox One X	129.6	66.0	24,090

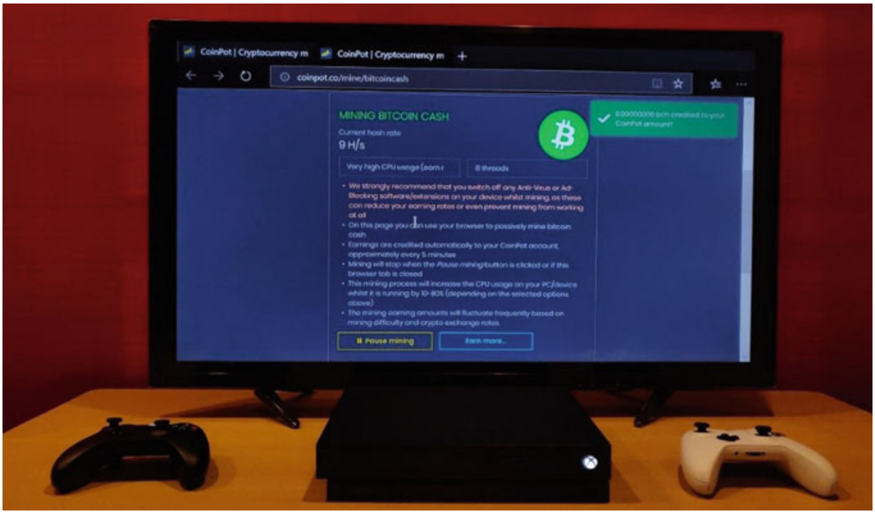
Table 8 Console life span

Console	Avg. life span
PlayStation 4	7
PlayStation 4 Pro	9 (considered)
Xbox One	5
Xbox One X	10 (considered)

Although these data are from users who primarily use the consoles for gaming, an estimate of half the duration is considered as the life span of the consoles when mining.

5 Results

Neither of the PlayStation consoles nor the Xbox consoles supports third-party cryptocurrency mining applications. To tackle this, a concept known as web mining is used. Web mining is a technology that utilizes a JavaScript-based interface between the host and the website to exchange mining information. Instead of the conven-


CoinPot | Cryptocurrency m
CoinPot | Cryptocurrency m
coinpot.co/mine/bitcoincash

MINING BITCOIN CASH

Current hash rate

9 H/s

Very high CPU usage (com·r

8 threads

- We strongly recommend that you switch off any Anti-Virus or Ad-Blocking software/extensions on your device whilst mining, as these can reduce your earning rates or even prevent mining from working at all
- On this page you can use your browser to passively mine bitcoin cash
- Earnings are credited automatically to your CoinPot account, approximately every 5 minutes
- Mining will stop when the *Pause mining* button is clicked or if this browser tab is closed
- This mining process will increase the CPU usage on your PC/device whilst it is running by 10-80% (depending on the selected options above)
- The mining earning amounts will fluctuate frequently based on mining difficulty and crypto exchange rates

0.00000006 bch credited to your CoinPot amount!

Fig. 5 Bitcoin Cash mining in Xbox One X

tional application-based mining that involved the maintenance of the cryptocurrency blockchain in the local system, the website only uses the hashing power of the host system to solve the blockchain and award the user's account with cryptocurrency based on the work done by the console. All the consoles under study possess native web browsers and support Wi-Fi standard IEEE 802.11 a/b/g/n/ac and an Ethernet speed of at least 100 Mbps. These requirements are sufficient to perform browser mining, as the block size is only 1 MB. Since web mining only allows the consoles to mine as a stand-alone system, the profitability of pool mining cannot be expected from this solution.

The website CoinPot [7] is used to mine cryptocurrency using the CPU of the host system. The website provides a user interface to configure the parameters of the mining such as CPU usage limit, thread limit. The website configurations are set to utilize high CPU usage and 1 thread per CPU core (Fig. 5). As the cryptocurrencies are mined, the obtained rewards are subsequently deposited to the account's crypto wallet.

The following tables show the summary of expected profit/loss from the consoles when mined using them. All the income calculations are based on the currency values mentioned in Table 5. Since these values are subjected to change, the following calculations provide an approximate estimation.

5.1 PlayStation 4

Life span expectancy—**3.5 years**. The Profit/Loss estimation of PlayStation 4 is elaborated in Table 9.

5.2 PlayStation 4 Pro

Life span expectancy—**4.5 years**. The Profit/Loss estimation of PlayStation 4 Pro is elaborated in Table 10.

5.3 Xbox One

Life span expectancy—**2.5 years**. The Profit/Loss estimation of Xbox One is elaborated in Table 11.

Table 9 PlayStation 4 profit/loss estimation

Crypto currency	Monthly income (₹)	Monthly profit (₹)	Monthly outcome (profit/loss)	Net profit (₹)	Net outcome (profit/loss)
Bitcoin core	0	0	NA	−25,555	Loss
Bitcoin cash	0	0	NA	−25,555	Loss
Doge coin	0	0	NA	−25,555	Loss
Lite coin	0	0	NA	−25,555	Loss
Dash	0	0	NA	−25,555	Loss

Monthly profit = monthly income − power consumption amount
Net profit = console price − income gained in the life span

Table 10 PlayStation 4 Pro profit/loss estimation

Crypto currency	Monthly income (₹)	Monthly profit (₹)	Monthly outcome (profit/loss)	Net profit (₹)	Net outcome (profit/loss)
Bitcoin core	0	0	NA	−31,500	Loss
Bitcoin cash	0	0	NA	−31,500	Loss
Doge coin	0	0	NA	−31,500	Loss
Lite coin	0	0	NA	−31,500	Loss
Dash	29.9	−261.100	Loss	−45,596	Loss

Monthly profit = monthly income − power consumption amount
Net profit = console price − income gained in the life span

Table 11 Xbox One profit/loss estimation

Crypto currency	Monthly income (₹)	Monthly profit (₹)	Monthly outcome (profit/loss)	Net profit (₹)	Net outcome (profit/loss)
Bitcoin core	0.0087	0.0087	Profit	−25,075	Loss
Bitcoin cash	1.9377	1.9377	Profit	−25,070	Loss
Doge coin	0.5254	0.5254	Profit	−25,060	Loss
Lite coin	20.0365	20.0365	Profit	−24,473	Loss
Dash	0.3903	0.3903	Profit	−25,063	Loss

Monthly profit = monthly income − power consumption amount
Net profit = console price − income gained in the life span

5.4 Xbox One X

Life span expectancy—**5 years**. The Profit/Loss estimation of Xbox One X is elaborated in Table 12.

From these observations, it can be seen that the Xbox One and the Xbox One X (for Lite Coin) console can provide a monthly profit. But when considering the life span expectancy of the consoles, all the gaming consoles provide only a loss to the investment. These gaming consoles are expected to only perform in their area of expertise, gaming. These losses can also be accounted for by their closed system-on-chips (SOCs) that only allow the utilization of the CPU rather than a combination of their CPU and GPU.

6 Conclusion

In this article, the performance of mainstream gaming consoles is tested and analyzed for cryptocurrency blockchain mining. The article also provides an analysis of the

Table 12 Xbox One X profit/loss estimation

Crypto currency	Monthly income (₹)	Monthly profit (₹)	Monthly outcome (profit/loss)	Net profit (₹)	Net outcome (profit/loss)
Bitcoin core	1.4262	−64.5371	Loss	−45,839	Loss
Bitcoin cash	5.1902	−60.8098	Loss	−45,648	Loss
Doge coin	2.0234	−63.9765	Loss	−45,837	Loss
Lite coin	200.100	134.1484	Profit	−33,950	Loss
Dash	2.1646	−63.8353	Loss	−45,829	Loss

Monthly profit = monthly income − power consumption amount
Net profit = console price − income gained in the life span

approximate estimate of profit that can be expected from mining using the consoles. From the conducted experiment, the following conclusions can be drawn.

1. The gaming consoles are extremely poor performers when utilized to mine cryptocurrency.
2. The gaming console will ultimately produce a loss of the initial invested capital by a large margin.
3. This low performance and efficiency are characterized by the inability of the gaming consoles to utilize the GPU present in them to mine effectively.
4. The PlayStation 4 and the PlayStation 4 Pro consoles produce a loss of the initial as well as the periodic (electricity cost) investment.
5. If the initial investment is not considered, then the Xbox consoles produce a small margin monthly profit.
6. The Xbox One console produces its highest profit of approximately ₹ 20 every month and the Xbox One X produces its only and the highest profit of approximately ₹ 134.

7 Limitations

Although this article proposes an approximate estimate of the profits and losses, they are highly susceptible to change. This is because cryptocurrency exchange rates are highly correlated with the popularity among the masses that mine the particular cryptocurrency. This could result in the investment of time and money on the mining operation for a particular currency, only to produce a futile result with a minimal amount of low-value exchange rate. Another assumption made in this article is that the consoles are to perform mining operation every hour of every day. This could contradict the practicality of carrying out such an operation, considering external influencing factors such as electricity outage, overheating, system crash, and defective console.

8 Future Works

At the time of writing this article, the Sony Entertainment Company and Microsoft Corporation have already confirmed the future release of their next-generation gaming consoles, named PlayStation 5 and Xbox Series X. These consoles are expected to have extremely high gaming performances. Although they would be undoubtedly futile cryptocurrency miners, it is expected that they would perform better than the current consoles. Even though the Xbox consoles cannot be fully used to bring about better mining performance, the existence of homebrew in the PlayStation platform could be utilized in developing software that can take advantage of the highly sophisticated GPU present in them.

References

1. Statista Bitcoin—Statistics & Facts. https://www.statista.com/topics/2308/bitcoin/. Accessed 3 Mar 2020
2. Statista distribution of cryptocurrency mining pools worldwide 2018, by region. https://www.statista.com/statistics/731449/geographical-distribution-of-mining-pools/. Accessed 15 Mar 2020
3. blockchain.com Blockchain size. https://www.blockchain.com/charts/blocks-size. Accessed 1 Apr 2020
4. blockchain.com Total number of transactions. https://www.blockchain.com/charts/n-transactions-total. Accessed 1 Apr 2020
5. Statista video game consoles. https://www.statista.com/outlook/14070000/100/video-game-consoles/worldwide. Accessed 10 Mar 2020
6. Statista installed base of Sony and Microsoft game consoles worldwide from 2012 to 2019. https://www.statista.com/statistics/697187/installed-base-of-sony-and-microsoft-game-consoles/. Accessed 11 Mar 2020
7. CoinPot homepage. https://coinpot.co. Accessed 7 May 2020
8. Clay J et al (2018) A power analysis of cryptocurrency mining: a mobile device perspective. In: The 16th annual conference on privacy, security, and trust (PST)
9. Barkatullah J et al (2015) Goldstrike 1: CoinTerra's first-generation cryptocurrency mining processor for bitcoin. IEEE Micro 35(2):68–76
10. Iyer SG et al (2018) GPU and CPU accelerated mining of cryptocurrencies and their financial analysis. In: The 2nd international conference on I-SMAC (IoT in social, mobile, analytics, and cloud) (I-SMAC), pp 599–604

Component Species Prediction of Birds with Song Spectrum Features Using Machine Learning

M. Shyamala Devi, P. Swathi, Ayesha Jahangir, A Ravindra Reddy, Mannem Prudhvinadh, and M. Naga Sai Tharun

Abstract The purpose of maintaining the ecological balance in the environment is much essential for the proper maintenance of the earth's properties. Birds are the wonderful world creation that directly connects with the atmosphere that performs conservation monitoring of the earth's features. To address this problem, this paper has the following contribution toward bird species prediction. Firstly, the bird's song numeric dataset with 16,626 rows and 172 feature components from the UCI machine repository is subjected to data preprocessing. Secondly, the raw dataset is applied to all the classifiers to analyze the performance metrics before and after feature scaling. Thirdly, the data is reduced by PCA reduction analysis with 100, 75, and 50 components and then fitted to all the classifiers to analyze the performance metrics before and after feature scaling. Fourthly, the data is reduced by LDA reduction analysis with 100, 75, and 50 components and then fitted to all the classifiers to analyze the performance metrics before and after feature scaling. Experimental results show that the Bagging classifier has achieved an accuracy of 98%, random forest with 96%, and logistic with 96% for the raw dataset. After reducing the raw dataset with PCA, the random forest classifier retains 96% accuracy for 100, 75, and 50 component reduction. Similarly, the LDA reduced the dataset with 100, 75, and 50 components and attained 96% accuracy for the logistic regression classifier.

Keywords Machine learning · Classifier · PCA · LDA · Accuracy

1 Introduction

The species of birds can be identified from the song spectrum details of the birds. The exact classification tools for the bird's species prediction still remain a challenging issue in addition to noise-robustness and scalability. Machine learning can be used toward the prediction of the bird's species along with dimensionality reduction to handle the high-dimensional data. Several projects are in progress for automatic

M. Shyamala Devi (✉) · P. Swathi · A. Jahangir · A. Ravindra Reddy · M. Prudhvinadh · M. Naga Sai Tharun
Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_29

bird species recognition based on bird vocalization. Bird sounds are presented and tested through various configurations and hyperparameters. The machine learning framework offers tools for translating annotations into datasets that can be used to train a computer to identify a bird's species.

2 Literature Review

Automatic bird species recognition is done by using the bird's sound that performs feature extraction from a signal. The 13-dimensional vector representing the given frame is calculated for each frame. The interspecific success of 81.2% has been reached [1]. To properly differentiate and compare the RGB spectrograms and grayscale spectrograms, the lower layers should be trained on grayscale images [2]. The machine learning model was inconsistent in identifying classes that were closely related to each other due to close resemblance in their feature list like pigeon and dove [3]. The use of short audio segments with high amplitude outperforms the use of the complete audio records in the species identification task [4]. Textures of syllables in audio spectrograms have noticeable discerning capabilities among bird species and these are used for bird species classification [5, 6]. Classifiers are used to automate the classification of reptiles and other water species [7]. Acoustic modeling for recognition of bird species from audio field recordings is decomposed into isolated segments, corresponding to detected sinusoids. Each segment is represented by a sequence of the frequency and normalized magnitude values of the sinusoid [8].

A handy tool as birdwatchers to admire the beauty of birds, it developed a deep learning platform to assist users in recognizing 27 species of birds [9]. An automatic bird identification system is required for offshore wind farms in Finland. Radar is used to detect flying birds, but external details are needed for real identification [10]. A convolutional neural network-based deep learning approach was used for bird song classification [11]. A Gaussian mixture model was used for bird classification based on their sound patterns [12]. Ensemble learning along with the coefficient may result in the best accuracy [13]. A deep learning neural network technique is used for identifying bird species with normal accuracy [14]. The distinction between the birds and other species can be done by an automated segmentation algorithm using neural networks [15].

.

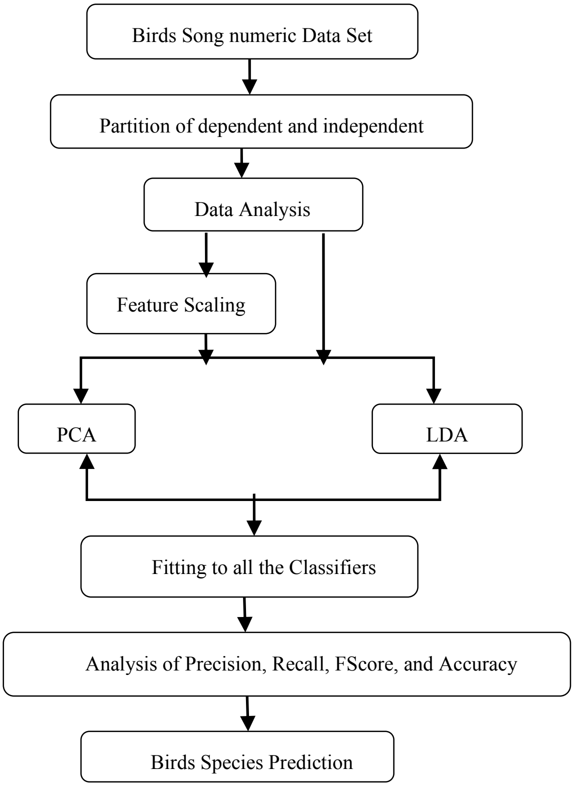
3 Our Contributions

3.1 Architectural Flow

This paper has the following contributions toward predicting the bird's species with the overall workflow shown in Fig. 1.

i. Firstly, the bird's song numeric dataset with 16,626 rows and 172 feature components from the UCI machine repository is subjected to data preprocessing.
ii. Secondly, the raw dataset is applied to all the classifiers to analyze the performance metrics before and after feature scaling.
iii. Thirdly, the data is reduced by PCA with 100, 75, and 50 components and then fitted to all the classifiers to analyze the performance metrics before and after feature scaling.
iv. Fourthly, the data is reduced by LDA with 100, 75, and 50 components and then fitted to all the classifiers to analyze the performance metrics before and after feature scaling.

Fig. 1 Overall architecture flow of the work


Birds Song numeric Data Set
Partition of dependent and independent
Data Analysis
Feature Scaling
PCA
LDA
Fitting to all the Classifiers
Analysis of Precision, Recall, FScore, and Accuracy
Birds Species Prediction

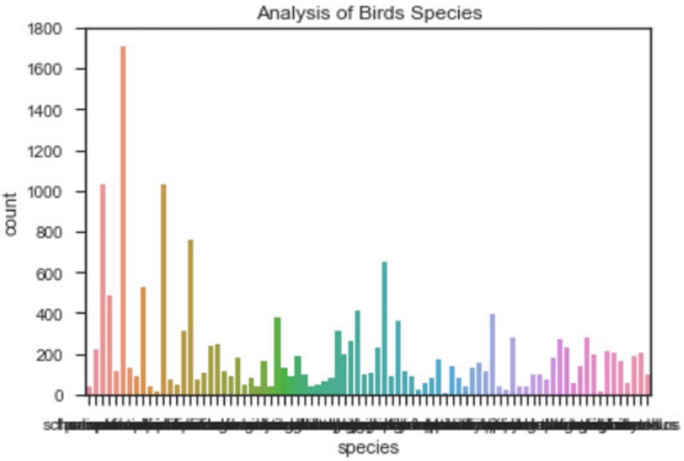


Fig. 2 Birds species dataset information and bird species analysis

4 Implementation Setup

4.1 Dataset Exploratory Analysis

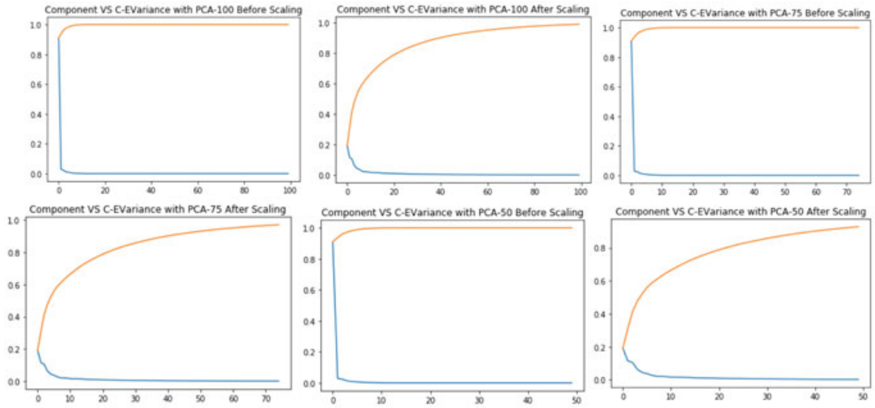
The bird's song numeric dataset with 16,626 rows and 172 feature components from the UCI machine repository is subjected to data preprocessing. The dataset information is shown in Fig. 2. The python scripting language is coded in Spyder editor with Anaconda navigator for implementation. The analysis of the bird's species in the dataset is shown in Fig. 2.

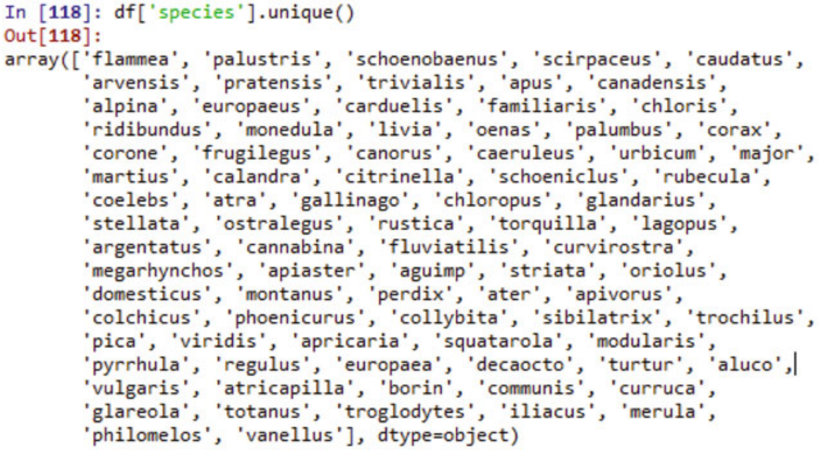
5 Results and Discussion

The data is reduced by principal component analysis, and component-wise relation with the explained variance for 100, 75, and 50 components is shown in Fig. 3. The birds species are shown in Fig. 3.

The raw dataset is subjected to all the classifiers, like logistic regression, K-nearest neighbor, kernel support vector machine, decision tree, random forest, gradient boosting, AdaBoost, Ridge, RidgeCV, SGD classifier, passive-aggressive, and the Bagging classifier, and the performance metrics is analyzed before and after feature scaling. The scaled and non-scaled performance analyses of all the classifiers are shown in Figs. 4 and 5.

The scaled and non-scaled performance analyses of the principal component analysis of reduced dataset with 100 components of all the classifiers are shown in Figs. 6 and 7.





```
In [118]: df['species'].unique()
Out[118]:
array(['flammea', 'palustris', 'schoenobaenus', 'scirpaceus', 'caudatus',
       'arvensis', 'pratensis', 'trivialis', 'apus', 'canadensis',
       'alpina', 'europaeus', 'carduelis', 'familiaris', 'chloris',
       'ridibundus', 'monedula', 'livia', 'oenas', 'palumbus', 'corax',
       'corone', 'frugilegus', 'canorus', 'caeruleus', 'urbicum', 'major',
       'martius', 'calandra', 'citrinella', 'schoeniclus', 'rubecula',
       'coelebs', 'atra', 'gallinago', 'chloropus', 'glandarius',
       'stellata', 'ostralegus', 'rustica', 'torquilla', 'lagopus',
       'argentatus', 'cannabina', 'fluviatilis', 'curvirostra',
       'megarhynchos', 'apiaster', 'aguimp', 'striata', 'oriolus',
       'domesticus', 'montanus', 'perdix', 'ater', 'apivorus',
       'colchicus', 'phoenicurus', 'collybita', 'sibilatrix', 'trochilus',
       'pica', 'viridis', 'apricaria', 'squatarola', 'modularis',
       'pyrrhula', 'regulus', 'europaea', 'decaocto', 'turtur', 'aluco',
       'vulgaris', 'atricapilla', 'borin', 'communis', 'curruca',
       'glareola', 'totanus', 'troglodytes', 'iliacus', 'merula',
       'philomelos', 'vanellus'], dtype=object)
```

Fig. 3 Component-wise versus cumulative explained variance of PCA and birds species

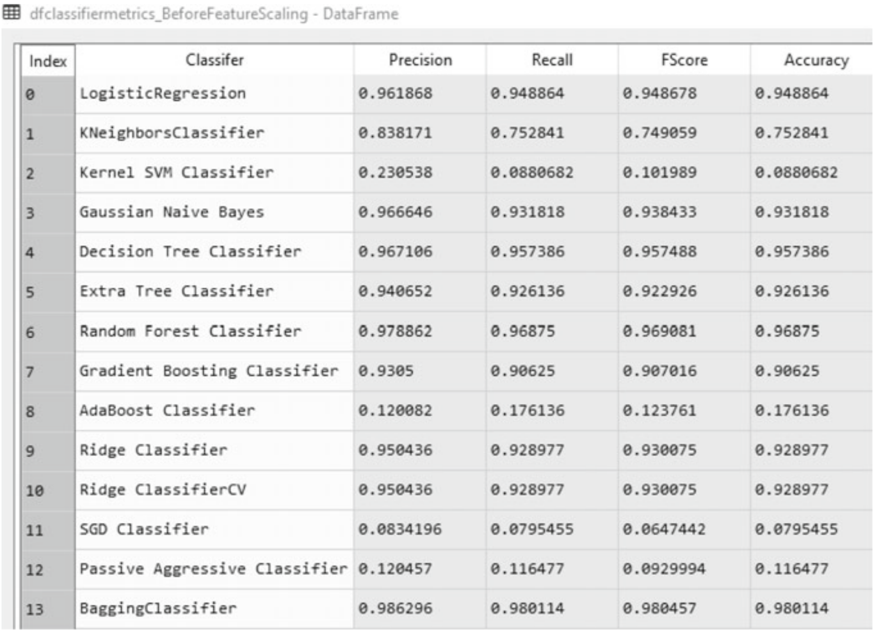
The scaled and non-scaled performance analyses of the principal component analysis of reduced dataset with 75 components of all the classifiers are shown in Figs. 8 and 9.

The scaled and non-scaled performance analyses of the principal component analysis of reduced dataset with 50 components of all the classifiers are shown in Figs. 10 and 11.

The scaled and non-scaled performance analyses of the linear discriminant analysis reduced the dataset with 100 components of all the classifiers are shown in Figs. 12 and 13.

The scaled and non-scaled performance analyses of the linear discriminant analysis of reduced dataset with 75 components of all the classifiers are shown in Figs. 14 and 15.

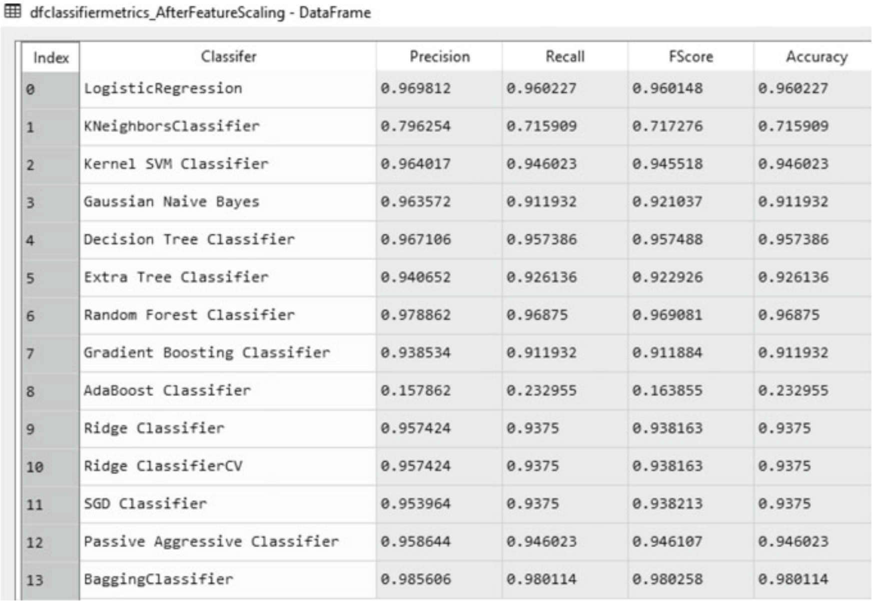
The scaled and non-scaled performance analyses of the linear discriminant analysis of reduced dataset with 50 components of all the classifiers are shown in Figs. 16 and 17.



dfclassifiermetrics_BeforeFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.961868	0.948864	0.948678	0.948864
1	KNeighborsClassifier	0.838171	0.752841	0.749059	0.752841
2	Kernel SVM Classifier	0.230538	0.0880682	0.101989	0.0880682
3	Gaussian Naive Bayes	0.966646	0.931818	0.938433	0.931818
4	Decision Tree Classifier	0.967106	0.957386	0.957488	0.957386
5	Extra Tree Classifier	0.940652	0.926136	0.922926	0.926136
6	Random Forest Classifier	0.978862	0.96875	0.969081	0.96875
7	Gradient Boosting Classifier	0.9305	0.90625	0.907016	0.90625
8	AdaBoost Classifier	0.120082	0.176136	0.123761	0.176136
9	Ridge Classifier	0.950436	0.928977	0.930075	0.928977
10	Ridge ClassifierCV	0.950436	0.928977	0.930075	0.928977
11	SGD Classifier	0.0834196	0.0795455	0.0647442	0.0795455
12	Passive Aggressive Classifier	0.120457	0.116477	0.0929994	0.116477
13	BaggingClassifier	0.986296	0.980114	0.980457	0.980114

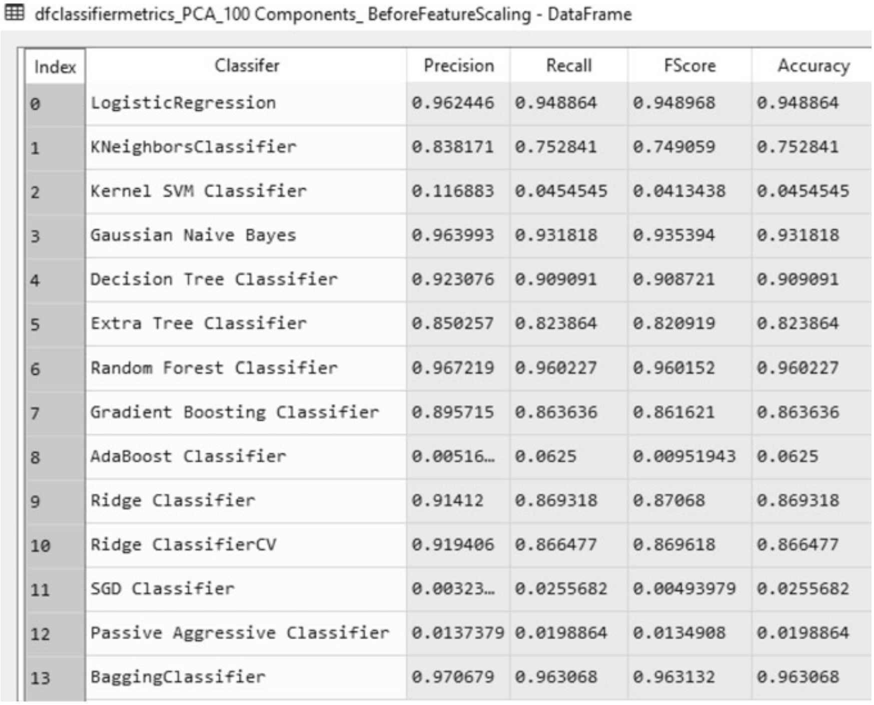
Fig. 4 Non-scaled performance metrics of classifiers without dimensionality reduction



dfclassifiermetrics_AfterFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.969812	0.960227	0.960148	0.960227
1	KNeighborsClassifier	0.796254	0.715909	0.717276	0.715909
2	Kernel SVM Classifier	0.964017	0.946023	0.945518	0.946023
3	Gaussian Naive Bayes	0.963572	0.911932	0.921037	0.911932
4	Decision Tree Classifier	0.967106	0.957386	0.957488	0.957386
5	Extra Tree Classifier	0.940652	0.926136	0.922926	0.926136
6	Random Forest Classifier	0.978862	0.96875	0.969081	0.96875
7	Gradient Boosting Classifier	0.938534	0.911932	0.911884	0.911932
8	AdaBoost Classifier	0.157862	0.232955	0.163855	0.232955
9	Ridge Classifier	0.957424	0.9375	0.938163	0.9375
10	Ridge ClassifierCV	0.957424	0.9375	0.938163	0.9375
11	SGD Classifier	0.953964	0.9375	0.938213	0.9375
12	Passive Aggressive Classifier	0.958644	0.946023	0.946107	0.946023
13	BaggingClassifier	0.985606	0.980114	0.980258	0.980114

Fig. 5 Scaled performance metrics of classifiers without dimensionality reduction



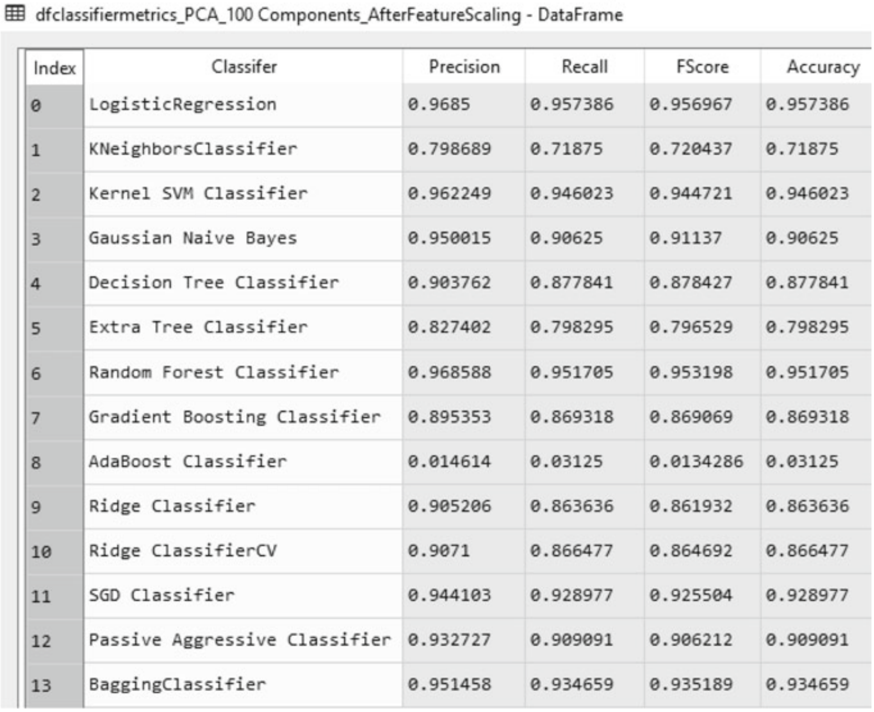
dfclassifiermetrics_PCA_100 Components_ BeforeFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.962446	0.948864	0.948968	0.948864
1	KNeighborsClassifier	0.838171	0.752841	0.749059	0.752841
2	Kernel SVM Classifier	0.116883	0.0454545	0.0413438	0.0454545
3	Gaussian Naive Bayes	0.963993	0.931818	0.935394	0.931818
4	Decision Tree Classifier	0.923076	0.909091	0.908721	0.909091
5	Extra Tree Classifier	0.850257	0.823864	0.820919	0.823864
6	Random Forest Classifier	0.967219	0.960227	0.960152	0.960227
7	Gradient Boosting Classifier	0.895715	0.863636	0.861621	0.863636
8	AdaBoost Classifier	0.00516…	0.0625	0.00951943	0.0625
9	Ridge Classifier	0.91412	0.869318	0.87068	0.869318
10	Ridge ClassifierCV	0.919406	0.866477	0.869618	0.866477
11	SGD Classifier	0.00323…	0.0255682	0.00493979	0.0255682
12	Passive Aggressive Classifier	0.0137379	0.0198864	0.0134908	0.0198864
13	BaggingClassifier	0.970679	0.963068	0.963132	0.963068

Fig. 6 Non-scaled performance metrics of PCA with 100 components

6 Conclusion

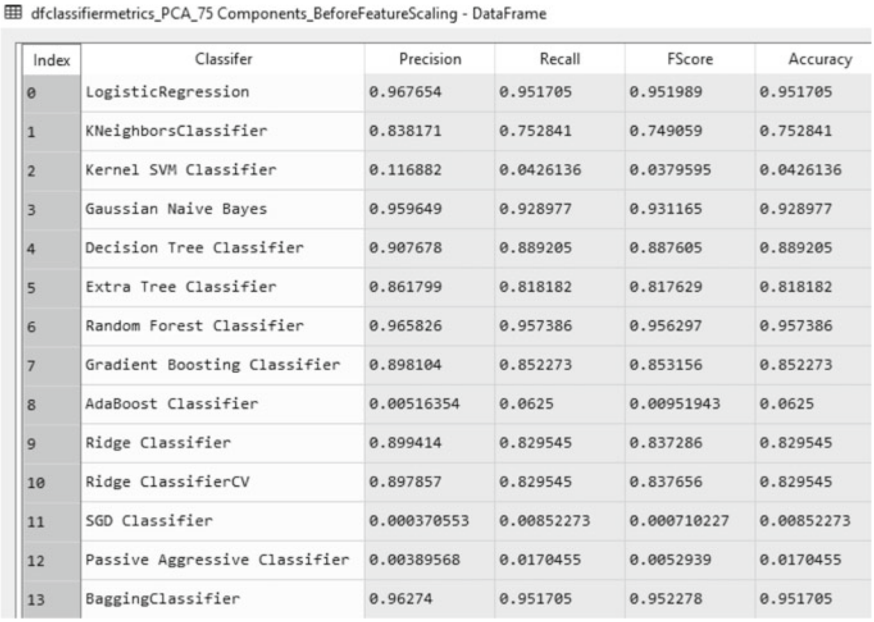
This paper attempts to predict the bird's species classification based on the song spectrum features. Since the dataset has 172 features, the accuracy is analyzed by fitting the classifiers with the raw dataset. An attempt is made to examine whether the same accuracy can be retained even after reducing the dimension of the dataset with the principal component analysis and linear discriminant analysis. Experimental results show that the Bagging classifier has achieved an accuracy of 98%, random forest with 96%, logistic with 96% for the raw dataset. After reducing the raw dataset with PCA, the random forest classifier retains 96% accuracy for 100, 75, and 50 component reduction. Similarly, the LDA-reduced dataset with 100, 75, and 50 components attains a 96% accuracy for logistic regression classifier.



dfclassifiermetrics_PCA_100 Components_AfterFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.9685	0.957386	0.956967	0.957386
1	KNeighborsClassifier	0.798689	0.71875	0.720437	0.71875
2	Kernel SVM Classifier	0.962249	0.946023	0.944721	0.946023
3	Gaussian Naive Bayes	0.950015	0.90625	0.91137	0.90625
4	Decision Tree Classifier	0.903762	0.877841	0.878427	0.877841
5	Extra Tree Classifier	0.827402	0.798295	0.796529	0.798295
6	Random Forest Classifier	0.968588	0.951705	0.953198	0.951705
7	Gradient Boosting Classifier	0.895353	0.869318	0.869069	0.869318
8	AdaBoost Classifier	0.014614	0.03125	0.0134286	0.03125
9	Ridge Classifier	0.905206	0.863636	0.861932	0.863636
10	Ridge ClassifierCV	0.9071	0.866477	0.864692	0.866477
11	SGD Classifier	0.944103	0.928977	0.925504	0.928977
12	Passive Aggressive Classifier	0.932727	0.909091	0.906212	0.909091
13	BaggingClassifier	0.951458	0.934659	0.935189	0.934659

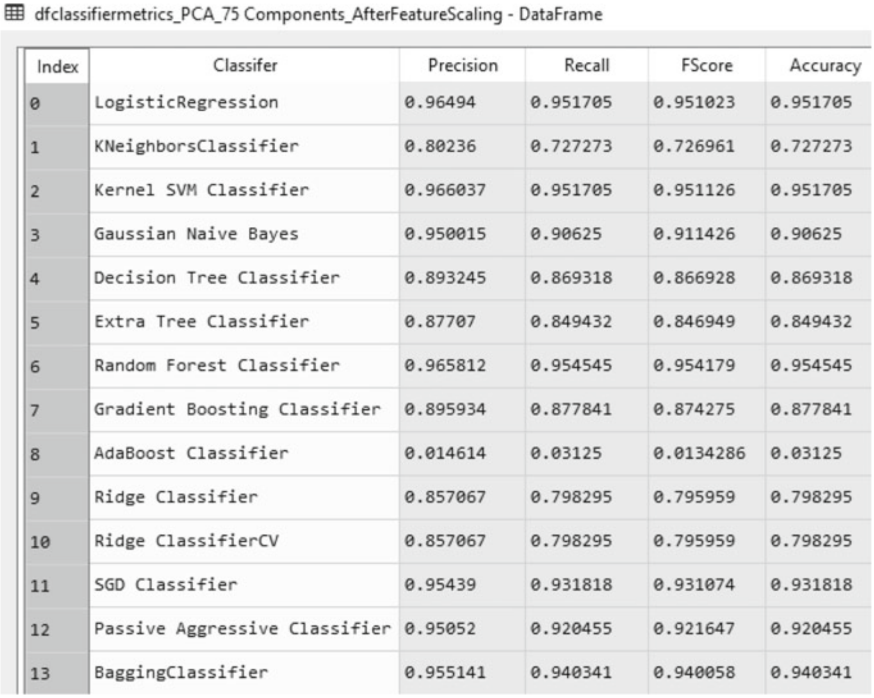
Fig. 7 Scaled performance metrics of PCA with 100 components



dfclassifiermetrics_PCA_75 Components_BeforeFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.967654	0.951705	0.951989	0.951705
1	KNeighborsClassifier	0.838171	0.752841	0.749059	0.752841
2	Kernel SVM Classifier	0.116882	0.0426136	0.0379595	0.0426136
3	Gaussian Naive Bayes	0.959649	0.928977	0.931165	0.928977
4	Decision Tree Classifier	0.907678	0.889205	0.887605	0.889205
5	Extra Tree Classifier	0.861799	0.818182	0.817629	0.818182
6	Random Forest Classifier	0.965826	0.957386	0.956297	0.957386
7	Gradient Boosting Classifier	0.898104	0.852273	0.853156	0.852273
8	AdaBoost Classifier	0.00516354	0.0625	0.00951943	0.0625
9	Ridge Classifier	0.899414	0.829545	0.837286	0.829545
10	Ridge ClassifierCV	0.897857	0.829545	0.837656	0.829545
11	SGD Classifier	0.000370553	0.00852273	0.000710227	0.00852273
12	Passive Aggressive Classifier	0.00389568	0.0170455	0.0052939	0.0170455
13	BaggingClassifier	0.96274	0.951705	0.952278	0.951705

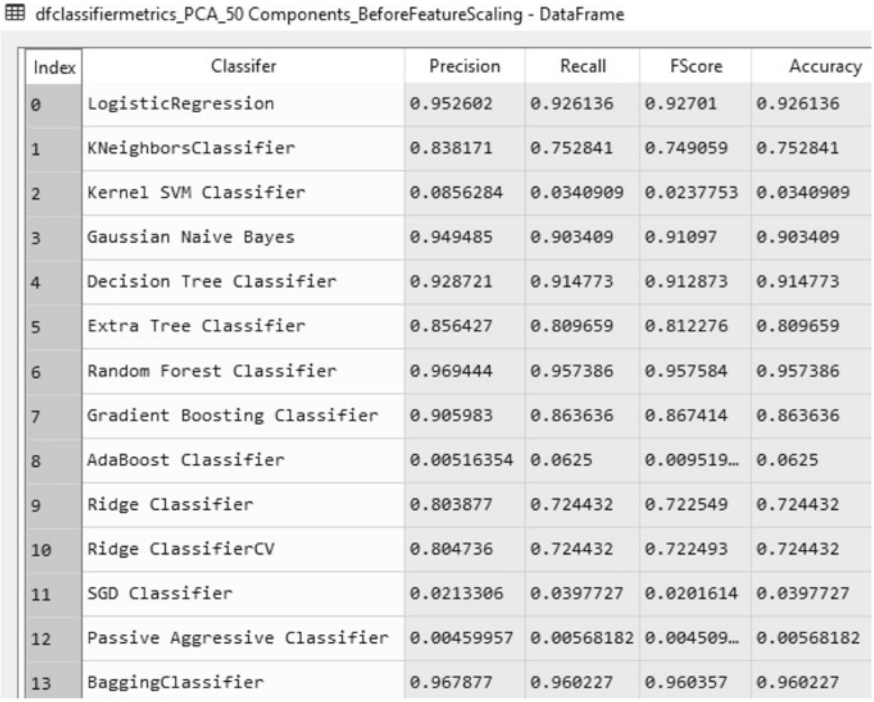
Fig. 8 Non-scaled performance metrics of PCA with 75 components



dfclassifiermetrics_PCA_75 Components_AfterFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.96494	0.951705	0.951023	0.951705
1	KNeighborsClassifier	0.80236	0.727273	0.726961	0.727273
2	Kernel SVM Classifier	0.966037	0.951705	0.951126	0.951705
3	Gaussian Naive Bayes	0.950015	0.90625	0.911426	0.90625
4	Decision Tree Classifier	0.893245	0.869318	0.866928	0.869318
5	Extra Tree Classifier	0.87707	0.849432	0.846949	0.849432
6	Random Forest Classifier	0.965812	0.954545	0.954179	0.954545
7	Gradient Boosting Classifier	0.895934	0.877841	0.874275	0.877841
8	AdaBoost Classifier	0.014614	0.03125	0.0134286	0.03125
9	Ridge Classifier	0.857067	0.798295	0.795959	0.798295
10	Ridge ClassifierCV	0.857067	0.798295	0.795959	0.798295
11	SGD Classifier	0.95439	0.931818	0.931074	0.931818
12	Passive Aggressive Classifier	0.95052	0.920455	0.921647	0.920455
13	BaggingClassifier	0.955141	0.940341	0.940058	0.940341

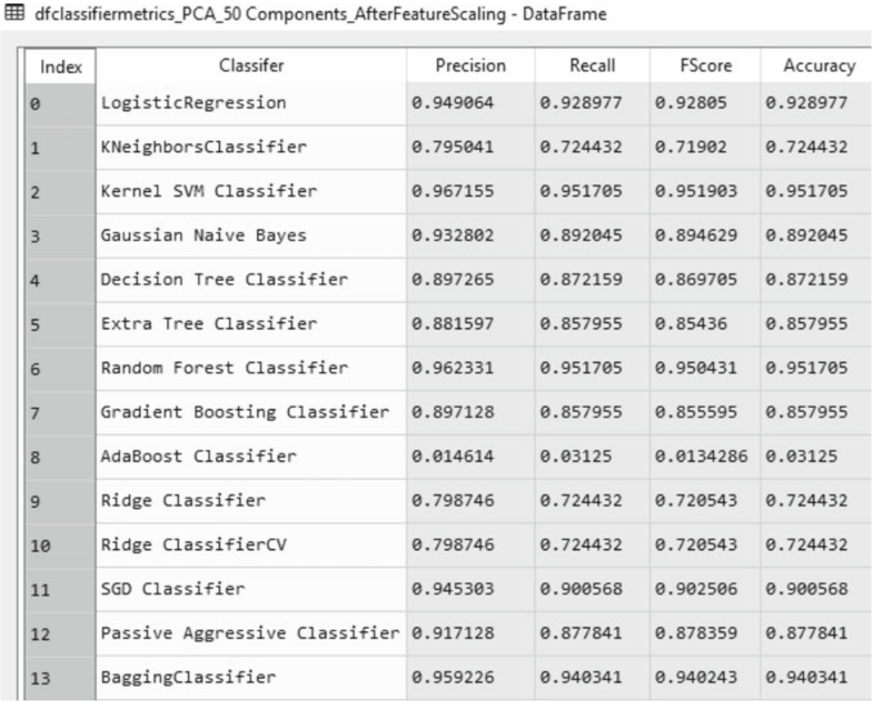
Fig. 9 Scaled performance metrics of PCA with 75 components



dfclassifiermetrics_PCA_50 Components_BeforeFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.952602	0.926136	0.92701	0.926136
1	KNeighborsClassifier	0.838171	0.752841	0.749059	0.752841
2	Kernel SVM Classifier	0.0856284	0.0340909	0.0237753	0.0340909
3	Gaussian Naive Bayes	0.949485	0.903409	0.91097	0.903409
4	Decision Tree Classifier	0.928721	0.914773	0.912873	0.914773
5	Extra Tree Classifier	0.856427	0.809659	0.812276	0.809659
6	Random Forest Classifier	0.969444	0.957386	0.957584	0.957386
7	Gradient Boosting Classifier	0.905983	0.863636	0.867414	0.863636
8	AdaBoost Classifier	0.00516354	0.0625	0.009519…	0.0625
9	Ridge Classifier	0.803877	0.724432	0.722549	0.724432
10	Ridge ClassifierCV	0.804736	0.724432	0.722493	0.724432
11	SGD Classifier	0.0213306	0.0397727	0.0201614	0.0397727
12	Passive Aggressive Classifier	0.00459957	0.00568182	0.004509…	0.00568182
13	BaggingClassifier	0.967877	0.960227	0.960357	0.960227

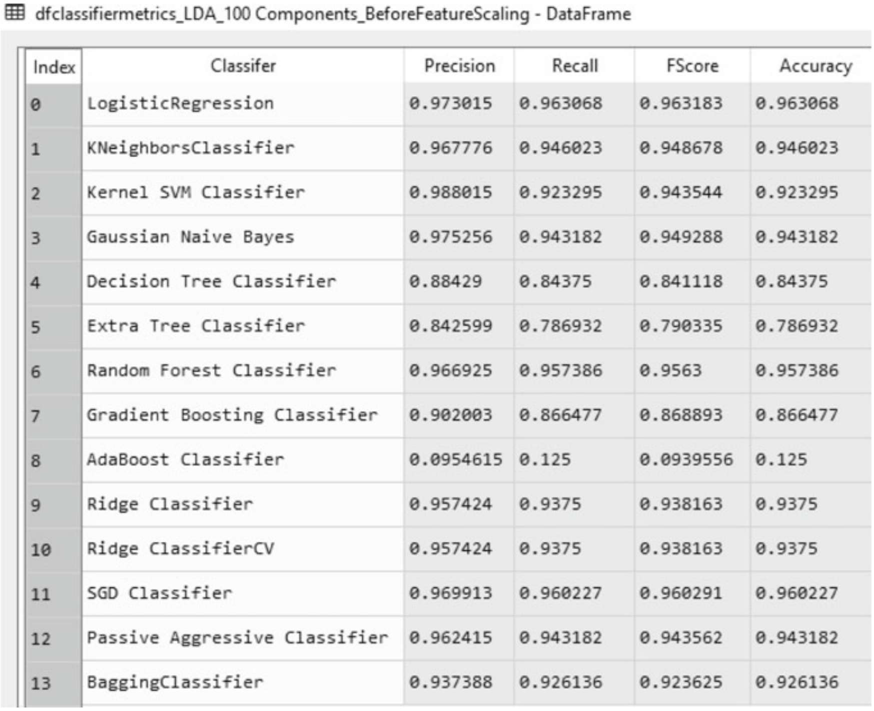
Fig. 10 Non-scaled performance metrics of PCA with 50 components



dfclassifiermetrics_PCA_50 Components_AfterFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.949064	0.928977	0.92805	0.928977
1	KNeighborsClassifier	0.795041	0.724432	0.71902	0.724432
2	Kernel SVM Classifier	0.967155	0.951705	0.951903	0.951705
3	Gaussian Naive Bayes	0.932802	0.892045	0.894629	0.892045
4	Decision Tree Classifier	0.897265	0.872159	0.869705	0.872159
5	Extra Tree Classifier	0.881597	0.857955	0.85436	0.857955
6	Random Forest Classifier	0.962331	0.951705	0.950431	0.951705
7	Gradient Boosting Classifier	0.897128	0.857955	0.855595	0.857955
8	AdaBoost Classifier	0.014614	0.03125	0.0134286	0.03125
9	Ridge Classifier	0.798746	0.724432	0.720543	0.724432
10	Ridge ClassifierCV	0.798746	0.724432	0.720543	0.724432
11	SGD Classifier	0.945303	0.900568	0.902506	0.900568
12	Passive Aggressive Classifier	0.917128	0.877841	0.878359	0.877841
13	BaggingClassifier	0.959226	0.940341	0.940243	0.940341

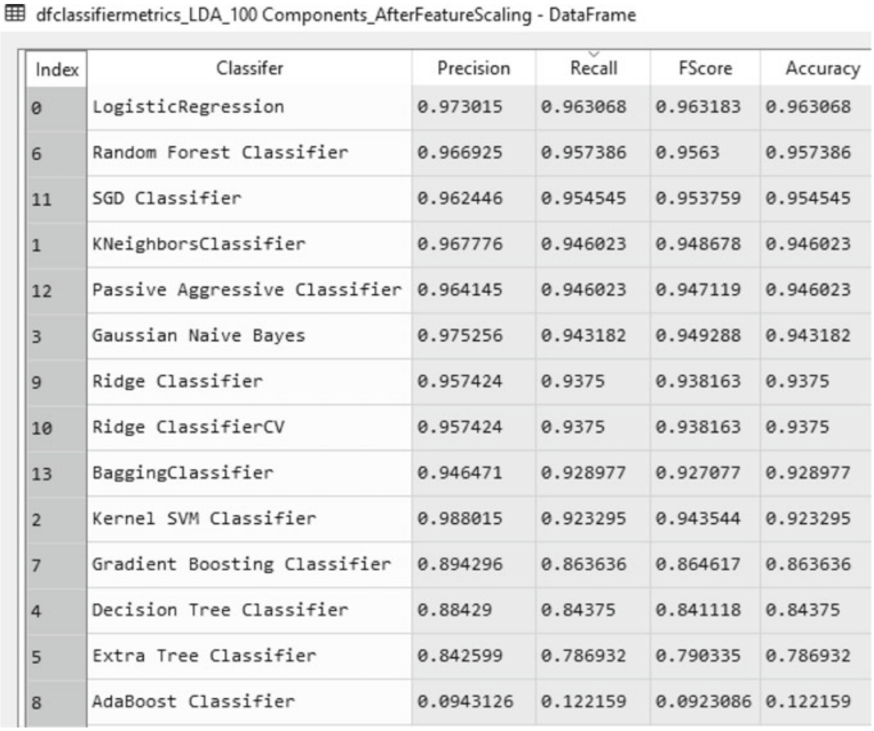
Fig. 11 Scaled performance metrics of PCA with 50 components



dfclassifiermetrics_LDA_100 Components_BeforeFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.973015	0.963068	0.963183	0.963068
1	KNeighborsClassifier	0.967776	0.946023	0.948678	0.946023
2	Kernel SVM Classifier	0.988015	0.923295	0.943544	0.923295
3	Gaussian Naive Bayes	0.975256	0.943182	0.949288	0.943182
4	Decision Tree Classifier	0.88429	0.84375	0.841118	0.84375
5	Extra Tree Classifier	0.842599	0.786932	0.790335	0.786932
6	Random Forest Classifier	0.966925	0.957386	0.9563	0.957386
7	Gradient Boosting Classifier	0.902003	0.866477	0.868893	0.866477
8	AdaBoost Classifier	0.0954615	0.125	0.0939556	0.125
9	Ridge Classifier	0.957424	0.9375	0.938163	0.9375
10	Ridge ClassifierCV	0.957424	0.9375	0.938163	0.9375
11	SGD Classifier	0.969913	0.960227	0.960291	0.960227
12	Passive Aggressive Classifier	0.962415	0.943182	0.943562	0.943182
13	BaggingClassifier	0.937388	0.926136	0.923625	0.926136

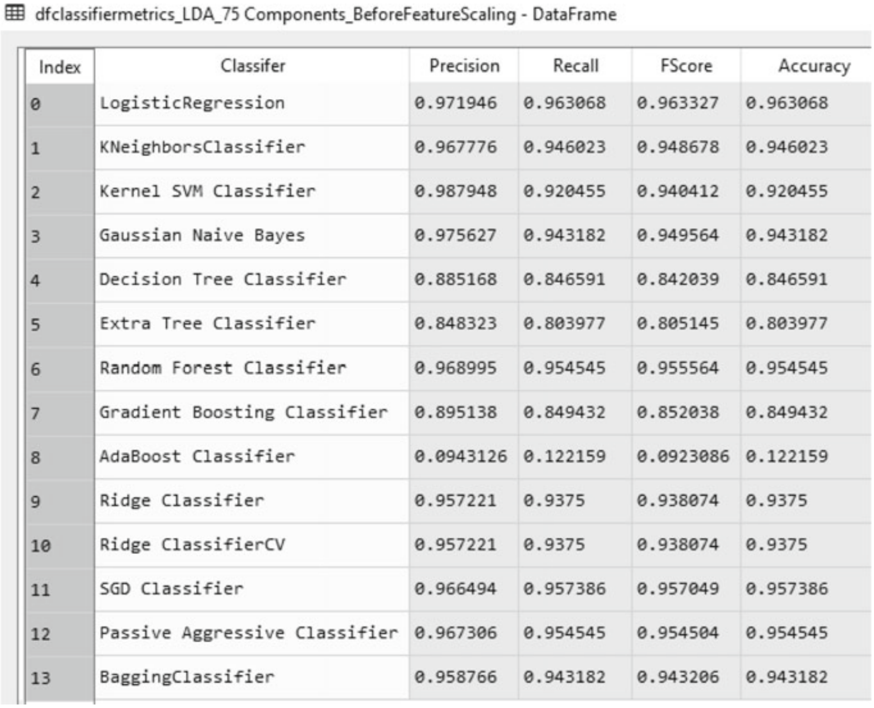
Fig. 12 Non-scaled performance metrics of LDA with 100 components



dfclassifiermetrics_LDA_100 Components_AfterFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.973015	0.963068	0.963183	0.963068
6	Random Forest Classifier	0.966925	0.957386	0.9563	0.957386
11	SGD Classifier	0.962446	0.954545	0.953759	0.954545
1	KNeighborsClassifier	0.967776	0.946023	0.948678	0.946023
12	Passive Aggressive Classifier	0.964145	0.946023	0.947119	0.946023
3	Gaussian Naive Bayes	0.975256	0.943182	0.949288	0.943182
9	Ridge Classifier	0.957424	0.9375	0.938163	0.9375
10	Ridge ClassifierCV	0.957424	0.9375	0.938163	0.9375
13	BaggingClassifier	0.946471	0.928977	0.927077	0.928977
2	Kernel SVM Classifier	0.988015	0.923295	0.943544	0.923295
7	Gradient Boosting Classifier	0.894296	0.863636	0.864617	0.863636
4	Decision Tree Classifier	0.88429	0.84375	0.841118	0.84375
5	Extra Tree Classifier	0.842599	0.786932	0.790335	0.786932
8	AdaBoost Classifier	0.0943126	0.122159	0.0923086	0.122159

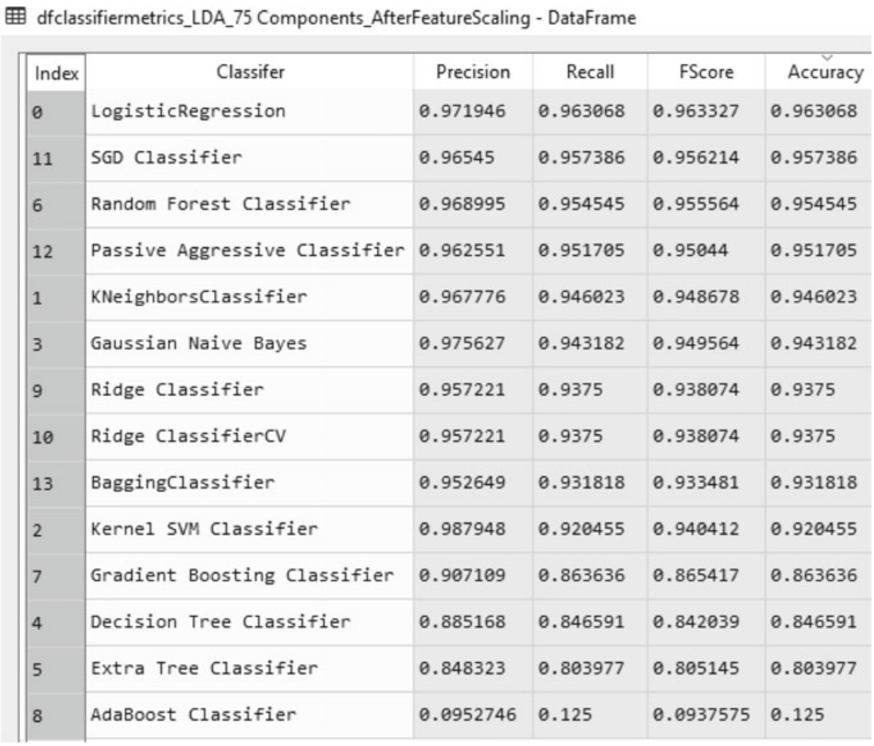
Fig. 13 Scaled performance metrics of LDA with 100 components



dfclassifiermetrics_LDA_75 Components_BeforeFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.971946	0.963068	0.963327	0.963068
1	KNeighborsClassifier	0.967776	0.946023	0.948678	0.946023
2	Kernel SVM Classifier	0.987948	0.920455	0.940412	0.920455
3	Gaussian Naive Bayes	0.975627	0.943182	0.949564	0.943182
4	Decision Tree Classifier	0.885168	0.846591	0.842039	0.846591
5	Extra Tree Classifier	0.848323	0.803977	0.805145	0.803977
6	Random Forest Classifier	0.968995	0.954545	0.955564	0.954545
7	Gradient Boosting Classifier	0.895138	0.849432	0.852038	0.849432
8	AdaBoost Classifier	0.0943126	0.122159	0.0923086	0.122159
9	Ridge Classifier	0.957221	0.9375	0.938074	0.9375
10	Ridge ClassifierCV	0.957221	0.9375	0.938074	0.9375
11	SGD Classifier	0.966494	0.957386	0.957049	0.957386
12	Passive Aggressive Classifier	0.967306	0.954545	0.954504	0.954545
13	BaggingClassifier	0.958766	0.943182	0.943206	0.943182

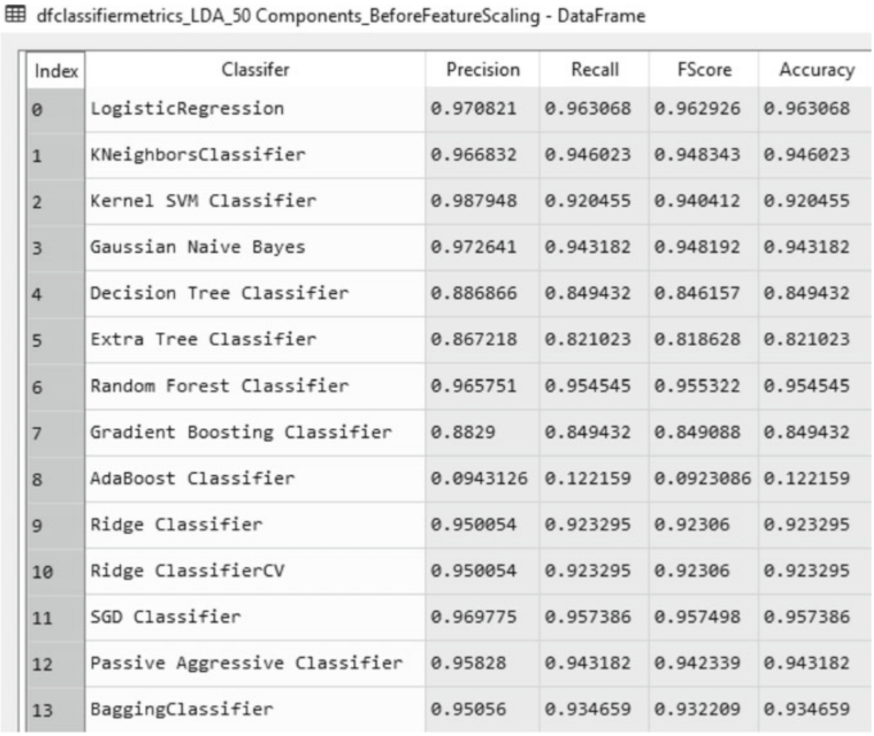
Fig.14 Non-scaled performance metrics of LDA with 75 components



dfclassifiermetrics_LDA_75 Components_AfterFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.971946	0.963068	0.963327	0.963068
11	SGD Classifier	0.96545	0.957386	0.956214	0.957386
6	Random Forest Classifier	0.968995	0.954545	0.955564	0.954545
12	Passive Aggressive Classifier	0.962551	0.951705	0.95044	0.951705
1	KNeighborsClassifier	0.967776	0.946023	0.948678	0.946023
3	Gaussian Naive Bayes	0.975627	0.943182	0.949564	0.943182
9	Ridge Classifier	0.957221	0.9375	0.938074	0.9375
10	Ridge ClassifierCV	0.957221	0.9375	0.938074	0.9375
13	BaggingClassifier	0.952649	0.931818	0.933481	0.931818
2	Kernel SVM Classifier	0.987948	0.920455	0.940412	0.920455
7	Gradient Boosting Classifier	0.907109	0.863636	0.865417	0.863636
4	Decision Tree Classifier	0.885168	0.846591	0.842039	0.846591
5	Extra Tree Classifier	0.848323	0.803977	0.805145	0.803977
8	AdaBoost Classifier	0.0952746	0.125	0.0937575	0.125

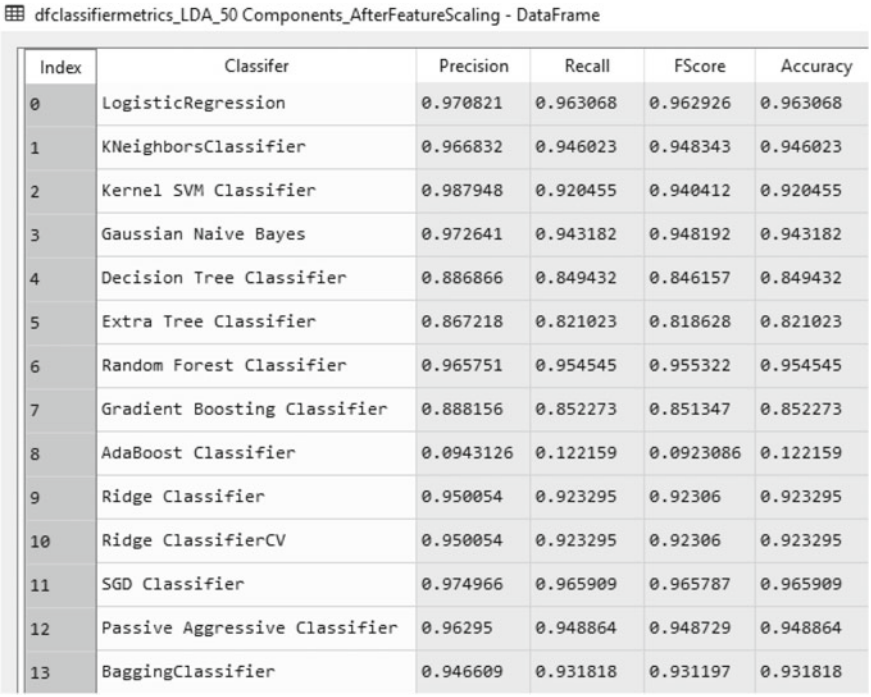
Fig. 15 Scaled performance metrics of LDA with 75 components



dfclassifiermetrics_LDA_50 Components_BeforeFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.970821	0.963068	0.962926	0.963068
1	KNeighborsClassifier	0.966832	0.946023	0.948343	0.946023
2	Kernel SVM Classifier	0.987948	0.920455	0.940412	0.920455
3	Gaussian Naive Bayes	0.972641	0.943182	0.948192	0.943182
4	Decision Tree Classifier	0.886866	0.849432	0.846157	0.849432
5	Extra Tree Classifier	0.867218	0.821023	0.818628	0.821023
6	Random Forest Classifier	0.965751	0.954545	0.955322	0.954545
7	Gradient Boosting Classifier	0.8829	0.849432	0.849088	0.849432
8	AdaBoost Classifier	0.0943126	0.122159	0.0923086	0.122159
9	Ridge Classifier	0.950054	0.923295	0.92306	0.923295
10	Ridge ClassifierCV	0.950054	0.923295	0.92306	0.923295
11	SGD Classifier	0.969775	0.957386	0.957498	0.957386
12	Passive Aggressive Classifier	0.95828	0.943182	0.942339	0.943182
13	BaggingClassifier	0.95056	0.934659	0.932209	0.934659

Fig. 16 Non-scaled performance metrics of LDA with 50 components



dfclassifiermetrics_LDA_50 Components_AfterFeatureScaling - DataFrame

Index	Classifer	Precision	Recall	FScore	Accuracy
0	LogisticRegression	0.970821	0.963068	0.962926	0.963068
1	KNeighborsClassifier	0.966832	0.946023	0.948343	0.946023
2	Kernel SVM Classifier	0.987948	0.920455	0.940412	0.920455
3	Gaussian Naive Bayes	0.972641	0.943182	0.948192	0.943182
4	Decision Tree Classifier	0.886866	0.849432	0.846157	0.849432
5	Extra Tree Classifier	0.867218	0.821023	0.818628	0.821023
6	Random Forest Classifier	0.965751	0.954545	0.955322	0.954545
7	Gradient Boosting Classifier	0.888156	0.852273	0.851347	0.852273
8	AdaBoost Classifier	0.0943126	0.122159	0.0923086	0.122159
9	Ridge Classifier	0.950054	0.923295	0.92306	0.923295
10	Ridge ClassifierCV	0.950054	0.923295	0.92306	0.923295
11	SGD Classifier	0.974966	0.965909	0.965787	0.965909
12	Passive Aggressive Classifier	0.96295	0.948864	0.948729	0.948864
13	BaggingClassifier	0.946609	0.931818	0.931197	0.931818

Fig. 17 Scaled performance metrics of LDA with 50 components

References

1. Stastny J, Munk M, Juranek L (2018) Automatic bird species recognition based on birds vocalization. Audio Speech Music Proc 2018:1–19
2. Incze A, Jancso HB, Szilagyi Z, Farka A (2018) Bird Sound recognition using a convolutional neural network. In: 16th International symposium on intelligent systems and informatics on proceedings, pp 295–300. https://doi.org/10.1109/SISY.2018.8524677
3. Jadhav Y, Patil V, Parasar D (2020) Machine learning approach to classify birds on the basis of their sound, In: 2020 International conference on inventive computation technologies, pp 69–73, Coimbatore. https://doi.org/10.1109/ICICT48043.2020.9112506
4. Evangelista TLF, Priolli TM, Silla CN, Angélico BA, Kaestner CAA (2014) Automatic segmentation of audio signals for bird species identification. In: International symposium on multimedia on proceedings, Taichung, pp 223–228
5. Towhid MS, Rahman MM (2017) Spectrogram segmentation for bird species classification based on temporal continuity. In: 20th International conference of computer and information technology, pp 1–4, Dhaka. https://doi.org/10.1109/ICCITECHN.2017.8281775
6. Aljarah I, Faris H, Mirjalili S (2019) Evolving neural networks using bird swarm algorithm for data classification and regression applications. Cluster Comput J 1317–1345. https://doi.org/10.1007/s10586-019-02913-5
7. Acevedo M, Corrada Bravo C, Corrada Bravo H, Villanueva-Rivera L, Aide TM (2009) Automated classification of bird and amphibian calls using machine learning: a comparison of methods. Ecol Inf 206–214

8. Jancovic P, Köküer M (2019) Bird Species recognition using unsupervised modeling of individual vocalization elements. In: IEEE/ACM transactions on audio, speech, and language processing, vol 27, no 5, pp 932–947
9. Huang Y, Basanta H (2019) Bird Image retrieval and recognition using a deep learning platform. IEEE Access 7:66980–66989
10. Niemi J, Tanttu JT (2018) Deep learning case study for automatic bird identification. Science 8–2018:1–15
11. Toth BP, Czeba B, Convolutional neural networks for large-scale bird song classification in noisy environment. http://ceur-ws.org/Vol-1609/16090560
12. Supriya PR, Bhat S, Shivani SS (2018) Classification of birds based on their sound patterns using GMM and SVM classifiers. Int Res J Eng Technol 4708
13. Guo X, Liu Q (2017) A comparison study to identify birds species based on bird song signals. In: ITM Web of conferences
14. Pillai SK, Raghuwanshi MM, Shrawankar U (2019) Deep learning neural network for identification of bird species. Lect Notes Netw Syst 291–298
15. Koops HV, van Balen J, Wiering F (2015) Automatic segmentation and deep learning of bird sounds. Lect Notes Comput Sci

P2P Traffic Identification Using Machine Learning and Feature Selection Techniques

Md. Sarfaraj Alam Ansari, Kunwar Pal, Mahesh Chandra Govil, Prajjval Govil, and Adarsh Srivastava

Abstract Classification of Internet traffic based on its applications is significantly important for better network management. The existing approaches for P2P traffic identification are port-based, payload-based, behaviour-based, and heuristic-based. In the present scenario, the method based on intrinsic techniques is not efficient to classify Internet traffic due to the use of dynamic port numbers, masquerading techniques and low accuracy. Therefore, the research is moved to the development of hybrid approaches. In this paper, hybrid approaches are proposed for the classification of Internet traffic into P2P and non-P2P traffic. The proposed approaches use machine learning algorithms with feature selection techniques such as analysis of variance and principal component analysis. Port-based labelling of data is used for the training of classifiers in the proposed system. The proposed approaches are investigated on standard UNIBS dataset, and the simulation and modelling are performed in python environment. The comparative study of the proposed hybrid approaches shows that random forest with ANOVA outperforms other approaches presented in this paper, resulting in 96.06% accuracy. The proposed approach provides better accuracy as compared to earlier reported similar approaches in the literature.

Keywords Internet traffic · Peer-to-peer (P2P) · Feature selection · Classifier · Machine learning

Md. S. A. Ansari (✉) · M. C. Govil · A. Srivastava
Department of Computer Science and Engineering, National Institute of Technology Sikkim, Ravangla 737139, India
e-mail: sarfaraj@nitsikkim.ac.in

A. Srivastava
e-mail: b180001@nitsikkim.ac.in

K. Pal
Department of Computer Science and Engineering, Dr B R Ambedkar National Institute of Technology Jalandhar, Jalandhar 144011, India
e-mail: kunwarp@nitj.ac.in

P. Govil
Department of Computer Science and Engineering, JK Lakshmipat University, Jaipur 302026, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_30

1 Introduction

The Internet can be considered as the system of various interconnected networks. Its enormous growth over the years has made the sharing of data like text, audio or video easier and has lead to an increase in network traffic. The traditional client–server system was efficient for multimedia services; however, over the time, an unprecedented increase in content streaming has forced the research community to look for a more efficient solution. The inception of P2P network can be attributed as a solution to various problems in traditional client–server models like congestion, scalability, bandwidth limitation and poor QoS [1, 2]. In the past few years, the popularity of P2P network has raised rapidly because it is robust, scalable, cost-effective and accurate. Further, resource sharing is also one of the major achievements in P2P networks which have contributed to its efficiency and power, satisfying QoS parameters [3]. In P2P network, the content sharing among peers has also resulted in symmetric flow of traffic [4]. Increment of users increases the services delivered by P2P networks unlike in client–server system where increment of users' resulting in the decrement of network performances [5]. The performance of the P2P network predominantly depends on the overlay construction [6–8] scheduling scheme [9, 10] and selfish peer [11]. An enormous amount of work on overlay and scheduling scheme has been already done [12–14].

The popularity and usage of P2P applications are increasing day by day due to its inherent advantages mentioned above. The P2P applications dominate over the Internet as compared to other protocol applications such as HTTP, FTP and SMTP. Presently nearly 70% of the Internet traffic is P2P and consumes a major portion of bandwidth [15]. Service providers have the policies to maintain service quality that are not sufficient to control P2P system traffic effectively. The Internet traffic characterization achieved a research temperament due to the need of provisioning and planning of network capacity, service quality of applications, fault analysis, anomaly detection, billing, etc. A fair distribution of bandwidth among P2P and other applications is necessary for QoS [16, 17]. It motivated us to review the literature and concluded that the classification of Internet traffic is one of the important requirements.

Many methods have been proposed for Internet traffic classification such as port-based, payload-based, statistics or behaviour-based and heuristics-based which are mainstream approaches. Traditionally, the port-based techniques are used to classify Internet traffic based on well-known port numbers, due to the ease of deployment. The application like DNS or SMTP uses specific ports statically and therefore yields high accuracy. Moore and Papagiannaki [18] reveal that 30–70% of traffic generated by P2P applications uses random port numbers and masquerading techniques that make port-based identification difficult. Park et al. [19] highlighted an important fact that although it provides low classification accuracy, this method is still relevant in the Internet backbone due to its scalability and minimal computational overheads. Hence, the port-based approach plays a determining role to give a direction when combined with other methods to make a hybrid approach for identifying the P2P traffic.

It is observed that all the techniques have their own limitations. A study of payload-based method reveals that it suffers from high computational overhead and user privacy issues. The applications which have similar behaviour are difficult to analyse with the behavioural-based approach. In case of statistical-based method, the numerical attributes do not always provide high-quality training data. Thus, integration of different techniques may provide the desired accuracy and this has led us to explore hybrid methods.

The paper is structured as follows: a brief literature survey on the related work is presented in Sect. 2. Section 3 describes the proposed framework, the methodology involved and the dataset used for the Internet traffic classification. Section 4 covers the experimental setup and performance evaluation. Finally, Sect. 5 put the necessary concluding remarks based on evaluated results and its' future research directions.

2 Related Works

This section provides a brief overview of hybrid approaches used for P2P traffic identification. The port-based technique [18, 20] as discussed earlier is simple to use. The purely port-based traffic classification techniques have become less effective due to the increase of dynamic port number usage, masquerading and encryption techniques [21]. However, Jefferey et al. [22] have advocated that port-based techniques are still useful and can provide better results. The present trend in the research community is to use hybrid approaches that combine various techniques from different domains such as machine learning [23–26], genetic algorithm and neural network [27] with intrinsic port-based, payload-based and behavioural-based methods.

Jefferey et al. [22] demonstrated that the AutoClass classifier outperformed the Naive Bayes and accuracy achieved was 91.19%. Yan et al. [23] also used ML-based techniques and accuracy reported on the UNIBS dataset is 93.9%. Raahemi et al. [24] used the very fast decision tree (CVFDT) ML technique and obtained 95% accuracy. They collected their own dataset and determined the performance for every 10,000 examples. The labelling of the training set was done using the port-based technique. In [27], the author used genetic algorithm and neural network for traffic identification and claimed to achieve 96% accuracy on their own dataset consisting of 32,767 sample records. They also labelled the dataset based on default port numbers of popular P2P applications.

Draper-Gil et al. [25] considered time-related features of captured VPN traffic and used C4.5 and KNN for classification. Results have shown that the time-related features are a good choice and it achieves more than 80% accuracy. However, C4.5 performed better for encrypted traffic characterization. Further, Saber et al. [26] proposed a similar approach but used PCA for feature selection and classify the combined over and under-sampling of data set of VPN and non-VPN using SVM. The proposed approach resulted in accuracies of 95.6%, 93.9%, 94.9% while taking the flow time-outs of 30s, 60s and 120s, respectively. It demonstrated that PCA improved the efficiency. However, the reported efficiency was higher for shorter flows. Junior et al. [28] proposed another ML-based approach using ANOVA as a feature selection technique. They achieved P2P classification accuracy of 90%.

In this paper, hybrid approaches are proposed to categorize Internet traffic by leveraging the advantages of various methods discussed above. The proposed approaches are a hybridisation of port-based methods, feature selection techniques and ML Algorithms. The salient contribution of this research work is as follows:

- Service port numbers are extracted from the datasets which collected by running P2P applications and prepared a comprehensive list of port numbers ($\approx$ 22,000) to label the training dataset.
- Data preprocessing and features extraction were performed.
- Various feature subsets were selected using ANOVA and PCA.
- Quantitative analysis has been done by combining the five ML algorithms with ANOVA and PCA to yield different hybrid approaches.
- Critical assessment is done by comparing the similar works mentioned in the literature.

3 Methodology

An overview of the proposed methodology is demonstrated by Fig. 1. The main objective of our proposal is to explore a hybrid approach to achieve high accuracy. The methodology contains data collection, preprocessing, feature extraction, labelling, feature selection and ML algorithm are presented systematically in the framework.

3.1 Data Collection

A sample dataset has been collected for the extraction of service port numbers. The sample dataset is collected by running dedicated P2P applications which include BitTorrent, YuppTV, PPTV, BBC, Funshion, Vuze, tubi, Miro, Skype, AajTak, QQplayer, iQIYI, μTorrent, Hotstar, Tribler, YouTube and Gnutella. The National Institute of Technology Sikkim ICT infrastructure was used. The institute provided more than 10 public IP addresses to collect the traces. Private IP addresses are also being used when we captured the data from the peers at the Computer Network Laboratory of the Institute. $Wireshark$ is used to collect the sample dataset and saved it as PcapNG file. Desktop PCs with the processor @3.20 GHz in Windows environment were used for the purpose. The collected dataset was analysed, and a list of P2P ports was prepared to be used for labelling of training data.

However, for the sake of comparative analysis, the popular dataset $UNIBS$ was obtained from the author [29, 30]. The $UNIBS$ dataset was generated from work-

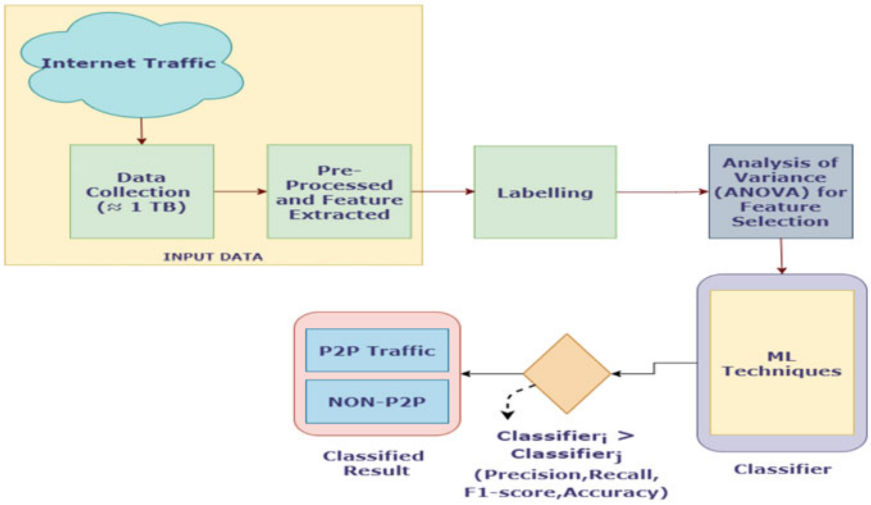

Internet Traffic
Data Collection (≈ 1 TB)
Pre-Processed and Feature Extracted
INPUT DATA
Labelling
Analysis of Variance (ANOVA) for Feature Selection
ML Techniques
Classifier
$\text{Classifier}_i > \text{Classifier}_j$ (Precision,Recall, F1-score,Accuracy)
P2P Traffic
NON-P2P
Classified Result

Fig. 1 Framework for proposed methodology

stations located at the University of Brescia (UNIBS) in Italy in September and October 2009. Tcpdump is used to capture these traces which include the classes, such as Web, Mail, P2P, SKYPE and others. Details of UNIBS traces are mentioned in Table 1. This dataset is used in our work to determine the performances of proposed approaches.

3.2 *Preprocessing of Data and Feature Extraction*

Preprocessing of data is an important task as it handles and filters the input data by removing redundant, duplicate, irrelevant and/or noisy features to create a set of patterns. Python scripts were written for the preprocessing of data and feature extraction. In preprocessing, flow information is determined from the collected raw data. A flow consists of many packets with the same source IP address, destination IP address, protocol, source port and destination port. Since network link adopts bidirectional communication between source and destination. The packets in a flow are divided into uplink and downlink packets. The next important task is the feature extraction

Table 1 Details of $UNIBS$ datasets

Dataset	Data size
unibs20090930.anon	317 MB
unibs20091001.anon	236 MB
unibs20091002.anon	1.94 GB

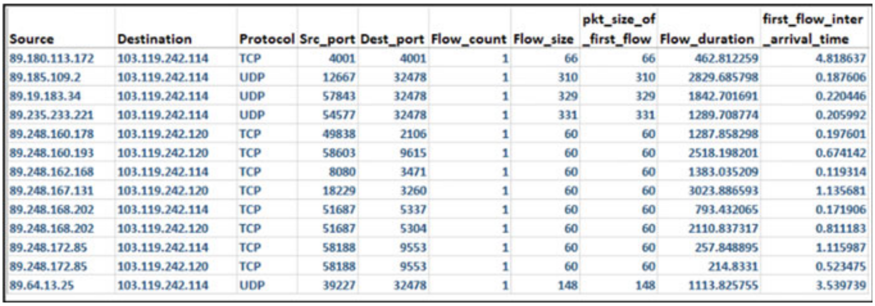
Table 2 List of extracted traffic features from the UNIBS datasets

Feature No.	Feature name	Description
1	Src_ip	Source IP address
2	Dst_ip	Destination IP address
3	Protocol	Transaction protocol (TCP, UDP)
4	Src_port	Source port address
5	Dst_Port	Destination port address
6	Flow_count	Nos. of times a particular flows appears
7	Flow_size	Total sent or received data by a particular flow
8	Pkt_size_of_first_flow	Size of a packet when it appears first in a flow
9	Flow_duration	Total flow duration
10	Flow_inter_arrival_time	Inter arrival time of flows

for identifying the network traffic. The python code was written, and the features were extracted as given in Table 2. A snapshot of the obtained features/attributes from the flow after preprocessing is also given in Fig. 2.

3.3 Feature Selection Techniques

Machine learning in generally works on a large and concise dataset. But using data of huge dimensionality has various pitfalls among which the major one is the curse of dimensionality [31]. It increases the computation time, makes data preprocessing and exploratory data analysis (EDA) more convoluted. This is due to the presence of redundant features in the dataset and inconsistencies in the features. Techniques



Source	Destination	Protocol	Src_port	Dest_port	Flow_count	Flow_size	pkt_size_of_first_flow	Flow_duration	first_flow_inter_arrival_time
89.180.113.172	103.119.242.114	TCP	4001	4001	1	66	66	462.812259	4.818637
89.185.109.2	103.119.242.114	UDP	12667	32478	1	310	310	2829.685798	0.187606
89.19.183.34	103.119.242.114	UDP	57843	32478	1	329	329	1842.701691	0.220446
89.235.233.221	103.119.242.114	UDP	54577	32478	1	331	331	1289.708774	0.205992
89.248.160.178	103.119.242.120	TCP	49838	2106	1	60	60	1287.858298	0.197601
89.248.160.193	103.119.242.120	TCP	58603	9615	1	60	60	2518.198201	0.674142
89.248.162.168	103.119.242.114	TCP	8080	3471	1	60	60	1383.035209	0.119314
89.248.167.131	103.119.242.120	TCP	18229	3260	1	60	60	3023.886593	1.135681
89.248.168.202	103.119.242.114	TCP	51687	5337	1	60	60	793.432065	0.171906
89.248.168.202	103.119.242.120	TCP	51687	5304	1	60	60	2110.837317	0.811183
89.248.172.85	103.119.242.114	TCP	58188	9553	1	60	60	257.848895	1.115987
89.248.172.85	103.119.242.120	TCP	58188	9553	1	60	60	214.8331	0.523475
89.64.13.25	103.119.242.114	UDP	39227	32478	1	148	148	1113.825755	3.539739

Fig. 2 Snapshot of a sample data set after preprocessing in csv format

are required to filter out significant features needed for training purposes. There are various approaches available for the purpose such as chi square test, analysis of variance (ANOVA), principal component analysis (PCA), evolutionary algorithms such as genetic algorithms (GAs) and particle swarm optimization (PSO). In this work, ANOVA and PCA are considered as feature selection techniques.

3.4 Machine Learning Algorithms

Machine learning algorithms are extensively used in almost all domains and so in network traffic identification. In this research work, the proposed approach also makes use of ML techniques to improve the efficiency of P2P traffic identification. A large number of ML algorithms are available in the literature. However, in the present work we have used decision tree (DT), random forest (RF), naive Bayes (NB), K-nearest neighbour (KNN), support vector machine (SVM). These techniques are rewritten to comply with our simulation environment. Here the implementation was done using python. A comparative analysis of the hybrid approaches designed has been done to determine the best solution among all. The performance evaluation parameters used are precision, recall, f1-score and accuracy. Details are discussed in Sect. 4.

3.5 Port Analysis and Labelling the Data

A training data set is essentially required in ML-based approaches. In the proposed approaches to identify the P2P traffic, the labelling of training data is carried out based on the port number. As revealed from the literature survey, the port-based approaches are better suited for data labelling due to ease of implementation and high accuracy. The process of port analysis and labelling the dataset is presented graphically through Fig. 3 for ease of understanding.

For labelling the training datasets, a list of known P2P port numbers is prepared considering both source and destination port. The list includes well-known, registered, ephemeral ports. The list prepared from our own collected dataset was further extended by including the port numbers gathered from the literature [3, 27, 34–36] which are used in similar research. The list of port numbers prepared by us is very large as compared to the list used by other researchers. The extracted port numbers in the prepared list are more than 22,000 and are not possible to report here. Therefore, a glimpse of the port number list is given in Tables 3 and 4.

Further, while extracting the port numbers, it is observed that the service port numbers were repeating in multiple P2P applications due to dynamic port assignment. However, in preparing the list of port numbers, it was ensured that it belongs to P2P traffic as we have only run P2P applications while collecting the sample dataset.

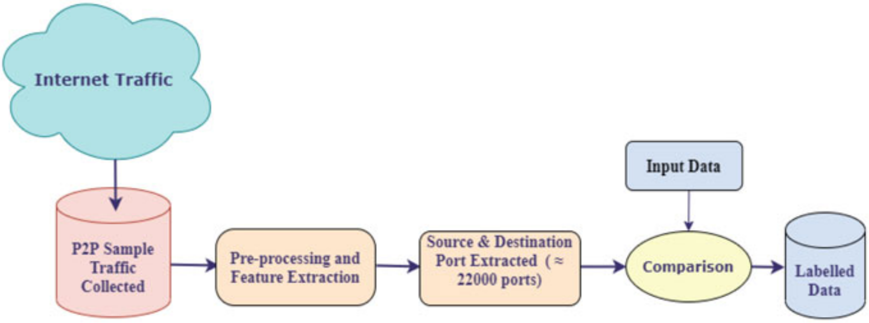

Internet Traffic
P2P Sample Traffic Collected
Pre-processing and Feature Extraction
Source & Destination Port Extracted (≈ 22000 ports)
Input Data
Comparison
Labelled Data

Fig. 3 Ports analyses and labelling the data

Table 3 Port numbers used by popular P2P applications

P2P applications	Port numbers
BitTorrent	6881–6889
Edonkey (eMule, xMule)	2323, 3306, 4242, 4500, 4501, 4661–4674, 4677, 4678, 4711, 4712, 7778
Gnutella	6346, 6347
FastTrack	1214, 1215, 1331, 1337, 1683, 4329
DirectConnect (DC++)	411, 412, 1364–1383, 4702, 4703, 4662
Napster (File Navigator, WinMx)	5555, 6666, 6677, 6688, 6699–6701, 6257
Freenet	19114, 8081
Blubster	41,170-41,350
GoBoogy	5335
HotLine	5500–5503
ICQ	5190
IRC	7000, 7514, 6667
XMPP	5222, 5269
SoulSeek	2234, 5534
QNext	5235–5237

Based on port numbers, the UNIBS dataset has labelled into two classes P2P and non-P2P.

4 Performance Evaluation

To carry out simulations and comparative analysis, the datasets were converted into CSV files. The flow-based features were extracted by using the codes written in Python. The simulation was done on GPU (4 Cores), with processor@3.80GHz and 64 GB Memory. To get a better feature subset, ANOVA and PCA have been

Table 4 Port numbers extracted from the sample dataset which was collected by running P2P applications

P2P applications	Port numbers
Skype	57,290, 56,091, 41,900, 55,303, 61,976, 62,015, 45,220, 59,774, 16,130, 10,131, 34,625, 25,406, 8999, 35,133, …
Funshion	64,018, 1153, 48,413, 52,347, 38,859, 56,481, 13,257, 29,560, 48,403, 4501, 61,651, 54,289, 44,403, 29,471, 60,308, …
Miro	50,542–50,544, 53,778–53,785, 61,969, 61,970, 50,545, 53,791, 51,549, 51,556, 50,527, 37,385, 51,579, 50,528, …
BitTorrent	40,283, 23,791, 57,605, 34,923, 51,084, 62,383, 55,300–55,308, 34,319, 37,192, 41,011, 45,177, 6771, 41,843 …
Tubi	50,366, 50,367, 50,932, 50,582, 50,587, 50,933–50,939, 50,596, 50,597, 50,604, …
PPTV	50,299, 50,300–50,310, 50,073, 50,072, 5041, …
YuppTV	56,213–56,224, 55,795, 56,000, 55,711, …
AajTak	50,975–50,978, 64,116, 64,117, 50,979, 50,980–50,984, 55,548, 55,549, 50,809, …
YouTube	54,980, 54,979, 50,762, 50,763–50,767, 53,026, 50,768–50,771, 61,049, 61,082, …
Vuze	13,398, 50,614, 57,208, 57,211, 57,212, 57,214, 57,215, 57,218, 57,263, 57,369, 57,126, 57,232, 59,794, 27,175, …
BBC	52,310, 52,311–52,315, 61,196, 33,419, 63,738, 18,340, 39,701, 56,727, 49,183, 50,270, 19,702, …
Hotstar	50,489, 50,490–50,495, 50,769, 61,046, 61,079, 61,090, 63,803, 63,802, 50,496, 50,497, …
Tribler	1130, 35,140, 53,736, 35,190, 35,175, 51,122, 9206, 35,120, 35,130, 51,044, 35,080, 58,476, 2105, 24,934, 24,935, …
Gnutella	63,432, 59,650, 6602, 6791, 50,088, 9216, 47,655, 15,398, 39,961, 6312, 11,553, 10,381, 17,983, 55,088, 9812, …
iQIYI	50,486–50,488, 50,568, 50,481, 50,569, 50,528–50,530, 50,570, 50,571, 50,475, 50,505, 50,428, 50,533, …

implemented. The ML techniques import the selected feature subset for identifying P2P traffic from the given dataset. The proposed hybrid techniques were evaluated based on the following performance matrices [32].

- **True Positive** (T^+): The instances belongs to P2P class, and it is classified correctly.
- **True Negative** (T^-): Instances correctly classified as non-P2P.
- **False Positive** (F^+): Number of P2P that are classified as the non-P2P.
- **False Negative** (F^-): Number of non-P2P that are classified as the P2P.

The metrics based on the above are used for evaluation of performance of classifiers and are defined below:

- **Accuracy**: It measures the capability of classifier to identify positive and negative cases. The overall accuracy of a classifier is estimated by dividing the total correctly classified positives and negatives by the total number of samples.

$$\text{Accuracy} = \frac{T^+ + T^-}{T^+ + T^- + F^+ + F^-} \tag{1}$$

It is difficult to evaluate the performance of the classifier based on accuracy only if the dataset is imbalanced and has large no of (+)ve and (–)ve cases. In that case, the importance is given to the more popular evaluation metrices. Hence, recall and precision are also commonly used [33] metrics for evaluating classifiers. Details as follows:

- **Recall**: It is the true positive rate of a classifier which is estimated by the ratio of the correctly classified positives upon the total positive count. Therefore, recall represents the percentage of overall positive cases present in the dataset. Recall is also called sensitivity [24].

$$\text{Recall} = \frac{T^+}{T^+ + F^-} \tag{2}$$

- **Precision**: It is the false positive rate or false alarm rate of a classifier which is estimated by the ratio of incorrectly classified negatives by the total negatives.

$$\text{Precision} = \frac{T^+}{T^+ + F^+} \tag{3}$$

- **F1-score**: This is the harmonic mean of precision and recall. It is used as a statistical measure to rate performance.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

4.1 Results and Discussion

Table 5 lists the values of precision, recall, f1-score and accuracy obtained after extensive simulation. The performance analysis of the ANOVA and PCA technique and different machine learning algorithms were done and discussed below.

When simulation run for two features, the ANOVA selects *Source Port and Flow Duration*. The accuracy of DT, RF and KNN is more than 91%, whereas accuracy of naive Bayes and SVM ranges from 88 to 89%. In case of PCA, the accuracy ranges between 87 and 92% for all the classifier. The RF with ANOVA outperform other classifiers, and accuracy achieved is 93.66%. Similarly, when we consider three attributes, the ANOVA picked up the Source Port, Flow Duration and Flow Inter Arrival Time. The DT, RF and NB with ANOVA have provided the better result 92.71%, 93.64% and 87.92%, respectively, whereas KNN and SVM furnished the enhanced accuracy of 92.52% and 89.63% for PCA when considered three features.

The simulation was also carried out for four, five and six features also. The results are mentioned in Table 5. It can be observed that almost all the cases ANOVA outperform the PCA. When comparing ML approaches, the RF performs better among all the classifier used in the present research work. It can also be inferred from Table 5 that the accuracy of RF and DT is comparably higher for four and more features. The maximum accuracy achieved is more than 96% for random forest and ANOVA combination, and the selected features are Source Port, Destination Port, Flow Count, First Packet Size of Flow, Flow Duration and Flow Inter Arrival Time. It was tested further for more number of features but as accuracy was low, hence, not reported. For ease of understanding, Figs. 4 and 5 have been drawn and represented the comparison among the classifier in terms of accuracy of P2P traffic identification with different feature subset.

The result obtained in the proposed approach to classify P2P traffic is compared with the similar approaches as reported above. Our proposed approach has achieved accuracy more than 96% using ANOVA feature selection technique and RF classifier. Yan et al. [23] have achieved 93.9% flow accuracy on same UNIBS dataset and approach used based on flow behaviour. The accuracy achieved by Jeffrey et al. [22] 91.70% on AUCK-IV sub-dataset and with limited port numbers. Saber et al. [26] have claimed the accuracy of 93.9% and 94.9% when it takes the flow time-outs of 60s and 120s by using PCA feature selection technique and SVM classifier. Mohammadi et al. [27] have claimed a similar accuracy on his dataset but the approaches used are based on genetic algorithm and KNN classifier on a comparably small dataset. The above comparative analysis reveals that the proposed approaches outperform the reported similar hybrid approaches. It is also noted that the feature selection techniques contributed to enhance the performance of the proposed model.

In the present work, we consider around 20 P2P application and collected more than 22000 number of service port from the captured traces. Further, the list was extended by collecting more registered port from the literature for labelling the training dataset. We feel that the list prepared may be extended to achieve better accuracy.

Table 5 Results: considering UNIBS (unibs20091001) dataset

ML techniques	Precision		Recall		F1-score		Accuracy	
	ANOVA	PCA	ANOVA	PCA	ANOVA	PCA	ANOVA	PCA
Feature selected: 02			Feature's name: Src_Port and Flow_duration					
Decision Tree	93	90	93	90	93	90	92.85	89.85
Random Forest	93	91	94	92	94	92	93.66	91.83
Naive Bayes	85	83	88	87	85	84	88.17	87.49
KNN	91	91	92	92	91	92	91.69	91.77
SVM	88	89	89	89	86	86	89.11	89.29
Feature selected:03			Feature's name: Src_Port, Flow_IAT and Flow_duration					
Decision Tree	93	92	93	92	93	92	92.71	91.59
Random Forest	93	93	94	93	93	93	93.64	93.23
Naive Bayes	85	83	88	87	85	84	87.92	87.49
KNN	90	92	91	93	90	92	90.65	92.52
SVM	88	89	89	90	87	87	89.34	89.63
Feature selected:04			Feature's name: Src_Port, Dst_Port, Flow_duration and Flow_IAT					
Decision Tree	95	93	95	93	96	93	95.45	93.12
Random Forest	96	94	96	94	96	94	95.79	94.43
Naive Bayes	85	83	88	87	85	84	87.97	87.48
KNN	95	93	94	93	95	93	94.50	93.29
SVM	89	89	90	90	89	87	90.49	89.83
Feature selected:05			Feature's name: Src_Port, Dst_Port, Flow_duration, First_Pkt_Size_in_flow and Flow_IAT					
Decision Tree	95	93	95	93	95	93	95.44	93.22
Random Forest	96	94	96	95	96	94	95.92	94.55
Naive Bayes	85	82	88	86	85	84	87.98	85.82
KNN	94	93	94	94	94	93	94.33	93.53
SVM	90	89	91	90	89	88	90.78	90.42
Feature selected:06			Feature's name: Src_Port, Dst_Port, flow_count, First_Pkt_Size_in_flow, Flow_duration and Flow_IAT					
Decision Tree	95	93	95	93	95	93	95.31	93.17
Random Forest	96	94	96	95	96	95	**96.06**	94.62
Naive Bayes	85	82	88	86	85	84	87.91	85.56
KNN	94	93	94	94	94	93	94.49	93.57
SVM	90	90	91	91	89	89	90.79	90.56

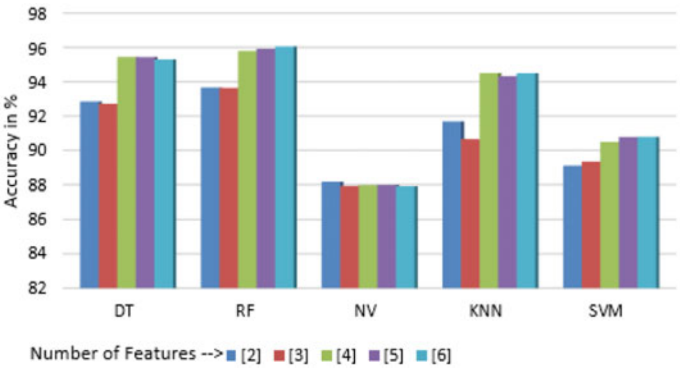


Fig. 4 Comparison of accuracy obtained with the different number of features selected using ANOVA and classified by DT, RF, NV, KNN and SVM on UNIBS dataset

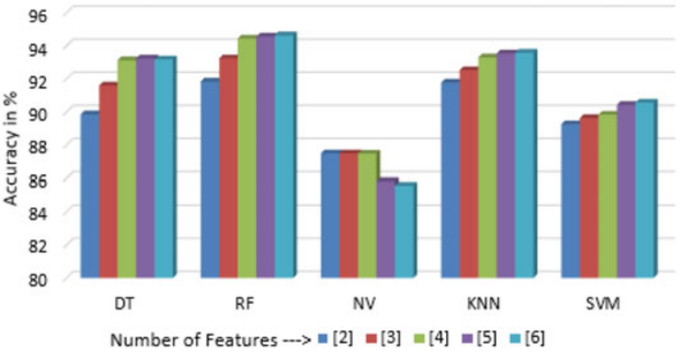


Fig. 5 Comparison of accuracy obtained with the different number of features selected using PCA and classified by DT, RF, NV, KNN and SVM on UNIBS dataset

5 Conclusion and Future Direction of Work

To improve network performance and traffic management, the traffic identification is required. In this work, we have studied the effect of feature selection and machine learning techniques for the identification of P2P traffic and proposed hybrid approaches by amalgamating port-based, feature selection and machine learning techniques. Port-based approach is used to label the dataset, the ANOVA and PCA assisted in selection of the best feature subset among all extracted features. The five ML techniques are combined with ANOVA and PCA to yield different hybrid approaches. The extensive simulation is performed to compare all the approaches developed using different combinations. The results have been analysed, and it is concluded that the random forest classifier with ANOVA outperforms the other proposed approaches and similar approaches reported in the literature. The maximum accuracy achieved is more than 96% of accuracy to correctly identify the P2P traffic.

During the execution of this work, it is realized that simply identifying P2P traffic may not be sufficient, the need for fine-grained classification may emerge in near future. A generic approach may also need to be developed which can identify the new applications as well as existing P2P applications so that network traffic can be managed in a better way. Further, the study can be extended towards other hybrid approaches to classify P2P traffic and address the challenges of P2P applications such as selfish peer, botnet and the flash crowd.

References

1. Thampi SM (2013) A review on P2P video streaming. arXiv preprint arXiv:1304.1235
2. Pal K, Govil MC, Ahmed M, Chawla T (2019) A survey on adaptive multimedia streaming. In: Recent trends in communication networks. IntechOpen
3. Gomes JV, Inácio PRM, Pereira M, Freire MM, Monteiro PP (2013) Detection and classification of peer-to-peer traffic: a survey. ACM Comput Surv (CSUR) 45(3):1–40
4. Bhatia M, Rai MK (2017) Identifying P2P traffic: a survey. Peer-to-Peer Netw Appl 10(5):1182–1203
5. Marfia G, Pau G, Di Rico P, Gerla M (2007) P2P streaming systems: a survey and experiments. ST J Res 1–4
6. Pal K, Govil MC, Ahmed M (2015) A new hybrid approach for overlay construction in P2P live streaming. In: 2015 international conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 431–437
7. Pal K, Govil MC, Ahmed M (2019) FLHyO: fuzzy logic based hybrid overlay for P2P live video streaming. Multimed Tools Appl 78(23):33679–33702
8. Pal K, Govil MC, Ahmed M (2019) Utilization-based hybrid overlay for live video streaming in P2P network. Recent findings in intelligent computing techniques. Springer, Singapore, pp 331–338
9. Pal K, Govil MC, Ahmed M (2018) Slack time-based scheduling scheme for live video streaming in P2P network. Int J Commun Syst 31(2):e3440
10. Pal K, Govil MC, Ahmed M (2018) Priority-based scheduling scheme for live video streaming in peer-to-peer network. Multimed Tools Appl 77(18):24427–24457
11. Jin Y, Kesidis G, Shin J, Kocak F, Yi Y (2014) Impacts of selfish behaviors on the scalability of hybrid client-server and peer-to-peer caching systems. IEEE/ACM Trans Netw 23(6):1818–1831
12. Pal K, Govil MC, Ahmed M (2016) Comparative analysis of new hybrid approach for overlay construction in P2P live streaming. In: International conference on emerging research in computing, information, communication and applications. Springer, Singapore, pp 239–250
13. Pal K, Ahmed M, Govil MC (2017) Utilization based hybrid overlay approach for P2P live streaming: a comparative analysis. In: Proceedings of the international conference on parallel and distributed processing techniques and applications (PDPTA). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), pp 46–51
14. Pal K, Ahmed M, Govil MC (2017) A distinctive analysis of new hybrid overlay approach with classical overlay approaches for P2P live streaming. In: Proceedings of the international conference on parallel and distributed processing techniques and applications (PDPTA). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), pp 32–38
15. Liu H, Feng W, Huang Y, Li X (2007) A peer-to-peer traffic identification method using machine learning. In: 2007 international conference on networking, architecture, and storage (NAS 2007). IEEE, pp 155–160

16. Haq IU, Ali S, Khan H, Khayam SA (2010) What is the impact of P2P traffic on anomaly detection? International workshop on recent advances in intrusion detection. Springer, Berlin, Heidelberg, pp 1–17
17. Ansari MSA, Govil MC, Pal K, Samaddar (2020) Tools and techniques in optimization of network resources. In: 2020 3rd international conference on emerging technologies in computer engineering: machine learning and internet of things (ICETCE). IEEE, pp 1–7
18. Moore AW, Papagiannaki K (2005) Toward the accurate identification of network applications. In: International workshop on passive and active network measurement. Springer, Berlin, Heidelberg, pp 41–54
19. Park B, Won Y, Chung JY, Kim M-s, Hong JW-K (2013) Fine-grained traffic classification based on functional separation. Int J Netw Manag 23(5):350-*381
20. Saroiu S, Gummadi KP, Dunn RJ, Gribble SD, Levy HM (2002) An analysis of internet content delivery systems. ACM SIGOPS Oper Syst Rev 36(SI):315–327
21. Roughan M, Sen S, Spatscheck O, Duffield N (2004) Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification. In: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, pp 135–148
22. Erman J, Mahanti A, Arlitt M (2006) Qrp05-4: Internet traffic identification using machine learning. In: IEEE Globecom 2006. IEEE, pp 1–6
23. Yan J, Wu Z, Luo H, Zhang S (2013) P2P traffic identification based on host and flow behaviour characteristics. Cybern Inf Technol 13(3):64–76
24. Raahemi B, Zhong W, Liu J (2008) Peer-to-peer traffic identification by mining IP layer data streams using concept-adapting very fast decision tree. In: 2008 20th IEEE international conference on tools with artificial intelligence, vol 1. IEEE, pp 525–532
25. Draper-Gil G, Lashkari AH, Mamun MSI, Ghorbani AA (2016) Characterization of encrypted and VPN traffic using time-related. In: Proceedings of the 2nd international conference on information systems security and privacy (ICISSP), pp 407–414
26. Saber A, Fergani B, Abbas M (2018) Encrypted traffic classification: combining over-and under-sampling through a PCA-SVM. In: 2018 3rd international conference on pattern analysis and intelligent systems (PAIS). IEEE, pp 1–5
27. Mehdi M, Bijan R, Ahmad A, Hossein M, Babak N (2011) Genetic-based minimum classification error mapping for accurate identifying Peer-to-Peer applications in the internet traffic. Expert Syst Appl 38(6):6417–6423
28. Junior GPS, Maia JEB, Holanda R, de Sousa JN (2007) P2P traffic identification using cluster analysis. In: 2007 first international global information infrastructure symposium. IEEE, pp 128–133
29. Maurizio D, Francesco G, Luca S (2011) Quantifying the accuracy of the ground truth associated with Internet traffic traces. Comput Netw 55(5):1158–1167
30. Gringoli F, Salgarelli L, Dusi M, Cascarano N, Risso F, Claffy KC (2009) Gt: picking up the truth from the ground for internet traffic. ACM SIGCOMM Comput Commun Rev 39(5):12–18
31. Verleysen M, François D (2005) The curse of dimensionality in data mining and time series prediction. In: International work-conference on artificial neural networks. Springer, Berlin, Heidelberg, pp 758–770
32. Olson DL, Delen D (2008) Advanced data mining techniques. Springer
33. Nguyen TTT, Armitage G (2008) A survey of techniques for internet traffic classification using machine learning. IEEE Commun Surv Tutor 10(4):56–76
34. Internet Assigned Numbers Authority (IANA) [Online]. https://www.iana.org/assignments/service-names-port-numbers. Accessed 10 June 2020
35. Michelle C, Lars E, Joe T, Magnus W, Stuart C (2011) Internet Assigned Numbers Authority (IANA) procedures for the management of the service name and transport protocol port number registry. RFC 6335:1–33
36. Karagiannis T, Broido A, Faloutsos M, Claffy KC (2004) Transport layer identification of P2P traffic. In: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, pp 121–134

Spotted Hyena Optimization (SHO) Algorithm-Based Novel Control Approach for Buck DC–DC Converter-Fed PMBLDC Motor

Deepak Paliwal and Dhanesh Kumar Sambariya

Abstract In this study, we implemented an SHO algorithm-based PI controller design for a buck DC–DC converter fed by a PMBLDC motor. This study was carried out using the new control system using current multiplier control. In the traditional system, the PMBLDC motor was operated using two loop mechanisms; first for the control of the converter and second for the control of rpm. In this study, first, we designed the DC-link control of the motor and removed two control loops and the mechanism subjected to lower total harmonic distortions. In addition to improving the power quality and speed control of the motor, we proposed the SHO algorithm to optimize PI controllers in the inner loop and outer loop for the buck converter. The problem is considered as multi-objective optimization resolved using the weighted sum technique. All results obtained have been performed using MATLAB-16b.

Keywords Buck converter · PI controller · Permanent magnet brushless DC motor · Spotted hyena optimization algorithm

1 Introduction

Over the last few years, the growing complexity of real-world problems has given rise to the need for superior evolutionary algorithms. They were used to achieve the best possible strategies for real-life engineering issues [1–3]. They became more popular because of their effectiveness and difficulty relative to other established classical methods [4, 5]. The motivation behind this work is the application potential of nature-inspired electrical drives.

Higher efficiency, lower maintenance and broad speed range are the key aspects of the PMBLDC engine that make it the most suitable drive for low and medium-power applications in the industry [6]. PMBLDC engines have a broad spectrum of applications opportunities such as electrical vehicles [7, 8], aviation [9], home appliances [10], health care [11], robotic systems [12] and renewable power [13, 14].

D. Paliwal (✉) · D. K. Sambariya
Electrical Engineering, University Department, Rajasthan Technical University, Kota, Rajasthan 324010, India
e-mail: dpaliwal.phd17@rtu.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_31

The buck converter is a DC–DC switching converter that converts output voltage to lower than the input voltage. The name of the step-down converter comes from the fact that the input voltage is lower than the input voltage like a step-down transformer [15, 16].

Generally, the PMBLDC motor drive is integrated into a single-phase AC supply through the diode bridge rectifier for low and medium power applications, which decreases its operating costs [17–19]. This configuration draws the pulsed input current from the AC mains, which induces higher THD and low power factor. So power factor correction converters are used to solve this problem [20, 21].

There are two PFC converter control modes, namely current multiplier control and voltage follower control. The voltage follower control mode is lacking due to the disadvantage of higher current stress on the switches of the converters [18, 22].

In this study, we used the SHO algorithm [23] to optimize the PI controller to accomplish THD reduction and speed control of the PMBLDC motor as an objective function based on the weighted sum technique for solving multi-objective optimization [24, 25]. The traditional control using a two-loop control is shown in Fig. 1 and the planned work shall be outlined as shown in Fig. 2. The article is organized into five sections. The problem formulation is shown in Sect. 2. Section 3 presents the controller design and SHO algorithm. All the results are shown in Sect. 4. At last, the article conclusion is provided in Sect. 5 followed by references.

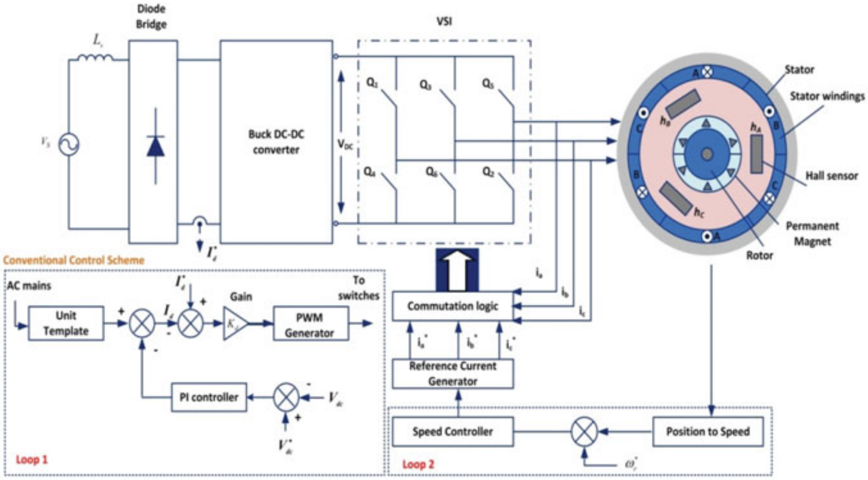


Fig. 1 Conventional control scheme of buck PFC-fed PMBLDC motor

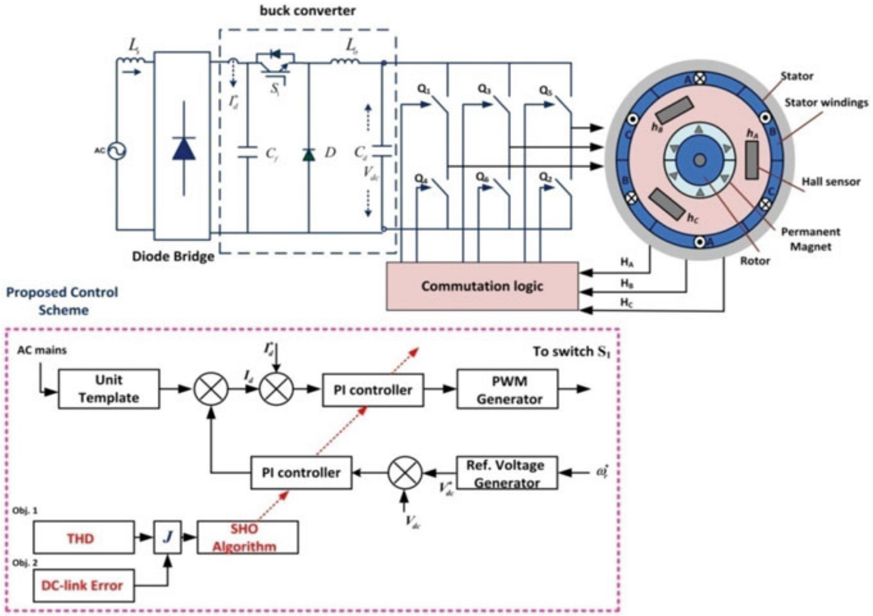


Fig. 2 Proposed control scheme of buck PFC converter-fed PMBLDC motor

2 Problem Formulation

2.1 Mathematical Modelling of Buck Converter

When switch S is “ON”, the diode acts in reverse bias and the current flows through the inductor as shown in Fig. 3a. When switch S is “OFF” the diode work in the forward bias mode as shown in Fig. 3b. In this work, the values of parameters are as follows: $L_o = 2$ µH, $C_o = 1500$ µF and filter $C_f = 7.5$ µF.

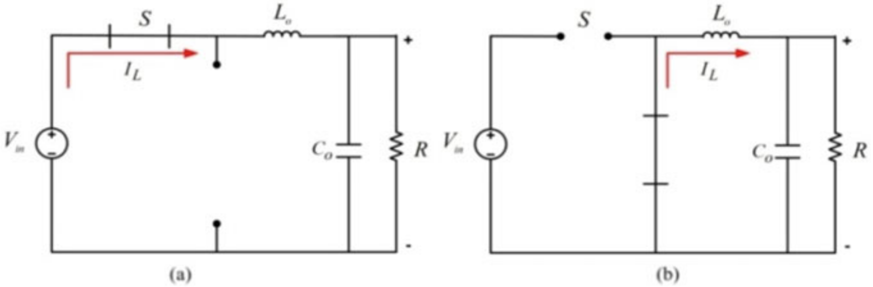


Fig. 3 Operation of buck converter (**a**) when switch S is “ON” and (**b**) when switch S is “OFF”

$$V_o = D.V_{in} \tag{1}$$

$$L_o = V_o(1 - D)/(\Delta I_{Lo}).f_s \tag{2}$$

$$C_o = (1 - D)/(8L_o f_s^2)(\Delta V_{Co}/V_o) \tag{3}$$

2.2 Mathematical Modelling of VSI-Fed PMBLDC Motor

The modelling of PMBLDC motor can be represented as a set of differential equations as follows:

$$pi_x = \frac{(v_x - i_x.R - e_x)}{(L_s + M)} \tag{4}$$

$$p\omega_r = \frac{(P/2)(T_e - T_l - B.\omega_r)}{(J)} \tag{5}$$

$$p\theta = \omega_r \tag{6}$$

$$e_x = K_b.f_x(\theta).\omega_r \tag{7}$$

$$\begin{array}{ll} f_a(\theta) = 1 & for\, 0 < \theta < 2\pi/3 \\ f_a(\theta) = 1\{(6/\pi)(\pi - \theta)\} - 1 & for 2\pi/3 < \theta < \pi \\ f_a(\theta) = -1 & for\, \pi < \theta < 5\pi/3 \\ f_a(\theta) = \{(6/\pi)(\pi - \theta)\} + 1 & for\, 5\pi/3 < \theta < 2\pi \end{array} \tag{8}$$

where p = differential operator (d/dt), i_x = current in phase x, L_s = self-inductance / phase, R = resistance of motor winding / phase, P = number of poles, ω_r = motor speed in radian / second, J = inertia, T_l = load torque, T_e = electromagnetic torque, e_x = back emf as a function of θ, $f_x(\theta)$ = function of rotor position..

3 Controller Design and Algorithm

3.1 Objective Function

The objective function is based on the weighted-sum technique for solving multi-objective optimization as a single objective function. In this method, each objective function is assigned a weight on a priority basis and at last, each objective is combined to make a single objective function. For this work, equal weight is assigned to every objective.

$$\begin{aligned} J_{Minimize} &= \sum_{i=1}^{i=2} w_i.|J_i| \\ &= w_1.J_1 + w_2.J_2 \\ &= \frac{0.5J_1}{100} + 0.5J_2 \\ &= 0.5(\frac{\sqrt{I_{rms}^2 - I_o^2 - I_{1\,rms}^2}}{I_{1\,rms}}) + 0.5\,ISE_{DC\,link\,Error} \\ &= 0.5(THD)\% + 0.5\int_0^{T_{sim}} \left|V_{dc}^*(t) - V_{dc}(t)\right|^2.dt \end{aligned} \tag{9}$$

$$\begin{aligned} Subjected\ \ To:\ \ & w_1 + w_2 = 1 \\ & 0 \le THD \le\ 3\% \\ & 0 \le V_{dc}^* - V_{dc} \le 50 \end{aligned}$$

3.2 Spotted Hyena Optimization (SHO) Algorithm

The spotted hyena optimization (SHO) algorithm is motivated by the social hierarchy of spotted hyenas and their hunting behaviour. Collaborative clusters can help to ensure successful cooperation among spotted hyenas. The superiority of the SHO algorithm shows 29 benchmark functions in terms of exploration and exploitation. The steps are shown in Fig. 4. In the following steps, the simulation of SHO can be represented by:

- Encircling the prey
- Hunting
- Attacking the prey (exploitation)
- Search for the prey(exploration)

Encircling prey

At this time, target prey is the better option for mathematically modelling the social hierarchy of spotted hyenas. The other quest agents can resume updating their positions until the best search has been decided.

The following equations represent the mathematical model of this behaviour:

$$\overrightarrow{S_h} = \left|\overrightarrow{B}.\overrightarrow{F_f}(y) - \overrightarrow{F(y)}\right| \tag{10}$$

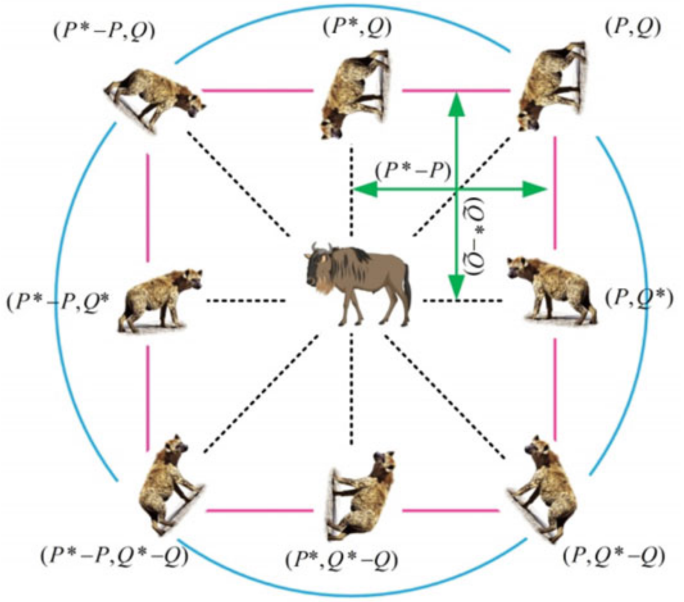


Fig. 4 “2D” representation of spotted hyena for attacking prey

$$\vec{F}(y+1) = \vec{F}_f(y) - \vec{E}.\vec{S}_h \tag{11}$$

$$\vec{B} = 2.r\vec{d}_1 \tag{12}$$

$$\vec{E} = 2\vec{h}.r\vec{d}_2 - \vec{h} \tag{13}$$

$$\vec{h} = 5 - (Iter. * (5/Max_{Iter.})) \tag{14}$$

where
$\vec{S}_h$ = distance between the prey and spotted hyena, y = current iteration, $\vec{B}$ and $\vec{E}$ = coefficient vectors, $\vec{F}_f$ = position vector of prey, $\vec{F}$ = position vector of spotted hyena, $\vec{h}$ = linearly decreased from 5 to 0.

As shown in Fig. 4, the spotted hyena (P, Q) will update its location towards the prey location $(P*, Q*)$ by changing the value of the vector $\vec{B}\,\vec{E}$.

Hunting

It is assumed that the best search agent has information about the position of prey. The following equations mathematically describe the behaviour of spotted hyenas as follows:

$$\overrightarrow{S}_h = \left|\overrightarrow{B}.\overrightarrow{F_h} - \overrightarrow{F}_k\right| \tag{15}$$

$$\overrightarrow{F}_k = \overrightarrow{F}_h - \overrightarrow{E}.\overrightarrow{S}_h \tag{16}$$

$$\overrightarrow{T}_h = \overrightarrow{F}_k + \overrightarrow{F}_{k+1} + \cdots + \overrightarrow{F}_{k+N} \tag{17}$$

$$N = count_{nos}(\overrightarrow{F}_h, \overrightarrow{F}_{h+1} \ldots (\overrightarrow{F}_h + H)) \tag{18}$$

$\overrightarrow{F}_h$ = location of first best-spotted hyena, $\overrightarrow{F}_k$ = location of other spotted hyenas. N = number of hyenas, $\overrightarrow{H}$ = random vector in [0.5, 1, $\overrightarrow{T}_h$ = Group of N number of best solutions, nos = the number of solutions and count all search agent solutions, after addition with $\overrightarrow{H}$.

Attacking prey (exploitation)

To strike the target, the arithmetic formulation is as follows:

$$\overrightarrow{F}(y+1) = \frac{\overrightarrow{T}_h}{N} \tag{19}$$

$\overrightarrow{F}(y + 1)$ saves and updates the best solution and changes all search agents location according to the best search agent.

Search for prey (exploration)

$\overrightarrow{E} > 1$ helps the spotted hyenas to shift away from the prey. $\overrightarrow{E} < 1$ pushes the spotted hyena unit to attack the prey. Here $\overrightarrow{B} > 1$ is considered to avoid local optima (Fig. 5).

4 Simulation Results

The simulation results are shown in Figs. 6, 7, 8, 9 and 10. To evaluate the performance of the SHO swarm-optimized controller, the results are compared with the conventional PI controller-based proposed scheme. Figure 6 shows the dynamic performance of drive for DC-link voltage. It shows that the SHO-PI-based drive has less time to settle the desired output voltage of the converter. Figure 7 shows the response of drive for unit step input to speed. The SHO-PI controller-based system has a lower settling time and lower overshoot compared to the conventional PI controller-based drive in Fig. 8. Figure 9 shows the THD values of the proposed schemes. The THD results are compared in the table. At last, Fig. 10 shows the fitness cost of the objective function and shows the qualitative performance of the algorithm for solving the objective function in terms of exploration and exploitation. Table 1 shows the controller gain values of the inner loop control and outer loop control of the proposed scheme. The THD values of proposed schemes are compared with the

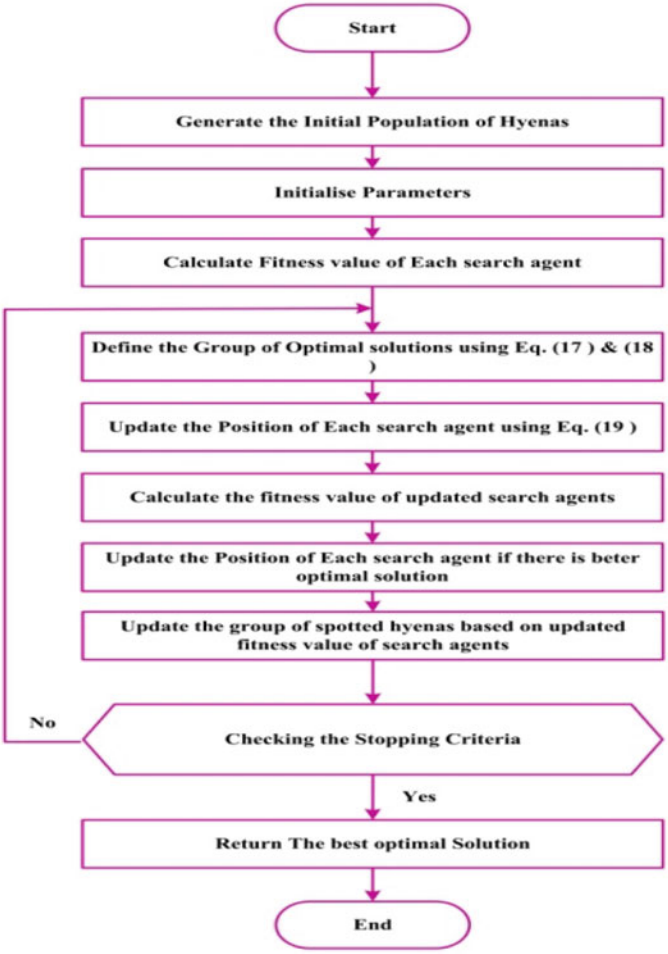

Start
Generate the Initial Population of Hyenas
Initialise Parameters
Calculate Fitness value of Each search agent
Define the Group of Optimal solutions using Eq. (17) & (18)
Update the Position of Each search agent using Eq. (19)
Calculate the fitness value of updated search agents
Update the Position of Each search agent if there is beter optimal solution
Update the group of spotted hyenas based on updated fitness value of search agents
No
Checking the Stopping Criteria
Yes
Return The best optimal Solution
End

Fig. 5 Flowchart of SHO algorithm

scheme available in the literature in Table 2. In Table 3 dynamic performance of the proposed schemes is compared. Table 4 shows the parametric setting of the algorithm used to propose work.

5 Conclusion

This work proposed DC-link control of the PMBLDC motor and its optimal control using the SHO algorithm. In this work, we reduced the input current THD and achieved speed control of the motor by improving its dynamic performance compared

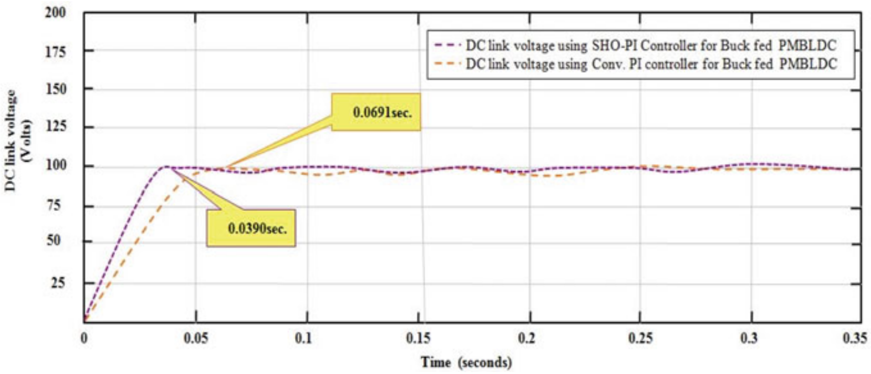


Fig. 6 Dynamic performance of DC-link voltage

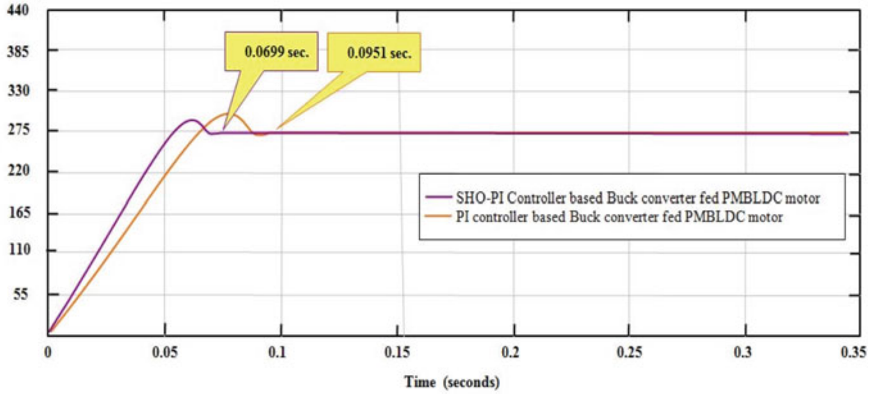


Fig. 7 Unit step response of speed for buck converter-fed PMBLDC motor

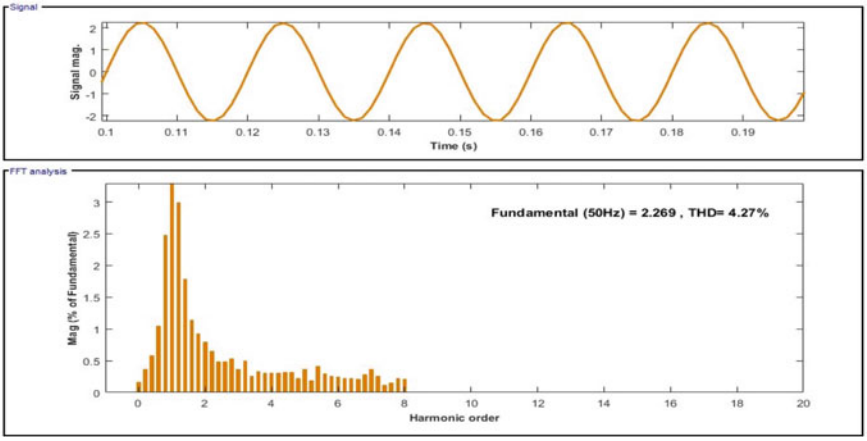


Fig. 8 THD spectrum of PI controller with DC-link control-based proposed scheme

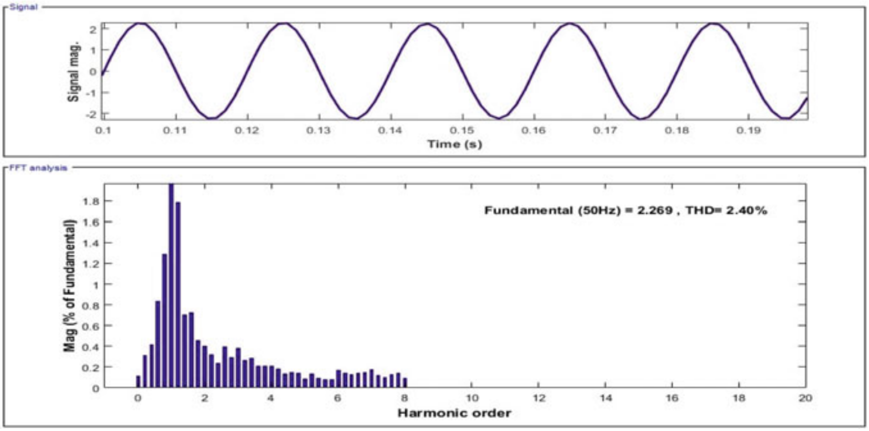


Fig. 9 THD spectrum of SHO-PI controller with DC-link control-based proposed scheme

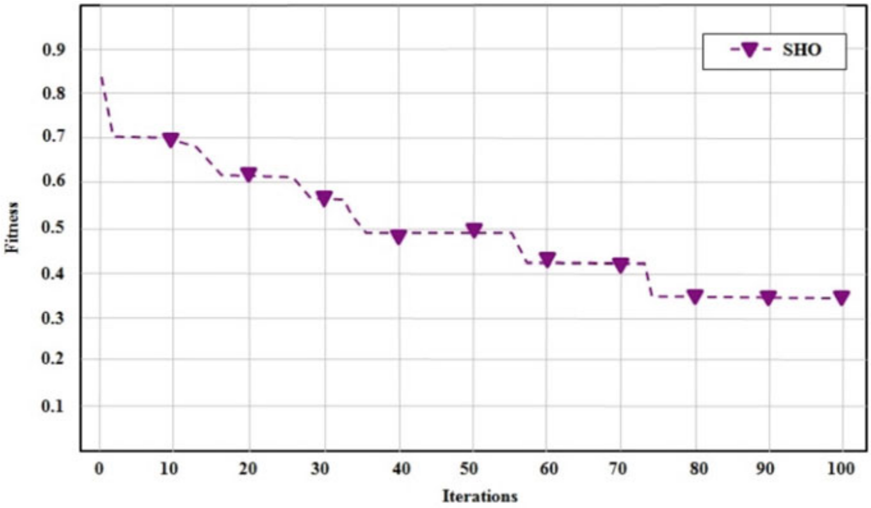


Fig. 10 Fitness graph of SHO algorithm for the proposed scheme

Table 1 Controller gain values

	Inner loop control parameters		Outer loop control parameters	
Controller	$K_{p,\text{ILC}}$	$K_{i,\text{ILC}}$	$K_{p,\text{OLC}}$	$K_{i,\text{OLC}}$
SHO-PI controller	1.988	1.965	0.721	1.413
Convt.-PI controller	NA	NA	0.185	1.85

Table 2 THD analysis of various schemes

Scheme	Input current THD %
SHO-PI with DC-link (Proposed)	2.40%
Conventional PI with DC-link (Proposed)	4.27%
Conventional PI without DC-link control [6]	10.55%

Table 3 Comparative performance analysis of proposed schemes

Controller	Dynamic response of DC-link voltage (s)	Speed control indices	
		Settling Time (S)	Overshoot %
SHO-PI with DC-link (Proposed)	0.0390	0.0699	4.363%
Conventional PI with DC-link (Proposed)	0.0691	0.0951	8.727%

Table 4 Parametric setting of SHO algorithm

Algorithm	Parameter	Value
Spotted hyena optimization (SHO) algorithm	Search agents	40
	Control parameter $\overrightarrow{h}$	[5,0]
	Constant $\overrightarrow{M}$	[0.5,1]
	Generations	100

to the system available in the literature. The findings indicate the feasibility of the proposed framework scheme. The work can be extended using hybrid algorithms as a future scope.

Appendix

Rated power = 1500 W, rated voltage = 400 V, rated speed = 1500 rpm, rated torque = 10 Nm, resistance = 2.8 Ω/phase, inductance = 5.21×10^{-3} H/phase, switching frequency = 20 kHz, back-emf constant = 1.23 V sec/radian.

References

1. A. S. Xin-She Yang, "Swarm Intelligence Algorithms: A Tutorial," ed. Boca Raton: CRC Press, 2020.
2. X.-S. Yang, Nature-inspired metaheuristic algorithms: Luniver press, 2010.

3. M. J. Ali Zilouchian, Intelligent Control Systems Using Soft Computing Methodologies: CRC Press 2001.
4. S. Mirjalili, J. S. Dong, and A. Lewis, Nature-inspired Optimizers: Theories, Literature Reviews and Applications vol. 811: Springer, 2020.
5. C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, Evolutionary algorithms for solving multi-objective problems vol. 5: Springer, 2007.
6. Singh B, Singh S (2010) Single-phase power factor controller topologies for permanent magnet brushless DC motor drives. IET Power Electronics 3:147–175
7. Lu J, Mallik A, Khaligh A (2017) Dynamic strategy for efficiency estimation in a CCM-operated front-end PFC converter for electric vehicle onboard charger. IEEE Transactions on Transportation Electrification 3:545–553
8. Nian X, Peng F, Zhang H (2014) Regenerative braking system of electric vehicle driven by brushless DC motor. IEEE Trans Industr Electron 61:5798–5808
9. Cao W, Mecrow BC, Atkinson GJ, Bennett JW, Atkinson DJ (2011) Overview of electric motor technologies used for more electric aircraft (MEA). IEEE Trans Industr Electron 59:3523–3531
10. Hsiao H-C, Hsiao C-Y, Huang Y-H, Chien Y-K, Zheng Y-W (2018) "Design and Economical Evaluation of Small-Capacity Motor Used in Household Appliances by Taguchi Method," in. IEEE Student Conference on Electric Machines and Systems 2018:1–6
11. Santhosh P, Vijayakumar P (2017) Performance Study of BLDC Motor Used in Wireless Medical Applications. Wireless Pers Commun 94:2451–2458
12. S. S. Patel, B. Botre, K. Krishan, K. Kaushal, S. Samarth, S. Akbar, et al., "Modeling and implementation of intelligent commutation system for BLDC motor in underwater robotic applications," in 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), 2016, pp. 1–4.
13. Dursun M, Ozden S (2012) Application of solar-powered automatic water pumping in Turkey. International Journal of Computer and Electrical Engineering 4:161
14. Madichetty S, Pullaguram D, Mishra S (2019) A standalone BLDC based solar air cooler with MPP tracking for improved efficiency. CSEE Journal of Power and Energy Systems 5:111–119
15. B. S. Sanjeev Singh, "Single-Phase SEPIC Based PFC Converter for PMBLDCM Drive in Air-Conditioning System."
16. G. Spiazzi, "Analysis of buck converters used as power factor preregulators," PESC97. Record 28th Annual IEEE Power Electronics Specialists Conference. Formerly Power Conditioning Specialists Conference 1970–71. Power Processing and Electronic Specialists Conference 1972, vol. 1, pp. 564–570 vol.1, 1997.
17. S. Singh and B. Singh, "Voltage-controlled PFC SEPIC converter fed PMBLDCM drive for an air-conditioner," in 2010 Joint International Conference on Power Electronics, Drives and Energy Systems & 2010 Power India, 2010, pp. 1–6.
18. Singh B, Singh S, Chandra A, Al-Haddad K (2011) Comprehensive Study of Single-Phase AC-DC Power Factor Corrected Converters With High-Frequency Isolation. IEEE Trans Industr Inf 7:540–556
19. V. B. Bhim Singh, "An Improved Power Quality Based Sheppard–Taylor Converter Fed BLDC Motor Drive," Journal of The Institution of Engineers (India): Series B, vol. Volume 96, pp. pp 327–337 2014.
20. B. Singh, "A PFC Based BLDCM Drive for Low Power Household Appliances AU - Bist, Vashist," EPE Journal, vol. 24, pp. 21–30, 2014/06/01 2014.
21. V. B. Bhim Singh, "Power factor correction (PFC) converters feeding brushless DC motor drive," International Journal of Engineering, Science and Technology, vol. Vol. 7, No. 3, 2015, pp. 65–75, 2015.
22. B. Singh and V. Bist, "DICM and DCVM of a PFC-based SEPIC-fed PMBLDCM Drive," IETE Journal of Research, vol. 59, pp. 141–149, 2013/03/01 2013.
23. G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," Advances in Engineering Software, vol. 114, pp. 48–70, 2017/12/01/ 2017.

24. Kim IY, De Weck O (2006) Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. Struct Multidiscip Optim 31:105–116
25. G. P. Rangaiah, Multi-objective optimization: techniques and applications in chemical engineering vol. 1: World Scientific, 2009.

Imbalanced Data Classification Using Hybrid Under-Sampling with Cost-Sensitive Learning Method

Khan Md. Hasib, Md. Imran Hossain Showrov, Jubayer Al Mahmud, and Kamruzzaman Mithu

Abstract The topic of class inequalities and class integration has been a significant problem in supervised learning. To optimize cost function, most supervised learning algorithms presume that the classes being considered are a similar cardinality and that supposition does not matter to imbalanced datasets in the case of sub-optimal classification results. Consequently, various methods have been proposed that correct imbalanced datasets such as under-sampling, over-sampling, ensemble-based methodologies, and cost-sensitive learning. However, the lack of information triggers the under-sampling and over-sampling to endure higher runtime and potential overfitting. Also, there is a disadvantage for a cost-sensitive method to handle insufficiently specified costs. To solve this issue, a new hybrid HUSCSL-Boost framework has been proposed to manipulate imbalanced data through three key steps: data cleaning, data balancing, and cost accumulation. HUSCSLBoost uses Tomek-Link algorithms for eliminating noise data in the first stage and then creates several balanced subsets applying the random under-sampling. Finally, the CSLBoost method involves each balance subsets like any boost iteration, such as RUSBoost. The inclusion of a cost concept into the value update calculation for each case is based on its hardness to minimize the lack of information introduced by undertaking tests. HUSCSBoost was applied to 27 imbalanced datasets, and the outcome is considerably better than the current best-performing RUSBoost method.

Keywords Class imbalance · Sampling · Boosting · Cost-sensitive learning · RUSBoost

K. Md. Hasib
Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh

Md. I. H. Showrov (✉)
Institute of Computer Science, Bangladesh Atomic Energy Commission, Dhaka, Bangladesh

J. Al Mahmud
Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh

K. Mithu
Department of Computer Science and Engineering, Bangladesh University, Dhaka, Bangladesh

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_32

1 Introduction

Data imbalance is a classified problem where the distribution of classes within groups is not consistent [1]. Current machine learning algorithms, including decision tree [2], SVM [3], ANN [4], random forests [5], and also extreme multi-label classification algorithms [6] perform sub-optimal data-balance quality. Many of the real-life challenges are imbalanced, like detection by credit card theft [7], facial recognition [8], medical diagnostic [9], and in industrial applications such as the Boeing production line [10] where supervised learning is used. These less faults will be assessed, as they can lead towards catastrophic effects for a single defective component. When these problems occur, a framework needs to be established to evaluate the information mismatch problem and adequately resolve the issue. This is why researchers have built different ways that allow current classifiers to deal with classification tasks that demonstrate class imbalance [11].

We can classify these approaches into sampling techniques, ensemble-based methods, and cost-sensitive strategies. The sampling techniques can maximize the quantity of minority (over-sampling) and perhaps decrease the rate of quantities of the dominant component class (under-sampling) to such a degree that the proportion of divergences decreases and preparation data is more balanced [11]. Cost-sensitive approaches supply the minority with high loss or damage suffered and label examples that are further inserted into the cost mechanism of the underlying classifier to reduce [11]. Integrating such expense concepts reduces the prejudice of the classifiers toward the ruling class and puts a stronger emphasis on the proper learning of the meaning of the minority [12]. Approaches such as Bagging [13] and Boosting [14] utilize several baseline classifying instances and incorporate their information to foresee the explanatory variables. Data collection strategies or costing measures are incorporated into community approaches to tackling social inequalities, which have been very successful [15, 16]. But still, the above methods implemented either sampling or cost-effective steps to minimize the impact of class imbalances and became the target of either information loss or unequal distribution of costs.

This article suggests the hybrid under-sampling approach to counter class imbalance with the cost-sensitive learning process (HUSCSLBoost). To remove conflicting datasets, we first use Tomek-Link and construct a range of stable categories using RUS. Then use the cost-sensitive learning boosting method (CSLBoost) to separate the majority and minority class instances mostly from raw data and to organize the class label instances toward k clusters using the k-mean grouping [16], and to obtain the average values of each balanced subset. Nevertheless, the values would differentiate between stable, boundary, and external examples of the majority and minority groups and come up with the requisite fundamental classes to a more precise reflection of either the majority or the minority values. Our proposed model's reliability is shown to increase the outcomes of 27 imbalanced datasets compared with RUSBoost.

The rest of this article is organized as follows. Section 2 establishes the foundation for our analysis and reasoning, Sect. 3 introduces the methodology, and experimental results are summarized in Sect. 4. Finally, in Sect. 5, we conclude.

2 Literature Review

Over the past period, sampling techniques, bagging, and boosting-based ensemble approaches, and cost-sensitive learning techniques were used to handle imbalanced binary classification problems [17, 18]. Sun et al. [19] suggested an ensemble strategy to resolve binary-class disparity issues by turning an imbalanced binary learning mechanism into several balanced learning processes. This framework splits majority class instances into many classes, where every other section has a comparable rate of samples of the minority group. There were various standardized datasets generated here. Any balanced dataset was then used to construct a binary classifier. After this, such binary classifiers have been combined to form an ensemble classifier to classify additional data.

Chawla et al. [20] suggested the methodology of over-sampling called the SMOTE algorithm. Over-sampling of a minority class is carried out by creating simulated minority class instances rather than over-sampling. Instead of data space recruiting nearest neighbors, SMOTE made simulated samples by operating on a function space. Their findings found that the addition of over-sampling and sub-sampling was optimal for the receiver operating characteristic (ROC) area. Estabrooks et al. [21] recommended merging the method of over-sampling with under-sampling. SMOTE-Tomek is a hybrid type SMOTE (Synthetic Minority Oversampling Technique) and Tomek-Link [1]. SMOTE-ENN is the alternative solution to the combination of SMOTE and ENN (nearest neighbors, published) [1].

Seiffert et al. [15] suggested RUSBoost for imbalance detection. For every AdaBoost repetition, RUSBoost implements random under-sampling [14]. However, in the face of imbalanced datasets too, RUSBoost can suffer knowledge loss. The random portion of under-sampling [22] is why several majority class instances are discarded in each iteration. Sun et al. proposed three approaches for optimizing the classification of AdaC1, AdaC2, and Imbalancing Data Sets [12] AdaC3, based on the cost-sensitive analysis. These methods assign a higher expense to minority class cases. If an indication of a minority class is mislabeled, the value may be greater than the misclassified majority class case. In comparison, when a minority sample is categorized properly, it is lighter in value than a correctly identified majority. Thus, a sufficient understanding of minority instances becomes more relevant in the teaching cycle of AdaBoost to reduce the effect of gender discrimination. As a result, adequate minority schooling stresses the teaching phase of AdaBoost to minimize the consequences of class imbalance. Most of these methods assign equivalent costs to many other instances with the same form in terms of the imbalance ratio between groups. The local features of the data points are not taken into consideration.

Many researchers incorporated locality information in their methodology in several recent initiatives to treat unbalanced datasets. There is a proposal of an over-sample strategy for ADASYN [21], which would take note of the rate of major group samples in existing minority samples along with generating additional analytic tests for those with more major neighbors so that the learning protocol for the more complicated minority instances can be based on. Blaszczynski et al.

suggested regional-and-over-all balanced bagging [23] to have nearby specifics for the majority of situations by way of UnderBagging. In this approach, the amount of issues is less minor; the bagging variants are more commonly picked in their immediate region samples from synthetic minorities using only safe minority instances. Bunkhumpornpat et al. suggested Secure-Level-SMOTE [24], and his model generates instances from synthetic minorities utilizing only safe minority bodies. Han et al. proposed that [25] BorderlineSMOTE uses just border minority bodies to construct synthesized minorities. For the minority instances, Napierala et al. used locality specifics to identify them as open, small, uncommon, and external [26]. Both of these methods note that minority and majority locality specifics are essential and can be included in the learning process of imbalanced classifications.

3 Proposed Model and Methodology

Throughout this segment, we have introduced an under-sampling-based ensemble solution (HUSCSLBoost) to reduce the effects of data induced by under-sampling and solve class imbalances known as HUSCSLBoost (hybrid under-sampling and cost-responsive methods of learning).

3.1 Data Cleaning

Data cleaning is a method to remove corrupt or distorted info when the majority and minority group’s circumstances intersect [1]. The Tomek-Links algorithm scans those instance pairs because it excludes data from each pair in the majority class. This algorithm aims to clarify minority–majority boundaries. Tomek-Links was used as a data cleaning tool to delete noisy data [27] (Fig. 1).

3.2 Data Balancing

After using Tomek-Links, we generated several balance subsets without replacing a majority class sample with random under-sampling (RUS). The majority of samples have been split across many bags so that they are equivalent to minority samples utilizing a non-substituted sampling method.

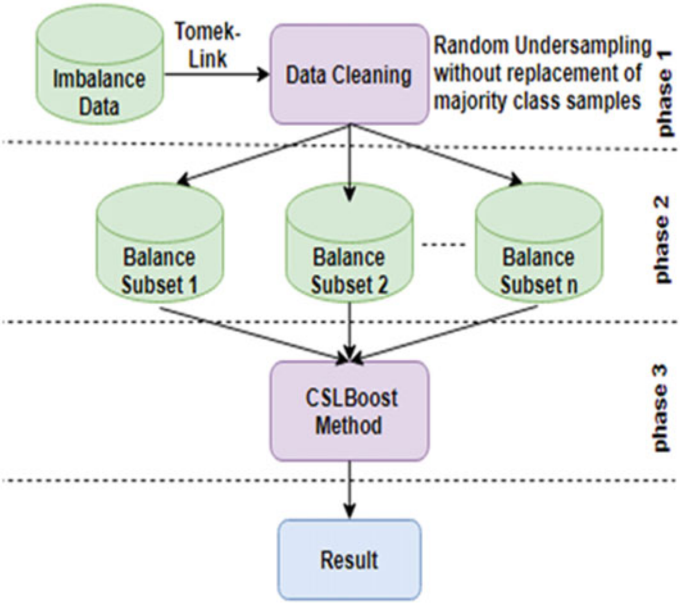


Fig. 1 The proposed model (HUSCSLBoost) for handling imbalanced data

3.3 Cost Accumulation

In this approach, the CSLBoost method (pseudocode II) checks the Extra_Value Allocation specified classes in pseudocode I until the iteration is boosted. Misclassified Value++ and Classified Value-- are returned by this process to decrease and raise. Misclassified Value++ is used for misclassified instances during the iteration under analysis within the exponential rate of the value-update method. Thus Classified Value—— is added for correctly classified instances. As a consequence, when misclassified, the value of cases with larger divisional value++ will quickly rise, and when correctly classified, the value—— of cases with more contentious values will decrease. The hyperparameter optimization determines the parameter K. These descriptive surveys are then combined to create equivalent datasets of instances of the minority population.

Pseudocode I Extra_Value Allocation *(dataset U,κ)*

```
for every status $u_i$ ∈ in the training set, U do
    find k means clustering for the instances
    $N_c$ ← Count of corresponding class
    $N_r$ ← Count of reverse class
    if $N_c$ = = 0 then
        Misclassified_Instances_Value$^{++}$(i) ← $\phi$
        Classified_Instances_Value$^{--}$(i) ← $1/N_r$
    else if $N_r$ = = 0 then
        Misclassified_Instances_Value$^{++}$ (i) ← $1/N_c$
        Classified_Instances_Value$^{--}$ (i) ← $\phi$
    else
        Misclassified_Instances_Value$^{++}$(i) ← $1/N_c$
        Classified_Instances_Value$^{--}$(i) ← $1/N_r$
    end if
end for
Return the Misclassified_Value$^{++}$ and Classified Value$^{--}$
```

The CSLBoost approach considers a collection of algorithmic decision trees and classifies additional samples using any tree vote. In the beginning, any sample is initialized along with the same value, 1/q, where q is the cumulative amount of samples of training. The values of the samples become balanced as per the form in which they have been made. If an instance has been identified right, the value is such, or if it has been misclassified, the value is expanded. The value of a sample reveals how hard it can be to explain. For measurement of the sample error rate Ti, in Eq. 2, we count the values of the incorrect instances as Qi. If you misclassify a case, err(Ui) is one. If not, err(Ui) is negative (where Ui is properly labeled). If an instance correctly classifies Ui in ith iteration, its value will be multiplied by error (Ti)/1 - error (Ti). The values of all instances are then uniform (including wrongly labeled instances). We subtract it for normalizing the value from several old values separated by the number of new values. As a consequence, it is the values of the misclassified samples. If the Model Ti error rate is 0.5, we abandon Ti and remove a new generation Qi subdata package.

Pseudocode II CSLBoost *(dataset = (U, Z))*

```
1: q = count of instances
2: k = count of boosting iterations
3: (Misclassified_Value$^{++}$. Classified_Value$^{--}$) = Extra_Value Allocation(dataset, $\kappa$)
4: initialize value $u_i \in Q$ to 1/q
5: for i = 1 to k do
6:     create balance dataset $Q_i$ with distribution Q using Under-sampling(dataset)
7:     Decision Tree $T_i$ from $Q_i$ employing
8:     loss_sum ← $\Sigma_{T_i(u_i) \neq z_i} Q^t_i$ misclassified_value$^{++}$ (i)
9:     acc_sum ← $\Sigma_{T_i(u_i) = z_i} Q^t_i$ classified_value$^{--}$ (i)
10:    accumulate the error rate of $T_i$, error($T_i$)
11:    if error ($T_i$) $\geq$ 0.5 then
12:        go back to step 5 and try again
13:    end if
14:    for each $u_i \in Q_i$ that correctly classified do
15:        accumulate value of $u_i$ by $\alpha_t$ ← (error($T_i$)/1-error($T_i$)) //update values
16:    end for
17:    for i = 1 to m do
18:        if $z_i \neq T_i(u_i)$ then
19:            $Q_{t+1}(i)$ ← $Q_t$ e-$_{\alpha t.zi.mi(ui).\text{misclassified\_value}^{++}\ (i)}$
20:        else if
21:            $Q_{t+1}(i)$ ← $Q_t$ e-$_{\alpha t.zi.mi(ui).\text{classified\_value}^{--}\ (i)}$
22:        else
23:            $v_i$ ← log((1-error($T_i$)/error($T_i$))   // value of the classifier's vote
24:            c ← $T_i$ (unew)                            // class prediction by Ti
25:            add $v_i$ to value for class c
26:    end for
27:    normalize Q
28: end for
29: Return the largest value
```

Our proposed model integrates an innovative approach to sampling and cost-sensitive learning. CSLBoost also determines the majority of cases by the distribution of value. The majority and minority groups greatly overlap, sometimes with rather unequaled datasets [28]. Substantial numbers of undefined and unusual pluralities that increase their risk of misclassification may be discarded. By tracking the bulk of these cases by value, CSLBoost overcomes this issue and focuses more on their learning. This is its specific function to mitigate the loss of information after cleaning noisy data and to create a variety of balanced subsets.

4 Experiment Setup

4.1 Datasets and Experimental Setup

We have been utilizing 27 binary imbalanced datasets of a specific imbalance ratio (IR) and are open to the public on the UCI / LIBSVM / repository [29]. There are also datasets on [30] accessible.

4.2 Evaluation Metrics in Imbalance Domain

For machine learning algorithms, predictive precision is usually known as the tool of evaluation. The collection of the right measurement criteria plays a significant role in the issue of data imbalances. Positive and negative class samples from a given classifier are seen in the uncertainty matrix.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{1}$$

$$\text{ErrorRate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{2}$$

$$\text{Recall or Sensitivity} : \text{TP}_{\text{rate}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{Specifivity} : \text{TN}_{\text{rate}} = \frac{\text{TN}}{\text{FP} + \text{TN}} \tag{4}$$

$$\text{FP}_{\text{rate}} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{5}$$

$$\text{FN}_{\text{rate}} = \frac{\text{FN}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Positive Predicted Value} : \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

The receiver operating characteristics (ROC) graph is one of the most common metrics used to measure the utility of classification over unbalanced results. AUC describes the results of learning algorithms by one amount and is not minority class discrimination. The method of constructing ROC space is to map the TPrate (Y-axis) to the FP rate (X-axis) on a two-dimensional diagram. To measure the AUC, we need to get the graphic area as follows:

$$\mathrm{AUC} = \frac{1 + \mathrm{TPrate} - \mathrm{FPrate}}{2} \tag{8}$$

The classifier allows some mistakes in identifying groups when the AUROC is smaller than 1. The misclassified meaning is lower if AUROC is equivalent to 1 [31].

4.3 *Results and Comparison*

We correlated the proposed HUSCSLBoost method's efficiency with RUSBoost using 27 imbalanced datasets with differing imbalance ratios; 20 times for each research protocol, the average cost estimate result was obtained 10 times the cross-validation of each data collection. The average value results for accuracy, recall, and f1 are shown in Tables 1 and 2. Again, RUSBoost only generates one stable subclass by taking sample sizes from the majority class, even though there is a number of stable system subset (randomly without replacement). We also found that the unequal datasets were noisy or skewed, so the Tomek-Link algorithm was initially used to wipe up the noisy data. From Tables 1 and 2, we see that our suggested approach (HUSCSLBoost) has outperformed the RUSBoost approach in 72 cases out of 135 instances following implementation in the 27 benchmark datasets.

5 Conclusion

Class inequality is prominent in many problems related to real-world classification. In this paper, we develop an ensemble approach that is cost-effective similar to RUSBoost. Still, the resulting loss of knowledge is not being affected, and the outputs are satisfactory to date. Several models using tenfold cross-validation are performed to equate test outcomes with other existing approaches and obtain optimal results on a particular performance assessment parameter. While comparing with the popular under-sampling-based ensemble method RUSBoost, our model provides a better result in most cases. HUSCSLBoost is cost-effective, defines a standard cost distribution system, fails to enforce any false framework, and considers external problems like low minority networks and group overlap. Yet we did not use the feature selection approach for datasets with large imbalances. In the future, we have an idea to collaborate with other cost allocation systems and strive to minimize runtime by enhancing our HUSCSLBoost model's performance.

Table 1 Average cost calculation and AV results of RUSBoost

Dataset	IR	RUSBoost				
		tenfold				
		Cost calculation	Accuracy	Recall	Precision	F1 Score
ecoli	8:6:1	0.8488	0.9643	0.9643	0.9626	0.9630
optical_digits	9.1:1	0.9326	0.9915	0.9915	0.9915	0.9913
satimage	9.3:1	0.8528	0.8912	0.8912	0.9031	0.8406
pen_digits	9.4:1	0.9532	0.9989	0.9989	0.9989	0.9989
abalone	9.7:1	0.8754	0.8909	0.8909	0.7937	0.8395
sick_euthyroid	9.8:1	0.9564	0.9153	0.9153	0.8378	0.8748
spectrometer	11:1	0.9258	0.9474	0.9474	0.9397	0.9378
car_eval_34	12:1	0.9724	0.9815	0.9815	0.9811	0.9812
isolet	12:1	0.9466	0.9882	0.9882	0.9880	0.9881
us_crime	12:1	0.8310	0.9359	0.9359	0.9300	0.9196
yeast_ml8	13:1	0.5386	0.9207	0.9207	0.8476	0.8826
scene	13:1	0.7630	0.9053	0.9053	0.8196	0.8603
libras_move	14:1	0.8035	0.9778	0.9778	0.9783	0.9757
thyroid_sick	15:1	0.9485	0.9321	0.9321	0.8689	0.8994
coil_2000	16:1	0.6583	0.9434	0.9434	0.8900	0.9159
arrhythmia	17:1	0.6415	0.9454	0.9454	0.8938	0.9189
solar_flare_m0	19:1	0.6415	0.9454	0.9454	0.8938	0.9189
oil	22:1	0.7533	0.9532	0.9532	0.9086	0.9303
car_eval_4	26:1	0.9747	0.9931	0.9931	0.9931	0.9927
wine_quality	26:1	0.7598	0.9461	0.9461	0.8951	0.9199
letter_img	26:1	0.9706	0.9944	0.9944	0.9944	0.9942
yeast_me2	28:1	0.8994	0.9704	0.9704	0.9416	0.9557
webpage	33:1	0.9094	0.9879	0.9879	0.9875	0.9867
ozone_level	34:1	0.7979	0.9700	0.9700	0.9410	0.9553
mammography	42:1	0.8789	0.9839	0.9839	0.9817	0.9818
protein_homo	111:1	0.9424	0.9949	0.9949	0.9949	0.9939
abalone_19	130:1	0.7510	0.9904	0.9904	0.9810	0.9857

Table 2 Average cost calculation and AV results of HUSCSLBoost

Dataset	IR	HUSCSLBoost				
		tenfold				
		Cost calculation	Accuracy	Recall	Precision	F1 Score
ecoli	8:6:1	**0.9995**	0.9642	0.9642	0.9626	0.9630
optical_digits	9.1:1	**0.9823**	**0.9914**	**0.9915**	**0.9915**	**0.9913**
satimage	9.3:1	**0.9973**	0.8906	0.8906	**0.7932**	0.8391
pen_digits	9.4:1	**0.9999**	**0.9993**	**0.9993**	**0.9993**	**0.9993**
abalone	9.7:1	**0.9898**	**0.8910**	**0.8910**	0.7937	**0.8401**
sick_euthyroid	9.8:1	0.9630	0.9153	0.9153	**0.8378**	0.8748
spectrometer	11:1	**0.9860**	**0.9474**	**0.9474**	0.9397	**0.9378**
car_eval_34	12:1	**0.9773**	0.9815	0.9815	0.9811	0.9812
isolet	12:1	**0.9858**	**0.9867**	**0.9867**	**0.9864**	**0.9865**
us_crime	12:1	**0.9425**	**0.9379**	**0.9379**	**0.9328**	**0.9229**
yeast_ml8	13:1	0.7799	0.9207	0.9207	0.8476	0.8826
scene	13:1	0.8869	0.9053	0.9053	0.8196	0.8603
libras_move	14:1	**0.9988**	**0.9778**	**0.9778**	**0.9783**	**0.9757**
thyroid_sick	15:1	**0.9689**	0.9321	0.9321	0.8689	0.8994
coil_2000	16:1	**0.9717**	0.9434	0.9434	0.8900	0.9159
arrhythmia	17:1	**0.9944**	**0.9469**	**0.9469**	**0.8966**	**0.9211**
solar_flare_m0	19:1	**0.9572**	0.9454	0.9454	**0.8938**	**0.9189**
oil	22:1	0.9346	0.9532	0.9532	0.9086	0.9303
car_eval_4	26:1	**0.9997**	**0.9931**	0.9931	0.9931	0.9927
wine_quality	26:1	**0.9989**	0.9461	0.9461	**0.8952**	**0.9201**
letter_img	26:1	**0.9999**	**0.9948**	**0.9948**	**0.9948**	**0.9946**
yeast_me2	28:1	0.9410	0.9704	0.9704	**0.9416**	**0.9558**
webpage	33:1	**0.9490**	0.9879	0.9879	0.9875	0.9867
ozone_level	34:1	**0.9557**	0.9700	0.9700	**0.9410**	**0.9553**
mammography	42:1	**0.9205**	**0.9840**	**0.9840**	**0.99817**	**0.9818**
protein_homo	111:1	0.9409	0.9949	0.9949	09,949	0.9939
abalone_19	130:1	**0.9998**	0.9904	0.9904	**0.9810**	**0.9857**

Bold significance denotes like the proposed method, HUSCSLBoost outperformed the RUSBoost method in 72 cases out of 135

References

1. Popel MH, Hasib KM, Ahsan Habib S, Faisal Muhammad Shah F (2018)A hybrid under-sampling method (HUSBoost) to classify imbalanced data. In: 2018 21st International conference of computer and information technology (ICCIT), Dhaka, Bangladesh. IEEE, pp 1–7
2. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

3. Hopfield JJ (1988) Artificial neural networks. IEEE Circuits Devices Mag 4(5):3–10
4. Safavian SR, Landgrebe D (1991) A survey of decision tree classifier methodology. IEEE Trans Syst Man Cybern 21(3):660–674
5. Breiman JL (2001) Random forests. Mach Learn 45(1):5–32
6. Kumar P, Dubey VK, Showrov MIH (2019) A comparative analysis on various extreme multi-label classification algorithms. In: 2019 4th International conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT), Mysuru, India, pp 265–268
7. Phua C, Alahakoon D, Lee V (2004) Minority report in fraud detection: classification of skewed data. Sigkdd Explorations, Researchgate
8. Liu Y-H, Chen Y-T (2005) Total margin based adaptive fuzzy support vector machines for multi-view face recognition. In: 2005 IEEE international conference on systems, man and cybernetics, vol 2. IEEE, pp 1704–1711
9. Ginsburg S, Ali S, Lee G, Basavanhally A, Madabhushi A (2013) Variable importance in nonlinear kernels (VINK): classification of digitized histopathology. Med Image Comput Assist Interv
10. Riddle P, Segal R, Etzioni O (1991) Representation design and bruteforce induction in a Boeing manufacturing domain. Appl Articial Intell 8:125–147
11. Ahmed S, Rayhan F, Mahbub A, Jani MR, Shatabda S, Farid D (2019) LIUBoost: locality informed under-boosting for imbalanced data classification. In: Proceedings of IEMIS 2018, vol 2. https://doi.org/10.1007/978-981-13-1498-8_12
12. Sun Y, Kamel MS, Wong AK, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn 40(12):3358–3378
13. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
14. Freund Y, Schapire RE (1995) A desicion-theoretic generalization of on-line learning and an application to boosting. In: European conference on computational learning theory. Springer, pp 23–37
15. Seiffert C, Khoshgoftaar TM, Van Hulse J, Napolitano A (2010) Rusboost: A hybrid approach to alleviating class imbalance. IEEE Transa Syst Man Cybern Part A Syst Hum 40(1):185–197
16. Chawla NV, Lazarevic A, Hall LO, Bowyer KW (2003) Smoteboost: Improving prediction of the minority class in boosting. In: European conference on principles of data mining and knowledge discovery. Springer,, pp 107–119
17. Beyan C, Fisher R (2015) Classifying imbalanced data sets using similarity based hierarchical decomposition. Pattern Recogn 48(5):1653–1672
18. Pozzolo AD, Caelen O, Bontempi G (2015) When is undersampling effective in unbalanced classification tasks? In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 200–215
19. Sun Z, Song Q, Zhu X, Sun H, Xu B, Zhou Y (2015) A novel ensemble method for classifying imbalanced data. Pattern Recogn 48(5):1623–1637
20. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
21. Estabrooks A, Jo T, Japkowicz N (2004) A multiple resampling method for learning from imbalanced datasets. Comput Intell 20:1836
22. Liu A, Ghosh J, Martin CE (2007) Generative oversampling for mining imbalanced datasets. In: DMIN, pp 66–72
23. Błaszczy´nski J, Stefanowski J, Idkowiak Ł (2013) Extending bagging for imbalanced data. In: Proceedings of the 8th international conference on computer recognition systems CORES 2013. Springer, pp 269–278
24. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2009) Safe-levelsmote: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. Adv Knowl Discov Data Mining 475–482
25. Han H, Wang W-Y, Mao B-H (2005) Borderline-smote: a new oversampling method in imbalanced data sets learning. Adv Intell Comput. 878–887

26. Napierala K, Stefanowski J (2016) Types of minority class examples and their influence on learning classifiers from imbalanced data. J Intell Inf Syst 46(3):563–597
27. Elhassan A, Al-Mohanna, Shoukri (2016) Classification of imbalance data using tomek link (T-Link) combined with random undersampling (RUS) as a data reduction method. 1(2):11. ISSN 2472–1956
28. Prati RC, Batista G, Monard MC et al (2004) Class imbalances versus class overlapping: an analysis of a learning system behavior. In: MICAI, vol 4. Springer, pp 312–321
29. Rong-En Fan, LIBSVM Data: classification, regression, and multilabel. https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets
30. Lemaitre G, Nogueira F, Aridas CK, Oliveira, Imbalanced dataset for benchmarking. Zenodo. https://doi.org/10.5281/zenodo.61452
31. Galar, M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2011) A review on ensembles for the class imbalance problem: bagging, boosting, and hybrid-based approaches. IEEE

Analyzing the Errors in Channel Sensing and Negotiation in Cognitive Radio H-CRAN

Annesha Das, Asaduzzaman, and Ashim Dey

Abstract A heterogeneous cloud radio access network with low power nodes has emerged as an attractive cost-effective solution to the problem of enormous growth in data over the cellular network. Ever-increasing low power nodes such as femtocells in macrocell's coverage area for indoor communication cause severe cross-tier interference to the umbrella macrocell network. One of the promising paradigms to avoid this interference is cognitive radio-enabled heterogeneous cloud radio access network architecture. In this paper, two mechanisms, channel sensing and channel negotiation, for allocating channels to femtocell users have been proposed suitable for this architecture. After identifying an idle slot by channel sensing, a femtocell user requests the base station pool for free channel suggestions and senses the listed channels later. Femtocell user equipment does not wait for the next slot to sense another channel on identifying an occupied channel in a slot. Two and three types of error in sensing and negotiation, respectively, have been defined. Poisson traffic model is developed for generating macrouser traffic. Throughput has been analyzed by varying macrouser arrival rate, the number of femtocell users, the average service time of macrouser, and sensing time considering the errors in sensing and negotiation mechanisms. Simulation results have shown that the maximum throughput for error-free sensing and negotiation in case of low macrouser traffic is 27 Mbps. A difference of 10 Mbps between maximum throughput without any error and throughput with sensing and negotiation errors is also observed.

Keywords Heterogeneous cloud network · Femtocell user · Cross-tier interference · Cognitive radio · Sensing and negotiation errors · Throughput analysis

A. Das (✉) · Asaduzzaman · A. Dey
Chittagong University of Engineering and Technology, Chittagong 4349, Bangladesh
e-mail: annesha@cuet.ac.bd

Asaduzzaman
e-mail: asad@cuet.ac.bd

A. Dey
e-mail: ashim@cuet.ac.bd

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_33

1 Introduction

The necessity for high-speed and high-quality data applications is growing rapidly in the fifth-generation (5G) cellular network along with a massive increase in capital expenditures and energy consumption [1, 2]. As an optimistic technique to meet the phenomenal rise in mobile data traffic, heterogeneous networks (HetNets) have drawn great interest. Service providers are turning to HetNets consisting of cells of different sizes (femto, pico, micro, and macro) to support ubiquitous coverage, high data rates, and a satisfactory level of quality of service (QoS) [3]. The small cells aka low power nodes (LPN, e.g., femtocell, picocell, etc.) can reduce the load of immensely growing wireless data traffic from macrocells and prolong the battery life of devices. However, too dense deployment of LPNs in the macrocell area will create acute cross-tier and intra-tier interferences reducing performance gains of HetNets.

Meanwhile, for achieving better energy efficiency (EE) and spectral efficiency (EE) across cellular networks, cloud radio access network (C-RAN) has emerged as a major breakthrough [2]. In a usual C-RAN, the traditional base station (BS) is divided into a base station unit (BSU), which is located in the cloud, and an antenna unit (AU), which is located at the cell site. AUs communicate with BSU through a wired (e.g., optical fiber)/wireless (e.g., millimeter-wave) front-haul network. BSUs are clustered in a centralized location from where they can manage the AUs and it is shared by different cell sites. In C-RAN, centralized processing of baseband signal is carried out in the base station unit (BSU) pool, while radio frequency functions are allotted in AUs [4]. Specifically, C-RANs can decrease the expense of deploying base stations because service providers only have to establish new AUs and append them to the centralized BSU pool to extend the network coverage [2].

Combining the hierarchical structure of HetNets to increase spectrum reuse and the advantages of C-RAN to centrally control networks at lower cost, heterogeneous cloud radio access networks (H-CRANs), as shown in Fig. 1, are proposed to further enhance network performance [5]. The main incentive of H-CRAN is to mitigate interference and simplify LPNs by attaching them to centralized signal processing clouds through high-speed front-haul connections [6]. To provide extended capacity and coverage, AUs can be installed on every floor of a building. By reducing the number of BSUs in ultra-dense HetNets, C-RAN decreases the power and energy consumption in HetNets [4].

According to surveys, data traffic up to 70% and voice traffic up to 50% come from indoor locations [7]. But at indoors, path loss, shadowing, and wall penetration can deteriorate QoS greatly. Femtocells in H-CRAN are low cost and easily deployable solution for overcoming indoor communication barriers. But too many co-channel deployment of femtocells will create severe interference to macrocells if there is no proper frequency planning for femtocells. Although the intra-tier interference among AUs can be reduced by centralized cooperative processing in the BSU pool of H-CRAN, the cross-tier interference between femtocell AU (FAU) and macrocell AU (MAU) is still a major impediment to fully utilize the potentials of H-CRANs [5].

Considering this, the main contributions of this paper are:

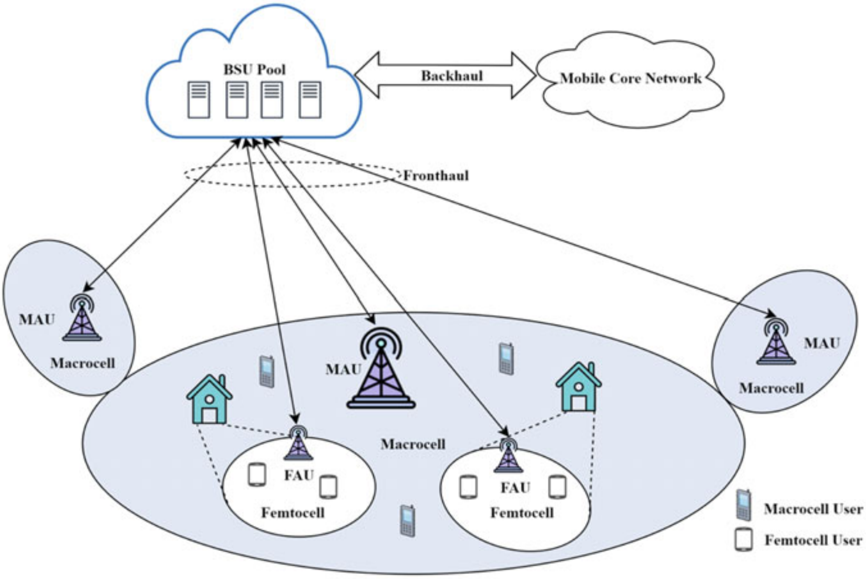


Fig. 1 Heterogeneous cloud radio access network (H-CRAN) (BSU: base station unit, MAU: macrocell antenna unit, FAU: femtocell antenna unit)

- To develop CR-based channel sensing and channel negotiation mechanism for femtocells users' spectrum allocation.
- To define the errors in channel sensing and negotiation mechanism and analyze the effect of those errors on throughput varying different metrics.

The remaining of this paper is outlined as follows: Related works are discussed in Sect. 2. The proposed CR-based interference mitigation scheme including system requirements, channel sensing, and negotiation process is described in Sect. 3. Errors in sensing and negotiation are introduced in Sect. 4. Results obtained by varying different metrics in the presence and absence of sensing and negotiation errors are presented in Sect. 5, and finally, the paper is concluded in Sect. 6.

2 Related Literature

Numerous studies have proposed many solutions for mitigating the interference problem. In [8], a detailed survey of various interference management issues for 5G networks has been presented considering inter-cell interference, HetNets, D2D, coordinated scheduling, and coordinated multi-point (CoMP). It has been stated that the use of static or dynamic interference cancellation approaches with improved resource allocation can lessen interferences in high-density HetNets. The CoMP transmission and reception have been presented for alleviating co-channel inter-cell interference of cell edge users [4, 6]. In joint CoMP transmission, several AUs constitute a coordinated AU cluster and assist the users jointly. But CoMP has few drawbacks in practical networks as its performance demotes with increasing density of LPNs. Studies in [1, 6] have proposed a cloud computing-based cooperative radio resource management (CC-CRRM) approach in which all AUs in H-CRAN are connected to a dedicated entity that is responsible for centralized radio resource allocation. With CC-CRRM, AUs can share radio resources among them and the cross-tier interference between femtocells and macrocell can be minimized. The difficulty with both CoMP and CC-CRRM is in handling a huge amount of data regarding traffic demands and channel state information (CSI). Authors in [9] have proposed to include strict and soft fractional frequency reuse (FFR) in 5G HetNets to obtain lower inter-channel and intra-channel interference. They have divided the whole-cell coverage area into the center and edge parts. Users of these parts are differentiated by comparing measured signal to interference and noise ratio (SINR) with a threshold. Results have shown that strict and soft FFR outperform without FFR scenarios in terms of throughput and outage probability.

In none of these techniques, user equipment (UE) is aware of the channel state and radio resource demands of other users sharing the radio network which reduces the efficiency of radio resource utilization. Deployment of cognitive radio (CR) in orthogonal frequency division multiple access-based (OFDMA) femtocell network as described in [10] can optimize network performance by maximizing capacity, facilitating sub-carrier selection, and mitigating interference in dense HetNets scenario. In [5], the authors have proposed a CR-based cooperation framework to reduce inter-tier interference in H-CRAN where FAU acts as a relay for multiple macrocell users (MUs) and gets MUs' fraction of time slot as a reward. Cooperative FAU-MU pairs are selected by a two-sided cooperator selection algorithm. This algorithm depends on the preference list of MUs and FAUs based on transmission rates and assigned resources of MUs. In [7], non-orthogonal multiple access (NOMA) with 5G cognitive femtocell has been investigated. In the power domain NOMA, more than one user can share at the same time, frequency, and code at different power levels. The authors have proposed a pairing algorithm between strong and weak users and a channel allocation algorithm based on the difference in channel gain and distance from femtocell base station. A higher sum rate for femtocell users (FUs) is achieved as compared to conventional OMA and CR-OMA. The authors in [11] have proposed a CR-based overlay interference mitigation strategy for femtocells in H-CRAN, which includes sensing, negotiation, data transmission, and acknowledgment. They have used the femtocell network architecture studied in [12].

In this study, we use CR technology based on overlay spectrum access to opportunistically utilize the unused spectrum of the macro network without interfering with it. Using time-slotted OMA transmission, FUs' secondary access to the unused spectrum involves detecting idle slots which are accomplished by channel sensing and channel negotiation as in [11, 12]. In [11, 12], FUs stop sensing for idle slots in the current slot duration and wait for the next slot when they detect that slot is not present in the list given by the femtocell base station unit (FBSU) in the channel negotiation process. Different from this in our study, FUs continue sensing in the current slot duration.

In most of the articles stated above, simulations have been accomplished with the assumption of perfect spectrum sensing. In practical cognitive wireless networks, perfect sensing is immensely tough to achieve [13–16]. Sensing errors have been investigated in [12–16]. Mostly these errors are spectrum-sensing errors based on miss detection and false alarm probabilities. These errors can exacerbate the co-channel interference in the macro-femto network and degrade the performance of MUs which is not intended. So, these errors must be investigated to fully utilize the addition of CR in H-CRAN architecture. In our study, FUs' performance is analyzed taking into account errors both in channel sensing and negotiation process. Errors in channel sensing happen in the first process of identifying an idle channel for FU's transmission. Similarly, errors in channel negotiation happen in the final process of getting transmission opportunity. Errors in channel sensing and negotiation are described later in Sect. 4.

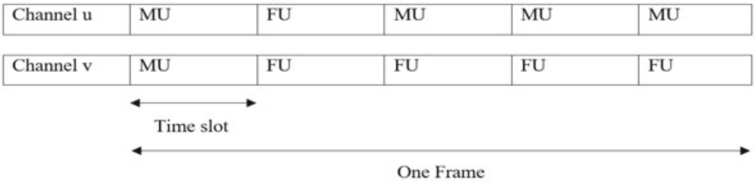
3 Proposed CR-Based Interference Mitigation Scheme

3.1 System Requirements

In H-CRAN, FAU and FBSU are separated. FBSUs are centralized in the BSU pool. As FUs use the same channels used by MUs in co-channel deployment causing co-channel or cross-tier interference, cognitive radio is introduced in the femtocell network. For this, a spectrum sensing (SS) engine is added to the FAU at the cell site which is capable of identifying the frequency channel in use and those available for transmission [11]. Also, the central BSU pool has a cognitive radio engine so that the SS engine in FAU can send channel information to the BSU pool. As the BSU pool will have an overall view about spectrum usage, it can allocate any free channel to the FUs from its umbrella macrocell.

Any FU device must be cognitive radio-enabled so that it can sense its umbrella macrocell's licensed spectrum to identify spatiotemporally available frequency channels. Here, MUs are primary users and FUs are secondary users. MUs will get higher priority in occupying free channels. MAU will provide voice service to all users whether they are in femtocell or macrocell. But it will provide data-oriented services to MU only. On the other hand, only FU will get data-oriented secondary services. FU will always act as a secondary user always in case of accessing data services to avoid any unwanted interference to MU. If any user comes in the range of femtocell, he will act as a FU.

OFDMA time-slotted transmission is considered here. Radio spectrum is divided into equal portions called channels, and the duration in every channel is partitioned into time slots also called bursts. A definite number of slots comprise a frame as shown in Fig. 2. FUs will sense channels in each time slot before using them for secondary



Channel u	MU	FU	MU	MU	MU
Channel v	MU	FU	FU	FU	FU

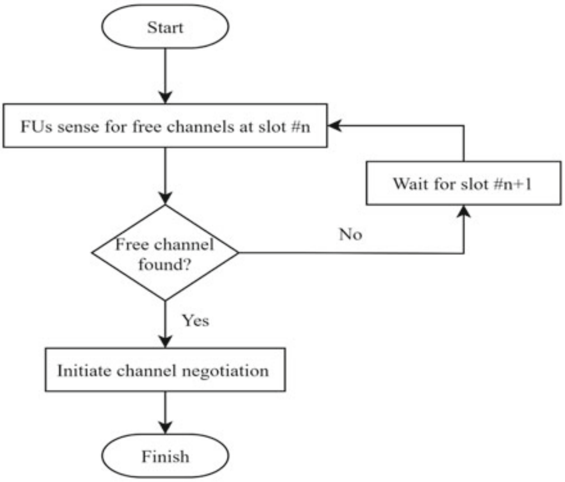
Fig. 2 Primary and secondary transmissions in different slots of a frame (MU: macrocell user, FU: femtocell user)

access for a certain sensing time. To access idle slot for transmission without interrupting MU, FU devices are proposed to follow two mechanisms: channel sensing first and then channel negotiation.

3.2 Channel Sensing

Whenever a FU needs a data-oriented service, the channel sensing process is activated as shown in the flowchart of Fig. 3. At first, CR-enabled FU equipment will search for free channels not occupied by MU at the beginning of a slot of a frame. It does this so that it can send a request to the FBSU in the BSU pool by using the idle slot for initiating channel negotiation. On each slot, it senses all channels one by one. If it finds a channel to be busy, it senses the next channel. It carries on sensing until a free channel is found or it finds that all channels are occupied on that slot. In the first case, after identifying an idle slot in any channel, the channel negotiation process is initiated. In the second case, if FU equipment sees that all channels are busy, it

Fig. 3 Channel sensing mechanism


Start
FUs sense for free channels at slot #n
Wait for slot #n+1
Free channel found?
No
Yes
Initiate channel negotiation
Finish

waits for the next slot to start channel sensing again. Channel sensing process flow is inspired by the works in [11, 12].

3.3 Channel Negotiation

At the beginning of channel negotiation as shown in the flowchart of Fig. 4, FU sends a request to FBSU in the BSU pool through FAU using the idle slot identified by channel sensing. It sends a request asking for information about which channels are occupied and which channels are idle. In response to this request, the BSU sends information on which channels are not occupied by MU and suggests a list

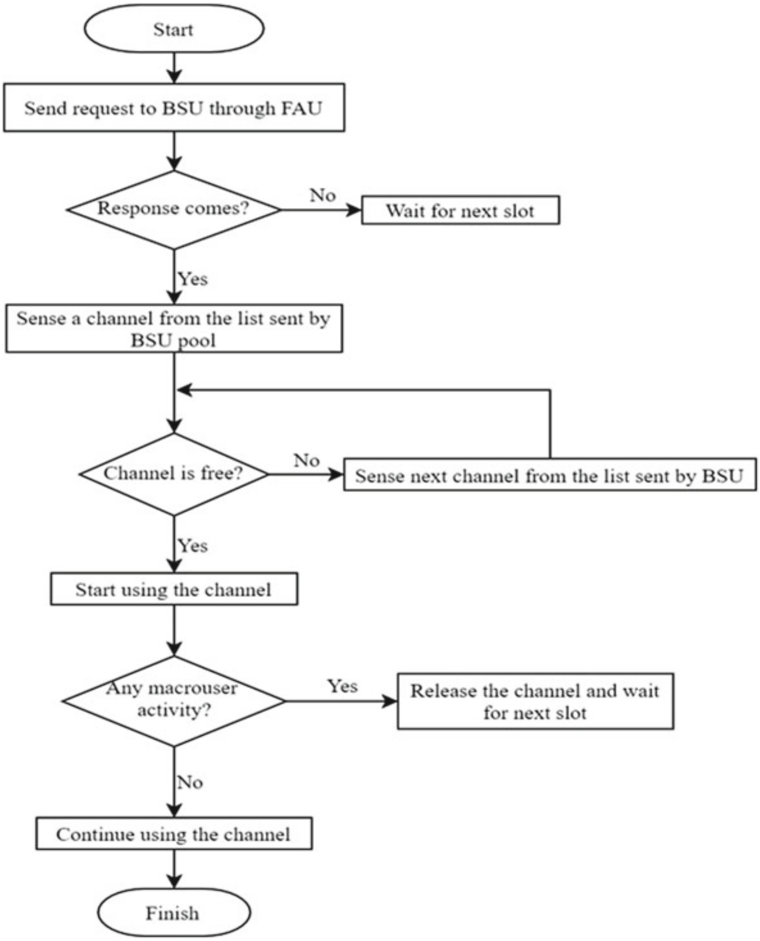

Start
Send request to BSU through FAU
Response comes?
No
Wait for next slot
Yes
Sense a channel from the list sent by BSU pool
Channel is free?
No
Sense next channel from the list sent by BSU
Yes
Start using the channel
Any macrouser activity?
Yes
Release the channel and wait for next slot
No
Continue using the channel
Finish

Fig. 4 Channel negotiation mechanism

of available channels to be sensed later. The FU device then senses the channels from the list sent by BSU one by one. It senses again to ensure that there are no transmissions from any MU or FU. If it finds an idle channel from the list sent by BSU, it starts transmitting data using that channel and one slot is added to the total number of idle slots observed by FU.

On a slot duration, FU is considered to be able to start using only one channel but it can use multiple slots on different channels at a time. If no channel is available on the list given by BSU, it waits for the next slot and resends the request if any channel is found idle by the channel sensing process. Once FU occupies a channel, he can use the channel forever unless there is any transmission from MU on that channel. While a FU is using a channel, if any MU begins transmission, FU then immediately releases the channel. While using a channel, FU keeps on checking MU activity to avoid any interference, and also it gets notified by BSU through FAU if any MU transmission is found.

4 Errors in Channel Sensing and Negotiation

Spectrum sensing performance is generally measured by the probability of miss detection and false alarm. The probability of false alarm refers to the probability that the secondary user decides a channel is busy when it is idle actually and the probability of miss detection refers to the probability that the secondary user decides a channel is idle when the primary user transmission is actually present there [12–16]. Thus, FU's device can identify a busy slot as a free slot or identify a free slot as a busy slot or also may not get the correct response from BSU which will result in errors in sensing and negotiation mechanism. The performance of the two proposed mechanisms is evaluated considering these errors.

4.1 Errors in Channel Sensing

Errors in sensing occur when FU senses channels for sending a request to BSU. In channel sensing, two types of error can occur which are false alarm and miss detection.

- **False Alarm.** It indicates the situation when a channel is idle, i.e., no MU is using that channel but FU equipment detects that the channel is occupied by someone. This error decreases the number of idle slots observed by FU as well as the utilization of idle slots and also degrades throughput.
- **Miss Detection.** It indicates the situation when MU or FU is transmitting on a channel but FU equipment detects that the channel is idle. As a result, FU tries to send a request to BSU using that channel causing collision or interference to

the user using that channel. When miss detection occurs on a slot duration, it is counted as a collision, not an idle slot.

4.2 Errors in Channel Negotiation

Channel negotiation errors occur when the list of idle channels is sent by BSU and FU senses channels one after another from the list. For analyzing negotiation error, errors in these steps are taken into account separately. In channel negotiation, three errors can occur which are as follows:

- **Getting False Response from FBS.** Sometimes, it may happen that there are available channels. FU sends a request to FBSU in the BSU pool for a list of free channels for secondary access but BSU may not respond and may not send the list of available channels due to the heavy load of requests.
- **False Alarm.** False alarm in channel negotiation occurs when FU identifies an idle channel as a busy channel while sensing the list of available channels sent by BSU.
- **Miss Detection.** Miss detection in channel negotiation occurs when FU identifies a busy channel as an idle channel while sensing the list of channels sent by BSU.

5 Performance Analysis

For performance analysis of the proposed CR-based sensing and negotiation, a random traffic generation model is developed in MATLAB where MU's arrival on each slot follows Poisson distribution and MU's service time or holding time follows an exponential distribution. Throughput or network throughput is the measure of successful data transfer within a time duration through a communication channel. Here, FUs' throughput is calculated based on the number of idle slots that they get to use for secondary transmission in the presence and absence of errors. After assigning channels to MU, idle slots for secondary transmission are identified by channel sensing and channel negotiation mechanisms as described earlier. While identifying idle slots, the probability of miss detection and false alarm are taken into account for analyzing channel sensing errors while the probability of getting a false response, miss detection, and false alarm are considered for analyzing channel negotiation errors. For error-free reliable sensing and negotiation, values of these error probabilities are considered as zero. These values are taken from Table 1 in [12] except probability of getting a false response whose value is taken as 0.20. FUs equally share a total of 16 channels of bandwidth (W) of 2000 kHz. In all the cases of calculating throughput, signal-to-noise ratio (SNR) has been kept at 10 dB. The number of slots per frame (D) is taken as 8. The average number of idle slots observed by a FU per frame (I) is given by

Table 1 Input sets 1 and 2

Parameter	Input set 1	Input set 2
Slot duration (ms)	10	10
Sensing time (µs)	50	50
Macrouser arrival rate	1	2
Average service time of macrouser (ms)	120	120

$$I = \frac{\text{Total idle slots observed}}{\text{Number of femtocell user} \times \text{Number of frames}} \tag{1}$$

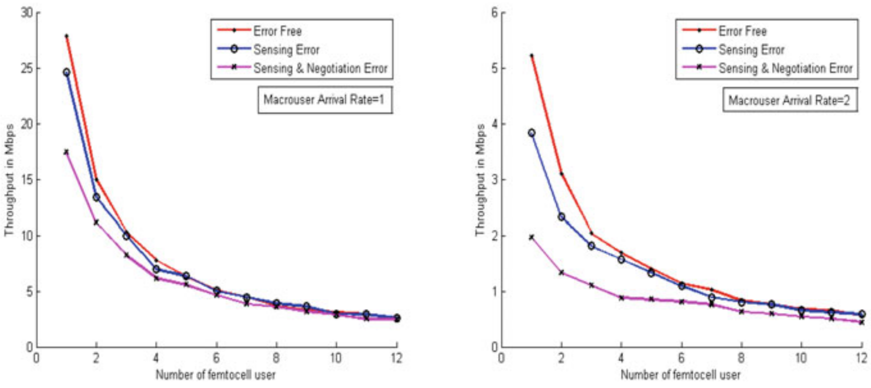
Using Shannon bound for throughput calculation, the femtocell user throughput (T) in bits per second (bps) is given by [11, 12]

$$T = \eta_{\text{sense}} \frac{I}{D} \times W \log_2(1 + SNR) \tag{2}$$

Here, W is applied in Hz and η_{sense} is sensing efficiency, which is given by [11, 12]

$$\eta_{\text{sense}} = 1 - \frac{\text{Sensing time}}{\text{Slot duration}} \tag{3}$$

To analyze the effect of MU arrival rate on throughput, the input sets 1 and 2 of Table 1 are used and the number of FU is increased from 1 to 12. Figure 5 shows that maximum of 27 and 5.2 Mbps throughput is achieved for MU arrival rate of 1 and 2, respectively. In the case of error-free sensing and negotiation, the average data rate is 7.47 Mbps for MU arrival rate of 1, and 1.58 Mbps for the arrival rate



(a)

(b)

Fig. 5 Throughput in terms of number of FU for macrouser arrival rate of **(a)** 1 and **(b)** 2

of 2, as shown in Fig. 5a, b, respectively. The average data rate is lower in case of errors in both mechanisms. Throughput does not vary much when the number of FU is greater than 10. Throughput is higher when the number of FU is less than 6. It is because when the number of FU is greater than 6, there remains only a small number of idle slots for each FU. Every user observes less number of idle slots for secondary transmission when the number of FU increases. Thus, throughput degrades.

Input sets in Table 2 are considered to analyze the effect of the average service time of MU on the throughput of FU. As shown in Fig. 6, for error-free sensing and negotiation, the maximum data rate is 16.44 Mbps for an arrival rate of 1, and 6.87 Mbps for an arrival rate of 2. For erroneous sensing, the maximum data rate of 16 Mbps for an arrival rate of 1 and 6.43 Mbps for an arrival rate of 2 is observed. For erroneous sensing and negotiation mechanism, the maximum data rate is 14.63 Mbps for an arrival rate of 1 (Fig. 6a) and 4.89 Mbps for an arrival rate of 2 (Fig. 6b). In both Fig. 6a, b, the throughput sharply degrades as the average service time of MU is increased. Because the channels remain occupied for a longer duration when MU's service time increases, thus the chances of identifying idle slots for secondary transmission also decrease which contributes to the fall of throughput of FU. Due to sensing error, about 3% fall and due to sensing and negotiation error, about 11% fall in throughput is observed.

Table 2 Input sets 3 and 4

Parameter	Input set 3	Input set 4
Slot duration (ms)	10	10
Sensing time (μs)	50	50
Macrouser arrival rate	1	2
Number of femtocell user	4	4

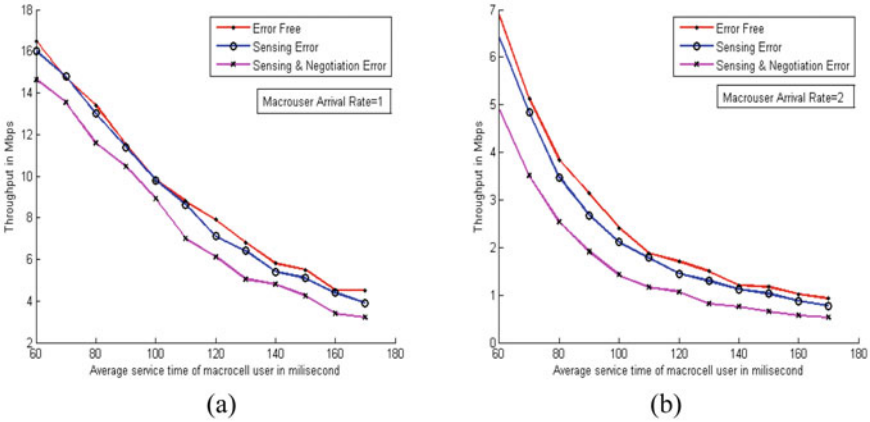


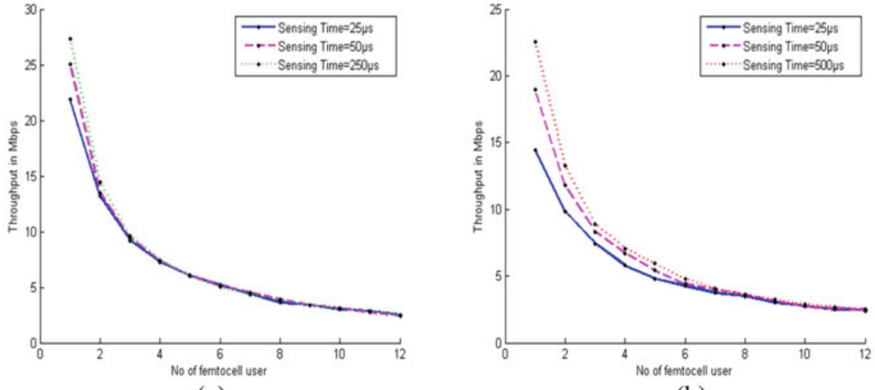
Fig. 6 Throughput in terms of average service time of MU for MU arrival rate of **(a)** 1 and **(b)** 2

Table 3 Input sets 5, 6, 7, and 8

Parameter	Input Set 5	Input Set 6	Input Set 7	Input Set 8
Slot duration (ms)	10	10	10	10
Sensing time (μs)	25	50	250	500
Probability of false alarm	0.73	0.63	0.34	0.21
Average service time of macrouser (ms)	120	120	120	120
Macrouser arrival rate	1	1	1	1

To analyze the effect of sensing time on FU's throughput, input sets in Table 3 are considered. The probability of false alarm decreases with an increase in sensing time. The number of FUs is varied from 1 to 12 as a femtocell can support 4–16 users. The throughput observed by a FU due to sensing error for input sets 5, 6, and 7 and due to sensing and negotiation error for input sets 5, 6, and 8 is plotted in Fig. 7a, b, respectively. Maximum of 27 and 23 Mbps throughput is achieved for FU in case of sensing errors and both errors, respectively, for greater sensing time as shown in Fig. 7. Throughput falls as an average number of idle slots identification decreases when sensing time decreases and the probability of false alarm increases. This happens because the more the probability of false alarm, the more chances the FUs do not get idle slots for secondary transmission even if idle slots are available. It means that the rate of identifying idle slots decreases as sensing time decreases.

There is not much difference in throughput for any sensing time whether only sensing error occurs (Fig. 7a) or both errors (Fig. 7b) when the number of FU is greater than 6 because only a few idle slots are left idle then. A higher number of idle slots is observed when the number of FU is less than 6. As the number of FU



(a) (b)

Fig. 7 Throughput in terms of number of FUs due to **(a)** sensing errors for input sets 5, 6, 7, and **(b)** sensing and negotiation errors for input sets 5, 6, and 8

increases, less idle slots are left for secondary transmission for each FU, and thus throughput degrades.

In mechanisms proposed in [11, 12], in channel negotiation after getting a response from BSU with the list of available channels, FUs have to wait for the next slot on identifying that the current channel is unavailable in the list sent by BSU. So although there are available channels, FU cannot use them for transmission till the next slot. In our proposed mechanisms, in channel negotiation FU senses channels one by one from the list sent by BSU pool immediately after getting the response from the pool. Even if the channel used for requesting BSU pool is not present in the list, FU does not need to wait for the start of the next slot to sense another channel. Thus utilization of idle slots increases. Usually, for Rayleigh fading channels, 25–50 microseconds (μs) sensing time is necessary to sense a channel of 200 kHz [17]. In most of our analysis, 2000 kHz bandwidth is used because in cloud RAN, extremely high-frequency bandwidth can be exploited [11]. 250–500 microseconds (μs) sensing time is enough to sense a channel of 2000 kHz.

For sensing one channel in a slot duration of 10 ms (ms):

$$\text{Percentage (\%) of sensing time} = \frac{250 \times 10^{-6}}{10 \times 10^{-3}} \times 100 = 2.5$$

In the case of our proposed mechanisms, FU has to sense all the 16 channels in a slot duration in the worst case. For sensing 16 channels in a slot duration of 10 ms (ms)

$$\text{Percentage (\%) of sensing time} = 2.5 \times 16 = 40$$

So, 60% of a slot duration is left for transmission, whereas 97.5% of a slot duration is left for transmission if one channel has to be sensed. But this problem does not affect the throughput of FU much because once channel negotiation is successful and transmission starts, FU does not have to sense all channels in every slot duration and the list of channels always contain less than 16 channels. Also, the throughput does not vary significantly with an increase in sensing time when the number of FU is greater than 6.

6 Conclusion

In this work, we have proposed a CR-based two-step process comprising channel sensing and channel negotiation to get idle slots for FUs' secondary transmission without hampering the QoS of MUs. The impact of errors in these mechanisms has been investigated by comparing throughput considering two kinds of error in channel sensing and three kinds of error in channel negotiation. Simulation results have shown that around 27 Mbps of maximum throughput can be achieved by using proposed mechanisms in H-CRAN femtocell. The addition of CR in H-CRAN reduces the possibility of interference between macro and femto networks in the case of co-channel deployment of femtocells. And the responsibilities of the centralized BBU pool are reduced because femtocell devices with spectrum sensing engine can autonomously detect free channels for transmission and hence simplify spectrum management. But errors in sensing and negotiation must be controlled as they can greatly affect the performance of FUs causing about 10 Mbps fall in maximum throughput.

References

1. Peng M, Li Y, Zhao Z, Wang C (2015) System architecture and key technologies for 5G heterogeneous cloud radio access networks. IEEE Netw 29:6–14. https://doi.org/10.1109/MNET.2015.7064897
2. Peng M, Sun Y, Li X, Mao Z, Wang C (2016) Recent advances in cloud radio access networks: system architectures, key techniques, and open issues. IEEE Commun Surv Tutorials 18:2282–2308. https://doi.org/10.1109/comst.2016.2548658
3. Al-Turjman F, Ever E, Zahmatkesh H (2019) Small cells in the forthcoming 5G/IoT: traffic modelling and deployment overview. IEEE Commun Surv Tutorials 21:28–65. https://doi.org/10.1109/COMST.2018.2864779
4. Zhang H, Jiang C, Cheng J, Leung V (2015) Cooperative interference mitigation and handover management for heterogeneous cloud small cell networks. IEEE Wirel Commun 22:92–99. https://doi.org/10.1109/MWC.2015.7143331
5. Tang Y, Yang P, Wu W, Mark J (2019) Cooperation-Based interference mitigation in heterogeneous cloud radio access networks. In: ICC 2019—2019 IEEE international conference on communications (ICC). IEEE, Shanghai, pp 1–6. https://doi.org/10.1109/ICC.2019.8761280
6. Peng M, Li Y, Jiang J, Li J, Wang C (2014) Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies. IEEE Wirel Commun 21:126–135. https://doi.org/10.1109/mwc.2014.7000980
7. Budhiraja I, Tyagi S, Tanwar S, Kumar N, Guizani M (2018) CR-NOMA based interference mitigation scheme for 5G femtocells users. In: 2018 IEEE global communications conference (GLOBECOM). IEEE, Abu Dhabi, pp 1–6. https://doi.org/10.1109/GLOCOM.2018.8647354
8. Qamar F, Hindia M, Dimyati K, Noordin K, Amiri I (2019) Interference management issues for the future 5G network: a review. Telecommun Syst 71:627–643. https://doi.org/10.1007/s11235-019-00578-4
9. Sahu G, Pawar S (2020) An approach to reduce interference using FFR in heterogeneous network. SN Comput Sci 1. https://doi.org/10.1007/s42979-020-0092-y
10. Ghosh J, Jayakody D, Qaraqe M (2018) Downlink capacity of OFDMA-CR based 5G femtocell networks. Phys Commun 29:329–335. https://doi.org/10.1016/j.phycom.2018.04.016
11. Meerja K, Shami A, Refaey A (2015) Hailing cloud empowered radio access networks. IEEE Wirel Commun 22:122–129. https://doi.org/10.1109/MWC.2015.7054727
12. Meerja K, Ho P, Wu B, Yu H (2013) Media access protocol for a coexisting cognitive femtocell network. Comput Netw 57:2961–2975. https://doi.org/10.1016/j.comnet.2013.06.017
13. Zhang H, Jiang C, Mao X, Chen H (2016) Interference-Limited resource optimization in cognitive femtocells with fairness and imperfect spectrum sensing. IEEE Trans Veh Technol 65:1761–1771. https://doi.org/10.1109/TVT.2015.2405538
14. Kumar A, Thakur P, Pandit S, Singh G (2019) Analysis of optimal threshold selection for spectrum sensing in a cognitive radio network: an energy detection approach. Wireless Netw 25:3917–3931. https://doi.org/10.1007/s11276-018-01927-y
15. Zhang H, Nie Y, Cheng J, Leung V, Nallanathan A (2017) Sensing time optimization and power control for energy efficient cognitive small cell with imperfect hybrid spectrum sensing. IEEE Trans Wireless Commun 16:730–743. https://doi.org/10.1109/TWC.2016.2628821

16. Wang W, Yang H (2018) Effect of imperfect spectrum sensing on slotted secondary transmission: energy efficiency and queuing performance. IEEE Trans Cognitive Commun Networking 4:764–772. https://doi.org/10.1109/TCCN.2018.2874457
17. Ghasemi A, Sousa, ES (2005) Collaborative spectrum sensing for opportunistic access in fading environments. In: First IEEE international symposium on new frontiers in dynamic spectrum access networks. DySPAN 2005. IEEE, Baltimore, pp. 131–136. https://doi.org/10.1109/DYSPAN.2005.1542627

A New Fairness Model Based on User's Objective for Multi-user Multi-processor Online Scheduling Problem

Debasis Dwibedy and Rakesh Mohanty

Abstract In multi-user multi-processor online scheduling, resources are shared among competing users, and fairness is considered to be a major performance criterion for resource allocation by the scheduler. Fairness ensures equality of resource sharing among the users. According to our knowledge, fairness based on the user's objective has neither been thoroughly explored nor a formal model has been well-defined in the literature. In this article, we propose a new fairness model for *Multi-user Multi-processor Online Scheduling Problem (MUMPOSP)*. We introduce and formally define quantitative fairness measures for an online scheduling algorithm based on optimization of makespan as an user's objective. Furthermore, we define unfairness and absolute fairness for an online scheduling algorithm. Lower bound results are shown for absolute fairness in a scheduling framework of equal length jobs. We show that our proposed fairness model can also measure algorithmic fairness by considering well-known optimality criteria such as sum of completion times, weighted sum of completion times and sum of flow times.

Keywords Multi-user system · Scheduling · Makespan · Performance measure · Fairness

1 Introduction

Fairness is an important performance criterion for a scheduler. Particularly, in multi-user systems, where several users compete for a set of resources (e.g., processor, memory) in order to achieve their objectives, the scheduler must guarantee fairness with respect to allocation of resources and user's objective. Though fairness has been studied based on resource allocation policies in the literature, there is less attention to devise a quantitative well-defined measure of fairness based on user's objectives.

User's objective as a fairness parameter has been motivated from the prevalent use of Web servers in client–server networking, grids and clusters in high-performance computing (HPC). Edge nodes in edge computing, service-oriented systems (SoS)

D. Dwibedy (✉) · R. Mohanty
Veer Surendra Sai University of Technology, Burla 768018, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_34

and supercomputers [1]. Unlike the traditional computing systems such as personal computer, the *SoS* supports multiple users. The users compete for system's resources for execution of their respective jobs. The most popular cluster scheduler *MAUI* [2] and the well-known *BOINC* platform [3] deal with a number of competing users, where each user submits a set of jobs simultaneously and desires *minimum time of completion (makespan)* for its submissions. A non-trivial challenge for the scheduler is to schedule jobs of multiple users in such a way that each user obtains its desired makespan.

Multi-user Multi-processor Online Scheduling Problem (MUMPOSP)

- **Inputs**: We are given a set $M = \{M_1, M_2, \ldots, M_m\}$ of m identical processors and a set of n jobs, where $m \geq 2$ and $n >>> m$. Let U_r represents a *user*, where $1 \leq r \leq k$ and J^r is the sequence of jobs requested by *user* U_r, where $J^r = (J_i^r | 1 \leq i \leq n_r)$ such that $J = \bigcup_{r=1}^{k} J^r$, $\sum_{r=1}^{k} n_r = n$ and $J^x \cap J^y = \phi$, where $x \neq y$ and $1 \leq x, y \leq k$. The processing time of job J_i^r is p_i^r, where $p_i^r \geq 1$.
- **Output**: A schedule (S) in which makespan for each U_r is denoted by $C_{\max}^r = \max\{c_i^r | 1 \leq i \leq n_r\}$, where c_i^r is the completion time of job J_i^r
- **Objective**: Minimization of $C_{\max}^r$, $\forall U_r$.
- **Constraint**: The scheduler can receive a batch of at most r jobs at any time step, and the jobs must be irrevocably scheduled before the arrival of next batch of jobs, where $1 \leq r \leq k$.
- **Assumption**: Jobs are independent and are requested from k parallel users, where $k \geq 2$.

Illustration of MUMPOSP. For simplicity and basic understanding of the readers, we illustrate an instance of *MUMPOSP* for scheduling of n jobs that are submitted by k users in Fig. 1. Here, $\{M_1, M_2, \ldots, M_m\}$ represent m identical machines and $\langle U_1, U_2, \ldots, U_{k-1}, U_k \rangle$ denote job sequences for k users, where each user has $\frac{n}{k}$ number of jobs. Jobs are submitted in batches online, where a batch is constructed after receiving exactly one job from each user (as long as a user has an unscheduled job). A batch consists of at least one job. Therefore, we have at least 1 batch, where $k = n$ and at most $n - k + 1$ batches, where any one of the users U_r has $n_r = n - k + 1$, and remaining users have exactly one job each. Each U_r seeks to obtain a minimum value for its makespan ($C_{\max}^r$) as the output, rather than the overall makespan ($C_{\max}$) of the system. Hence, it is indispensable for the scheduler to be fair while optimizing the $C_{\max}^r$, $\forall U_r$.

Representation of MUMPOSP. By following general framework $\alpha|\beta|\gamma$ of Graham et al. [4], we represent MUMPOSP as MUMPOSP(k, $P_m|C_{\max}^r$), where P_m denotes m identical machines and k represents number of users.

Perspectives of Fairness. Fairness has been considered and studied as a major performance criterion for scheduling algorithms in multi-user systems [5, 6] from two perspectives such as allocation of resources to the users and user's objective. Fairness of an algorithm with respect to resource allocation guarantees uniform allocation

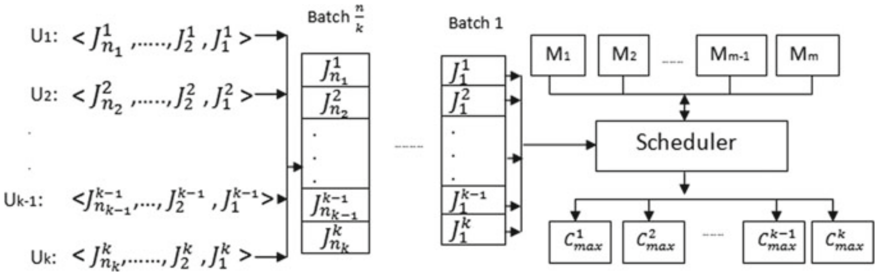

U1: <J^1_{n_1},……,J^1_2, J^1_1>
U2: <J^2_{n_2},……,J^2_2, J^2_1>
Uk-1: <J^{k-1}_{n_{k-1}},…,J^{k-1}_2, J^{k-1}_1>
Uk: <J^k_{n_k},……,J^k_2, J^k_1>
Batch n/k
J^1_{n_1}
J^2_{n_2}
J^{k-1}_{n_{k-1}}
J^k_{n_k}
Batch 1
J^1_1
J^2_1
J^{k-1}_1
J^k_1
M1
M2
Mm-1
Mm
Scheduler
C^1_max
C^2_max
C^{k-1}_max
C^k_max

Fig. 1 Illustration of MUMPOSP for k users with equal number of jobs

of resources to the competing users [7]. The resources to be shared are application-dependent. For example, in client-server networking, the resources such as link bandwidth, network delay and specific time quantum can be shared [8, 9], whereas in case of *HPC* systems, the resources such as processors, memory and time slices can be shared [10, 11].

Algorithmic fairness based on user's objective is evaluated by the objective values achieved for respective users. An equality in the obtained objective values for a user ensures fairness of a scheduling algorithm. It is important for a fairness measure to define the equality for quantifying gap of an achieved objective value from the defined equality.

Related Work. Fairness as a quantitative performance measure based on resource allocation was studied by Jain et al. [7]. A set of properties for an ideal fairness measure was defined, and a fairness index $F(x)$ was proposed for resource allocation schemes. $F(x)$ is defined as follows: if any scheduling algorithm assigns resources to k competing users such that rth user gets an allocation of x_r. Then,

$$F(x) = \frac{(\sum_{r=1}^{k} x_r)^2}{\sum_{r=1}^{k} {x_r}^2}, \quad \text{where } x_r \geq 0.$$

The value of $F(x)$ is bounded between 0 and 1 to show percentage of fairness and discrimination of a resource allocation scheme for each user. Fairness based on sharing of resources such as processors, memory, system clock and system bus in multi-programmed multi-user system was well studied in [10–12]. Some recent works on fairness in scheduling online jobs on multi-user systems can be found in [13, 14]. To the best of our knowledge, fairness of online scheduling algorithms based on user's objective has not been exhaustively studied and explored the literature.

In [15–18], *stretch* matrix has been considered as a user's objective-based fairness measure for resource scheduling algorithms in multi-user systems. Here, *Stretch* (d^r_A) has been defined as a degradation factor in the objective value obtained by any algorithm A for each user U_r. Let us consider V^r_A be the objective value achieved by algorithm A and V^r_{OPT} be the optimum objective value for respective U_r. Then, stretch has been defined as follows:

$$d_A^r = \frac{V_A^r}{V_{\mathrm{OPT}}^r}$$

The objective of any scheduling algorithm is to incur an equal stretch for each U_r to ensure fairness. Stretch matrix guarantees fairness based on equality in achieved objective values. However, it fails to depict the exact value of fairness per user as well as overall fairness of a scheduling algorithm. Stretch matrix does not capture the discrimination of a scheduling algorithm for the deprived users. Therefore, it is quintessential to define a formal fairness measure based on user's objective.

Our Contributions. We propose a novel model to evaluate fairness of online algorithms in the *Multi-user Multi-processor Online Scheduling Problem (MUMPOSP)*. We introduce and formally define quantitative fairness measures in our proposed model by considering optimization of makespan as user's objective. Furthermore, we define unfairness and absolute fairness of an online scheduling algorithm. We obtain lower bound results for the absolute fairness for a framework of m identical machines with equal length jobs. We show that our proposed model can be served as a framework for measuring algorithmic fairness by considering other optimality criteria such as sum of completion times, weighted sum of completion times and sum of flow times.

2 Our Proposed Fairness Model

We develop a new model, in which we define five quantitative measures to ensure algorithmic fairness. Instead of considering the resource allocation at the input level, our model considers the achieved value of user's makespan at the output level to determine the fairness of a scheduling algorithm. The model captures the issues of relative and global parameters for fairness by a *Fairness Index (FI)*. The issues of unfairness is captured by a *Discrimination Index (DI)*. The *FI* includes fairness parameters such as *Relative Fairness (RF)* and *Global Fairness (GF)*. Higher value of any fairness parameter indicates more fair algorithm. The *DI* includes unfairness measures such as *User Discrimination Index (UDI)*, *Global Discrimination Index (GDI)* and *Relative Discrimination Index (RDI)*. Lower value of any unfairness measure indicates higher degree of fairness of the algorithm. Before defining fairness and unfairness parameters, we illustrate our novel model and discuss the characteristics of a good fairness model as follows.

Illustration of Our Proposed Fairness Model. We illustrate our proposed fairness model as shown in Fig. 2. The model quantitatively defines the fairness of an online scheduling algorithm by taking into account the makespan ($C_{\max}^r$) of individual user in the *MUMPOSP* setup.

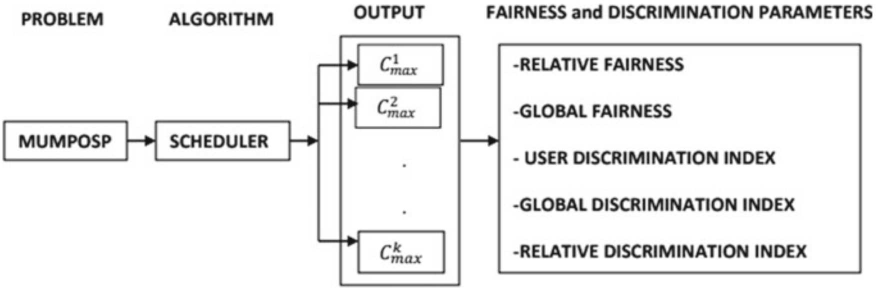

PROBLEM
ALGORITHM
OUTPUT
FAIRNESS and DISCRIMINATION PARAMETERS
MUMPOSP
SCHEDULER
C^1_{max}
C^2_{max}
C^k_{max}
-RELATIVE FAIRNESS
-GLOBAL FAIRNESS
- USER DISCRIMINATION INDEX
-GLOBAL DISCRIMINATION INDEX
-RELATIVE DISCRIMINATION INDEX

Fig. 2 A fairness model based on user's objective

2.1 Characteristics of a Good Fairness Model

A fairness model evaluates the performance of a scheduling algorithm based on the achieved makespan for each user. Recall that in [15–18], *Stretch* was considered as a user's objective-based fairness measure. For instance, if a scheduling algorithm A obtains makespans for three users as 5, 10, and 15, respectively, where their respective optimum makespans are 1, 5 and 10, then *stretch* defines the following degradation factors for respective users: $d^1_A = 5$, $d^2_A = 2$, and $d^3_A = 1.5$.

Before formally defining fairness and unfairness parameters, we present the characteristics of a good fairness model as follows. A good fairness model must be:

- *Finitely Bounded*—The fairness of a scheduling algorithm is bounded within a finite interval, preferably between 0 and 1 for meaningful representation of fairness with respect to each user.
- *Consistent*—If any change in the scheduling policy results in different makespans for at least one user, then the change in the fairness parameters must be reflected for the concerned users as well as in the overall fairness of the policy.
- *Independent of Input Size*—It is applicable to any number of users with any number of jobs and machines.
- *Independent of Scale*. It must be able to measure fairness irrespective of units of measurement of processing time of the jobs such as seconds or milliseconds, microseconds or nanoseconds. The measuring unit must be uniform or inter convertible.

In addition to the above-mentioned properties, we also consider relative and overall fairness as an essential feature to develop our fairness parameters. We believe that the model must represent *relative equality* among achieved objective values for the users to show fairness of an algorithm for each user. For example, the users may not seek equal makespan as a gesture of fairness; however, they expect from an online scheduling algorithm to obtain an equal ratio between the *desired makespan (optimum value)* to the achieved makespan for all users. The value obtained by an algorithm for relative equality leads to *relative fairness* with respect to each user.

Also, the model must show *overall fairness* of an algorithm with respect to all users, which can lead to the comparison of the fairness of different scheduling policies.

2.2 Our Proposed Fairness and Unfairness Parameters

By considering the above-mentioned desirable properties, we now define formal measures of fairness and unfairness for MUMPOSP as follows.

Let A be an online scheduling algorithm. If algorithm A schedules jobs of k competing users on m identical processors such that rth user obtains a makespan of C_A^r, then we define the following fairness parameters.

Definition 1 The **Relative Fairness (RF)** obtained by algorithm A for any user U_r is defined as:

$$\mathrm{RF}(C_A^r) = \frac{C_{\mathrm{OPT}}^r}{C_A^r}, \quad \text{where } C_{\mathrm{OPT}}^r = \frac{\sum_{i=1}^{n_r} p_i^r}{m} \tag{1}$$

Corollary 1 *The* ***Relative Fairness Percentage (RFP)*** *for any user U_r obtained by algorithm A is defined as:*

$$RFP(C_A^r) = RF(C_A^r) \cdot 100 \tag{2}$$

Definition 2 The **Global Fairness (GF)** of algorithm A for k users is defined as:

$$\mathrm{GF}(C_A, k) = \frac{1}{k} \cdot \sum_{r=1}^{k} (\mathrm{RF}(C_A^r)) \tag{3}$$

Corollary 2 *The* ***Global Fairness Percentage (GFP)*** *of any algorithm A for k users is defined as:*

$$GFP(C_A, k) = GF(C_A, k) \cdot 100 \tag{4}$$

If algorithm A schedules jobs of k competing users such that rth user obtains a makespan of C_A^r, then we define **Fairness Index** for algorithm A represented by 2-tuple with two parameters such as RF and GF as follows

$$\mathrm{FI}(C_A, k) = \langle \{\mathrm{RF}(C_A^r) | 1 \le r \le k\}, \mathrm{GF}(C_A, k)\rangle \tag{5}$$

Example 1 Let us consider three departments {CSE, MAT, PHY} of a University as three users $\{U_1, U_2, U_3\}$, submitting jobs by MUMPOSP model to a centralized supercomputer (having 2 identical machines) in order to finish their respective projects at the earliest. Let us denote the job sequences of U_1, U_2, and U_3 as $U_1 = \langle J_1^1/1, J_2^1/2\rangle$, $U_2 = \langle J_1^2/3, J_2^2/4\rangle$ and $U_3 = \langle J_1^3/5, J_2^3/6\rangle$, respectively. Suppose that the supercomputer runs an online scheduling algorithm *Alg* that schedules the jobs of U_1, U_2 and U_3 and obtains $C_{\mathrm{Alg}}^1 = 11$, $C_{\mathrm{Alg}}^2 = 9$ and $C_{\mathrm{Alg}}^3 = 10$, then we have, $\mathrm{RF}(C^1_{\mathrm{Alg}}) = \frac{1.5}{11} = 0.13$ and $\mathrm{RFP}(C^1_{\mathrm{Alg}}) = 13\%$, $\mathrm{RF}(C^2_{\mathrm{Alg}}) = \frac{3.5}{9} = 0.38$ and $\mathrm{RFP}(C^2_{\mathrm{Alg}}) = 38\%$, $\mathrm{RF}(C^3_{\mathrm{Alg}}) = \frac{5.5}{10} = 0.55$ and $\mathrm{RFP}(C_{\mathrm{Alg}}) = 55\%$. Therefore, we have $\mathrm{GF}(C_A, 3) = 0.35$ and $\mathrm{GFP}(C_A, 3) = 35\%$.

Definition 3 The **Unfairness** of algorithm A for MUMPOSP with respect to each user U_r is defined by **User Discrimination Index** as:

$$\mathrm{UDI}^r_A = 1 - \mathrm{RF}(C^r_A) \tag{6}$$

Definition 4 The **Overall Unfairness** of algorithm A for k users is defined by **Global Discrimination Index** as:

$$\mathrm{GDI}(C^r_A, k) = 1 - \mathrm{GF}(C^r_A, k) \tag{7}$$

Definition 5 The **Realtive Discrimination Index (RDI)** of any algorithm A for MUMPOSP with respect to each user U_r is defined as:

$$\mathrm{RDI}^r_A = \begin{cases} \mathrm{GF}(C^r_A, k) - \mathrm{RF}(C^r_A), & \text{if } \mathrm{RF}(C^r_A) < \mathrm{GF}(C^r_A, k) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

If algorithm A schedules jobs of k competing users such that rth user obtains a makespan of C^r_A, then we define **Discrimination Index** for algorithm A as 3-tuple with three parameters such as UDI, GDI and RDI as follows.

$$\mathrm{DI}(C_A, k) = \langle\ \{\mathrm{UDI}^r_A \mid 1 \le r \le k\}, \mathrm{GDI}(C^r_A, k), \{\mathrm{RDI}^r_A | 1 \le r \le k\}\ \rangle \tag{9}$$

Example 2 Let us consider algorithm A results in relative fairness for U_1, U_2, U_3 and U_4 as 0.6, 0.6, 0.6 and 0.2 respectively. We now have $\mathrm{GF}(C^r_A, 4) = 0.5$. Therefore, $\mathrm{UDI}^1_A = 1 - 0.6 = 0.4$, $\mathrm{UDI}^2_A = 1 - 0.6 = 0.4$, $\mathrm{UDI}^3_A = 1 - 0.6 = 0.4$, $\mathrm{UDI}^4_A = 1 - 0.2 = 0.8$, $\mathrm{GDI}(C^r_A, 4) = 1 - 0.5 = 0.5$ and $\mathrm{RDI}^4_A = 0.5 - 0.2 = 0.3$.

3 Absolute Fairness and Lower Bound Results

We define absolute fairness as a quantitative measure and provide lower bound results of absolute fairness in generic *MUMPOSP* setting with equal length jobs. Let A be an online scheduling algorithm for the setup MUMPOSP (k, $P_m|C^r_{\max}$).

Definition 6 Algorithm A achieves **Absolute Fairness** if $\mathrm{RF}(C^r_A)$ is same $\forall U_r$, where $1 \le r \le k$.

Lemma 1 *If any algorithm A incurs $RDI^r_A = 0$, $\forall U_r$, then it achieves absolute fairness.*

Proof If $\mathrm{RDI}^r_A = 0, \forall U_r, 1 \le r \le k$, then by Eq. (8), we have

$$\mathrm{RF}(C^r_A) \ge \mathrm{GF}(C^r_A, k) \tag{10}$$

By Eqs. (3) and (10), we can infer that

$$\mathrm{RF}(C^r_A) = \mathrm{GF}(C^r_A, k), \forall U_r.$$

Therefore, Lemma 1 holds true. □

Definition 7 Any Algorithm A is b-**fair**, if it achieves $\mathrm{RF}(C^r_A) = b$ for all U_r, where $1 \le r \le k$ and $0 < b \le 1$.

Theorem 1 *Any online algorithm A achieves absolute fairness in the setup MUMPOSP (k, $P_2|C^r_{\max}$) such that $\frac{C^r_{OPT}}{C^r_A} \ge \frac{1}{k}$, $\forall U_r$, where $k \ge 2$ and $1 \le r \le k$.*

Proof Let us consider an instance of MUMPOSP (k, $P|C^r_{\max}$), where $k = 2$. We analyze two cases based on n_r as follows.

Case 1: $n_1 \ne n_2$.
Case 1(a): If the first job pair (J^1_1, J^2_1) is scheduled on different machines. Let us consider the following instance $U_1 : \langle J^1_2/2, J^1_1/1\rangle, U_2 : \langle J^2_1/1\rangle$, where each job is specified by its processing time. Assigning $J^1_1/1$ and $J^2_1/1$ to machines M_1 and M_2, respectively, followed by the assignment of $J^1_2/2$ to either of the machines such that $C^1_A = 3$ and $C^2_A = 1$, where $C^1_{\mathrm{OPT}} \ge 1.5$ and $C^2_{OPT} \ge 0.5$. Therefore, we have $\frac{C^1_{\mathrm{OPT}}}{C^1_A} \ge \frac{1}{2}$ and $\frac{C^2_{\mathrm{OPT}}}{C^2_A} \ge \frac{1}{2}$.
Case 1(b): If the first job pair (J^1_1, J^2_1) is scheduled on the same machine. Let us consider the following instance $U_1 : \langle J^1_3/2, J^1_2/1, J^1_1/1\rangle, U_2 : \langle J^2_2/2, J^2_1/1\rangle$. If the first job pair ($J^1_1/1$, $J^2_1/1$) is scheduled either on machine M_1 or on M_2, then by assigning the next pair of jobs (J^1_2, J^2_2) to the same or different machines, followed by the assignment of job $J^1_3/2$ such that $C^1_A = 4$ and $C^2_A = 3$, where $C^1_{\mathrm{OPT}} \ge 2$ and $C^2_{\mathrm{OPT}} \ge 1.5$. Therefore, we have $\frac{C^1_{\mathrm{OPT}}}{C^1_A} \ge \frac{1}{2}$ and $\frac{C^2_{\mathrm{OPT}}}{C^2_A} \ge \frac{1}{2}$.
Case 2: $n_1 = n_2$.
Case 2(a): If the first job pair (J^1_1, J^2_1) is scheduled on different machines. Let us consider the following instance $U_1 : \langle J^1_3/2, J^1_2/1, J^1_1/1\rangle, U_2 : \langle J^2_3/2, J^2_2/2, J^2_1/1\rangle$. Assigning jobs $J^1_1/1$ and $J^2_1/1$ to machines M_1 and M_2 respectively, followed by the assignment of the subsequent jobs as shown in Fig. 3a, such that $C^1_A = 4$ and $C^2_A = 5$, where $C^1_{\mathrm{OPT}} \ge 2$ and $C^2_{\mathrm{OPT}} \ge 2.5$. Therefore, we have $\frac{C^1_{\mathrm{OPT}}}{C^1_A} \ge \frac{1}{2}$ and $\frac{C^2_{\mathrm{OPT}}}{C^2_A} \ge \frac{1}{2}$.
Case 2(b): If the first job pair (J^1_1, J^2_1) is assigned to the same machine. We consider the same instance of Case 2(a). Assigning $J^1_1/1$ and J^2_1 on either machine M_1 or on M_2, followed by the assignment of the subsequent jobs as shown in Fig. 3b such that $C^1_A = 4$ and $C^2_A = 5$. Therefore, we have $\frac{C^1_{\mathrm{OPT}}}{C^1_A} \ge \frac{1}{2}$ and $\frac{C^2_{\mathrm{OPT}}}{C^2_A} \ge \frac{1}{2}$. □

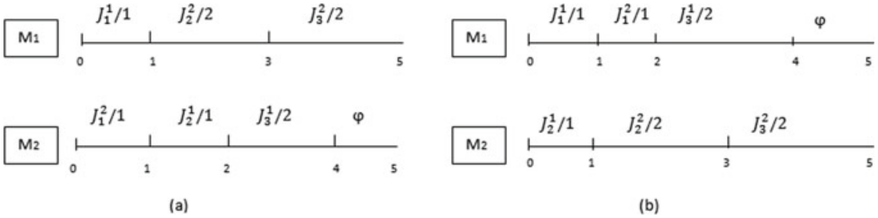


Fig. 3 Illustration of case 2

3.1 Results on Absolute Fairness in MUMPOSP with m Identical Machines for Equal Length Jobs

For ease of understanding, we analyze the lower bound of absolute fairness for any online algorithm in a generic *MUMPOSP* setting, where each user has equal number of jobs, and all jobs have equal processing time of x unit, where $x \geq 1$. The objective of each user is to obtain a minimum $C^r_{\max}$. We formally denote the problem as MUMPOSP $(k, P_m|p^r_i = x|C^r_{\max})$.

Lemma 2 *Let A be an online scheduling algorithm. In MUMPOSP $(k, P_m|p^r_i{=}x|C^r_{\max})$ with $k = b \cdot m$, algorithm A obtains $C^r_A \leq b \cdot \sum_{i=1}^{n_r} p^r_i$, for each U_r, respectively, where $1 \leq r \leq k$, $m \geq 2$ and $b \geq 1$.*

Proof We prove Lemma 2 by method of induction on number of jobs per user (n_r) as follows.

Induction Basis: Let us consider $k = m = 2$, $n_1 = n_2 = 1$ and $p^1_1 = p^2_1 = 1$. Clearly, $C^r_A = 1 \leq b \cdot 1 \cdot 1$, where $r = 1, 2$ and $b \geq 1$.

Induction Hypothesis: Let us consider $k = b \cdot m, n_r = \frac{n}{k} = y$, where $y \geq 1, b \geq 1$ and $n = \sum_{r=1}^{k} n_r$.

We assume that

$$C^r_A \leq b \cdot \sum_{i=1}^{n_r} p^r_i \leq b \cdot x \cdot y \tag{11}$$

Inductive Step: For $n_r = y + 1$ with $p^r_i = x, \forall J^r_i$. We have to show that $C^r_A \leq (y + 1) \cdot b \cdot x$.

By Eq. (11), we have $C^r_A = y \cdot b \cdot x$ with $n_r = y$. When we add extra one job to each user, we have by *Induction Basis* $C^r_A = b \cdot x \cdot y + (b \cdot x) = (y + 1) \cdot b \cdot x$. Therefore, Lemma 2 holds true. □

Lemma 3 *Any algorithm A is $\frac{1}{k}$-fair for MUMPOSP $(k, P_m|p^r_i = x|C^r_{\max})$ with $k = b \cdot m$, where $m \geq 2$ and $b \geq 1$.*

Proof By Lemma 2, we have

$$C^r_A \leq b \cdot \sum_{i=1}^{n_r} p^r_i, \forall U_r \tag{12}$$

We have the fair optimum bound as

$$C^r_{\text{OPT}} \geq \frac{\sum_{i=1}^{n_r} p^r_i}{m}, \quad \forall U_r \tag{13}$$

By Eqs. (12) and (13), we have

$$\frac{C^r_{\text{OPT}}}{C^r_A} \geq \frac{1}{k}, \ \forall U_r. \tag{14}$$

Therefore, *Lemma* 3 holds true. □

Lemma 4 *In MUMPOSP* $(k, P_m|p^r_i = x|C^r_{\max})$ *with* $k > m$*, algorithm A obtains* $C^r_A \leq \lceil \frac{n}{m} \rceil \cdot x$*, for each* U_r *respectively, where* $k \neq m \cdot b$ *for* $b \geq 1$.

Proof The correctness of Lemma 4 is shown by method of induction on n_r as follows.

Induction Basis: Let us consider $m = 2, k = 3, n_r = 1$ and $p^r_i = 1$. Now, we have $n = n_r \cdot k = 3$.

Clearly, $C^r_A \leq 2 = \lceil \frac{n}{2} \rceil \cdot 1$.

Induction Hypothesis: Let us consider $n_r = \frac{n}{k} = y$, $p^r_i = x$ and $k > m$ with $k \neq m \cdot b$ for $b \geq 1$. We assume that $C^r_A \leq \lceil \frac{n}{m} \rceil \cdot x, \forall U_r$.

Inductive Step: We show that $C^r_A \leq \lceil \frac{n+k}{m} \rceil \cdot x$ for $n_r = y + 1, \forall U_r$.

By our Induction Basis, for one extra job of each user U_r, where $1 \leq r \leq k$, algorithm A incurs an additional time of $\lceil \frac{k}{m} \rceil \cdot x$ for each U_r.

Therefore, $C^r_A \leq \lceil \frac{n}{m} \rceil \cdot x + \lceil \frac{k}{m} \rceil \cdot x \leq \lceil \frac{n+k}{m} \rceil \cdot x$

Thus, Lemma 4 holds true. □

Theorem 2 *Any Algorithm A is* $\frac{1}{k}$*-fair for MUMPOSP* $(k, P_m|p^r_i = x|C^r_{\max})$*, where* $k \geq m$ *and* $m \geq 2$.

Proof Theorem 2 holds true by Lemma 3 for $k = m \cdot b$, where $b \geq 1$.

By Lemma 4, we have

$$C^r_A \leq \lceil \frac{n}{m} \rceil \cdot x \tag{15}$$

By Eq. (13), we have $C^r_{\text{OPT}} \geq \frac{\frac{n}{k} \cdot x}{m}$.

Implies,

$$C^r_{\text{OPT}} \geq \frac{n \cdot x}{k \cdot m} \tag{16}$$

By Eqs. (14) and (15), we have

$$\begin{aligned} \frac{C^r_{\text{OPT}}}{C^r_A} &\geq \frac{\frac{n \cdot x}{k \cdot m}}{\frac{n \cdot x}{m}} \\ &\geq \frac{n \cdot x \cdot m}{n \cdot k \cdot m \cdot x} \geq \frac{1}{k}. \end{aligned}$$ □

4 Fairness Measure Using Flow Time and Completion Time as User's Objective

We show that our proposed Fairness Index can be served as a framework for measuring fairness of any algorithm based on well-known user's objectives such as *sum of completion times* (S^r), weighted sum of completion times (W^r) and sum of flow times (SF^r). Selection of an user's objective is application-dependent. For instance, users of interactive systems require optimized value for respective flow time f^r, where f_i^r of any J_i^r is the difference between its completion time c_i^r and arrival time t_i^r. We now define relative fairness measures based on the above-mentioned user's objectives, respectively, by our proposed *FI*.

- **Sum of Completion Times** (S^r): Here, the objective for each U_r is to obtain a minimum $S^r = \sum_{i=1}^{n_r} c_i^r$. The relative fairness for any U_r, obtained by any algorithm A based on S^r is defined as

$$R_A(S_A^r) = \frac{S_{\mathrm{OPT}}^r}{S_A^r}, \quad \text{where } S_{\mathrm{OPT}}^r \text{ is the optimum value for } S^r.$$

- **Weighted Sum of Completion Times** (W^r): Here, the c_i^r is associated with certain positive weight w_i^r. The objective for each U_r is to obtain a minimum $W^r = \sum_{i=1}^{n_r} w_i^r \cdot c_i^r$. The relative fairness for any U_r obtained by algorithm A based on W^r is defined as

$$R_A(W_A^r) = \frac{W_{\mathrm{OPT}}^r}{W_A^r} \quad \text{where, } W_{\mathrm{OPT}}^r \text{ is the optimum value for } W^r.$$

- **Sum of Flow Times** (SF^r): Here, each U_r wants a minimum value for respective $\mathrm{SF}^r = \sum_{i=1}^{n_r} f_i^r$, where f_i^{*r} is the desired value of f_i^r and $\mathrm{SF}_{\mathrm{OPT}}^r = \sum_{i=1}^{n_r} f_i^{*r}$. The relative fairness for any U_r obtained by algorithm A based on SF^r is defined as

$$R_A(\mathrm{SF}_A^r) = \frac{\mathrm{SF}_{\mathrm{OPT}}^r}{\mathrm{SF}_A^r}.$$

5 Concluding Remarks and Scope of Future Work

In this work, we make an attempt to address the non-trivial research challenge of defining a new fairness model with quantitative measures of algorithmic fairness for *Multi-user Multi-processor Online Scheduling Problem (MUMPOSP)* based on user's objective. We formally presented the *MUMPOSP* setting with an illustration followed by a discussion on perspectives of fairness in *MUMPOSP*. We have proposed a new fairness model and have defined five quantitative measures to ensure algorithmic fairness by considering minimization of makespan as the user objective.

Lower bound results on absolute fairness of an online scheduling algorithm have been shown in *MUMPOSP* setup with equal length jobs. We have shown how our proposed fairness measure can be served as a framework for measuring algorithmic fairness based on well-known user's objectives such as sum of completion times, weighted sum of completion times and sum of flow times.

Scope of Future Work. We assumed a ideal theoretical bound for C^r_{OPT}. It is still open to explore a realistic bound for C^r_{OPT}. A non-trivial challenge is to compare the fairness of any two online scheduling algorithms A and B, when global fairness of algorithms A and B are same, whereas relative fairness of A is more than that of B for some users or vice-versa. In this scenario, it is interesting to make a trade-off by considering the number of users and individual relative fairness for each user to compare the fairness of two different algorithms.

References

1. Emmott S, Rison S (2020) Towards 2020 science. Tech. Rep., Microsoft Research Cambridge, Working Group Research
2. Jackson D, Snell Q, Clement M (2001) Core algorithms of the Maui scheduler. In: Feitelson DG, Rudolph I (eds) 7th international workshop JSSPP. LNCS, vol 2221. Springer, Heidelberg
3. Anderson DP (2004) BOINC: a system for public-resource computing and storage. In: 5th IEEE/ACM international workshop on grid computing, pp 4–10
4. Graham RL, Lawer EL, Lenstra JK, Rinnooy kan AH (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann Discrete Math 5:287–326
5. Jaffe JM (1980) A decentralized optimal multiple user flow control algorithm. In: Proceedings of the international conference computer communications, Atlanta, GA
6. Jaffe JM (1981) Bottleneck flow control. IEEE Trans Commun COM-29(7):954–962
7. Jain RK, Chiu DMW, Hawe WR (1984) A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, TR-301
8. Bharath-Kumar K, Jaffe JM (1981) A new approach to performance-oriented flow control. IEEE Trans Commun COM-29(4):427–435
9. Sauve JP, Wong JW, Field JA (1980) On fairness in packet switching networks. In: Proceedings of the 21st IEEE Computer Society international conference, COMPCon 80, Washington, DC, pp 466–470
10. Kay J, Lauder P (1988) A fair share scheduler. Commun ACM 31(1):44–55
11. Feitelson DG (1997) Job scheduling in multi-programmed parallel systems (extended version). IBM Research Report, RC19790(87657), Second Revision
12. Vandierendonck H, Seznec A (2011) Fairness metrics for multi-threaded processors. IEEE Comput Archit Lett 10(1):4–7
13. Sun H, Hsu WJ, Cao Y (2014) Competitive online adaptive scheduling for sets of parallel jobs with fairness and efficiency. J Parallel Distrib Comput 74(3):2180–2192
14. Bian S, Huang X, Shao Z (2019) Online task scheduling for fog computing with multi-resource fairness. In: Proceedings of the 90th IEEE vehicular technology conference, Honolulu, HI, USA, pp 1–5
15. Bender MA, Muthukrishnan S, Rajaraman R (2002) Improved algorithms for stretch scheduling. In: Proceedings of the 13th annual ACM-SIAM symposium on discrete algorithms (SODA), pp 762–771

16. Legrand A, Su A, Vivien F (2006) Minimizing the stretch when scheduling flows of biological requests. In: Symposium on parallelism in algorithms and architectures (SPAA)
17. Saule E, Trystram D (2009) Multi users scheduling in parallel systems. In: Proceedings of IEEE international parallel and distributed processing symposium, Washington, DC, USA, pp 1–9
18. Pinheiro VG (2014) The management of multiple submissions in parallel systems: the fair scheduling approach. PhD Thesis, Institute of Mathematics and Statistics, University of Sao Paulo, Brazil

Att-PyNet: An Attention Pyramidal Feature Network for Hand Gesture Recognition

Gopa Bhaumik, Monu Verma, Mahesh Chandra Govil, and Santosh Kumar Vipparthi

Abstract This paper proposes a novel deep network Att-PyNet: an attention pyramidal feature network for hand gesture recognition. The proposed Att-PyNet comprises three feature streams: multi-scale feature extractor (MSFE), attention pyramid module, and locally connected (LC) layer. The MSFE is introduced to enrich the proposed Att-PyNet model with features of macro- and micro-edges by learning the complementary features of multi-receptive fields. The attention pyramid module is designed to carry forward the high-level features to the lower layer by maintaining the distinctive quality. The locally connected layer is adopted to enhance the discriminative capability of the proposed network by preserving the pertinent context information. The Att-PyNet is a computationally effective model as it holds very less 565K parameters than state-of-the-art models and can be easily deployed in a resource-constrained platform. The effectiveness of the proposed Att-PyNet is evaluated on three standard datasets: MUGD, Triesch, and ASL FingerSpelling. The quantitative and qualitative results validate that Att-PyNet outperforms the state-of-the-art hand gesture approaches.

Keywords Att-PyNet · Multi-scale feature extractor · Attention pyramid module · Locally connected · Hand gesture

1 Introduction

Computing technology greatly streamlined communication between man and computer via unprecedented advances in programming. But till date, these advances are only limited to direct explicit communication. Non-verbal communication, a major part of human interactions, has also received significant traction in the field of human–computer interaction. Human–computer interaction finds its use in variety

G. Bhaumik (✉) · M. C. Govil
National Institute of Technology Sikkim, Sikkim, India
e-mail: gopa.bhaumik09@nitsikkim.ac.in

M. Verma · S. K. Vipparthi
Malaviya National Institute of Technology Jaipur, Jaipur, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_35

of applications such as entertainment, gaming control, or communication in contactless environments. Extensive research has been done in the area of human–computer interaction, a field which practitioners believe will bring the next iterations of technological advancements. When computers are able to comprehend human words or even thoughts—the application possibilities are numerous. An important subset of the aforesaid research work is human–computer interaction by way of hand gesture recognition (HGR)—an alternative to input devices like mouse and keyboard. Hand gestures are one of the most common body language forms used for interaction and communication. Hand gesture recognition can be categorized in multiple ways based on observable features and acquisition techniques [1]. Depending on spatial and temporal relationships, gestures can be classified either as static or dynamic. In static hand gestures, the position of the hand remains fixed while in dynamic the position of hand changes continuously with respect to time. Based on the acquisition technique, HGR can be classified either as sensor-based or vision-based approaches.s The sensor-based approach uses gloves, gyroscope, accelerometer, etc., to capture the motion and orientation of the hand gesture whereas vision-based approach captures gesture images using camera, webcam, etc. In a sensor-based approach, the users are required to wear devices like gloves that make the process resource-intensive. But vision-based approach overcomes this limitation of the hardware.

In the past decades, several papers exploit handcrafted feature descriptors like LBP [2], HOG [3], and EXTRA [4] for HGR. However, these techniques involve intensive computation and are not fully robust. They are designed for specific tasks and are highly influenced by environmental conditions such as illumination variation and noise. Furthermore, with the introduction of a deep learning approach—the performance of the HGR system is increased as compared to the existing handcrafted feature extraction techniques [5–7]. The multiple layers in the deep learning techniques extract high-level features, which boost the efficiency of the HGR system. This has resulted in the extensive development of deep learning approaches for HGR.

Recently, researchers are increasingly employing convolutional neural networks for detection and recognition of human hand gestures. Yamashita et al. [5] develop a bottom-up structured deep convolutional neural network (CNN) for localization and classification by comprising a special layer for binary image extraction. The binary layer is introduced to ignore the background complexity and illumination changes. The identification of the hand region is the primary task in a vision-based hand gesture recognition system. Due to the cluttered background and illumination variation, it is difficult to extract the exact hand region from an image. Therefore, to solve this issue, Paul et al. [6] propose an HGR model by incorporating depth thresholding and histogram thresholding for hand segmentation. Zhang et al. [8] propose a two-stage hand gesture recognition system. In the first stage, hand key points are localized using convolutional pose machine, and in the second stage, fuzzy Gaussian mixture models are used to classify hand gestures. Hu et al. [9] design a hand gesture recognition system based on deep learning approach that controls flights of unmanned aerial vehicles (UAV). Neethu et al. [7] employ connected component analysis to segment the figure tips from the hand region. The segmented hand regions are then classified using CNN networks. Huang et al. [10] use skin color segmentation

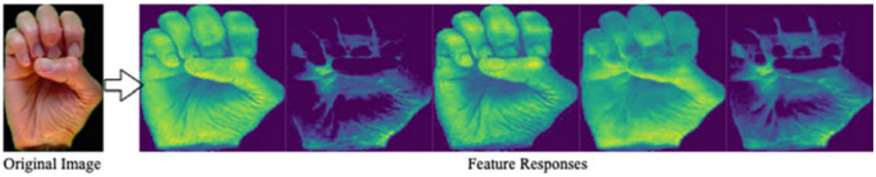


Fig. 1 Feature response generated by the Att-PyNet

for hand region extraction from hand gesture videos and classification is done using pyramidal pooling and attention mechanism, respectively. Nguyen et al. [11] propose a neural network based on symmetric positive definite (SPD) manifold learning for skeletal hand gesture recognition. The architecture uses spatial and temporal Gaussian aggregation of the joint features. Nuzzi et al. [12] develop a smart hand gesture recognition system for collaborative robots that use a faster R-CNN object detector to track the hands in RGB image. Wu et al. [13] propose a double channel CNN that processes both hand gesture and hand edge images in dual-stream mode. Pinto et al. [14] propose CNN-based classifier for static hand gesture recognition. The input gestures are preprocessed before the gestures are fed to the classifier. The preprocessing steps include morphological filtering, contour generation, polygonal approximation, and segmentation for better feature extraction.

Motivated by the existing CNN-based networks, we design a robust and portable Att-PyNet: an attention pyramidal feature network by introducing three units: multi-scale feature extraction module (MSFE), attention pyramid module, and LC layer, for precise hand gesture recognition. The main contribution of this paper is summarized as follows:

- An end-to-end Att-PyNet: An attention pyramidal feature network is proposed for hand gesture recognition. The Att-PyNet model consists of three feature encoding streams: MSFE, attention pyramid module, and LC layer.
- The MSFE is introduced to extract coarse and minute level edges by encoding multiple receptive fields and enhancing the learnability of the network.
- The proposed attention pyramid module is designed to endure high-quality features from high-level layers to low-level layers using a stack of multi-scale filters.
- The multi-scale receptive fields utilized in the network capture coarse to fine-grained features that are effective for discrimination.
- The LC layers are adopted in the proposed Att-PyNet to capture dominant features from the local neighborhood using local filters.
- Att-PyNet is a lightweight network as it has fewer trainable parameters, which also reduces the computational complexity of the HGR models.

The Att-PyNet is validated on three standard datasets: MUGD [15], Triesch [16], and ASL FingerSpelling [17] in person-dependent and person-independent setup. The experimental results and analysis demonstrate that Att-PyNet achieves better performance compared to the state-of-the-art approaches (Fig. 2).

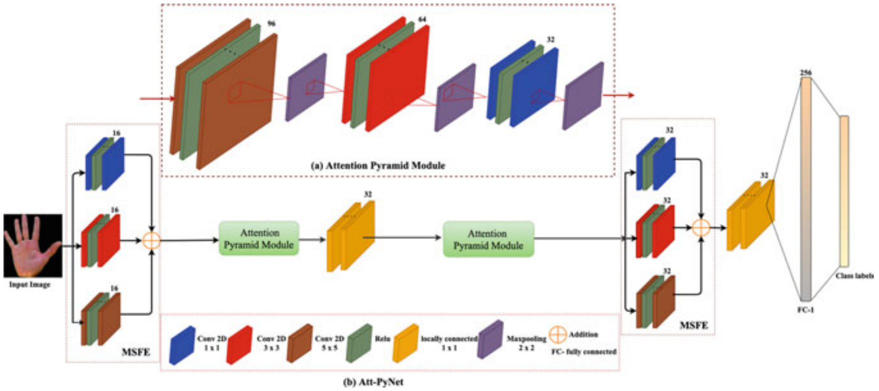


Fig. 2 Detailed workflow of the proposed network architecture for hand gesture recognition

2 Proposed Network

The existing literature indicates the strong capabilities of handcrafted feature-based approaches. However, these approaches are less robust and lack generalization. With the introduction of deep learning approaches, there is a significant improvement in the performance of the recognition system. Though the existing CNN-based approaches extract high-level features and are robust in scale variance, pyramid representation of features seems to achieve promising results as generic feature extractors [18]. Inspired by the literature, we designed Att-PyNet: an attention pyramidal feature network for efficient hand gesture recognition.

2.1 Attention Pyramidal Feature Network (Att-PyNet)

The proposed Att-PyNet comprises of three feature encoding streams: multi-scale feature extraction module (MSFE), attention pyramid module, and LC layer to learn the significant features for HGR as shown in Fig. 1.

2.1.1 Multi-scale Feature Extraction Module (MSFE)

The aim of designing MSFE is to extract the predominant features such as the fingertip, palm line, and shape which act as a differentiator to identify hand gestures and enhances the discriminability of the proposed Att-PyNet. The MSFE incorporated laterally connected multi-scale convolutional layers with a size of $1 \times 1, 3 \times 3$, and 5×5 to capture the multi-receptive fields. The resultant multi-receptive fields encoded macro- and micro-level edge information of hand gestures. Furthermore, laterally connected response feature maps are integrated to preserve the complementary features of multi-receptive fields. The complementary features allow proposed Att-PyNet to learn the fine edges extracted from different scales and boosts the performance of the network. Let $h(x, y)$ be an input image and $\varsigma_s^{d,f,f}$ represents the Conv function where d is the depth channels with kernel size $(f \times f)$, s is the stride. The output feature map of the MSFE (f_r) is calculated by Eq. 1

$$f_r^d = [\varsigma_1^{d,1,1}(h(x, y)) + \varsigma_1^{d,3,3}(h(x, y)) + \varsigma_1^{d,5,5}(h(x, y))] \tag{1}$$

2.1.2 Attention Pyramid Module

The attention pyramid module is introduced to maintain the feature quality from high-level layers to deep layers as features are diminished at lower layers due to repetitive operations. Therefore, the attention pyramid module enriches the lower layers features and plays a vital role in performance improvement of Att-PyNet The attention pyramid module consists of three stacks convolution layers with multi-scale receptive fields: $1 \times 1, 3 \times 3, 5 \times 5$, and depth channels 96, 64, 32, respectively. Retaining the same image size across the network requires high computing resources. To evade this, the attention pyramid module downsamples the feature maps by a scale of 2 at a proper interval and extracts meaningful information. The response of the attention pyramid module (p_r) is calculated using Eq. 2.

$$p_r = \varsigma_1^{32,5,5}\{\varsigma_1^{64,3,3}(\varsigma_1^{96,1,1}(f_r^d))\} \tag{2}$$

2.1.3 Locally Connected Layer

Typically, a locally connected (LC) layer increases the number of parameters in a network. Considering the size of the datasets, Att-PyNet employs only two locally connected layers. Unlike Conv layer, a locally connected layer uses a different set of filters at every pixel position in an image. Figure 3 represents the structures of a locally connected layer and standard convolution layer where the different colored lines in Fig. 3a denotes different set of filters and the same colored lines in Fig. 3b represent same set of filters. LC layer learns the local transformations of input feature when similar features present themselves differently in other locations. Thus, LC layer preserves the fine-grained features of the local region of the hand pose that plays a vital role in efficient hand gesture recognition. Mathematically, a locally connected layer can be represented by Eq. 3

$$y_l = \sum w_i x_i + b, \quad i \in \text{ receptive field} \tag{3}$$

where w stands for weights, x represents the input to the network, and b is the bias. The configuration of the proposed network is presented in Table 1. Thus, for a given

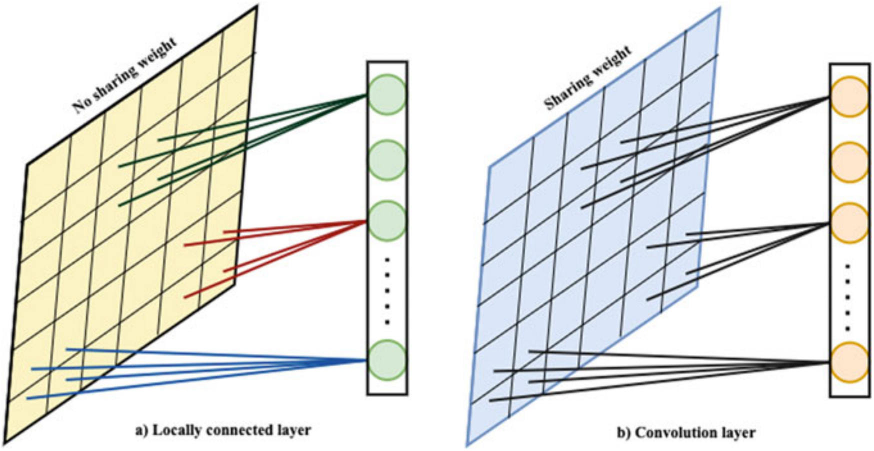


Fig. 3 Diagrammatic representation of **a** locally connected layer without weight sharing **b** convolution layer with shared weight

input image $h(x, y)$, the output of the proposed Att-PyNet can be calculated using Eqs. 4–5.

$$\text{Att}_{\text{out}} = \text{FC}^{256}[\xi_s^{32,1,1}(f_r^{32}(x_1))] \tag{4}$$

$$x_1 = P_r(\xi_1^{32,1,1}(P_r(f_r^{16}(h(x, y))))) \tag{5}$$

$\xi_s^{d,f,f}$ represents the Conv function where d is the depth channels with kernel size $(f \times f)$, s represents stride.

3 Experimental Results and Analysis

The proposed network is investigated on three benchmark datasets, in both person-dependent (PD) and person-independent (PI) setup. The qualitative and quantitative analyses clearly demonstrate the effectiveness of Att-PyNet as compared to the existing state-of-the-art approaches. In addition, the computational complexity of the proposed network is evaluated and discussed in this section.

3.1 Datasets

The experiment is conducted on three challenging datasets: Massey University gesture (MUGD) dataset, ASL fingerspelling (FS), and Jochen Triesch Static Hand Posture Database (Triesch). MUGD consists of 36 different postures performed by five different individuals (Part-1, Part-2, Part-3, Part-4, Part-5) captured in different

Table 1 Configuration of the proposed Att-PyNet

Layers		Filter	Output	Parameters
Input image		–	128 × 128	–
Conv 1.1		1 × 1	128 × 128 × 16	64
Conv 1.2		3 × 3	128 × 128 × 16	448
Conv 1.3		5 × 5	128 × 128 × 16	1216
Add		–	128 × 128 × 16	–
Block 1	Conv 2	1 × 1	128 × 128 × 96	1632
	Maxpooling	2 × 2	64 × 64 × 96	–
	Conv 3	3 × 3	64 × 64 × 64	55360
	Maxpooling	2 × 2	32 × 32 × 64	–
	Conv 4	5 × 5	32 × 32 × 32	51,232
	Maxpooling	2 × 2	16 × 16 × 32	–
Locally Connected Layer		1 × 1	16 × 16 × 32	270,336
Block 2	Conv 5	1 × 1	16 × 16 × 96	3168
	Maxpooling	2 × 2	8 × 8 × 96	–
	Conv 6	3 × 3	8 × 8 × 64	55,360
	Maxpooling	2 × 2	4 × 4 × 64	–
	Conv 7	5 × 5	4 × 4 × 32	51232
	Maxpooling	2 × 2	2 × 2 × 32	–
Conv 8		1 × 1	2 × 2 × 32	1056
Conv 9		3 × 3	2 × 2 × 32	9248
Conv 10		5 × 5	2 × 2 × 32	25,632
Add		–	2 × 2 × 32	–
Locally connected layer		1 × 1	2 × 2 × 32	4224
Fully connected layer-1		–	256	33,024
Fully connected layer-2		–	10	2570
Total				565,802

directions under illumination variations. FS dataset comprises of 24 static alphabets performed by five individuals against a complex background. The Triesch dataset contains 10 postures performed by 24 persons against light, dark, and complex background. In this paper, we have considered Triesch datasets with light and dark backgrounds to validate the effectiveness of the proposed Att-PyNet model under illumination variations.

3.2 Quantitative Analysis

The efficiency of the proposed Att-PyNet is measured in terms of accuracy and F1-score in PD and PI setup. The comparative analysis of the accuracy and F1-score of proposed Att-PyNet with the existing approaches: MobileNet [19], MobileNetV2 [20], ResNet50 [21], InceptionResNet [22], NasNetMobile [23], InceptionV3 [24], HandGes [25], DeepGestures [26] are presented in Tables 2 and 3. Att-PyNet achieves 2.73%, 19.73%, 3.56%, 4.62%, 10%, 51%, 70.89% gain in accuracy as compared to MobileNet, MobileNetV2, ResNet50, Incep-ResNet, NasNetMobile, InceptionV3, HandGes on MUGD (Part-1), 14.06%, 2.61%, 43.61%, 57.73% as compared to MobileNetV2, NasNetMobile, InceptionV3, HandGes on MUGD (Part-2) in PD setup. Further, the proposed network yields 5.28%, 6.39%, 18.47%, 25.14%, 63.19%, 64.03%, 89.86%, 0.14%; 0.7%, 10.28%, 14.45%, 33.89%, 65.98%, 66.12%, 91.12%, 3.62%; 40.83%, 35.97%, 18.02%, 15.97%, 34.72%, 58.89%, 78.61%, 6.94%; 6.1%, 36.59%, 19.76%, 16.34%, 20.73%, 15.12%, 54.63%, 65.85%; 7.5%, 38.0%, 13.75%, 22.25%, 11.50%, 14.0%, 59.0%, 55.50% gain on MUGD (Part-3), MUGD (Part-4), MUGD (Part-5), Triesch(L), Triesch(D) as compared to MobileNet, MobileNetV2, ResNet50, Incep-ResNet, NasNetMobile, InceptionV3, HandGes, DeepGestures, respectively. In ASL fingerspelling dataset, Att-PyNet gains 29.18%, 6.45%, 17.03%, 13.07%, 17.08%, 28.17%, 3.06% accuracy compared to MobileNetV2, ResNet50, Incep-ResNet, NasNetMobile, InceptionV3, HandGes, DeepGestures, respectively. Furthermore, in PI setup, Att-PyNet is investigated on MUGD dataset and found to gain 0.25%, 1.22%, 27.17%, 8.33%, 40.06%, 53.22% more accuracy compared to MobileNet, MobileNetV2, Incep-ResNet, NasNetMobile, InceptionV3, HandGes, respectively. Thus, it is evident from the quantitative analysis that the proposed Att-PyNet achieves better performance compared to existing approaches.

Table 2 Comparison of accuracy on MUGD, Triesch and ASL finger spelling datasets

Accuracy (%)									
SD								SI	
Networks	MUGD					Triesch		FS	MUGD
	Part-1	Part-2	Part-3	Part-4	Part-5	L	D		
MobileNet [19]	77.83	77.39	88.19	93.61	41.53	75.12	65.75	98.78	73.25
MobileNetV2 [20]	60.83	55.72	87.08	84.03	46.39	44.63	35.25	70.20	72.28
ResNet50 [21]	77.00	71.38	75.00	79.86	64.34	61.46	59.50	92.93	74.42
Incep-ResNet [22]	75.94	74.83	68.33	60.42	66.39	64.88	51.00	82.35	46.33
NasNetMobile [23]	70.56	67.17	30.28	28.33	47.64	60.49	61.75	86.31	65.17
InceptionV3 [24]	29.56	26.17	29.44	28.19	23.47	66.10	59.25	82.30	33.44
HandGes [25]	09.67	12.05	03.61	03.19	03.75	26.59	14.25	71.21	20.28
DeepGestures [26]	81.50	76.56	93.33	90.69	75.42	15.37	17.75	96.32	74.78
Att-PyNet	**80.56**	**69.78**	**93.47**	**94.31**	**82.36**	**81.22**	**73.25**	**99.38**	**73.50**

Table 3 Comparison of F1-score on MUGD, Triesch, and ASL fingerspelling datasets

F1-score (%)									
SD								SI	
Networks	MUGD					Triesch		FS	MUGD
	Part-1	Part-2	Part-3	Part-4	Part-5	L	D		
MobileNet [19]	76.95	76.31	87.98	93.47	38.61	75.17	64.91	98.78	71.63
MobileNetV2 [20]	59.49	53.07	87.03	83.86	44.91	43.44	35.79	70.26	70.53
ResNet50 [21]	76.53	69.70	74.83	79.61	63.11	60.84	58.49	92.94	73.03
Incep-ResNet [22]	73.96	72.52	73.00	64.68	68.32	64.75	50.31	82.29	45.02
NasNetMobile [23]	69.72	65.30	27.91	27.74	46.77	58.68	60.67	87.40	62.25
InceptionV3 [24]	33.66	26.65	31.88	29.91	24.29	65.78	58.09	82.33	31.33
HandGes [25]	05.49	08.70	01.13	00.71	00.10	20.18	10.40	70.20	12.93
DeepGestures [26]	86.54	74.77	92.87	90.51	74.62	11.53	13.14	96.32	71.15
Att-PyNet	**79.35**	**66.76**	**92.90**	**94.05**	**80.88**	**81.06**	**72.76**	**99.37**	**71.46**

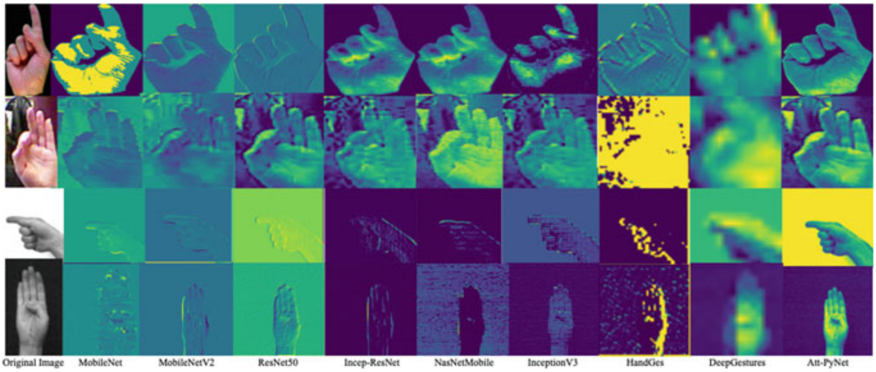


Fig. 4 Comparison of the feature response generated by the MobileNet, MobileNetV2, ResNet50, InceptionResNet, NasNetMobile, InceptionV3, HandGes, DeepGestures and Att-PyNet on MUGD, ASL Fingerspelling and Triesch (light and dark background) datasets

3.3 *Qualitative Analysis*

The qualitative representation of Att-PyNet with the existing networks over the three standard datasets: MUGD, ASL fingerspelling, and Triesch is demonstrated in Fig. 4. The comparison of the feature response of various gestures clearly shows that the proposed network preserves the most significant edges of hand postures that are responsible for describing the distinctive features between different hand postures for effective hand gesture recognition.

Table 4 Comparison of computational complexity of Att-PyNet with the existing networks

Networks	# parameters	# memory	# inference time
MobileNet [19]	5M	36.5 MB	8.26 s
MobileNetV2 [20]	3.5M	24.4 MB	15.26 s
Resnet-50 [21]	31M	208.4 MB	18.17 s
Incep-ResNet [22]	4M	40.5 MB	60.93 s
NasNetMobile [23]	4.8M	41.7 MB	10.55 s
InceptionV3 [24]	22M	179.3 MB	17.64 s
HandGes [25]	16 K	252 KB	1.57 s
DeepGestures [26]	11K	128KB	1.43 s
Att-PyNet	**565K**	**2.25 MB**	**5.11 s**

In the table, M represents millions, K represents thousands, MB represents megabytes, KB represents kilobytes, S represents seconds

3.4 Computational Complexity

Att-PyNet is a lightweight network with a minimum of 565K trainable parameters, which are less compared to MobileNet, MobileNetV2, ResNet50, InceptionResNet, NasNetMobile, InceptionV3, HandGes, and DeepGestures. The design structure of Att-PyNet allows the network to be implemented in a limited resource environment and is computationally efficient. Moreover, Att-PyNet allocates only 2.25 MB memory. The comparison of parameters, memory occupied, and the inference time of Att-PyNet with the existing approaches are presented in Table 4.

4 Conclusion

A lightweight end-to-end network named as Att-PyNet: An attention pyramidal feature network is proposed in this paper. The proposed Att-PyNet extracts rich semantic pyramidal features for hand gesture recognition by introducing three streams: MSFE, attention pyramid module, and LC layer. The MSFE is proposed to extract the local and global features from the multi-receptive fields. Therefore, the MSFE preserves the dominant edge variations of the hand region at the different scale level that enhances the discriminative ability of the network. Further, the attention pyramid module is introduced to carry forward the significant edge information from initial layers to downstream layers and maintain the feature quality in lower layers. The LC layer is embedded to encode the dominant features of the local neighborhood. Moreover, the proposed Att-PyNet is a memory and computationally efficient model as requires 565k parameters. The effectiveness of the proposed Att-PyNet is validated on three standard datasets. The experimental results show the effectiveness of the network compared to the existing approaches.

References

1. Pisharady PK, Saerbeck M (2015) Recent methods and databases in vision-based hand gesture recognition: a review. Comput Vis Image Underst 141:152–165
2. Muthukumar K, Poorani S, Gobhinath S (2017) Vision based hand gesture recognition for Indian sign languages using local binary patterns with support vector machine classifier. Adv Nat Appl Sci 11(6, SI):314–322
3. Misra A, Abe T, Deguchi K 2011 Hand gesture recognition using histogram of oriented gradients and partial least squares regression. In: MVA, pp 479–482
4. Bhaumik G, Verma M, Govil MC, Vipparthi SK (2020) EXTRA: an extended radial mean response pattern for hand gesture recognition. In: 2020 international conference on communication and signal processing (ICCSP), Chennai, India, 2020, pp 0640–0645. https://doi.org/10.1109/ICCSP48568.2020.9182207
5. Yamashita T, Watasue T (2014) Hand posture recognition based on bottom-up structured deep convolutional neural network with curriculum learning. In: 2014 IEEE international conference on image processing (ICIP), 2014. IEEE
6. Paul S, Bhattacharyya A, Mollah AF, Basu S, Nasipuri M (2020) Hand segmentation from complex background for gesture recognition. In: Emerging technology in modelling and graphics. Springer, Singapore, pp 775–782
7. Neethu PS, Suguna R, Sathish D (2020) An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks. Soft Comput 1–10
8. Zhang T, Lin H, Ju Z, Yang C (2020) Hand gesture recognition in complex background based on convolutional pose machine and fuzzy Gaussian mixture models. Int J Fuzzy Syst 1-12
9. Hu B, Wang J (2020) Deep learning based hand gesture recognition and UAV flight controls. Int J Autom Comput 17–29
10. Huang H, Chong Y, Nie C, Pan S (2019) Hand gesture recognition with skin detection and deep learning method. J Phys: Conf Ser 1213(2)
11. Nguyen XS, Brun L, Lezoray O, Bougleux S (2019) A neural network based on SPD manifold learning for skeleton-based hand gesture recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2019
12. Nuzzi C et al. (2019) Deep learning-based hand gesture recognition for collaborative robots. IEEE Instrum Meas Mag 22(2), 44–51
13. Wu XY (2019) A hand gesture recognition algorithm based on DC-CNN. Multimed Tools Appl 1–13
14. Pinto RF, Borges CDB, Almeida Antonio MA, Paula IC (2019) Static hand gesture recognition based on convolutional neural networks. J Electr Comput Eng
15. Barczak ALC, Reyes NH, Abastillas M, Piccio A, Susnjak T (2011) A new 2D static hand gesture colour image dataset for ASL gestures
16. Nicolas Pugeault RB ASL finger spelling dataset. http://personal.ee.surrey.ac.uk/Personal/N.Pugeault/index.php
17. Triesch J, von der Malsburg C (1996) Robust classification of hand postures against complex backgrounds. In: Proceedings of the second international conference on automatic face and gesture recognition. IEEE Computer Society Press, Killington, Vermont, USA, 14–16 Oct 1996, pp 170–175
18. Wang Y, Long M, Wang J, Yu PS (2017) Spatiotemporal pyramid network for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1529–1538
19. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
20. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520

21. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
22. Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence, 2017
23. Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8697–8710
24. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
25. Zhan F (2019) Hand gesture recognition with convolution neural networks. In: 2019 IEEE 20th international conference on information reuse and integration for data science (IRI). IEEE, pp 295–298
26. Mohanty A, Rambhatla SS, Sahay RR (2017) Deep gesture: static hand gesture recognition using CNN. In: Proceedings of international conference on computer vision and image processing. Springer, pp 449–461

Electroencephalogram-based Cognitive Load Classification During Mental Arithmetic Task

Aman Anand Rai and Mitul Kumar Ahirwal

Abstract Cognitive load also known as mental workload is one of the most important factors in the field of psychology to measure human performance during mental tasks. In this paper, classification of subjects has been done as a good and bad counter while performing mental arithmetic. Mathematical calculations evoke mental workload in the subjects. The concept of mental workload is used with electroencephalogram (EEG) signals for classification. EEG signal recordings are used to access the mental state of the subject by performing feature extraction. Several time domain features are extracted and classified through different classifiers. This initial study successfully classifies subjects in the said classes with 80% accuracy.

Keywords Cognitive load · EEG signals · Mental arithmetic task

1 Introduction

In general, the cognitive load (CL) is a quantitative measure of the amount of mental effort required to complete a task [1]. In many fields such as educational program evaluation, driver health review, analysis of the mental state of pilots and airline traffic dispatchers, CL assessment is required [2]. This will also be helpful to measure the mental fatigue or stress because of the continuous workload on employees. The CL must be optimal for any specific task to do correctly. If the CL exceeds its optimal limit, it can lead to mental fatigue. Measurement of CL is the first step in the process of maintaining optimal CL. It can be measured in different ways [3], and traditional measure is a manual questionnaire process, which is not a very accurate and reliable measure. In this several questions about the mental status are answered by a person and based on answers some rating is calculated. It can also be measured using physiological signals like EEG, electrocardiogram (ECG) signal, and eye movement tracking [4–7].

EEG signal is the measure of brain activity as electrical voltages changes due to synaptic excitations of the dendrites. EEG signal frequency spectrum varies from

A. A. Rai (✉) · M. K. Ahirwal
Maulana Azad National Institute of Technology, Bhopal, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_36

0.5 to 35 Hz. Delta frequency band (δ: less than 4 Hz), theta frequency band (θ: from 4 to 8 Hz), alpha frequency band (α: from 8 to 15 Hz), beta frequency band (β: from 15 to 32 Hz), and gamma frequency band (γ: greater than 32 Hz) are the five conventional frequency bands of EEG signal. It is the best method to measure CL because it reflects the actual status of the brain. EEG signal recording is also the cheapest option as compared to other brain imaging methods. But EEG has some limitations as it can get easily affected by artifacts or noises from eye movements, heartbeat, and breathing movements. These artifacts can be easily removed by signal processing methods [7, 8].

Several studies have been conducted for mental workload assessment using EEG signal [9]. Mental workload and CL detection through EEG signals were used in various fields like driver mental state measurement and educational tasks [10–14]. Multimedia education is in trend but multimedia resources with non-optimal CL can hamper the learning of students. Reduction in extraneous and intrinsic CL and increasing germane CL can increase the learning outcome of students [11]. Mental fatigue while driving can lead to road accidents and finding optimal CL can help in reducing accidents. In [12], a study was performed to find CL while driving and multitasking to find optimal CL. Mathematical cognitive task and visual search task are performed during lane changing in a driving simulator by subjects. Driver fatigue is classified by EEG signals in [13]. Eye movements also have been used for finding CL while driving task [14]. Following the same track of research, an attempt has been made to classify mental arithmetic task performance with the help of EEG signals.

In this study, an EEG signal dataset has been taken that includes EEG recordings during mental tasks [15]. This mental task is mental arithmetic for mathematic calculation (serial subtraction). Subjects performing this task were labeled as good and bad counters based on their performance, and the same is to be classified by EEG signal. The challenges associated with this study are the skewed dataset and feature selection. Under-sampling is used to overcome the skewness of the dataset and six features are extracted to make a comparison among them. The novelty of this study is the use of a unique dataset of its type, and this study is the first attempt in which classification of good and bad counters is performed through EEG.

The organization of the rest of the paper is as follows: In Sect. 2, methodology with four subparts that includes details about dataset, feature extraction, classification methods, and metrics for performance evaluation are given. Section 3 is reported with results and analysis. Conclusions are provided in Sect. 4.

2 Methodology

2.1 Dataset Description

The dataset consists of EEG signals recorded by a device having 23 electrodes; the device name is Neurocom Monopolar EEG 23-channel system [15]. The anterior frontal, frontal, central, parietal, occipital, and temporal are the locations over which electrodes are placed for data collection. For positioning the electrodes, the international 10/20 electrode placement scheme is followed. The sampling frequency is 500 Hz. The recordings include two segments which are artifact-free EEG signals of 3-min duration in the state of rest and 1-min duration while performing a mathematical task. Data of 36 participants were collected. Each subject (participant) is labeled as *good counter* (GC) or *bad counter* (BC) based on the subtraction task performed by them. In this study, out of 23 channels and 36 participants, only 19 channels and 20 participants' data are used for classification. EEG signal of 1-min duration is used for feature extraction which is corresponding to the mental calculation of subtraction. The subjects are classified into GC and BC classes based on these extracted features.

2.2 Feature Extraction

The following time-domain statistical features have been calculated.

Mean (F1): Mean value of all EEG samples, as given in Eq. (1):

$$\mu_X = \frac{1}{N}\sum_{n=1}^{N} X(n), \tag{1}$$

where μ_X is the mean of the EEG signal represented as X. N is the total number of samples.

Standard Deviation (F2): Standard deviation of EEG samples, as given in Eq. (2):

$$\sigma_X = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(X(n) - \mu_X)^2}, \tag{2}$$

where σ_X is the standard deviation and μ_X is the mean of the EEG signal.

First Difference Mean (F3): This is the mean of difference of two consecutive samples, as given in Eq. (3):

$$\delta_X = \frac{1}{N-1}\sum_{n=1}^{N-1} X(n+1) - X(n), \tag{3}$$

where δ_X is the mean of the first difference of EEG signal.

Second Difference (F4): This is the mean of difference of $(n+2)$th and nth sample, given in Eq. (4):

$$\gamma_X = \frac{1}{N-2}\sum_{n=1}^{N-2} X(n+2) - X(n), \tag{4}$$

where γ_X is the mean of the second difference of EEG signal.

Normalized First Difference (F5): This is the mean of the normalized first difference of EEG signal, as given in Eq. (5):

$$\overline{\delta}_X = \frac{1}{N-1}\sum_{n=1}^{N-1} \overline{X}(n+1) - \overline{X}(n), \tag{5}$$

where $\overline{\delta}_X$ is the mean of normalized first difference of EEG signal, and $\overline{X}(n)$ is the normalized signal and calculated as $\overline{X}(n) = \frac{X(n)-\mu_X}{\sigma_X}$.

Normalized Second Difference (F6): This is the mean of normalized second difference of EEG signal, as given in Eq. (6):

$$\overline{\gamma_X} = \frac{1}{N-2}\sum_{n=1}^{N-2} \overline{X}(n+2) - \overline{X}(n), \tag{6}$$

where $\overline{\gamma_X}$ is the mean of normalized second difference of EEG signal.

2.3 *Classification*

In this study, six classifiers for classifying the EEG signal into GC and BC classes have been implemented. These classifiers are discussed below.

Gaussian Naïve Bayes (GNB): It is a type of probabilistic classifier based on the Bayes' theorem. It works on the supervised learning approach. The features are assumed to be independent of each other and have equal weightage which makes the classifier naïve.

Decision Tree (DT): The decision tree uses decision rules derived from the data features for learning and predicting the class. It classifies the samples by decision-making based on some parameter on each node from root to leaf, with the leaf node predicting the class. In this model, the criteria for decision-making is the Gini index.

Stochastic Gradient Descent Classifier (SGD): In this classifier, a regularized linear classifier is implemented using a stochastic gradient descent algorithm. The loss function is used to train the model. The data samples are shuffled after each epoch.

Logistic Regression (LR): It is the simplest supervised learning classifier and is based on the sigmoid function. It is a linear classification model. Regularization is used to reduce overfitting. L2 regularization also known as ridge regularization is used in this classifier.

Support Vector Classifier (SVC): Support vector classifier separates the classes based on maximal margin hyperplane. This hyperplane is the decision boundary between two classes. The regularization parameter is set to 1 and the kernel function used is the radial basis function.

Artificial Neural Network (ANN): A simple multilayer perceptron model is used here, which consists of output layer, single hidden layer, and output layer. In the hidden layer five nodes are present. Rectified linear unit (ReLU) is used as an activation function along with Adam as an optimizer for the neural network. The neural networks with 5, 10, and 15 nodes in the hidden layer were tried for classification, out of which the neural network with five nodes gives better performance.

2.4 *Performance Measures*

For measuring the performance of classifiers, several parameters such as accuracy (ACC), precision (PR), recall (RC), and F1-score (F1S) are calculated. Five-fold cross-validation is applied. The dataset has an equal number of samples in each class. These measures are calculated by finding the value of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The formulas used for these performance measures are given in Eqs. (7)–(10):

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{7}$$

$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{8}$$

$$\mathrm{RC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{9}$$

$$F1S = 2 \times \left(\frac{\mathrm{RC} \times \mathrm{PR}}{\mathrm{RC} + \mathrm{PR}}\right). \tag{10}$$

3 Results and Analysis

The results obtained after performing the features extraction and classification are listed in Table 1. It is observed from the results that the normalized first difference with the decision tree classifier gives the best performance. The average of the performance measures is calculated with respect to classifiers as well as features. The average performance is shown as bar charts in Figs. 1 and 2. Figure 1 shows the comparative analysis of different classifiers based on the average of the performance measures, while Fig. 2 compares the different features. The average accuracy is best for the logistic regression classifier, as shown in Fig. 1. While in terms of features, the normalized first difference gives the best performance, as shown in Fig. 2.

Table 1 Performance measure for different features and classifiers

Features	Classifiers	ACC (%)	PR (%)	RC (%)	F1S (%)
Mean (F1)	GNB	35.00	16.70	20.00	18.00
	DT	25.00	13.30	20.00	16.00
	SGD	65.00	70.00	70.00	67.30
	LR	55.00	36.70	40.00	38.00
	SVC	45.00	26.70	50.00	34.70
	ANN	55.00	60.00	70.00	60.67
Standard deviation (F2)	GNB	60.00	56.70	70.00	58.70
	DT	50.00	40.00	50.00	43.30
	SGD	45.00	36.70	60.00	44.70
	LR	45.00	40.00	50.00	43.30
	SVC	45.00	40.00	40.00	36.70
	ANN	50.00	20.00	30.00	24.00
First difference (F3)	GNB	60.00	56.70	60.00	55.30
	DT	70.00	70.00	60.00	63.30
	SGD	50.00	30.00	60.00	40.00
	LR	75.00	70.00	90.00	78.00
	SVC	70.00	73.00	70.00	69.30
	ANN	50.00	50.00	100.00	66.00
Second difference (F4)	GNB	65.00	66.70	70.00	65.30
	DT	65.00	63.30	60.00	59.30
	SGD	50.00	30.00	60.00	40.00
	LR	75.00	70.00	90.00	78.00
	SVC	65.00	63.30	60.00	59.30

(continued)

Table 1 (continued)

Features	Classifiers	ACC (%)	PR (%)	RC (%)	F1S (%)
	ANN	50.00	50.00	100.00	66.00
Normalized first difference (F5)	GNB	65.00	66.70	70.00	65.30
	DT	**80.00**	**83.30**	**80.00**	**79.30**
	SGD	50.00	30.00	60.00	40.00
	LR	65.00	46.70	50.00	45.30
	SVC	70.00	73.30	70.00	69.30
	ANN	50.00	50.00	100.00	66.00
Normalized second difference (F6)	GNB	60.00	46.70	60.00	52.00
	DT	55.00	40.00	50.00	43.30
	SGD	50.00	30.00	60.00	40.00
	LR	75.00	70.00	90.00	78.00
	SVC	70.00	70.00	60.00	63.30
	ANN	50.00	50.00	100.00	66.00

The bold signifies the value with the best result. Out of all the comparative study the best overall value is highlighted using bold

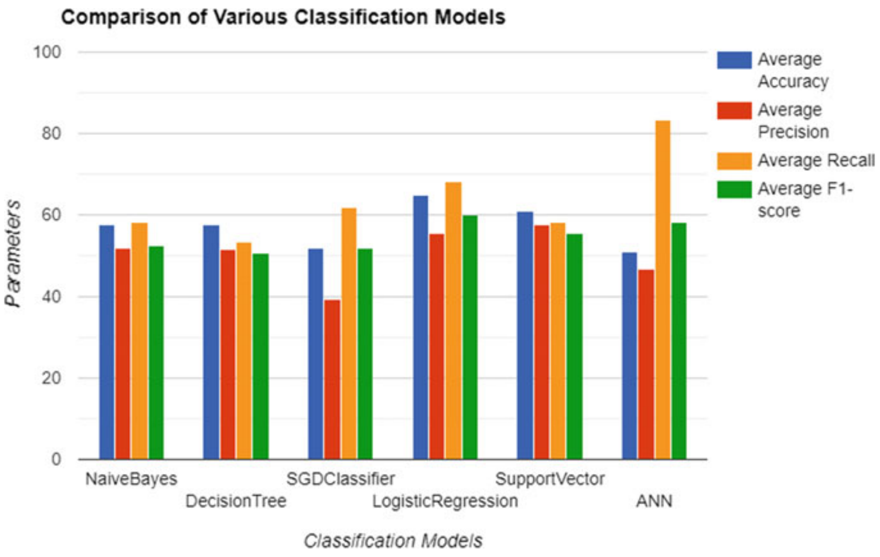


Fig. 1 Average accuracy of different classifiers

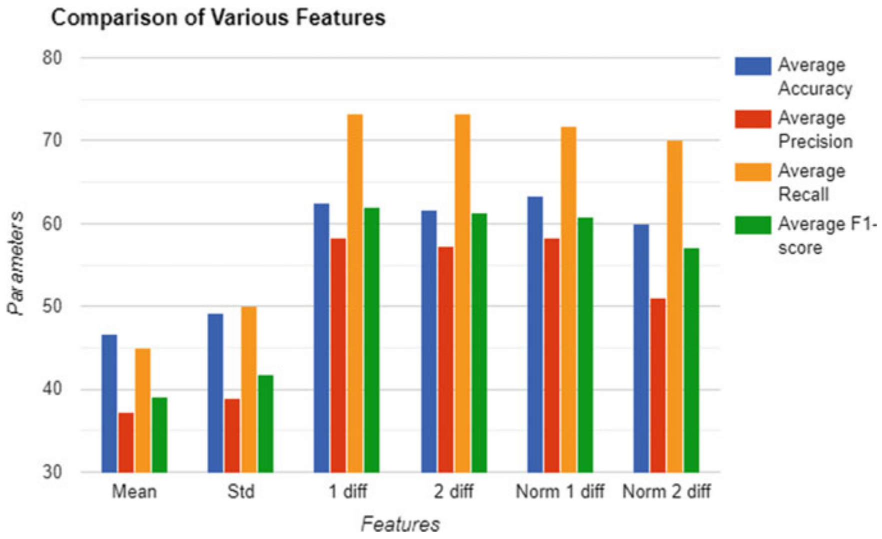


Fig. 2 Average accuracy of different features

4 Conclusions

In this study, the classification of subjects has been done as good counter and bad counter through their EEG signals recorded at the time of doing subtraction of numbers (a type of mathematical calculation) in their mind. For this, six statistical features are extracted for EEG signals. For classification six classifiers are used. Among all the features normalized first difference is found as the best feature with all classifiers with an average accuracy of 63.33% and average F1-score of 60.87%. The best classifier is logistic regression with an average accuracy of 65% and an average F1-score of 60.10% with all features. The highest accuracy of 80% is achieved by the combination of normalized first different and decision tree classifier. To further improve the accuracy and different performance measures, a complex feature extraction process can be used for signals in the time domain as well as in the frequency domain.

References

1. Gevins A (1997) High-resolution EEG mapping of cortical activation related to working memory: effects of task difficulty, type of processing, and practice. Cereb Cortex 7(4):374–385
2. Plechawska-Wójcik M, Tokovarov M, Kaczorowska M, Zapała D (2019) A three-class classification of cognitive workload based on EEG spectral data. Appl Sci 9(24):5340
3. Paas F, Tuovinen J, Tabbers H, Van Gerven PWM (2003) Cognitive load measurement as a means to advance cognitive load theory. Educ Psychol 38:63–72

4. Antonenko P, Paas F, Grabner R et al (2010) Using Electroencephalography to measure cognitive load. Educ Psychol Rev 22:425–438
5. Hossain D et al (2019) Cognitive load measurement using galvanic skin response for listening tasks. In: 2019 4th International conference on electrical information and communication technology (EICT), Khulna, Bangladesh, pp 1–4
6. Fowler A, Nesbitt K, Canossa A (2019) Identifying cognitive load in a computer game: an exploratory study of young children. In: 2019 IEEE conference on games (CoG), London, United Kingdom, pp 1–6
7. Ahirwal MK (2020) Analysis and identification of EEG features for mentalstress. In: 8th International conference on frontiers of intelligent computing: theory and applications (FICTA), NIT Karnataka, Surathkal, India, Jan 2020. (In Press)
8. Ahirwal MK, Kumar A, Singh GK (2013) EEG/ERP adaptive noise canceller design with controlled search space (CSS) approach in cuckoo and other optimization algorithms. IEEE/ACM Trans Comput Biol Bioinf 10(6):1491–1504
9. Ahirwal MK, Kumar A, Singh GK (2016) Study of ABC and PSO algorithms as optimized adaptive noise canceller for EEG/ERP. Int J Bio-Inspired Comput 8(3):170–183
10. Gavas R, Das R, Das P, Chatterjee D, Sinha A (2016) Inactive-state recognition from EEG signals and its application in cognitive load computation. In: 2016 IEEE International conference on systems, man, and cybernetics (SMC), Budapest, pp 003606–003611
11. Wang B, Wu F, Zhang S (2010) Reflections on the control of cognitive load in multimedia learning. In: 2010 Second international conference on multimedia and information technology, Kaifeng, pp 14–16
12. Putze F, Jarvis J, Schultz T (2010) Multimodal recognition of cognitive workload for multitasking in the car. In: 20th International conference on pattern recognition, Istanbul, pp 3748–3751
13. Hu J (2017) Automated detection of driver fatigue based on adaboost classifier with EEG signals. Front Comput Neurosci 11
14. Sega S, Iwasaki H, Hiraishi H, Mizoguchi F (2011) Applying qualitative reasoning to a driver's cognitive mental load. In: IEEE 10th International conference on cognitive informatics and cognitive computing (ICCI-CC'11), Banff, AB, pp 67–74
15. Zyma I, Tukaev S, Seleznov I, Kiyono K, Popov A, Chernykh M, Shpenkov O (2019) Electroencephalograms during mental arithmetic task performance. Data 4(1):14

Sniffing Android Malware Using Deep Learning

Anand Tirkey, Ramesh Kumar Mohapatra, and Lov Kumar

Abstract Android malware classification problem seems to have been solved with published AUC and F1 scores up to 0.99 or is it a facade, hiding an inherent problem? In this paper, we bring forward a novel method of recognising android malware using object-oriented software metrics-based dataset and deep learning. We realise that the real-world android malware is a minority class and its distribution according to 2017 Google's android security report, and Miller et al. [17] is estimated to be about 8–12%. The malware distribution in our dataset of 93K samples spanning over three years is around 10.9%. In this study, four data-sampling methods, six feature selection techniques and five deep learning networks with varying hidden layers are used over the imbalanced dataset of 93K samples. A total of 120 different machine-learned models are developed, and its classification potential is compared using area under ROC curve (AUC) metric. Finally, a machine-learned model obtained using upscale sampling (USD) data-sampling method applying significant set of metrics (SGM) feature selection technique over deep learning network with two hidden layers (DL2) yields a better AUC value of 0.893681.

Keywords Android malware detection · Machine learning · Object-oriented metrics

1 Introduction

Android OS market share in 2020 has been 85.4% and is expected to grow to 86.0% in 2021, according to International Data Corporation (IDC, USA). Currently, there are more than 2.5 billion active android devices and it is expected that another 1.5 billion units will be shipped by 2022 according to reports by Statista. Google's own 2019 statistics reveal that 42.1% of android devices running legacy OS versions are

A. Tirkey (✉) · R. K. Mohapatra
National Institute of Technology Rourkela, Rourkela, Odisha, India
e-mail: rkmohapatra@ieee.org

L. Kumar
BITS Pilani, Hyderabad Campus, Telangana, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_37

no longer supported. For the supported OS versions, Google publishes monthly OS updates which are then customised by the handset vendors for their respective android devices. Nohl et al. [18] argue that handset manufacturers act as a bottleneck for timely updating the android devices. Most device manufacturers prioritise software updates towards their flagship phones over the budget-friendly ones, which is a major part of the total sales. This disparity in receiving updates has caused the problem of OS fragmentation in android ecosystem. Google has introduced "Project Treble" that focuses on streamlining the process of customising software updates for handset manufacturers. Finally, it depends upon the mobile vendor whether they choose to push updates to all of its devices. Hence, zero-day exploits can affect millions of vulnerable devices because of android OS fragmentation. Google's built-in application permissions manage the allowed application permissions for the end-users meanwhile play protect regularly checks for malicious applications installed using Google Play Store. However, most of the end-users seldom understand the consequences of granting suspicious applications requesting critical permissions. Therefore, end-user privacy and security are always at risk either due to zero-day exploits or naively granting of critical permissions.

The rest of the paper is organised as follows. Section 2 discusses the related work. Section 3 describes the how the android application samples were selected for processing. Section 4 presents the methods used to obtain object-oriented metrics-based dataset and subsequently to create models using this dataset. Section 5 discusses the model's performance and interesting experimental observations. Section 6 addresses the research questions. Section 7 discusses possible threats to validity. Section 8 concludes the experiment and describes future work.

1.1 Objectives and Research Questions

The main objective of this study is to assess the importance of object-oriented software metrics, for android malware recognition. The following research questions (RQ) have been designed in order to identify, analyse and summarise the findings of the experiment proceedings:

- RQ1: Is there an interesting and significant distinction in the performances manifested by the four data-sampling techniques?
- RQ2: Is there a major difference in performance manifested by the six feature selection techniques?
- RQ3: How do the five classifiers fare in their discriminatory power as adjudged by AUC metrics? Do these classifiers vary greatly in their malware predictive performances?
- RQ4: Does increasing the hidden layers in deep learning, affect the malware recognition potential?

2 Related Work

Malware classification techniques are categorised into dynamic analysis and static analysis.

2.1 Static Analysis

In static analysis android application in reverse engineered and needful features are extracted such as permissions and API/system call sequence. Even though static analysis provides larger code coverage, malware programmers can employ encryption, obfuscation or polymorphic techniques in order to evade detection and analysis. We will be using static analysis in this experiment as it has less overhead costs as compared to dynamic analysis with comparable detection potential. Appice et al. [4] use permissions, api calls and network addresses as its features and classify using clustering and classification approach. Pektaş and Acarman [21] use opcode code sequence as its features and use deep neural networks for malware classification. Ding et al. [8] converts android apk into its bytecode file, and then it is transformed into a two-dimensional bytecode matrix which is used over CNN in order to recognise malware. Protsenko and Müller [22], Yermia et al. [27], Fan et al. [10], Tirkey et al. [26], Cen et al. [6], Arp et al. [5], Aafer et al. [1] are the malware detection solutions based on static analysis approach.

2.2 Dynamic Analysis

In dynamic analysis, android application is run in an android emulator, where the runtime behaviour of the applications is logged and is used as features. Alzaylaee et al. [3] use real devices to collect application behavioural information, and they have a dynamic stateful input generation approach based on deep learning. Millar et al. [16] use deep learning discriminative adversarial network (DAN) that classifies both obfuscated and unobfuscated, either benign or malware applications. They use three features datasets such as raw opcodes, permissions and API calls. Pektas et al. [20] uses API call graphs, which is transformed into a low-dimensional feature vector using graph embedding. Then deep neural network (DNN) is deployed to uncover malicious patterns. Surendran et al. [25] uses API calls, permissions and system calls as features and then uses tree augmented naive Bayes (TAN) to detect android malware.

The difficulty in detecting android malwares has driven the researchers towards exploring machine learning algorithms with promising malware discovery speed and automation. Shabtai et al. [24], Yermia et al. [27], Arp et al. [5], Dini et al. [9], Peiravian and Zhu [19], Rasthofer et al. [23] are the machine learning solutions that attempts to detect zero-day android malware. Deep learning is now emerging as promising technique for effective android malware detection as can be seen since Yuan et al. [28] proposed Droid-Sec which was one of the first to deploy deep learning over features extracted both from static analysis and dynamic analysis with 200 features that yielded an accuracy of 96.5% over 250 malware and 250 benign applications, Yuan et al. [29] proposed DroidDetector that analysed 20,000 benign and 1760 malware applications with an accuracy of 96.76%, Hou et al. [12] proposed Deep4MalDroid that extracts linux system calls using genymotion emulator, it yielded an accuracy of 93.68% over 1500 malware and 1500 benign applications and Hou et al. [13] proposed AutoDroid which extracts API calls and uses deep belief networks (DBN) yielding an accuracy of 95.98% based on experiments on 2500 malware and 2500 benign applications.

The aforementioned research works based on static and dynamic analysis fail to consider the ground truth, that android malware is a minority class in android ecosystem, and hence, this study takes into consideration this limitation while carrying out the experiments.

3 Experimental Dataset

We have obtained all the android application samples from Androzoo [2], archiving more than 12 million android application between 2010 and mid-2020. The Androzoo dataset tuple is of the format (sha256, sha1, md5, dex date, apk size, pkg name, vercode, vt detection, vt scan date, dex size, markets). Every android application within androzoo contains the metadata of VirusTotal service, especially the android package name (pkg name), package version (vercode), package date (dex date), VirusTotal scan date (vt scan date) and total number of antivirus companies that have tagged the android package (vt detection). Androzoo regularly updates its repository by crawling through the android marketplaces such as Google Play Store, Anzhi, AppChina amongst others. We choose this repository because of its colossal size and application availability for any period in time.

3.1 Benignware and Malware

Every android application in androzoo has a metadata "p" (vt detection metadata) that stores the number of antivirus that reported the particular application as a malware. For any benign application $p = 0$ and for a malware, we considered $p \geq 4$, considering Miller et al. [17] suggestions for a reliable ground truth. Around 13% of the Androzoo applications are grayware ($0 < p < 4$), which are left out from our datasets as these can either be a benign application or a malware, which can affect the performance of machine learning algorithms.

3.2 Selecting and Collecting Applications

Initially, we chose to collect and process a considerable number of applications for every month starting from January 2014. Due to time and space constraint required in processing the android applications, we limited the android application collection and processing till December 2016. Finally, a total of ninety-three thousand five hundred and forty-two android samples were collected and processed. In accordance with the 2017 Google Security report and Martina et al. [15], the android malware distribution can safely be assumed to be around 8–12%. In this study, the overall malware average is around 10%, whereas most of the month-wise android malware distribution is in between 8 and 12%. Miller et al. [17] state that the antivirus detections stabilise after approximately one year. Hence, we can affirm that the dataset is mature and represents the ground truth. The benign application and malware were collected from January 2014 to December 2016.

4 Research Methodology

The process of malware detection is shown in Fig. 1. The first step in creating a metrics-based dataset is acquiring considerable samples of android application packages (apks). Since these packages are basically compiled and archived Java class files, decompilation of apk into Java source code is necessary before analysing the extracted source code. The source code artefacts are then analysed, and Chidamber and Kemerer Java Metrics (CKJM) are obtained for each of the android applications. Then, aggregation methods are applied over these metrics which finally gives a tuple for each of the android applications for the metrics-based dataset. After the metrics-based dataset is built, it is then checked for sample imbalance between malware and benign classes. These imbalances are countered using four data-sampling techniques such as all sampling, random sampling, downscale sampling and upscale sampling. Since the datasets contain large number of features, six feature selection methods such as all metrics (AM), significant metrics (SGM), cross-correlation (CCR), principal component analysis (PCA), gini index (GINI) and information gain (INFOG) are deployed in order to obtain the smallest possible features that do not affect the performance of the dataset. These sampled datasets are then normalised using min-max normalisation, so that the values of the features are scaled between 0 and 1. Finally, five different deep neural networks are designed so as to build the models using the available datasets, the performance of each model is evaluated using AUC metric.

4.1 Object-Oriented Metrics Extraction and Aggregation

Initially, ninety-three thousand five hundred and forty-two android applications are collected using Androzoo [2] as described in Sect. 3. These android applications are

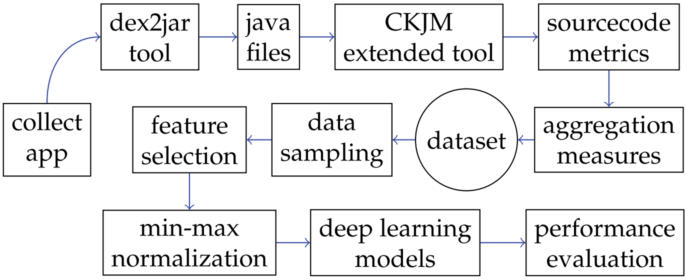


Fig. 1 Malware classification model

decompiled into Java source codes using dex2jar tool [11]. These Java artefacts are further processed in CKJM extended tool [14], and the respective eighteen object-oriented source code metrics are obtained as described in Table 1. Since an android application is composed of multiple classes and the CKJM extended tool provides metrics for every class, we obtain a (n × 18) feature vector for every application, where n represents the total number of classes the respective android application contains. Each aggregation method transforms a (n × 18) matrix into (1 × 18) vector. Since we have a total of ten aggregation methods as described in Table 2, we finally obtain ten different (1 × 18) vectors, which are merged together to form (1 × 180) feature vector. Hence, every android application is represented by (1 × 180) tuple in the metrics-based dataset as illustrated in Fig. 1.

Table 1 Object-oriented metrics

Measure of aggregation (MOA)	Depth of inheritance (DIT)	Efferent coupling (Ce)	Average method complexity (AMC)
LCOM3 afferent coupling (Ca)	Response for class (RFC)	Measure of functional abstraction (MFA)	
Coupling between objects (CBO)	Number of children in tree (NOC)	Inheritance coupling (IC)	Coupling between methods (CBM)
Lack of cohesion in methods (LCOM)	Number of public methods (NPM)	Cohesion among methods of class (CAM)	Weighted methods per class (WMC)
Lines of code (LOC)	Data access metric (DAM)		

Table 2 Aggregation measures

Min	Max	Mean
Standard deviation	Variance	Skew
First quartile	Second quartile	Third quartile
Kurtosis		

4.2 Data-Sampling Techniques

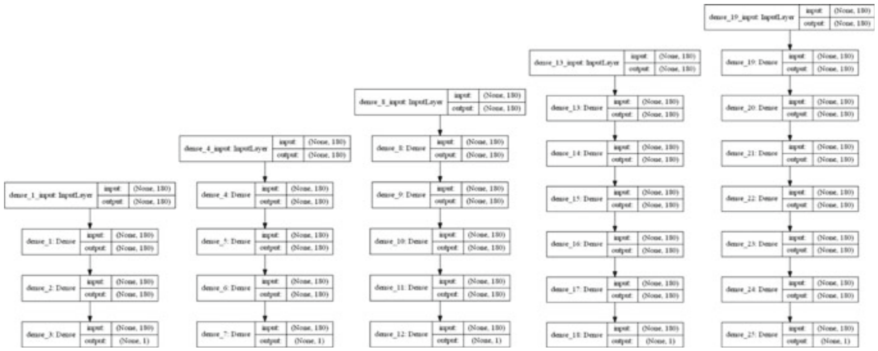
Out ninety-three thousand five hundred and forty-two android samples, it is observed that there are eighty-four thousand three hundred and fifty-seven benign applications and nine-thousand one-hundred and eighty-five malware applications. This disparity in benign and malware samples affects the real-world performance of machine-learned models, and hence, three data-sampling techniques, i.e., random sampling, upscale sampling and downscale sampling, are employed using SMOTE [7] analysis to overcome this class imbalance. Random sampling technique selects the samples from both the classes with equal probability, the dataset obtained using this technique is termed as random sampled dataset (RSD). Downscale sampling technique samples the greater-population class so as to equally balance with that of the lesser-population class, the dataset obtained using this technique is termed as downscale sampled dataset (DSD). Upscale sampling technique generates new samples so as to equally balance the lesser-population class with that of the greater-population class, the dataset obtained using this technique is termed as upscale sampled dataset (USD). Performance of classifiers using these sampling techniques is compared with unsampled original dataset (ORGD).

4.3 Feature Selection Techniques

Selecting minimum number of features meanwhile preventing classifier performance deterioration is desirables both in terms of reduced computation overhead and model complexity. Hence, six different feature selection techniques are used such as all metrics (AM), significant set of metrics (SGM), cross-correlation analysis (CCR), principal component analysis (PCA), gini index (GINI), information gain (INFOG). SGM is the set of source code metrics that are better predictors of malware, which is obtained by applying t-test over each source code metric and the metric having values less than 0.05 are selected. In CCR feature selection technique, when two features are closely correlated, then one feature is dropped and the features with low correlation in selected. PCA feature selection technique uses singular value decomposition of the data which projects the data into a lesser-dimensional space. GINI feature selection technique uses GINI ranking for choosing the best features. INFOG feature selection technique uses Shannon's entropy and chooses those features that maximise the overall information gain.

4.4 Classification Techniques

Five deep learning models with 2, 3, 4, 5 and 6 hidden layers, denoted as DL2, DL3, DL4, DL5 and DL6, respectively, with tenfold cross-validation, have been used in



(a) 2 hidden layers (DL2) (b) 3 hidden layers (DL3) (c) 4 hidden layers (DL4) (d) 5 hidden layers (DL5) (e) 6 hidden layers (DL6)

Fig. 2 Deep learning models

order to classify malware as illustrated in Fig. 2. Deep neural network has been used in keras with dropout parameter = 0.2. The hidden layers normally contain 180 input and output nodes when used with original dataset (ORGD), whereas upon using the dataset with reduced features, the number of nodes in the hidden layers are adjusted accordingly. The discriminatory power of these models is evaluated in order to choose the best classifier.

4.5 Performance Evaluation Metrics

Since we are dealing with a highly imbalanced dataset, choosing accuracy as the performance metric will not suffice as it fails to show the classifying potential of the minority classes. Whereas area under the ROC curve (AUC) performance metric is robust against class imbalance in a dataset. Hereafter, AUC metric is preferred over accuracy as the performance metric for all machine-learned models.

5 Experimental Results and Findings

Based on discussions in Sect. 4, we formulate and evaluate a null hypothesis H_0 : "*Machine-learned models composed using various data-sampling techniques, feature selection methods, classification algorithms and evaluated using AUC metric. Indicates no appreciable performance gain when compared against machine-learned models built using original dataset (ORGD)*".

5.1 *Analysing Data-Sampling Techniques*

In this experiment, three data-sampling techniques have been used as discussed in Sect. 4.2. Box-plots for ORGD, RSD, DSD and USD are shown in Fig. 3, their respective box-plot descriptive statistics, and p-values are shown in Table 3a, b and c, respectively. From Table 3a, it is observed that upscale sampled dataset (USD) yields a higher AUC value of 0.890398. Now, considering the four primary metrics-based dataset, a total of ${}^{4}C_{2} = 6$ unique pairs are possible. Analysing these six unique pairs at 0.05 significance level, we can reject null hypothesis H_0 if and only if the p-value is less than $0.05/6 = 0.0083$. In Table 3c, p-values less than 0.0083 are marked as "T"; otherwise, it is marked as "F". It can be deduced from Table 3c, that ORGD- and RSD-based datasets are similar amongst themselves and are significantly different from DSD- and USD-based datasets. Table 3a shows that USD-based datasets yield better AUC median values as compared to ORGD-, RSD- and DSD-datasets. Therefore, USD-based machine-learned models are expected to outperform the rest.

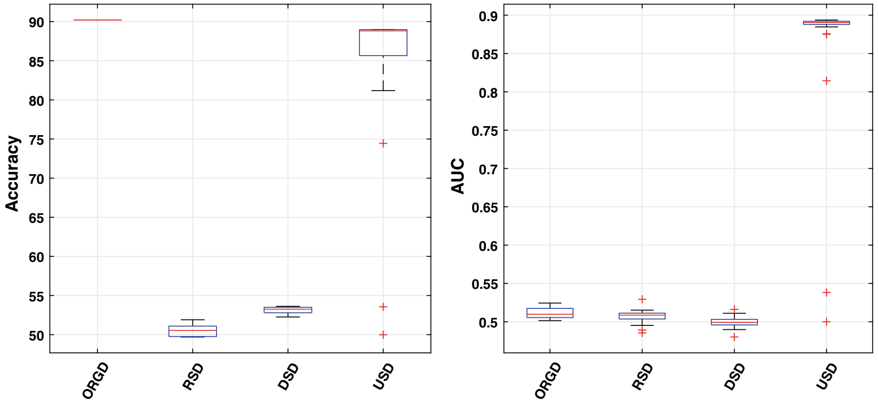


Fig. 3 Box-plots for data-sampling techniques

Table 3 Box-plot descriptive statistics and p-value for data-sampling techniques

	Min	Max	Mean	Median	Q1	Q3
ORGD	0.501471	0.524371	0.511288	0.50985	0.505442	0.517415
RSD	0.485473	0.5293	0.506899	0.508599	0.503565	0.511221
DSD	0.480115	0.51614	0.499436	0.49917	0.495848	0.50316
USD	0.5	0.893681	0.86235	**0.890398**	0.887995	0.892064

(a) AUC

	Min	Max	Mean	Median	Q1	Q3
ORGD	90.1972	90.1972	90.1972	**90.1972**	90.1972	90.1972
RSD	49.7006	51.9053	50.53892	50.54435	49.755	51.0887
DSD	52.2569	53.6239	53.1098	53.2499	52.7986	53.495
USD	49.9875	88.9875	84.83667	88.81875	85.65	88.9687

(b) Accuracy

	ORGD	RSD	DSD	USD
ORGD		F	T	T
RSD			T	T
DSD				T
USD				

(c) p-value

5.2 Analysing Feature Selection Methods

Six different feature selection methods have been used as discussed in Sect. 4.3. Box-plots for AM, SGM, CCR, PCA, GINI and INFOG are shown in Fig. 4, and their respective box-plot descriptive statistics and p-values are shown in Table 4a, b and c, respectively. Now, considering the six feature selection methods, a total of ${}^{6}C_2 = 15$ unique pairs are possible. Analysing these fifteen unique pairs at 0.05 significance level, we can reject null hypothesis H_0 if and only if the p-value is less than $0.05/15 = 0.0033$. In Table 4c, p-values less than 0.0033 is marked as "T" otherwise it is marked as "F". It can be deduced from Table 4c, that all datasets are similar amongst themselves and are not significantly different from each other. From Table 4a, it is observed that SGM-, PCA- and INFOG-based models perform at par with each other, yielding similar AUC values. Therefore, machine-learned models employing SGM, PCA and INFOG are expected to outperform the rest.

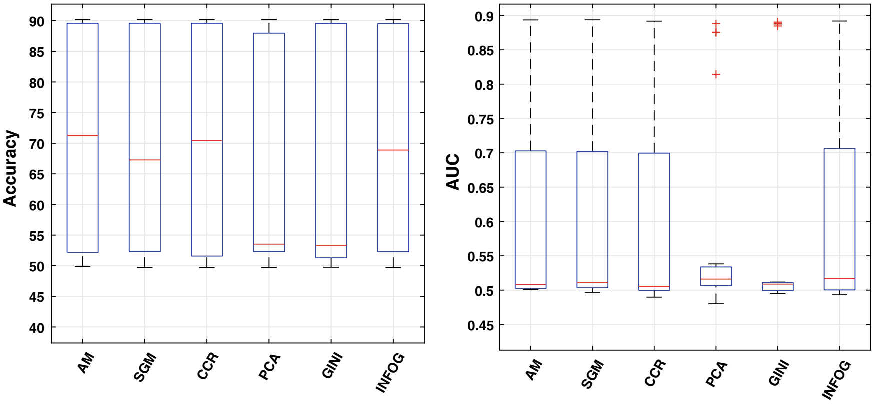


Fig. 4 Box-plots for feature selection techniques

Table 4 Box-plot descriptive statistics and p-value for feature selection techniques

	Min	Max	Mean	Median	Q1	Q3
AM	0.500845	0.893545	0.602697	0.508187	0.502569	0.702811
SGM	0.496917	0.893681	0.602972	**0.510707**	0.503424	0.702056
CCR	0.489793	0.891791	0.598791	0.505636	0.499708	0.699537
PCA	0.480115	0.887995	0.580083	**0.516086**	0.506663	0.533761
GINI	0.495298	0.890368	0.581032	0.50876	0.499116	0.511024
INFOG	0.493203	0.891936	0.604386	**0.517143**	0.500319	0.706324

(a) AUC

	Min	Max	Mean	Median	Q1	Q3
AM	49.8911	90.1972	70.77958	**71.27445**	52.19155	89.58925
SGM	49.7278	90.1972	70.35982	67.27365	52.32615	89.59235
CCR	49.7006	90.1972	70.39757	70.4617	51.57755	89.58925
PCA	49.7006	90.1972	67.37579	53.52565	52.3212	87.97045
GINI	49.755	90.1972	68.79169	53.34025	51.2929	89.57675
INFOG	49.7006	90.1972	70.31945	68.8881	52.29825	89.51735

(b) Accuracy

	AM	SGM	CCR	PCA	GINI	INFOG
AM		**F**	**F**	**F**	**F**	**F**
SGM			**F**	**F**	**F**	**F**
CCR				**F**	**F**	**F**
PCA					**F**	**F**
GINI						**F**
INFOG						

(c) p-value

5.3 Analysing Machine Learning Algorithms

Five different deep learning models have been used as discussed in Sect. 4.4. Box-plots for DL2, DL3, DL4, DL5 and DL6 used over all data-sampling techniques are shown in Fig. 5, their respective box-plot descriptive statistics are shown in Table 5a and b, respectively. Similarly, Box-plots for DL2, DL3, DL4, DL5 and DL6 were used only over upscaled sampling technique are shown in Fig. 6, and their respective box-plot descriptive statistics and p-values are shown in Table 6a, b and c, respectively. Upon observing and comparing Figs. 5 and 6, it is observed that upscaled sampling technique performs the best for deep learning algorithms. Now, considering the five deep learning models, a total of ${}^5C_2 = 10$ unique pairs are possible. Analysing these ten unique pairs at 0.05 significance level, we can reject null hypothesis H_0 if and only if the p-value is less than $0.05/10 = 0.005$. In Table VII(c), p-values less than 0.005 is marked as “T” otherwise it is marked as “F” . It can be deduced from Table 6c, that all datasets are similar amongst themselves and are not significantly different from each other. From Table 6a, it is observed that DL2- and DL6-based machine-learned models perform at par with each other, yielding similar AUC values. Therefore, machine-learned models employing either DL2, DL6 are expected to outperform the rest.

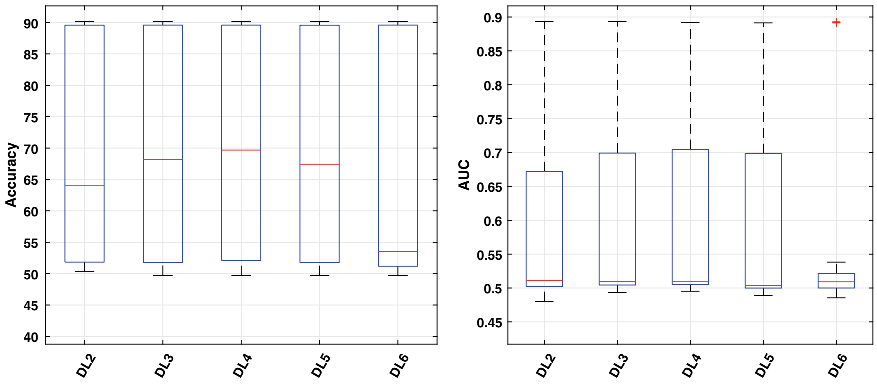


Fig. 5 Box-plots for classifiers (all sampling techniques)

Table 5 Box-plot descriptive statistics for classifiers (all sampling techniques)

	Min	Max	Mean	Median	Q1	Q3
DL2	0.480115	0.893681	0.599783	**0.510964**	0.502259	0.671869
DL3	0.493132	0.893665	0.602404	0.509771	0.504424	0.699148
DL4	0.495298	0.89222	0.60276	0.509215	0.505106	0.704476
DL5	0.489162	0.891302	0.599157	0.503473	0.499924	0.698465
DL6	0.485473	0.892785	0.570863	0.509142	0.5	0.52127

(a) AUC

	Min	Max	Mean	Median	Q1	Q3
DL2	50.2994	90.1972	69.99449	63.99205	51.834	89.58295
DL3	49.7278	90.1972	70.29124	68.21625	51.79245	89.58925
DL4	49.7006	90.1972	70.58106	**69.6838**	52.0811	89.58925
DL5	49.7006	90.1972	69.95559	67.33815	51.76455	89.57675
DL6	49.7006	90.1972	67.53086	53.52565	51.184	89.59235

(b) Accuracy

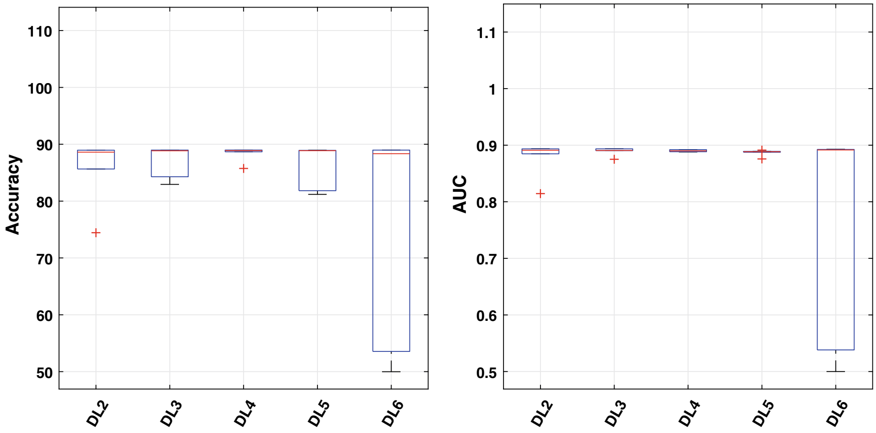


Fig. 6 Box-plots for classifiers (upscaled sampling only)

Table 6 Box-plot descriptive statistics for classifiers (upscaled sampling only)

	Min	Max	Mean	Median	Q1	Q3
DL2	0.814438	0.893681	0.87806	**0.891005**	0.884737	0.893497
DL3	0.875126	0.893665	0.889013	0.890686	0.890368	0.893545
DL4	0.887995	0.89222	0.890127	0.890024	0.888436	0.892064
DL5	0.875718	0.891302	0.88677	0.888473	0.887619	0.889034
DL6	0.5	0.892785	0.76778	**0.89159**	0.538222	0.892491

(a) AUC

	Min	Max	Mean	Median	Q1	Q3
DL2	74.4375	88.9687	85.875	88.61875	85.65	88.9563
DL3	82.9375	88.9813	87.14583	88.85	84.2812	88.975
DL4	85.7437	88.9813	88.35833	**88.8875**	88.6687	88.9813
DL5	81.1813	88.9563	86.44063	88.875	81.8312	88.925
DL6	49.9875	88.9875	76.36355	88.3375	53.5563	88.975

(b) Accuracy

	DL2	DL3	DL4	DL5	DL6
DL2		F	F	F	F
DL3			F	F	F
DL4				F	F
DL5					F
DL6					

(c) p-value

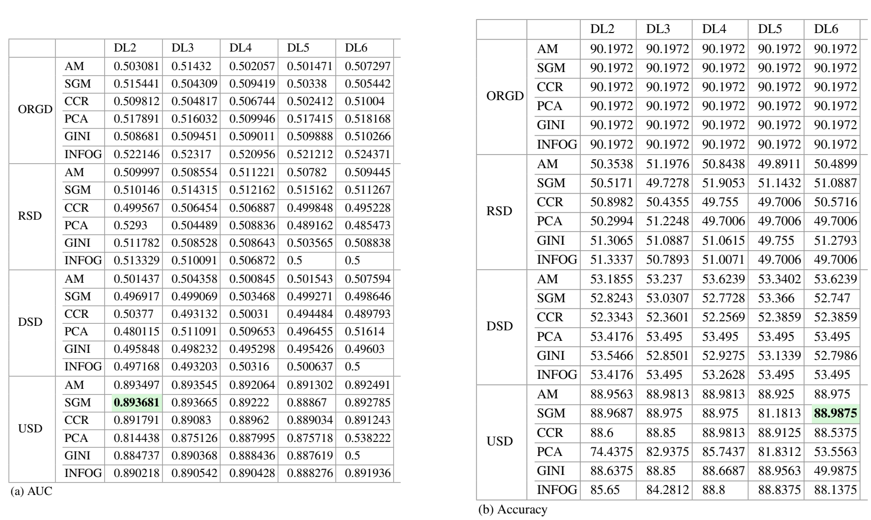
5.4 *Analysing Machine-Learned Models*

Upon analysing Sects. 5.1, 5.2 and 5.3, a total of one-hundred and twenty different machine-learned model's classifying power is evaluated using their respective AUC metric. Hence, it is expected that a upscaled sampling dataset (USD) applying any one of SGM, PCA, INFOG feature selection methods, modelled using either DL2 or DL6 classifier, will yield a better machine-learned model. This expectation is observed and confirmed from Table 7a and b, where USD-based dataset applying SGM feature selection technique over DL2 classifier, yields a better AUC of 0.893681.

6 Comparison of Results

Table 8 shows the comparison of the proposed work with previous works.

Table 7 Box-plot descriptive statistics for machine-learned models



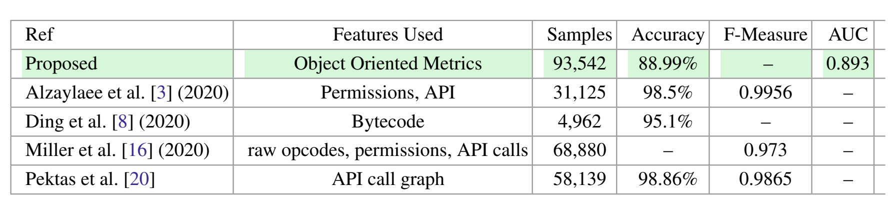
		DL2	DL3	DL4	DL5	DL6
ORGD	AM	0.503081	0.51432	0.502057	0.501471	0.507297
	SGM	0.515441	0.504309	0.509419	0.50338	0.505442
	CCR	0.509812	0.504817	0.506744	0.502412	0.51004
	PCA	0.517891	0.516032	0.509946	0.517415	0.518168
	GINI	0.508681	0.509451	0.509011	0.509888	0.510266
	INFOG	0.522146	0.52317	0.520956	0.521212	0.524371
RSD	AM	0.509997	0.508554	0.511221	0.50782	0.509445
	SGM	0.510146	0.514315	0.512162	0.515162	0.511267
	CCR	0.499567	0.506454	0.506887	0.499848	0.495228
	PCA	0.5293	0.504489	0.508836	0.489162	0.485473
	GINI	0.511782	0.508528	0.508643	0.503565	0.508838
	INFOG	0.513329	0.510091	0.506872	0.5	0.5
DSD	AM	0.501437	0.504358	0.500845	0.501543	0.507594
	SGM	0.496917	0.499069	0.503468	0.499271	0.498646
	CCR	0.50377	0.493132	0.50031	0.494484	0.489793
	PCA	0.480115	0.511091	0.509653	0.496455	0.51614
	GINI	0.495848	0.498232	0.495298	0.495426	0.49603
	INFOG	0.497168	0.493203	0.50316	0.500637	0.5
USD	AM	0.893497	0.893545	0.892064	0.891302	0.892491
	SGM	**0.893681**	0.893665	0.89222	0.88867	0.892785
	CCR	0.891791	0.89083	0.88962	0.889034	0.891243
	PCA	0.814438	0.875126	0.887995	0.875718	0.538222
	GINI	0.884737	0.890368	0.888436	0.887619	0.5
	INFOG	0.890218	0.890542	0.890428	0.888276	0.891936

(a) AUC

		DL2	DL3	DL4	DL5	DL6
ORGD	AM	90.1972	90.1972	90.1972	90.1972	90.1972
	SGM	90.1972	90.1972	90.1972	90.1972	90.1972
	CCR	90.1972	90.1972	90.1972	90.1972	90.1972
	PCA	90.1972	90.1972	90.1972	90.1972	90.1972
	GINI	90.1972	90.1972	90.1972	90.1972	90.1972
	INFOG	90.1972	90.1972	90.1972	90.1972	90.1972
RSD	AM	50.3538	51.1976	50.8438	49.8911	50.4899
	SGM	50.5171	49.7278	51.9053	51.1432	51.0887
	CCR	50.8982	50.4355	49.755	49.7006	50.5716
	PCA	50.2994	51.2248	49.7006	49.7006	49.7006
	GINI	51.3065	51.0887	51.0615	49.755	51.2793
	INFOG	51.3337	50.7893	51.0071	49.7006	49.7006
DSD	AM	53.1855	53.237	53.6239	53.3402	53.6239
	SGM	52.8243	53.0307	52.7728	53.366	52.747
	CCR	52.3343	52.3601	52.2569	52.3859	52.3859
	PCA	53.4176	53.495	53.495	53.495	53.495
	GINI	53.5466	52.8501	52.9275	53.1339	52.7986
	INFOG	53.4176	53.495	53.2628	53.495	53.495
USD	AM	88.9563	88.9813	88.9813	88.925	88.975
	SGM	88.9687	88.975	88.975	81.1813	**88.9875**
	CCR	88.6	88.85	88.9813	88.9125	88.5375
	PCA	74.4375	82.9375	85.7437	81.8312	53.5563
	GINI	88.6375	88.85	88.6687	88.9563	49.9875
	INFOG	85.65	84.2812	88.8	88.8375	88.1375

(b) Accuracy

Table 8 Comparison



Ref	Features Used	Samples	Accuracy	F-Measure	AUC
Proposed	Object Oriented Metrics	93,542	88.99%	–	0.893
Alzaylaee et al. [3] (2020)	Permissions, API	31,125	98.5%	0.9956	–
Ding et al. [8] (2020)	Bytecode	4,962	95.1%	–	–
Miller et al. [16] (2020)	raw opcodes, permissions, API calls	68,880	–	0.973	–
Pektas et al. [20]	API call graph	58,139	98.86%	0.9865	–

RQ1: Is there an interesting and significant distinction in the performances manifested by the four data-sampling techniques?

Considering the null hypothesis H_0 and analysing Sect. 5.1, it is observed that out of a total of six pairs, five pairs reject the null hypothesis and are marked by "T" in Table 3c. In case, a null hypothesis is rejected, it implies that the distinction identified between samples isn't by chance and the observation is statistically significant. Upon observing Table 3a, it is evident that the machine-learned models based on USD yield better AUC as compared to others. Therefore, USD-based models are interesting and manifest significant malware prediction potential as compared against ORGD, RSD and DSD.

RQ2: Is there a major difference in performance manifested by the six feature selection techniques?

Considering the null hypothesis H0 and analysing Sect. 5.2, it is observed that all of the fifteen pairs, accept the null hypothesis and is marked by "F" in Table 4c. In case, a null hypothesis is accepted, it implies that the feature selection techniques are similar amongst themselves and are not statistically significant. Upon observing Table 4a, it is evident that out of the six feature selection methods, SGM, PCA and INFOG perform at par with each other with almost similar AUC metric. Hence, SGM-, PCA- and INFOG-based models outperform AM-, CCR- and GINI-based models.

RQ3: How do the five classifiers fare in their discriminatory power as adjudged by AUC metrics ? Do these classifiers vary greatly in their malware predictive performances?

Considering the null hypothesis H_0 and analysing Sect. 5.3, it is observed that all of the ten pairs, accept the null hypothesis and is marked by "F" in Table 6c. In case, a null hypothesis is accepted, it implies that the classification techniques are similar amongst themselves and are not statistically significant. Upon observing Table 6a, it is evident that out of the five classification methods, DL2 and DL6 perform at par with each other with almost similar AUC metric. Hence, models applying DL2 and DL6 marginally outperform DL3-, DL4- and DL5-based models.

RQ4: Does increasing the hidden layers in deep learning, affect the malware recognition potential?

Upon observing Table 6a and Fig. 6, it is evident that the malware recognition potential varies meagerly. Hence, increasing the number of hidden layers does not contribute in higher predictive performance of machine-learned models.

7 Threats to Validity

This work depends upon the positive malware cases as declared by VirusTotal service. Hence, any false-positive case will be difficult to assess and handle if it isn't properly addressed by VirusTotal. Another possible threat is that, according to Miller et al. [17], it takes about a year for an application metadata to stabilise with VirusTotal and consequently with Androzoo. Hence, updating the machine-learned model for a recently discovered malware becomes difficult. This may render the machine-learned model useless against new strain or family of malware.

8 Conclusion and Future Work

In order to build a dataset that replicates the ground truth, it was established using 2017 Google's android security report and Miller et al. [17] that android malware is a minority class and it can be safely assumed that the distribution of android malware is in between 8 and 12%. For this experiment, we considered an overall malware distribution of 10–11% in our dataset spanning three years, whereas most of month-wise data have malware distribution between 8 and 12%. Data from January 2014 to December 2016 was collected from Androzoo [2] and is decompiled into its respective Java source code using dex2jar tool. These java artefacts are then processed further using CKJM extended tool [11] in order to extract object-oriented software metrics. These software metrics are then aggregated using ten aggregation measures such as min, max, mean, Q1, Q2, Q3, standard deviation, variance, skew and kurtosis which is merged together to form a (1 × 180) dimensional vector, that forms the tuple of our metrics-based dataset. Then various data-sampling techniques are employed to mitigate the class imbalance present in the original dataset such as random sampling, downscale sampling and upscale sampling. Box plot descriptive statistics for data-sampling techniques points out that USD outperforms other data-sampling methods. After new datasets are sampled using the original dataset, it is desirable to reduce the total number of features in order to reduce training time, prevent model overfitting, lower the model complexity and lessen computational overheads. A total of six different feature selection techniques are used such as all metrics (AM), significant metrics (SGM), cross-correlation analysis (CCR), principal component analysis (PCA), gini index (GINI) and information gain (INFOG). Box-plot descriptive statistics for feature selection techniques point that, SGM, PCA and INFOG performs at par with each other. Subsequently, the feature-reduced dataset is then normalised using min-max normalisation, that transforms all the feature values in between 0 and 1. This normalised dataset is then used over different deep learning models, and the respective model's performance evaluation is done using AUC metric. A total of five deep learning models are used such as DL2 (2 hidden layers), DL3 (3 hidden layers), DL4 (4 hidden layers), DL5 (5 hidden layers) and DL6 (6 hidden layers). Box-plot descriptive statistics for classifiers point out that DL2 and DL6 perform at par with each other as compared to other techniques. Finally, a model using USD sampled dataset applying SGM feature selection technique over DL2 performs better yielding an AUC of 0.893681.

This work is a binary classification problem, where all families of malware are merged into one class. In future, malware classification into different malware families will be considered.

References

1. Aafer Y, Du W, Yin H (2013) DroidAPIMiner: mining api-level features for robust malware detection in android. In: International conference on security and privacy in communication systems. Springer, pp 86–103
2. Allix K, Bissyandé TF, Klein J, Le Traon Y (2016) Androzoo: collecting millions of android apps for the research community. In: 2016 IEEE/ACM 13th working conference on mining software repositories (MSR). IEEE, pp 468–471
3. Alzaylaee MK, Yerima SY, Sezer S (2020) DL-Droid: deep learning based android malware detection using real devices. Comput Secur 89:101663
4. Appice A, Andresini G, Malerba D (2020) Clustering-aided multi-view classification: a case study on android malware detection. J Intell Inf Syst 1–26
5. Arp D, Spreitzenbarth M, Hubner M, Gascon H, Rieck K, Siemens C (2014) Drebin: effective and explainable detection of android malware in your pocket. NDSS 14:23–26
6. Cen L, Gates CS, Si L, Li N (2014) A probabilistic discriminative model for android malware detection with decompiled source code. IEEE Trans Depend Secure Comput 12(4):400–412

7. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
8. Ding Y, Zhang X, Hu J, Xu W (2020) Android malware detection method based on bytecode image. J Ambient Intell Hum Comput 1–10
9. Dini G, Martinelli F, Saracino A, Sgandurra D (2012) MADAM: a multi-level anomaly detector for android malware. In: International conference on mathematical methods, models, and architectures for computer network security. Springer, pp 240–253
10. Fan M, Liu J, Wang W, Li H, Tian Z, Liu T (2017) DAPASA: detecting android piggybacked apps through sensitive subgraph analysis. IEEE Trans Inf Forens Secur 12(8):1772–1785
11. Fora PO (2014) Beginners guide to reverse engineering android apps. In: RSA conference, pp 21–22
12. Hou S, Saas A, Chen L, Ye Y (2016) Deep4MalDroid: a deep learning framework for android malware detection based on linux kernel system call graphs. In: 2016 IEEE/WIC/ACM international conference on web intelligence workshops (WIW). IEEE, pp 104–111
13. Hou S, Saas A, Chen L, Ye Y, Bourlai T (2017) Deep neural networks for automatic android malware detection. In: Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, pp 803–810
14. Jureczko M, Spinellis D (2010) Using object-oriented design metrics to predict software defects. Models and methods of system dependability. Oficyna Wydawnicza Politechniki Wrocławskiej, pp 69–81
15. Lindorfer M, Volanis S, Sisto A, Neugschwandtner M, Athanasopoulos E, Maggi F, Platzer C, Zanero S, Ioannidis S (2014) Andradar: fast discovery of android applications in alternative markets. In: International conference on detection of intrusions and malware, and vulnerability assessment. Springer, pp 51–71
16. Millar S, McLaughlin N, Martinez del Rincon J, Miller P, Zhao Z (2020) DANdroid: a multi-view discriminative adversarial network for obfuscated android malware detection. In: Proceedings of the tenth ACM conference on data and application security and privacy, pp 353–364
17. Miller B, Kantchelian A, Tschantz MC, Afroz S, Bachwani R, Faizullabhoy R, Huang L, Shankar V, Wu T, Yiu G, et al (2016) Reviewer integration and performance measurement for malware detection. In: International conference on detection of intrusions and malware, and vulnerability assessment. Springer, pp 122–141
18. Nohl K, Lell K (2018) Mind the gap: uncovering the android patch gap through binary-only patch level analysis. In: HITB security conference
19. Peiravian N, Zhu X (2013) Machine learning for android malware detection using permission and API calls. In: 2013 IEEE 25th international conference on tools with artificial intelligence. IEEE, pp 300–305
20. Pektaş A, Acarman T (2020) Deep learning for effective android malware detection using API call graph embeddings. Soft Comput 24(2):1027–1043
21. Pektaş A, Acarman T (2020) Learning to detect android malware via opcode sequences. Neurocomputing 396:599–608
22. Protsenko M, Müller T (2014) Android malware detection based on software complexity metrics. In: International conference on trust, privacy and security in digital business. Springer, pp 24–35
23. Rasthofer S, Arzt S, Bodden E (2014) A machine-learning approach for classifying and categorizing android sources and sinks. In: NDSS, vol 14. Citeseer, p 1125
24. Shabtai A, Kanonov U, Elovici Y, Glezer C, Weiss Y (2012) “andromalyâŁž: a behavioral malware detection framework for android devices. J Intell Inf Syst 38(1):161–190
25. Surendran R, Thomas T, Emmanuel S (2020) A tan based hybrid model for android malware detection. J Inf Secur Appl 54:102483
26. Tirkey A, Mohapatra RK, Kumar L (2019) Anatomizing android malwares. In: 2019 26th Asia-Pacific software engineering conference (APSEC). IEEE, pp 450–457
27. Yerima SY, Sezer S, Muttik I (2015) High accuracy android malware detection using ensemble learning. IET Inf Secur 9(6):313–320

28. Yuan Z, Lu Y, Wang Z, Xue Y (2014) Droid-sec: deep learning in android malware detection. In: Proceedings of the 2014 ACM conference on SIGCOMM, pp 371–372
29. Yuan Z, Lu Y, Xue Y (2016) Droiddetector: android malware characterization and detection using deep learning. Tsinghua Sci Technol 21(1):114–123

Vertical Fusion: A Distributed Learning Approach for Vertically Partitioned Data

Anirudh Kasturi, Ameya Salankar, Sharath S. Chandra, and Chittaranjan Hota

Abstract Over the last couple of years, distributed learning has gained tremendous focus primarily because of the exponential rise in the computing capabilities of handheld devices and their ability to generate enormous data. However, the majority of the existing distributed learning algorithms focus on horizontally partitioned data, where each client has a subset of the entire data containing all the features and their corresponding labels. In this paper, we present a communication-efficient approach to learn from data that is partitioned vertically, i.e., each client holds a subset of the dataset's attributes. The proposed algorithm tries to find the distributions of each feature residing on the client, and the server trains a global model on the sample data generated from these distributions. Our experiments on multiple datasets show that the proposed approach can achieve accuracies similar to that of a centralized learning setup without transmitting the actual data to the server and using only one communication round.

Keywords Fusion learning · Federated learning · Vertically partitioned data · Distributed learning

1 Introduction

Machine learning has played a significant role in numerous applications because of its ability to derive meaningful information and observations from vast data spanning

A. Kasturi (✉) · A. Salankar · S. S. Chandra · C. Hota
BITS Pilani, Hyderabad Campus, Hyderabad, India
e-mail: p20170403@hyderabad.bits-pilani.ac.in

A. Salankar
e-mail: f20170182@hyderabad.bits-pilani.ac.in

S. S. Chandra
e-mail: f20170219@hyderabad.bits-pilani.ac.in

C. Hota
e-mail: hota@hyderabad.bits-pilani.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_38

across different domains such as traffic, health care and recommendation systems. A machine learning model is typically trained over a distributed data group that can be accessed by one or more data providers. While distributed algorithms have been developed to speed up the training, the training data itself is still collected and centrally stored in a data center. These algorithms can briefly be divided into two groups, $horizontal$ and $vertical$, based on how the data is partitioned. In a horizontally partitioned scenario, data is split horizontally and distributed across multiple nodes such that all nodes share the same set of features. Majority of the currently available distributed learning algorithms have been proposed on horizontally partitioned data [2, 7, 12]. In contrast to horizontal partitioning, communication-efficient learning strategies designed for vertically partitioned are much lower. The vertically partitioned data paradigm is especially difficult for distributed data processing in an IoT environment where feature data is spread across multiple clients. This data needs to be combined to gain some meaningful insights. With the current increase in smart devices, especially in IoT, research in distributed learning algorithms for vertically partitioned data has picked up pace. Some of such uses cases include predicting the final quality of a product in the early stages of production using different process parameters and measurements at different production stages [10], prediction of traffic flow from various sensor devices located at different places [15], diagnosing patients health from different data sources such as fitness tracker devices, diet monitoring apps, and blood sample data. All the above-mentioned use cases require data to be transferred to a central server for processing. This leads to high communication costs and serious privacy concerns, especially when healthcare data is involved. The relationship between accuracies and communications costs is still a concern, and open research questions still exist. The key issue is how communication-efficient algorithms can be built while retaining adequate accuracies. To summarize, any vertically partitioned algorithm needs to address the following two problems:

- Reduce the amount of data that is transferred to a central server from the clients and achieve similar accuracies in comparison with a centralized framework.
- Preserve the privacy of the user while building a distributed model.

To address the above two challenges, we propose a novel $verticalfusion$ algorithm that computes the distributions of each feature located at the clients. A central server generates sample points from these parameters and trains a global model on these sample points. We show that our approach achieves similar accuracies to that of a centralized framework and, at the same time, reduces the amount of data transmitted over the network as only the distribution parameters are passed to the server. This, in turn, addresses the privacy concerns of sensitive data at the clients as the actual data is not transferred.

The main challenge with distributed learning in a vertically partitioned scenario is that each client will not have enough information to predict the target variable. They would only contain only a subset of all the features. In some of the earlier works, although not communication efficient, they try to process the data locally at the clients, and the new values are merged with the help of a central coordinator. The works presented in [13, 18] have used support vector machines (SVMs) where the kernel matrices are communicated and have shown that the global kernel matrix is separable. In [18], each client calculates its local kernel matrix, which is transmitted to the central server. The server trains a centralized SVM on the full kernel matrix, and this approach can preserve privacy since the original data is not revealed. Another method proposed by [6] uses random projections. This helps in reducing the feature set, which is transmitted to the clients.

Authors in [1] have tried to combine SVM for vertically partitioned data with alternating direction method of multipliers (ADMM) framework. They have introduced an additional variable to solve the ADMM and have shown that the computation of this variable can be split at the component level. This means that each part of the variable can be calculated independently of each other. Some of the other works involved running a logistic regression model in which one node contains the features, whereas the other has the labels [16]. A linear regression model was built on vertically partitioned data using a hybrid multi-party computation [5]. Recent works by [14] proposed a privacy-preserving machine learning approach using secret sharing, garbled circuits and oblivious transfer.

More recent approaches involved merging of local predictions and have proved to be more communication efficient as only the weight vector is transmitted to the server. One such strategy was to train the primal SVM problem using stochastic gradient descent (SGD), where the global model is learned by combining the predictions of individual clients. Another more interesting work in this direction for horizontally partitioned data was presented in [9], where each client trains a model locally, and the model weights are averaged with the help of a central server. Authors in [11, 17] have tried to leverage this idea and implement a variation of this for the vertically partitioned data scenario.

2 Preliminaries

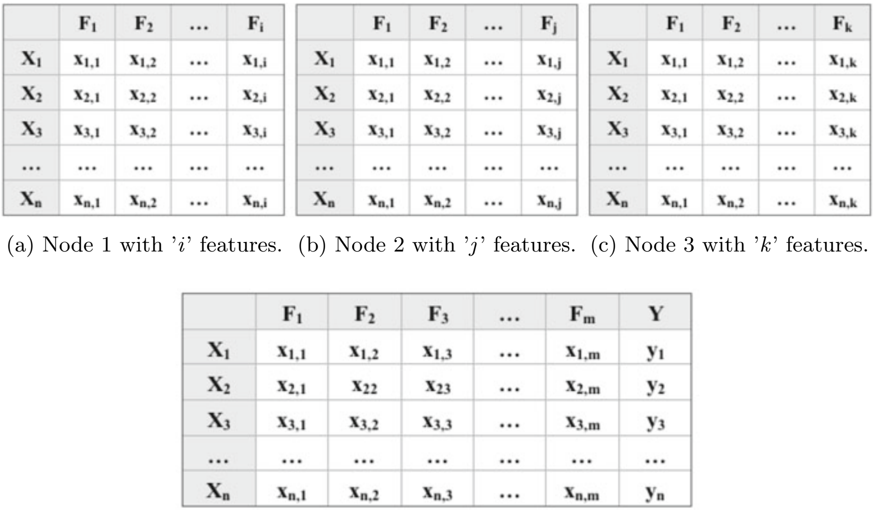
In this section, we provide a brief overview on fusion learning which forms the building block for our proposed approach.

2.1 Fusion Learning

In our earlier work, we have proposed the fusion learning algorithm [8] that tries to learn a global model by sending distribution parameters along with model parameters to a central server. The major advantage of such an approach over other approaches such as federated learning [9] is that the number of communication rounds required to learn the global model is reduced to just one, thereby greatly reducing the overall communication overhead involved when compared with most of the other distributed learning approaches. In this technique, all clients train a local model on the locally available data and obtain the model parameters, θ. Along with this, they also compute the distributions of each feature ψ. This is done by first calculating the probability values (p-values) using K-S [3] test on the feature data against each distribution in the initial set, and the maximum out of these is chosen since it denotes the best fit. Once the distribution is chosen for each feature, the parameters corresponding to these features are sent to a central server along with the model parameters. The central node generates sample points from the inputs received from each client, and the labels are computed using the model weights, thus forming a global dataset. Once these points are generated, a new global machine learning model is trained on this data. The global parameters are now sent back to all the clients. This approach brings down the number of communication rounds to just one as compared to the other existing distributed learning algorithms.

3 Proposed Approach

In our current work, we propose a *vertical fusion* algorithm that can address the challenge of learning from vertically partitioned data using our previously proposed fusion learning algorithm [8]. In a vertically partitioned distributed environment, features of the dataset are split across different clients. Consider a scenario as shown in Fig. 1, where the total number of features (m) is split between three nodes. Node 1 has i features, node 2 has j features, and node 3 has k features where $m = i + j + k$. The central server tries to learn a global model from the clients who, instead of transmitting their data, send only the distribution parameters.



	F_1	F_2	...	F_i
X_1	$x_{1,1}$	$x_{1,2}$	...	$x_{1,i}$
X_2	$x_{2,1}$	$x_{2,2}$	...	$x_{2,i}$
X_3	$x_{3,1}$	$x_{3,2}$	...	$x_{3,i}$
...	...	...	...	...
X_n	$x_{n,1}$	$x_{n,2}$	...	$x_{n,i}$

(a) Node 1 with 'i' features.

	F_1	F_2	...	F_j
X_1	$x_{1,1}$	$x_{1,2}$	...	$x_{1,j}$
X_2	$x_{2,1}$	$x_{2,2}$	...	$x_{2,j}$
X_3	$x_{3,1}$	$x_{3,2}$	...	$x_{3,j}$
...	...	...	...	...
X_n	$x_{n,1}$	$x_{n,2}$	...	$x_{n,j}$

(b) Node 2 with 'j' features.

	F_1	F_2	...	F_k
X_1	$x_{1,1}$	$x_{1,2}$	...	$x_{1,k}$
X_2	$x_{2,1}$	$x_{2,2}$	...	$x_{2,k}$
X_3	$x_{3,1}$	$x_{3,2}$	...	$x_{3,k}$
...	...	...	...	...
X_n	$x_{n,1}$	$x_{n,2}$	...	$x_{n,k}$

(c) Node 3 with 'k' features.

	F_1	F_2	F_3	...	F_m	Y
X_1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,m}$	y_1
X_2	$x_{2,1}$	x_{22}	x_{23}	...	$x_{2,m}$	y_2
X_3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{3,m}$	y_3
...	...	...	...	...	...	...
X_n	$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	...	$x_{n,m}$	y_n

(d) Total data with 'm' features.

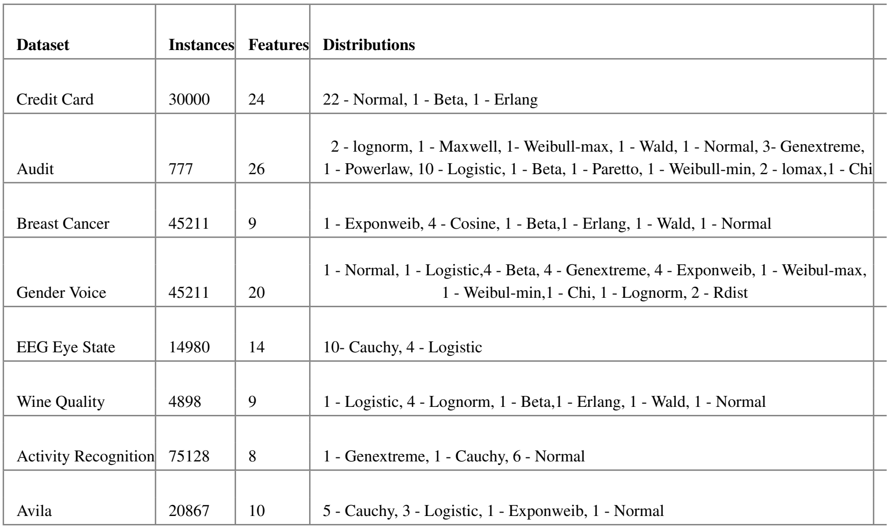
Fig. 1 Vertical partition of a dataset with m features divided into three different nodes

Algorithm 1 Vertical FL

```
At Client:
for $i \in \{1 \ \ to \ \ K \ \ \}\forall \ \ features$ do
    for $j \in \{1 \ \ to \ \ S \ \ \}\forall \ \ distributions$ do
        Compute $p - value$ $(p_j)$ using K-S test
    end for
    $\psi_i$ = max $(p_1,...,p_1)$
end for
Transmit $\Psi$
At Server:
//train the model on the local data
for $p \in \{1 \ \ to \ \ P \ \ \}\forall \ \ epochs$ do
    for $d \in \{1 \ \ to \ \ n \ \ \}\forall \ \ N$ do
        $\theta^{p+1} = \theta^p - \eta\nabla L(\theta^p)$
        where $\theta$ = weight vector, $\eta$ = learning rate,
        $L$ = Emperical loss function
    end for
end for
for $i \in \{1 \ \ to \ \ C \ \ \}\forall \ \ clients$ do
    for $f \in \{1 \ \ to \ \ F \ \ \}\forall \ \ features$ do
        Generate $D_f$ from $\psi_f$
    end for
end for
$D_x = \bigcup_{i=1}^{C} \bigcup_{f=1}^{F} D_f$
$D_y = F(D_x, \theta)$ //compute labels for the above data
for $p \in \{1 \ \ to \ \ P \ \ \}\forall \ \ epochs$ do
    for $d \in \{1 \ \ to \ \ S \ \ \}\forall \ \ D_s$ do
        $\theta^{p+1} = \theta^p - \eta\nabla L(\theta^p)$
    end for
end for
```

It is essential that in any vertically partitioned scenario, some data is needed at the server side in order to predict any global event. In the first step of our approach, a machine learning model is trained on the locally available data at the server. Next, each node computes the distribution of each feature. For example, node 1 computes the distributions for the features 1 to i represented by the set ψ_1 to ψ_i, node 2 represents the distribution set ψ_1 to ψ_j, and node 3 represents ψ_1 to ψ_k. The distributions of each feature are computed using the fusion algorithm, as illustrated in Sect. 2.1. These distribution parameters from each client are sent to a central server. Sample points are generated by the server using these parameters, i.e., from the parameters received from node 1, and it generates points D_1 to D_i where D_i indicates data points for feature i. Similarly, sample points are generated for the remaining features received from multiple clients. The sample data across multiple features is now merged vertically. The predicted values for this sample data are calculated using the weights from the model generated by the server. This combined dataset helps us in creating a sample repository of data using the distributions of each feature from every client, as shown in Fig. 1d. The final step includes training a machine learning model on this new sample dataset at the server, as explained in Algorithm 1.

Table 1 Dataset description that shows the instance count, number of features and distribution of each feature



Dataset	Instances	Features	Distributions
Credit Card	30000	24	22 - Normal, 1 - Beta, 1 - Erlang
Audit	777	26	2 - lognorm, 1 - Maxwell, 1- Weibull-max, 1 - Wald, 1 - Normal, 3- Genextreme, 1 - Powerlaw, 10 - Logistic, 1 - Beta, 1 - Paretto, 1 - Weibull-min, 2 - lomax,1 - Chi
Breast Cancer	45211	9	1 - Exponweib, 4 - Cosine, 1 - Beta,1 - Erlang, 1 - Wald, 1 - Normal
Gender Voice	45211	20	1 - Normal, 1 - Logistic,4 - Beta, 4 - Genextreme, 4 - Exponweib, 1 - Weibul-max, 1 - Weibul-min,1 - Chi, 1 - Lognorm, 2 - Rdist
EEG Eye State	14980	14	10- Cauchy, 4 - Logistic
Wine Quality	4898	9	1 - Logistic, 4 - Lognorm, 1 - Beta,1 - Erlang, 1 - Wald, 1 - Normal
Activity Recognition	75128	8	1 - Genextreme, 1 - Cauchy, 6 - Normal
Avila	20867	10	5 - Cauchy, 3 - Logistic, 1 - Exponweib, 1 - Normal

Table 2 Comparison of accuracies between vertical fusion learning and central learning

	Vertical fusion	Central learning
Credit card	73.3	81.4
Audit	89.1	98.1
Breast cancer	97.1	97.8
Gender voice	95.4	97.7
EEG eye state	62.31	83.41
Wine quality	52.1	54.6
Activity recognition	90.1	97.8
Avila	55.8	72.6

4 Experimental Results

Our experimental settings, datasets on which the experiments were carried out and the comparison of the results obtained using vertical fusion with those from a central learning setup are presented in this section. We also present certain trade-offs associated with this approach.

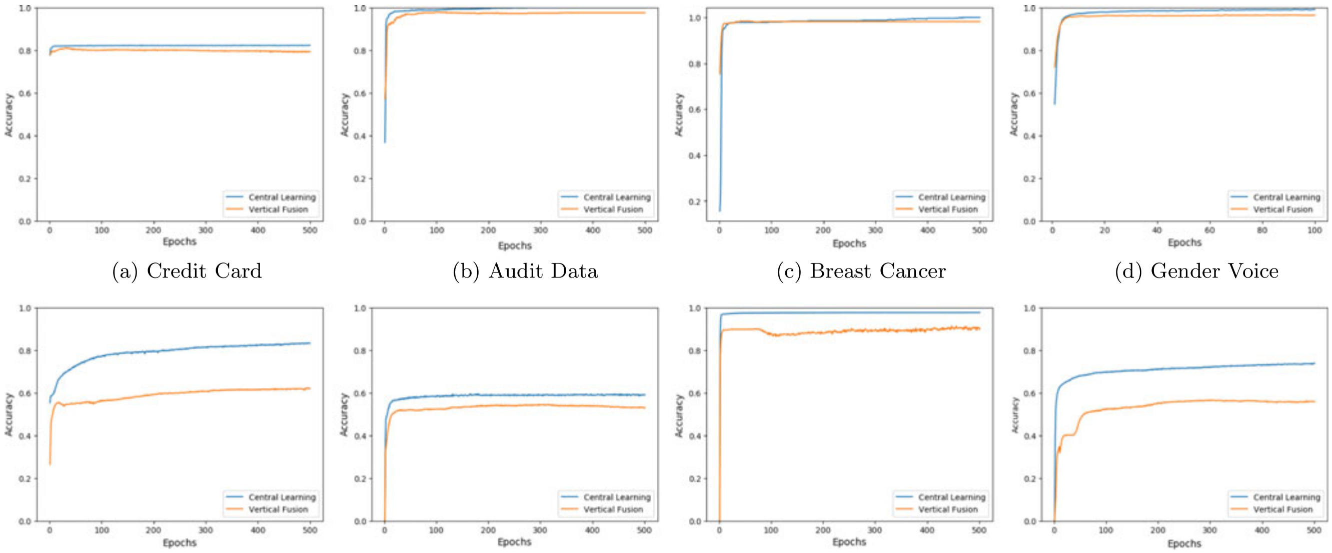
4.1 Experimental Settings

The setup consists of a simple client-server model where the number of clients is set to four. The datasets that are used for the experiments are credit card, audit, breast cancer, gender voice, EEG eye state, wine quality, activity recognition and Avila datasets. These are available at [4]. The number of features and instances of these datasets is summarized in Table 1. All the features of each dataset have been split uniformly across four clients. We have used a total of 25 distributions to compare and find the distribution of each feature. The overall feature distributions for the four datasets have also been summarized in Table 1. It is important to note that the division of the number of features across multiple clients is independent of the number of clients involved in the learning process as the purpose of the clients is to only find the distributions of each feature. We use a simple two-layered multi-layer perceptron at the server where each hidden layer contains ten hidden nodes to train the locally available data. Once the model parameters, θ, are computed, the server generates sample points using the distribution parameters from each feature. These points are merged vertically, and the model parameters (θ) are used to find the labels of this new sample dataset. A machine learning model with the same configuration, as described above, is now used to train the combined dataset (Fig. 2).

4.2 Results

All previously proposed solutions on vertically partitioned data either had the clients transmit their data to the server or frequently update their local model parameters, both of which are not communication efficient. Our approach sends only the distribution parameters, which is only a fraction of data compared to either the weight vector or the total data. It is also interesting to note that our approach needs only one round of communication to the server.

The accuracies achieved through our proposed model are summarized in Table 2. We have compared our accuracies with a central setup, where the data from each client is transmitted to a central server. A major drawback with the central approach is the large amount of information each client has to transfer and, at the same time, raises some serious privacy concerns when sensitive data is moved across the network. As we can observe from Table 2, the accuracy obtained from the vertical fusion algorithm is compared with those from a centralized approach with a minimal drop of $\tilde{2}\%$ inaccuracies. The vertical fusion approach achieves 97.1% and 95.4%, 52.1% accuracy, whereas the centralized approach achieves 97.8%, 97.7% and 54.6% accuracy on breast cancer, gender voice and wine quality datasets where the difference is less than 2%. In credit card, audit and activity recognition datasets, we see the drop in accuracy to be less than 10%. And finally, in EEG eye state prediction and Avila datasets, the difference in accuracies is higher than 10%. This difference in accuracy can be attributed to the quality of the data generated by the server from the client's



(a) Credit Card

(b) Audit Data

(c) Breast Cancer

(d) Gender Voice

(e) EEG Eye State

(f) Wine Quality

(g) Activity Recognition

(h) Avila Voice

Fig. 2 Comparison of accuracies between vertical fusion and central learning approaches across eight different datasets

distributions. These accuracies can be further improved if clients can determine the distribution of each feature more accurately by considering more number of distributions in the initial setup. It would also be interesting to see how the accuracy varies when the number of points generated at the server increases.

5 Conclusion and Future Work

The goal of this paper is to reduce the amount of data transferred between clients and a server in order to learn a global model in a vertically partitioned scenario. We have shown that instead of transferring the actual data, we can transmit the distributions of each feature from each client, and the server is able to achieve similar accuracies to that of a central learning method by training the model on the sample points generated from these distributions. The current work is still in its nascent stage, and learning in vertically partitioned scenarios still poses some very challenging issues such as leakage of client information through the feature distributions, analysis of more intricate datasets, non-independent features and non-stationary feature distributions.

References

1. Boyd S, Parikh N, Chu E (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc
2. Das K, Bhaduri K, Kargupta H (2010) A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to-peer networks. Knowl Inf Syst 24(3):341–367
3. DeGroot M, Schervish M (2012) Probability and statistics. Addison-Wesley. https://books.google.co.in/books?id=4TlEPgAACAAJ
4. Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
5. Gascón A, Schoppmann P, Balle B, Raykova M, Doerner J, Zahur S, Evans D (2017) Privacy-preserving distributed linear regression on high-dimensional data. Proc Priv Enhancing Technol 2017(4):345–364
6. Heinze C, McWilliams B, Meinshausen N (2016) Dual-loco: distributing statistical estimation using random projections. In: Artificial intelligence and statistics, pp 875–883
7. Kamp M, Boley M, Keren D, Schuster A, Scharfman I. Communication-efficient distributed online prediction by decentralized variance monitoring. In: Proceedings of the European conference on machine learning and principles and practice of knowledge discovery (ECML/PKDD), pp 623–639
8. Kasturi A, Ellore AR, Hota C (2020) Fusion learning: a one shot federated learning. In: Proceedings of international conference in computational sciences. Springer, Berlin
9. Konecnỳ J, McMahan HB, Ramage D, Richtárik P (2016) Federated optimization: distributed machine learning for on-device intelligence
10. Konrad B, Lieber D, Deuse J. Challenges for data mining on sensor data of interlinked processes
11. Liu Y, Kang Y, Zhang X, Li L, Cheng Y, Chen T, Hong M, Yang Q (2019) A communication efficient vertical federated learning framework. arXiv preprint arXiv:1912.11187
12. Mallik R, Kargupta H (2011) A sustainable approach for demand prediction in smart grids using a distributed local asynchronous algorithm. In: CIDU, pp 1–15
13. Mangasarian OL, Wild EW, Fung GM (2008) Privacy-preserving classification of vertically partitioned data via random kernels. ACM Trans Knowl Discov Data (TKDD) 2(3):1–16

14. Mohassel P, Zhang Y (2017) Secureml: a system for scalable privacy-preserving machine learning. In: 2017 IEEE symposium on security and privacy (SP). IEEE, pp 19–38
15. Stolpe M, Liebig T, Morik K (2015) Communication-efficient learning of traffic flow in a network of wireless presence sensors. In: Proceedings of the workshop on parallel and distributed computing for knowledge discovery in data bases (PDCKDD), CEUR workshop proceedings, CEUR-WS
16. Wu S, Teruya T, Kawamoto J, Sakuma J, Kikuchi H (2013) Privacy-preservation for stochastic gradient descent application to secure logistic regression. In: The 27th annual conference of the Japanese Society for Artificial Intelligence, vol 27, pp 1–4
17. Yang K, Fan T, Chen T, Shi Y, Yang Q (2019) A quasi-newton method based vertical federated learning framework for logistic regression. arXiv preprint arXiv:1912.00513
18. Yu H, Vaidya J, Jiang X (2006) Privacy-preserving svm classification on vertically partitioned data. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, Berlin, pp 647–656

FedPeer: A Peer-to-Peer Learning Framework Using Federated Learning

Anirudh Kasturi, Raviteja Sivaraju, and Chittaranjan Hota

Abstract The computing power of personal devices has increased in the recent past, and the large amount of data is being generated by these devices. Owning to security concerns, research is currently done on training machine learning models locally on these devices. In order to train a global model, two different approaches have been proposed in the recent past. One, in which the models trained on the devices are aggregated at a central location, and the aggregated model is transmitted back to the devices and the other, where devices communicate the models among themselves without the need of a central server. Both these approaches have their own share of advantages and short comings. In this paper, we propose $FedPeer$ a new decentralized machine learning approach where the nodes are clustered based on confidence level of predicting the data. The higher confident nodes form a federated cluster, whereas the remaining nodes participate in a peer-to-peer decentralized learning along with a representative from the federated cluster. Our experiments show that this approach is proved to have a faster convergence rate and lower communication overhead when compared to the either a federated approach or a complete peer-to-peer method.

Keywords Distributed machine learning · Federated learning · Decentralized learning

1 Introduction

Current smart devices are generating large amounts of data both from users and the environment using of the help of underlying sensors [4]. Learning from this data without transferring it to a central server has become one of the major challenges of

A. Kasturi (✉) · R. Sivaraju · C. Hota
BITS Pilani, Hyderabad Campus, India
e-mail: p20170403@hyderabad.bits-pilani.ac.in

R. Sivaraju
e-mail: h20181030096@hyderabad.bits-pilani.ac.in

C. Hota
e-mail: hota@hyderabad.bits-pilani.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_39

distributed learning. Federated learning [5, 8], a recent advancement in distributed learning, has gained tremendous attention in the machine learning community as it was able to solve one of major challenges of distributed learning, i.e., learning without transferring actual data from the clients node to a central server. One of major challenges of this approach is the dependency on a centralized entity [7]. To overcome this problem, researchers have proposed the idea of a decentralized framework where there is no dependency on a centralized system. Instead, the nodes communicate among themselves and update their models in a distributed fashion. By exchanging the models, all peers develop a global model in collaboration. This avoids the bottleneck on the central entity in large networks and also provides security to the private data as an intruder cannot have a global model at any point of communication. Decentralized framework has great potential in an IoT environment. Although this model fares better than the federated approach in certain aspects, it brings up the challenge of network overhead as there will be more packets that will be transmitted during model exchange.

In this paper, we propose a hybrid framework where we combine the benefits of both federated learning approach and the decentralized framework where we group nodes with higher confidence value into a federated cluster. The remaining nodes form a decentralized setup and exchange models. In such a framework, a representative from the federated cluster interacts with the other nodes from the decentralized setup helping in faster convergence.

2 Related Work

A recent and a seminal work in the area of distributed machine learning was proposed by [8]. Their approach has found wide acceptance in a number of applications. The objective function formulated by them is:

$$\min_{w \in \mathbb{R}^d} f(w) \quad \text{where } f(w) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

which is the average of loss functions from various clients. These losses can be mean squared errors loss (liner regression), logistic loss function (logistic loss), hinge loss (SVM), etc. Each $f_i(w)$ refers to the loss function of agent "i". The central entity keeps track of all the agent's loss function and helps the each agent to minimize the global loss function locally.

Contrary to the federated setup where a central server is needed, a decentralized model does not need a central server. All the agents can learn a global model by peer-to-peer communication. Many solutions have been proposed to solve the problem of fully decentralized learning and optimization. Reference [9] proposed an approach where the goal is to minimize the summation of convex optimization functions of all the agents in the network. Another research in this direction was proposed by Ref.

[2] where agents communicate with their neighbors using the gossip protocol. The objective function is a pairwise objective function, i.e., the goal is to minimize the objective function formed with each neighbor. This method is useful for developing a common model for cluster of users who have similar interests. Pairwise objective function is defined as follows:

$$\min_{\theta} \frac{1}{n_2} \sum_{1 \leq i, j \leq n} f(\theta; x_i, x_j)$$

where i, j are two agents which are neighbors. Every agent pairs with its neighbor and keeps updating θ, and toward the end, all neighbors have a common model. Authors in [1] proposed a model which is not only decentralized, but they also give importance for the agents to learn a personalized model.

Though all the above-mentioned works succeed in training a global model in a decentralized fashion, personalization of models is not achieved completely as all the agents learn a single global model. Our approach tries to improve over these methods by defining an objective function with a trade-off parameter that can control the extent up to which each agent learns the global model and at the same does not loose its personalization. This is more common in scenarios where certain agents have more data and some with less data. Learning a global model might compromise the personalization of agents with higher data.

3 Problem Formulation and Algorithm

3.1 Proposed Loss Function

We assume that there are n agents in the network. Each agent has a set of data points $D_i = x_i, y_i$ such that $1 \leq j \leq m_i$, and all agents have p features. The goal of each agent is to learn model $\theta_i \in R_p$. The local loss function for each agent can be defined as $l(\theta; x_i, y_i)$. This is a convex loss function in θ_i. In a classification problem, we can take the logistic loss function as:

$$\theta_i^{\text{loc}} \in \arg\min_{\theta \in R^p} \left[\frac{1}{m_i} \sum_{j=1}^{m} l(\theta; x_i^j, y_i^j) + \lambda \theta^2 \right]$$

The above function is the general form of the local loss function along with the regularization term. If every agent tries to learn on its own, then the goal would be to minimize this function, but as this is the collaborative model, we need a different loss function that would meet the need of collaborative learning. The new loss function can be formed by:

$$\mathbb{Q}_{\mathbb{L}}(\theta) = \frac{1}{2}\sum_{i<j}^{n} W_{ij}|||\theta_i - \theta_j|||^2 + \mu\sum_{i=1}^{n} D_{ii}c_i\mathbb{L}_i(\theta_i; S_i)$$

where θ is stack of all parameters of all the agents in the network, i.e., $\theta = [\theta_1; \theta_2; \theta_3 ... \theta_n]$. W_{ij} is the weightage given to the relation between two agents. If two agents are similar, then $W_{ij} = 1$, else $W_{ij} = 0$. Similarity of the agents can be computed by taking the nearest neighbors or using $cosinesimilarity$ between the agents. The loss function can be viewed as the summation of global loss and local loss. $W_{ij}\|\theta_i - \theta_j\|^2$ minimizes the squared error between the parameters of the agents in pairwise manner. It implies that all agents will try to learn the same model. The right-hand side of summation part is purely local loss function. c_i is known as the confidence factor of the agent. This is dependent on the number of data points present on each agent. A higher number indicates a greater confidence factor, and its value lies between 0 and 1, i.e., $0 < c_i < 1$. D_i is the normalization factor with $D_{ii} = \sum W_{ij}$ where 1. μ is the hyperparameter which is responsible for the trade-off between learning a global model or local model at each agent. If μ is very high, then in the loss function, preference is given to the local loss function that implies agent tends to learn local model. If μ is less, then preference is given to the global loss function, and agent learns the global model. μ is important because it allows less confident agents to learn from their neighbors, and at same time, confident agent's model is not disturbed by other agents.

3.2 Minimization Algorithm

Decentralized coordinate descent algorithm is used to minimize the loss function introduced in the previous section. This algorithm allows agents to learn the model in a decentralized manner. Initialization step: At $t = 0$, all the agents' local parameters are initialized to some arbitrary values. The algorithm performs two steps:

- Update Step: Agent i updates its $\theta_i(t+1)$ based on the parameters $\theta_j(t)$ received from its neighbors.

$$\theta_i(t+1) = \theta_i(t) - (1/L_i)[\nabla\mathcal{Q}_{\mathcal{L}}(\theta_i(t))]$$

The above step is similar to the gradient descent step. $1/L_i$ can be interpreted as learning rate, and it is multiplied by the gradient of the global loss function. After calculating gradient, the equation turns out to be as:

$$(1-\alpha)\theta_i(t) + \alpha\Big(\sum_{j\in\mathcal{N}_\rangle} W_{ii}D_{ii}\theta_j(t) - \mu c_i\nabla\mathcal{L}_\rangle(\theta_i(t); S_i)\Big)$$

$$\text{where} \quad \alpha = \frac{1}{(1+\mu c_i L_i^{\text{loc}})} \quad \text{and} \quad 0 < \alpha \leq 1$$

- Broadcasting step: After updating its local parameters, agent i broadcasts its new parameters to its neighborhood.

Algorithm 1 FedPeer Learning

```
**Federated Update:**
for $i \in \{1 \;\; to \;\; C \;\}\forall \;\; clients$ do
    for $e \in \{1 \;\; to \;\; E \;\}\forall \;\; epochs$ do
        $\theta^e = \theta^e - \eta \nabla L(\theta^e)$
        $where \quad \theta = weight \quad vector, \eta = learning \quad rate, L_e = Loss$
    end for
    transfer model parameters to representative node
end for

**Decentralized Update:**
for $i \in \{1 \;\; to \;\; C \;\}\forall \;\; clients$ do
    for $e \in \{1 \;\; to \;\; E \;\}\forall \;\; epochs$ do
        $\theta^e = \theta^e - \eta \nabla L(\theta^e)$
    end for
    transfer model parameters to other nodes
end for
```

In the above steps, we can see that minimization of the loss function is occurring in collaboration with all the agents. If c_i value is more, then α value decreases. So clearly, the agent tends to learn its model based on the local loss function. If c_i value is less, the value increases which in turn makes the agent to drift away from the parameter values that would satisfy the local loss function, i.e., they try to learn the global model.

4 Proposed Approach

In the above decentralized algorithm, it has been observed that by modifying the algorithm, we can reduce the number of broadcasting messages among the nodes. We have seen that after updating step, each node broadcasts its updated value to its neighborhood. This step unnecessarily increases the message overhead in the network. We can avoid this by using the concept of federated learning. Decentralized learning eliminates the need of central server which is required in the federated learning which may lead to bottleneck at the server, but federated learning does not have the problem of broadcast messages. We come up with a solution which incorporates the advantages in both the federated learning and decentralized learning. In federated learning, group of clients interacts with a central server. These clients can be computationally powerful handheld devices or IoT devices. Based on the local data, they compute a local model. These local models are given to the central server. Central server performs aggregation of these models and develops a new global model. This global model is again given to clients, and they personalize these models by running SGD or any other optimization algorithm with their local data. This process continues till they converge. In decentralized learning, there is no central entity. All the devices communicate among each other and develop a global model. In these approaches, any new node will tend to learn and benefit from the global models. Even if they do not have sufficient data with them, they learn a reasonably good model. It has been observed that nodes with lesser data need to participate in the decentralized model more actively than the nodes which already have good amount of data. Because with the increase in data confidence of nodes increases, they tend to rely on their local model more than global model. By exploiting this observation, a new method of learning is proposed in order to reduce the broadcasting overhead. Based on the confidence of the nodes, clusters are formed. Nodes with higher confidence are formed into one cluster, and remaining lower-confidence nodes remain as it is in the network. We implement federated learning among the nodes in cluster. One of the nodes in this cluster is selected at random which collects the models of other nodes and aggregates them. This selected node participates in the decentralized learning. The size of clusters should be small as the capacity of nodes will be limited to perform aggregation of several models. With this modification, we are able to achieve lesser convergence time and also lesser number of broadcast messages.

As we can see from Fig. 1, the nodes on the left participate in federated learning and nodes on the right perform a decentralized training. The node in the center is the representative node. In federated learning, each client transfers its model parameter to the representative node. Any polling algorithm can be used to choose a node among confident nodes which act as the representative of those nodes. This representative

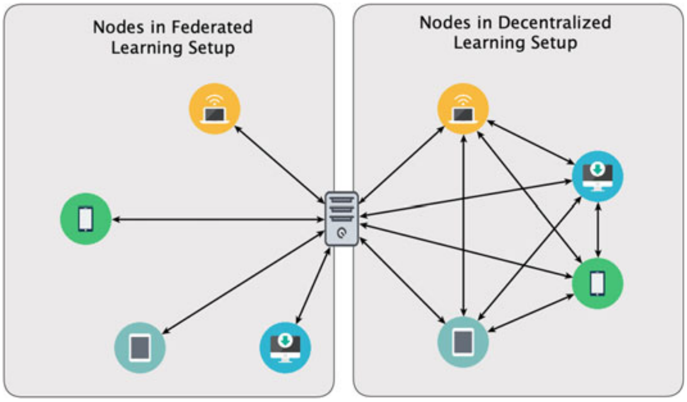

Nodes in Federated Learning Setup
Nodes in Decentralized Learning Setup

Fig. 1 Architectural diagram of a FedPeer learning system

node aggregates the model parameters and transfers them back to the clients. In the case of the decentralized training, every client transfers its model to every other node in the group as it involves broadcasting.

5 Experimental Results

In this section, we present our results along with the experimental settings we have used to achieve them.

5.1 Experimental Settings

We have experimented this approach on three different datasets for a classification problem. We have used two datasets MNIST [6] and Adult Income [3] along with synthetically generated dataset. The statistics of these datasets are summarized in Table 1. The MNIST database of handwritten digits has a training set of 42,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. Each example is an image of handwritten digits. Each image is of 28 × 28 pixel resolution. We have applied PCA on this dataset to reduce the number of features to 100 as 75% variance is retained by these 100 features. From the dataset, 16,000 data points are taken as the test data. Remaining data is split into the nine nodes each node having 7000, 3000, 4000, 50, 6000, 4000, 30, 20 and 40 points, respectively. Corresponding confidence factors for these nodes are 1, 0.4, 0.5, 0, 0.8, 0.5, 0, 0, 0. $c1 = 1$ $c2 = 0.4$. For the Adult dataset, we have used one-hot encoding to vectorize the categorical features resulting in a total of 95 features. A similar breakdown of data points has been done on the other two datasets.

5.2 Performance Metrics

The two metrics we have considered to compare our approach with a decentralized model are the total time taken to achieve the accuracy and the number of messages transferred between clients during the process.

5.3 Results

The overall results have been summarized in Table 2. We compare the total time taken, number of messages transferred and the accuracy of our proposed approach

Table 1 Dataset description

Dataset	Instances	Features	After PCA
MNIST	60,000	784	100
Adult income	48,842	95	95
Synthetic	30,000	20	20

Table 2 Total time taken and accuracy with hybrid fusion, fusion and federated learning

	Decentralized			FedPeer		
Datasets	Time (s)	Acc (%)	# of Msg	Time (s)	Acc (%)	# of Msg
MNIST	602	73	9000	210	72.4	3000
Adult income	101	77.4	9000	53	77.1	3000
Synthetic	730	77	9000	259	76.2	3000

to a decentralized framework. We can observe that the $FedPeer$ achieves similar accuracy when compared to a decentralized model but taking significantly lesser time and communication bandwidth. In a decentralized approach with 10 clients, the number of messages exchanged per epoch is 90, and at 100 epochs, this number stands at 9000, whereas in the proposed approach, only 5 clients participate in the decentralized training which brings down the number of messages exchanged to 2000. Since the remaining 5 clients participate in a federated learning model, the messages exchanged in this approach stand at 1000 as each client only talks to the server and vice versa. The total number of messages in this hybrid approach will, hence, be 3000 which is less than fifty percent of a decentralized model. The lesser communication between nodes has a significant impact on the amount of time taken to converge to the given accuracy. As we can see from Table 2, FedPeer takes nearly half the time when compared to a decentralized algorithm.

6 Conclusion

This paper shows that a hybrid model that uses both a federated and a decentralized setup yields much better results compared to a stand-alone decentralized framework. We have demonstrated that our proposed approach significantly reduces both the amount of time and the number of messages transmitted across the network. It is important to note that this method will perform similarly to a decentralized approach if all the nodes have no significant data points.

References

1. Bellet A, Guerraoui R, Taziki M, Tommasi M (2018) Personalized and private peer-to-peer machine learning. In: International conference on artificial intelligence and statistics, pp 473–481
2. Colin I, Bellet A, Salmon J, Clémençon S (2016) Gossip dual averaging for decentralized optimization of pairwise functions. arXiv preprint. arXiv:1606.02421
3. Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
4. Garcia Lopez P, Montresor A, Epema D, Datta A, Higashino T, Iamnitchi A, Barcellos M, Felber P, Riviere E (2015) Edge-centric computing: vision and challenges
5. Konecnỳ J, McMahan HB, Ramage D, Richtárik P (2016) Federated optimization: distributed machine learning for on-device intelligence
6. LeCun Y, Cortes C (2010) MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/
7. Li T, Sahu AK, Talwalkar A, Smith V (2020) Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag 37(3):50–60
8. McMahan HB, Moore E, Ramage D, Hampson S et al (2016) Communication-efficient learning of deep networks from decentralized data. arXiv preprint. arXiv:1602.05629
9. Ram SS, Nedić A, Veeravalli VV (2010) Distributed stochastic subgradient projection algorithms for convex optimization. J Optim Theory Appl 147(3):516–545

Weighted Road Network Distance-Based Data Caching Policy for Spatio-Temporal Data in Mobile Environment

N. Ilayaraja, J. Sherin, J. Sandra, and F. Mary Magdalene Jane

Abstract Mobile users expect context-aware spatio-temporal data while they use location-based services. Caching data at the mobile client reduces execution time and improves the performance of the services. An efficient cache replacement policy is considered important to retain the useful items and evict the others from the cache. This paper proposes a cache replacement policy considering the spatial and temporal factors for eviction. Network distance is considered as an important eviction parameter along with the properties of points of interest which are dynamic. The simulation results by varying the query interval, moving interval, and cache size are presented to verify that the proposed policy outperforms the existing FAR and PAID cache replacement policies.

Keywords Data Cachingcaching · Cache Replacementreplacement · Network Voronoivoronoi

1 Introduction

The demand for location-based services is increasing as the number of wireless users grows exponentially. Location-based spatial query processing offers real-time location information [1]. Existing studies [2–10] have investigated how to process the spatio-temporal queries in road networks by determining the shortest road network distance between points of interest. The main goal is to reduce latency and increase data availability.

Nearest neighbor query (NNQ) results with respect to a moving query point largely focus on techniques that utilize pre-computed network distances of the road network.

N. Ilayaraja (✉)
PSG College of Technology, Coimbatore, India
e-mail: nir.mca@psgtech.ac.in

J. Sherin · J. Sandra
University of Texas at Dallas, Dallas, USA

F. Mary Magdalene Jane
Dr. N. G. P. Arts and Science College, Coimbatore, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_40

Network Voronoi diagram (NVD) index structure is one of the important techniques for representing the road network [5–7, 11–14].

Data caching has proven to minimize latency and access costs and hence the cache replacement policy has to be efficient [3, 6, 15]. The existing cache replacement policies such as the temporal factors are considered for eviction in policies like least recently used LRU-k [14] that take into account the temporal factors, whereas the policy furthest away replacement (FAR) considers the spatial properties. The spatial and temporal properties are applied for eviction by the policy probability area inverse distance (PAID).

Euclidean distance has been used by the previous policies as a measure, whereas this work considers the road network connectivity and applies network distance. The shortest distance depends on the connectivity of the road network rather than the Euclidean [6].

In the proposed work, a new caching policy is derived on road network distance while answering nearest neighbor queries with modifications in spatial parameters. This paper proposes a weighted network distance-based cache replacement policy (WIND-CP) which considers the network distance between the client and a point of interest, POI service opening time (e.g., the time when a restaurant is open for service), the weighted density of the valid scope and weights on the road network direction and access frequency of the POI item. A ratio of the number of queries generated to the number of queries answered from cache is the cache hit ratio (CHR) which is the performance metric. Data availability is improved and access cost is reduced when the CHR is high.

The rest of the paper is organized as follows: In Sect. 2, we present the problem statement and related work in the spatial query processing in road network; Sect. 3 describes the proposed system design and terminologies used; Sect. 4 describes the system implementation; Sect. 5 shows the efficiency of proposed cache policy and extensive performance analysis; and finally, Sect. 6 concludes with future research directions.

2 Literature Survey and Model Description

Range Query and NNQ search are very popular location-dependent queries in LDIS to provide location-related POIs [10, 16]. Depending on how to measure the distance between query point and interest point, the NNQs are categorized as Euclidean distance NNQ and road network distance NNQ.

Studies have been carried out on NN and kNN queries and the algorithms proposed applied Euclidean spaces, the major setback being not applicable to road networks. Hence, these algorithms are not suitable for returning the right object to the user by approximating the exact distance between the user query point and the interest point's location. The authors of [17] proposed a spatial network database (SNDB) that generates a search region for every generator point that grows from the client location. References [2, 10] also identified a solution for the kNN queries in SNDB.

Finally, [5, 6, 9] identified a novel approach to efficiently address NN and kNN queries in SNDB based on the NVD. Reference [7] also identified the solution for processing continuous NNQ in the road networks on the air indexing.

Most of the existing location-dependent query processing techniques rely on client/server architecture in the context of location-based services (LBS). A client can raise an NN query to search for any of the interesting points/events for his need from his mobile device while on move.

When certain POI is searched by category in the maps, getting results that are very far from the current user's location will not be of much use. The main aim of the search is to obtain a POI that is very near to his location specified.

Even if there are POIs very close to the given location using Euclidean distance, the accessibility becomes ambiguous to the new user. The term closest may refer to the Euclidean distance between the two points or the shortest road route. If the Euclidean distance is considered, POI may appear closer, but the direct path may not be accessible using a motor vehicle for the user. Hence, by choosing a POI which is closer in terms of Euclidean distance, a path that is the closest may not be chosen.

For example, in the above scenario, a user is searching for the nearest hospital for his emergency need at query location "*Q*", as shown in Fig. 1. Three hospitals (*A*, *B*, and *C*) are located, which are closer to the user's query location. By using the Euclidean distance, we identify that the closest POI from the user's location "*Q*" is Hospital A (ED_1) and Hospital B (ED_2), whereas in terms of the road network, the distance to the Hospital A (AD_1) is longer than the distance to the Hospital B (AD_2) from query location "*Q*" as there is no appropriate route along the median distance. In this proposed work, in order to provide a more accurate distance to the POI, road network is to be implemented considering the factors such as traffic condition, number of signals/junctions, tollgates, etc., which will add additional time taken to reach the POI. For example, two road networks of unequal lengths have traffic on the longer route considerably less than the traffic on the shorter route. If the longer route is taken, the time to reach the POI will be shorter and vice versa. Thus by assuming

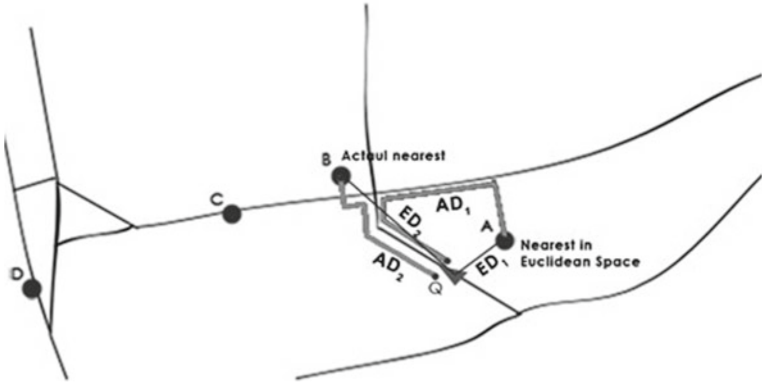


Fig. 1 Scenario representation using the road network

the factors mentioned above, the optimal route is the route that takes the least amount of time to reach the POI is identified.

In addition to the above factors, the operating time of POI is also considered significantly. For example, the working hours of the hospital POIs (*A*, *B*, and *C*) are from 09:00 to 20:00, 21:00, 22:00, respectively. The query time by the user is 19:40, suggesting Hospital A as an option using Euclidean distance is not appropriate. It is imperative to provide options B and C to become valuable to the user. Hospital B is suggested as the nearest hospital to the user upon road network. Further, if the conditions such as traffic and operating hours of the Hospital B have an impact on user's reaching time, he can avail the service of the Hospital C as an additional result to his query. In order to minimize the time consumption to send another query and receive the required results, by implementing this option in the proposed work, the user is not required to send another query to search for the nearest hospital.

3 Assumptions and Terminologies

The assumptions and terminologies used for modeling the proposed work are described in this section. Service area refers to the geographical coverage area, and item value is an instance of the item that is valid for a service area.

3.1 Valid Scope

Voronoi diagram is a partition of a plane into regions where each region has a generator point (POI) and all points in one region are closer to the respective point than to any other [5, 6, 18, 19]. Spatial networks (e.g., road networks) can be modeled as weighted graphs where the nodes of the graph represent road intersections or points of interest and the edges connecting the nodes represent the roads. The weights can be the distances between nodes or they can be the time it takes to travel between the nodes. We define valid scope as the network region in which a data item value is valid. When spatial networks are modeled as graphs, the valid scope can be represented as a subgraph. The valid scopes are generated using NVDs.

3.2 Network Distance

The least-cost distance from the position of a client (P) to a point of interest (P_i) is called the network distance. When the mobile client requests a location-dependent service, we assume that the user can choose the shortest/least-cost path from his current position to the POI object. The POI object which is far away from the user either in his direction of motion or in inverse direction will have a lower chance of using it in near future. This implies that a POI object with a higher network distance would be more probable for eviction.

3.3 Network Density

On road networks, the client has a higher chance of remaining in a region with densely connected roads and a larger area for a longer time than a region with a smaller area and sparsely connected roads as discussed by [20, 21]. They argued that the density of a valid scope has an impact on the access probabilities for different data values and the distances between roads and their degree of connectivity should be considered. The weighted network density (WD) is computed for every valid scope (network Voronoi cell) as defined by [20, 21] as follows:

$$\mathrm{WD} = \frac{2 \cdot \sum w_i}{n(n-1)} \tag{1}$$

where w_i represents the weight of each edge in the graph and n is the number of vertices in the graph.

4 Experimental Design

The proposed methods have been implemented systematically by calculating the cache replacement score for stale POI objects, estimated travel time to reach the POI object, and an access probability of POI type.

4.1 Proposed Replacement Policy

A cache replacement policy should choose an item with low access frequency, less network density, and longer network distance for eviction also considering the operating time of the POI object. The replacement of data items from the cache is done based on a cost function. The cost function in the policy (WIND-CP) of a data value j of an item is defined by

$$C_{i,j} = \frac{P_i \ \times \mathrm{WD}(V_{i,j}) \times A}{\mathrm{Nd}(V_{i,j})} \tag{2}$$

where WD is the weighted network density of the valid scope V, N_d is the network distance between the client's current position and the POI object considered, P_i is the access frequency, and *A* is the temporal validity of the POI object (difference between the current time and service closing time of the POI object). This policy will evict the data with a lesser cost during each replacement. The policy would choose the data with less access frequency, less network density, and greater network distance for eviction.

4.2 Shortest Distance and Estimated Travel Time Calculation

In this work, Dijkstra's algorithm is applied to find the least cost path from a single source to a single destination. The computational costs are ignored because they are considered pre-processing costs. Figure 2 represents graph notation of the road network with the three hospital POI objects, namely *A*, *B*, and *C*.

A road network is modeled as a graph *G* (<*V,E* >) that represents a road network. The graph node depicts a road intersection or junction, and the edge is the road between any two junctions. The network distance is represented as the weights of the edges. An NNQ on the road network, given a node *Q* and a dataset of clinics distributed on the nodes *V*, finds a clinic that is closest to the node *Q* with respect to network distance. The various weights like distance, average speed, and traffic condition are listed in Table 1. They can be used to compute the estimated travel time to reach the nearest clinic.

For the calculation of the estimated travel time from the query point "*Q*" to the POI object, the factor "traffic condition" is considered. Each route has an optimal average speed and based on that the time taken to cover the distance is calculated and the travel time is taken as the optimal time. The traffic condition is taken as "low", "moderate", and "heavy" according to the road chosen and the time of journey. Each traffic condition is given a percentage weight, and the optimal time obtained is multiplied by the percentage weights to get the actual time taken to reach the desired result.

Consider an identified route with a finite number of edges, then

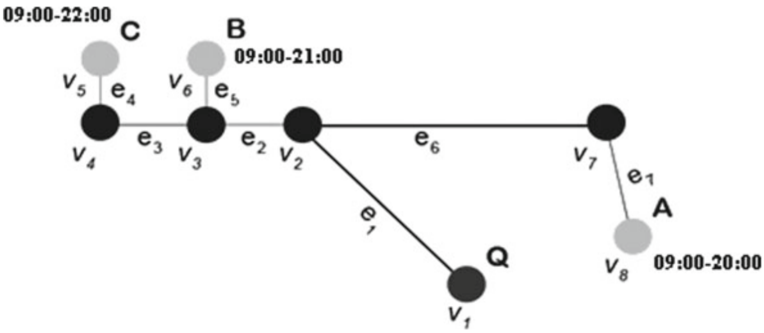


Fig. 2 Graph representation of road network (G) with hospital POI objects

Table 1 Weights of road network G

Edge	Connects vertices	Weight of edge in terms of distance (km)	Optimal average speed of the edge	Traffic condition of the edge
e_1	v_1, v_2	3	50	Medium
e_2	v_2, v_3	1	50	Heavy
e_3	v_3, v_4	1.5	50	Heavy
e_4	v_4, v_5	0.75	40	Low
e_5	v_3, v_6	0.75	40	Low
e_6	v_2, v_7	4	50	Heavy
e_7	v_7, v_8	1.5	40	Low

$$\text{Estimated Travel Time} = \sum_i \frac{d_i}{o_i} * \text{tc}_i \tag{3}$$

where

i denotes the subscript of the edges in the identified route,
d_i is the distance of the ith edge,
o_i is the optimum average speed of the ith edge,
tc_i is the traffic condition of the ith edge.

Consider Table 2 which contains route information to reach the POI (*A*, *B*, and *C*) from the user location *Q*.

The estimated travel time is calculated using the values taken from Tables 1 and 2 to reach each POI (*A*, *B*, and *C*). The weights for traffic condition are given, namely low (10% of the optimal average time), medium (20% of the optimal average time), and heavy (30% of the optimal average time). The estimated travel time from location "*Q*" to Hospital *A*, *B*, and *C* are approximately 13, 7 and 20 min, respectively.

Let the user query time be 19:40, the closest POI using the road network is B and the operating time of B is (9:00–21:00). The user has enough time to utilize the service of the B. For some of the POI categories, servicing time is the major constraint. If the time difference between the current time of the query issued and the closing time of the POI object is lesser, then the resultant POI object is not in use for the user. In that case, thus going to B and then searching for another nearest hospital in the new location would waste his valuable time. This can be avoided by considering the operating time of the POI objects as listed in Table 3 and adjusting the results such that the user will have ample time for being serviced. Hence, when

Table 2 Route to POI (*A*, *B*, and *C*) from location *Q*

POI	Route to POI
A	$\{e_1, e_6, e_7\}$
B	$\{e_1, e_2, e_5\}$
C	$\{e_1, e_2, e_3, e_4\}$

Table 3 Operating time of POI

POI	Opening time of POI (h)	Closing time of POI (h)
A	09:00	20:00
B	09:00	21:00
C	09:00	22:00

C is considered, the user will have enough time for utilizing the service, which is above the threshold value for the category hospital, so that Hospital C as an alternate POI object along with Hospital B is returned with the remaining closing time details.

4.3 Access Frequency Calculation

The access frequency is derived by applying the exponential aging method. For every item, running frequency (P_i) and the item's last access time ($t_{\text{last},\ i}$) are maintained. The current time is denoted by t_{current} and α is the weighing factor to represent the recent access. The running frequency is set to zero and updated by applying the formula [22, 23]:

$$P_i = \frac{\alpha}{(t_{\text{current}} - t_{\text{last},i})} + (1 - \alpha)P_i \tag{4}$$

5 Performance Evaluation

In this section, the proposed cache replacement policy (WIND-CP) is evaluated against PAID and FAR using the simulation model described in [20]. The performance is the CHR. The direction of user movement is not taken into consideration because if the user is simply cruising or if he wants to use an emergency service like a hospital or fuel station, he would prefer the closest one even if it is opposite to his current direction of motion. Query interval is tested from 20 to 200s, which is the time interval between two consecutive queries. As illustrated in Fig. 3, the performance decreases when the query interval is increased. The new policy (WIND-CP) shows a 22% improvement over PAID and 80% improvement over FAR.

By varying the moving intervals, the performance of the policies is studied. For longer moving intervals t, client's movement is less random and for smaller intervals the movement is rather random. WIND-CP performs better than other policies for small moving intervals where the user moves more randomly in a dense network. Our policy ensures that items in dense and large Voronoi cells are not evicted. Our policy behaves like PAID for large moving intervals (Highways) where the Euclidean distance is almost equal to the network distance. The hit ratio decreases drastically

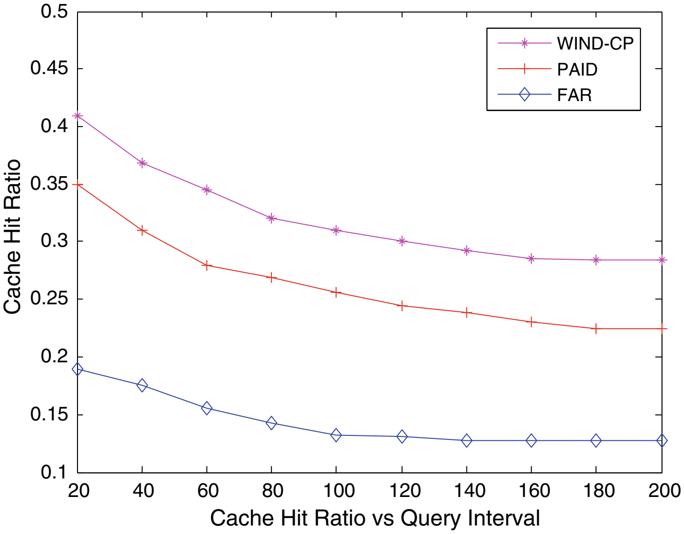


Fig. 3 CHR–query interval

for a relatively longer moving interval because the client has a higher possibility of leaving certain valid regions.

WIND-CP shows a 10% improvement over PAID and 80% improvement over FAR in Fig. 4. The effect of cache size on the performance of replacement policies is shown in Fig. 5. The performance of all policies improves with increased cache

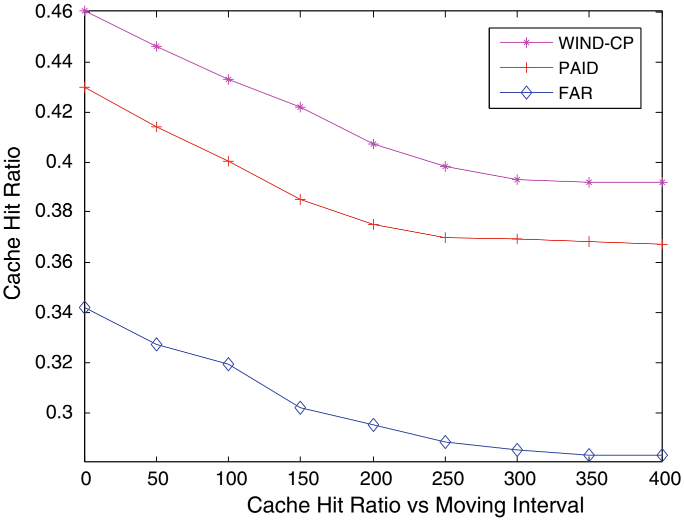


Fig. 4 CHR–moving interval

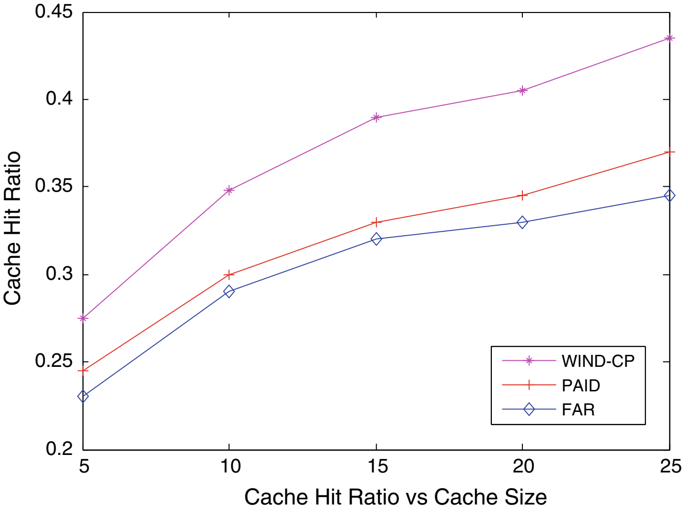


Fig. 5 CHR–cache size

size. WIND-CP policy performs up to 14% better than PAID and 20% better than FAR.

6 Conclusion

A new cache replacement policy WIND-CP that considers network distance and weighted density of the valid scope of a spatial network as important factors and operating time and access frequency as the temporal factors has been proposed. Simulation results have shown that the new policy WIND-CP improves substantially in performance than the existing policies FAR and PAID. Considering the deterministic values of network distance, traffic condition, and operating time of POI object, the overheads of sending a new query and receiving the results for the NNQ have been minimized. In the future, incorporating the random behavior in traffic condition and incorporating pre-fetching or hoarding mechanism into the proposed policy for answering NNQ may be studied.

References

1. Acharya S, Alonso R, Franklin M, Zdonik S (1995) Broadcast disks: data management for asymmetric communications environments. In: Proceeding of ACM SIGMOD conference on management of data, pp 199–210

2. International workshop on database and expert systems applications. IEEE, Vienna, Austria, pp 414–419
3. Hu L, Jing Y, Ku WS, Shahabi C (2012) Enforcing k nearest neighbor query integrity on road networks. In: Proceedings of international conference on advances in geographic information systems. ACM, New York, USA, pp 422–425
4. Ilayaraja N, Jane M (2015) Answering closest-pair nearest neighbor using voronoi diagram for location dependent information system in mobile environment. Int J Appl Eng Res 10(3):7133–7145
5. Ilayaraja N, Jane M, Safar M, Nadarajan R (2016) WARM based data pre-fetching and cache replacement strategies for location dependent information system in wireless environment. Wireless Personal Commun, pp 1811–1842
6. Kolahdouzan M, Cyrus Shahab I (2005) Alternative solutions for continuous k-nearest neighbor queries in spatial network databases. Geo Inform (9):321–341
7. Li Y, Li J, Shu L, Li Q, Li G, Yang F (2014) Searching continuous nearest neighbors in road networks on the air. Inf Syst 42:177–194
8. Papadias D, Zhang J, Mamoulis N, Tao Y (2003) Query processing in spatial network databases. In: Proceedings of VLDB. Berlin, Germany, pp 802–813
9. Safar M (2005) k-Nearest neighbor search in navigation systems. J Mob Inform Syst 1(3):207–224
10. Safar M (2008) Spatial queries in road networks based on PINE. J UCS 14(4):590–611
11. Demiryurek U, Shahabi C (2012) Indexing network voronoi diagrams. Database systems for advanced applications. LNCS, vol. 7238. Springer Berlin Heidelberg, pp 526–543
12. Lee DL, Lee WC, Xu J, Zheng B (2002) Data management in location-dependent information services. IEEE Pervasive Comput 1(3):65–72
13. Nutanong S, Tanin E, Ali ME, Kulik L (2010) Local network voronoi diagrams. In: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. ACM, New York, USA, pp 109–118
14. Okabe B, Boots K, Sugihara SN, Chiu (2000) Spatial Tessellation: Concepts and Applications of Voronoi Diagrams. 2nd edition, John Wiley and Sons Ltd, Chichester (2000).
15. Cheverst K, Davies N, Mitchell K, Friday A (2000) Experiences of developing and deploying a context-aware tourist guide: the GUIDE project. In: Proceedings of conference on mobile computing and networking. ACM, Boston,USA, pp 20–31
16. Dunham MH, Kumar V Location dependent data and its management in mobile databases. In: Proceedings of ninth
17. O'Neil E, O'Neil P (1993) The LRU-k page replacement algorithm for database disk buffering. ACM SIGMOD 22(2):296–306
18. Jane MM, Nouh Y, Nadarajan R, Safar M (2008) Network distance based cache replacement policy for location-dependent data in mobile environment. In: Proceedings of mobile data management workshops. IEEE, Beijing, China, pp 177–181
19. Kolahdouzan M, Cyrus Shahabi (2004) Voronoi-based k-nearest neighbor search for spatial network databases. In:Proceedings of the 30th VLDB conference. Toronto, Canada, pp 840–851
20. Jane MMF, Parameswaran R, Nadarajan R, Safar M (2008) PINE-guided cache replacement policy for location-dependent data in mobile environment. In: Proceedings of the international conference on pervasive technologies related to assistive environments, vol 282, pp 1–5. ACM, New York, USA
21. Shekhar S, Yoo JS (2003) Processing in-route nearest neighbor queries: a comparison of alternative approaches. In: Proceedings of ACM-GIS03, NewOrleans, USA
22. Zhao G, Xuan K, Rahayu W, Taniar D, Safar M, Gavrilova ML, Srinivasan B (2011) Voronoi-based continuous nearest neighbor search in mobile navigation. IEEE Trans Industr Electron 58(6):2247–2257
23. Zheng B, Xu J, Lee DL (2002) Cache invalidation and replacement strategies for location-dependent data in mobile environments. IEEE Trans Comput 51(10):1141–1153

An Experimental Analysis for Credit Card Fraud Detection with Imbalanced and Machine Learning Techniques

G. Anirudh and Upasana Talukdar

Abstract Credit card fraud is a criminal misdemeanour and causes harm to banks, persons as well as financial institutions. Hence, it is substantial to detect and prevent fraudulent activities. Several significant approaches have been put forward in the literature to detect different types of fraud cases. However, most of the existing approaches face different challenges. Among them, the class imbalance problem is the most common. This problem consists of unequal distribution of observations across the classes. In this study, the ratio fraudulent:non-fraudulent is very small, and this poses a challenge for traditional classification algorithms to detect fraudulent activities. This paper presents a rigorous experimental analysis to tackle the class imbalance problem. The experiments were carried out in three categories: (a) experiments with traditional machine learning algorithms, (b) experiments with ensemble methods, and (c) experiments with traditional machine learning algorithms after imbalanced data pre-processing techniques. The imbalanced data pre-processing techniques include undersampling, oversampling, and a combination of both. The performance of the solutions has been evaluated using five different metrics: $F1$-score, precision, recall, area under precision–recall curve (AUPR), and area under receiver operating characteristic curve (AUROC). The experimental results show improved performance in detecting fraudulent cases.

Keywords Credit card · Fraud · Imbalanced data · Machine learning

1 Introduction

Fraud detection is a criminal offence and concerns a large number of banks as well as financial institutes. Different types of fraud exist as statement fraud, credit card fraud, insurance fraud, etc. Credit card fraud can be defined as stealing the information of a credit card to make unauthorized purchases. It is ranked top among identity thefts and one of the common frauds. Each year, millions of credit cardholders fall victim to fraud that costs the country's economy.

G. Anirudh (✉) · U. Talukdar
Department of Data Science and Analytics, Central University of Rajasthan, Ajmer, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_41

Several challenges exist in detecting credit card fraud cases [1–3]. The class imbalance problem is one of the well-known and most critical challenge. It is defined as having an unequal distribution of the data [4], i.e. the ratio of fraudulent transactions is considerably smaller than the non-fraudulent ones. Such a problem poses a challenge in detecting and extracting fraudulent patterns. Because of the dominance of one class, existing machine learning algorithms may fail to detect fraudulent cases accurately. As a result, illegal transactions will be considered legal, causing severe financial harm and loss to institutions as well as individuals.

There are different ways to balance the distribution of class, and some of the common approaches are the use of cost-sensitive algorithms [5], ensemble methods [5], and re-sampling the data [6]. With the rapid advancement in deep learning, the use of unsupervised methods and sequence models to extract fraudulent cases has become more common [7, 8]. However, the drawback of using deep learning methods is they are computationally expensive.

This paper presents rigorous experimentation with state-of-the-art methods to improve the performance of detection and extraction of credit card fraudulent transactions with comparatively lesser features. The experiments were categorized into three categories: (a) experiments with traditional machine learning algorithms, (b) experiments with ensemble machine learning algorithms and (c) experiments with imbalanced data pre-processing techniques and traditional machine learning algorithms. The performance of the solutions has been evaluated using five different metrics: $F1$-score, precision, recall, area under receiver operating characteristic curve (AUROC), and area under precision–recall curve (AUPR).

2 Materials and Methods

2.1 Data Set Description

In this study, credit card fraud detection data set[1] has been used. The data set presents transactions made in September 2013 by European credit cardholders in two days. It has a total of 284,807 samples with 31 features. Out of 31 features, 28 features are principal components represented by V1, V2, . . ., and V28. Time and Amount have not been transformed into PCA. The outcome is the target variable, takes values 0 and 1, 0 represents non-fraudulent transactions, and 1 represents fraudulent transactions. The non-fraudulent transactions are 284,315, and fraudulent transactions are 492. The data set is highly imbalanced, the percentage of fraudulent cases is 0.172%, and non-fraudulent cases are 99.828%. Figure 1 shows the scatter plot and the bar plot of the credit card fraud data set employed in the study. The yellow dots in the scatter plot represent the fraudulent cases, and the blue dots represent the non-fraudulent cases.

[1] https://www.kaggle.com/mlg-ulb/creditcardfraud.

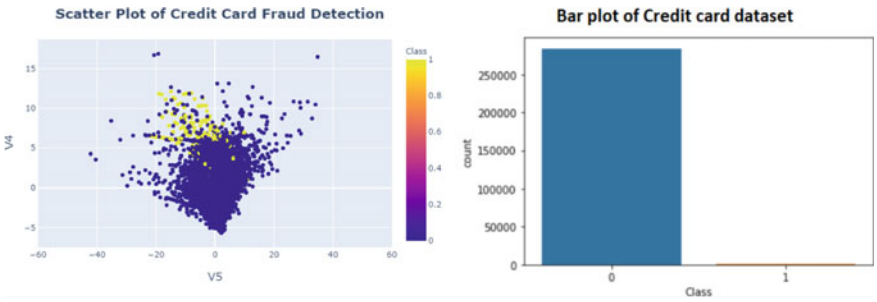


Fig. 1 Scatter plot and bar plot of the credit card fraud data set

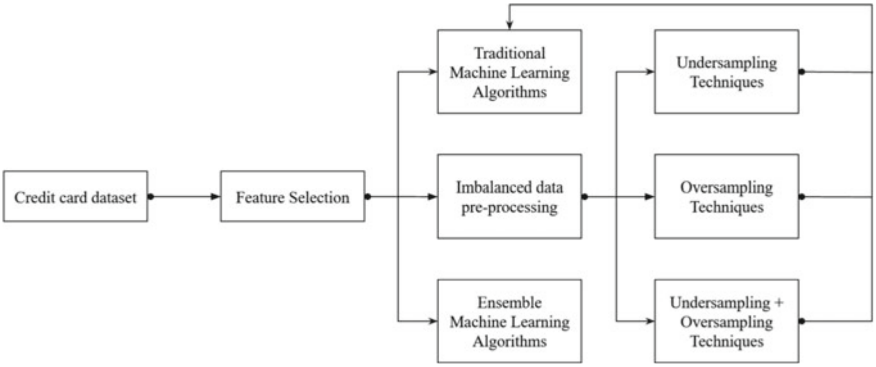

Credit card dataset
Feature Selection
Traditional Machine Learning Algorithms
Imbalanced data pre-processing
Ensemble Machine Learning Algorithms
Undersampling Techniques
Oversampling Techniques
Undersampling + Oversampling Techniques

Fig. 2 Three categories of experimentation carried out in the study

2.2 Methodology

This paper reported rigorous experimentation to tackle the class imbalance problem and adopted various state-of-the-art methodologies to attain better performance in the detection and extraction of fraudulent activities. Figure 2 portrays the methodology adopted in the paper. Feature selection has been employed as a first step to identify the most relevant features. After feature selection, the experiments were conducted in three different categories.

1. **Category 1—Experiments with traditional machine learning algorithms**: In this category, six different supervised classification algorithms are employed on our data set that includes logistic regression (LR), Gaussian Naive Bayes (GNB), support vector machine (SVM), decision tree (DT), K-nearest neighbour (KNN), and deep neural network (DNN).
2. **Category 2—Experiments with ensemble machine learning algorithms**: In this category, six different ensemble machine learning algorithms are applied that includes bagging [9], balanced bagging [10], random forest [11], AdaBoost [12], gradient boosting Classifier [13] and XGBoost [14].

3. **Category 3—Experiments with imbalanced data pre-processing and traditional machine learning algorithms**: Here, to tackle the class imbalance problem different undersampling techniques [15], oversampling techniques [15] and a combination of both have been employed before feeding it to the traditional machine learning algorithms. The undersampling techniques like random undersampling (RU), near miss-1 [16], near miss-2 [16], near miss-3 [16], Tomek links [17], edited nearest neighbours (ENN) [18], and one-sided selection (OSS) [19] are employed. On the other hand, oversampling techniques like random oversampling (RO), synthetic minority oversampling technique (SMOTE) [20], borderline SMOTE-1 [21], borderline SMOTE-2 [21], SVM-SMOTE [22], and adaptive synthetic sampling (ADASYN) [23] are employed in the study. Besides, a combination of different oversampling and undersampling techniques have also been investigated using stratified fivefold cross-validation.

3 Experimental Results and Discussions

3.1 Feature Selection

The data set consists of 31 features, of which 28 features are principal components (PCs) represented by V1, V2,. . ., and V28. The first few PCs have a higher signal-to-noise ratio, and later PCs may be dominated by noise in the data. We experimented with five classification algorithms for choosing the first k ($k < 28$) PCs, using $F1$-score. During our experiments, we noticed the first 7 PCs have shown 52.28% of the variation. LR and SVM achieved the highest $F1$-score for the first 7 PCs while KNN, GNB, and DT achieved the highest $F1$-score for the first 5 PCs (see Table 1). Since the majority of classifiers have achieved the highest $F1$-score for 5 PCs, we retained the first 5 PCs with 41.54% variation. 'Time' and 'Amount' features have not been transformed into PCA. The 'Time' feature denotes the seconds elapsed from the first transaction and every other transaction. The 'Amount' feature denotes the transaction amount. The range of the transaction amount in fraudulent is [0, 2125.87] and non-fraudulent is [0, 25691.16], where 0 is the least transaction in both the cases and maximum being respective upper bounds. We observed 99.78% (283,712 samples) of non-fraudulent transactions lie in the range of [0, 2125]. This makes it difficult for the model to differentiate fraudulent and non-fraudulent samples since most of the transactions lie in the range [0, 2125]. We checked the importance of 'Time' and 'Amount' features with our base models (see Table 1). We observed the majority of the classifiers show poor performance after the inclusion of Time, Amount, and Time + Amount with 5 PCs. This proves our intuition that both the features do not contribute significant information. Hence, we removed them. Our final data set has the first 5 PCs and the outcome variable. We evaluated all the techniques with 0.67:0.33 as train and test split in our study.

Table 1 Performance of traditional machine learning algorithms for the different combination of features

Algorithm	4 PCs	5 PCs	6 PCs	7 PCs	5 PCs + Time	5 PCs + Amount	5 PCs + Time + Amount
LR	0.508	0.518	0.518	0.714	0.543	0.533	0.543
KNN(5)	0.662	0.729	0.726	0.587	0.177	0.642	0.143
GNB	0.129	0.183	0.114	0.111	0.143	0.109	0.132
SVM	0.511	0.521	0.521	0.579	0.206	0.359	0.205
DT	0.578	0.61	0.538	0.538	0.608	0.561	0.593

Table 2 $F1$-score and AUPR for different k value of KNN

k value	$F1$-score	AUPR
1	0.673	0.675
2	0.673	0.734
3	0.725	0.757
4	0.732	0.759
5	0.726	0.761
6	0.729	0.759
7	0.72	0.758
8	0.707	0.753
9	0.693	0.75

3.2 Category 1: Traditional Machine Learning Algorithms

We applied six supervised classification algorithms on our data set that include LR, KNN, GNB, SVM, DT, and DNN, with class weights as 0.1 and 0.9 for class 0 and class 1.

For the KNN, we tried with different values of k ranging from 1 to 9 (shown in Table 2), we observed that $F1$-score is maximum in the case of $k = 4$, and AUPR is maximum in the case of $k = 5$. However, $F1$-scores do not vary much for k in the range of 3–7 and similarly AUPR, so we proceeded with $k = 5$ considering AUPR as an evaluation metric. For DNN, we built architecture with three layers with 5, 8, and 1 units of nodes. We used rectified linear unit (ReLU) for all the layers except for the last layer where we used the sigmoid function. We have used Adam optimizer, the batch size is 32, and the number of epochs is 20.

The performance of different machine learning algorithms has been illustrated in Table 3. It is seen that in terms of AUROC and recall, DNN was giving better performance, while in terms of AUPR, $F1$-score, and precision, KNN (5) performed best. Based on $F1$-score, we observed all the models except GNB was giving better performance. For further analysis, we did not consider GNB because of poor performance through most of the metrics. Also, implementing DNN after applying

Table 3 Performance of traditional machine learning algorithms

Algorithm	AUROC	AUPR	$F1$-score	Precision	Recall
Logistic regression (LR)	0.953	0.513	0.518	0.48	0.56
K-nearest neighbour (KNN) (5)	0.883	0.759	0.726	0.92	0.6
Gaussian Naive Bayes (GNB)	0.941	0.114	0.183	0.06	0.55
Support vector machine (SVM)	0.946	0.503	0.521	0.52	0.52
Decision tree (DT)	0.794	0.613	0.611	0.64	0.59
Deep neural networks (DNN)	0.968	0.627	0.54	0.44	0.69

Table 4 Performance of ensemble methods

Algorithm	AUROC	AUPR	$F1$-score	Precision	Recall
Bagging	0.777	0.664	0.525	0.65	0.44
Balanced bagging (BB)	0.786	0.681	0.616	0.63	0.6
Random forest (RF)	0.803	0.717	0.583	0.67	0.52
AdaBoost	0.801	0.7	0.594	0.7	0.52
Gradient boosting classifier (GBC)	0.83	0.745	0.562	0.69	0.47
XGBoost	0.816	0.728	0.583	0.67	0.52

the appropriate pre-processing technique is a tedious task, consumes more time, and computation power than LR, KNN, SVM, and DT. Hence, we did not consider DNN for further analysis.

3.3 Category 2: Ensemble Machine Learning Algorithms

To evaluate the performance of ensemble methods on imbalanced data, this study includes experiments with six different ensemble algorithms that include bagging, balanced bagging (BB), random forest (RF), AdaBoost, gradient boosting classifier (GBC), and XGBoost. The results are shown in Table 4 in terms of all five evaluation metrics. It is seen from the results that gradient boosting classifier performed best in terms of AUROC and AUPR. Balanced bagging performed best in terms of $F1$-score and recall, similarly AdaBoost in terms of precision. XGBoost attains better performance in terms of all the evaluation metrics.

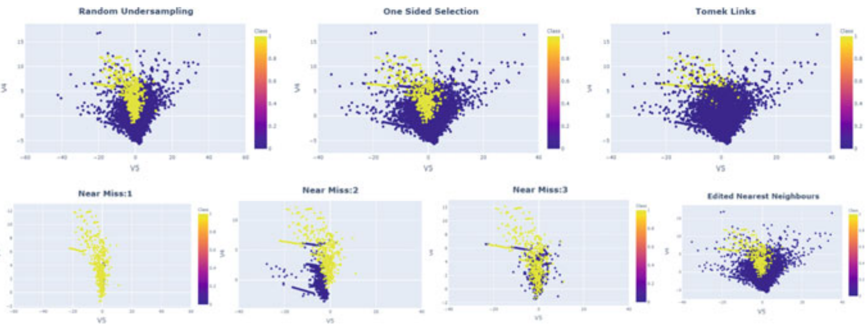


Fig. 3 Scatter plot of the credit card data set with different undersampling techniques

3.4 Category 3: Imbalanced Data Pre-processing and Traditional Machine Learning Algorithms

Keeping in view the aim of the paper to tackle the class imbalance problem with better results, this category of experimentation includes the amalgamation of the imbalanced data pre-processing and the traditional machine learning methods. Different undersampling and oversampling techniques are examined to pre-process the data. The undersampling techniques like random undersampling, near miss-1, near miss-2, near miss-3, Tomek links, edited nearest neighbours, and one-sided selection are employed to down-sample the data. On the other hand, oversampling techniques like random oversampling, SMOTE, borderline SMOTE-1, borderline SMOTE-2, SVM-SMOTE, and ADASYN are applied in the study to up-sample the data. Besides, a combination of different oversampling, and undersampling techniques has also been investigated using stratified fivefold cross-validation. Figure 3 shows the scatter plot of the credit card fraud data set employed after employing the different undersampling methods, Fig. 4 portrays the scatter plot after applying different oversampling methods, while Fig. 5 shows the scatter plot after employing combination methods using two features (V4 and V5). The yellow dots in the scatter plot represent the fraud cases, and the blue dots represent the non-fraud cases. It is seen from the figure that the ratio of fraudulent and non-fraudulent cases has improved compared to Fig. 1. This is believed to improve the performance of machine learning algorithms.

In undersampling, the non-fraudulent cases are down-sampled to the ratio of fraudulent cases, and the data set after undersampling results in 984 samples as the fraudulent cases in our data are 492. In oversampling, the fraudulent cases are up-sampled to the ratio of non-fraudulent cases, and the data set after oversampling results in 568,630 samples as the non-fraudulent cases in our data set are 284,315. Similarly, in combination methods, the data set is up-sampled first using oversampling techniques, and then transformed data is down-sampled using undersampling techniques. So, the number of samples will drastically increase in both cases. The combination techniques have resulted in different sample sizes for different meth-

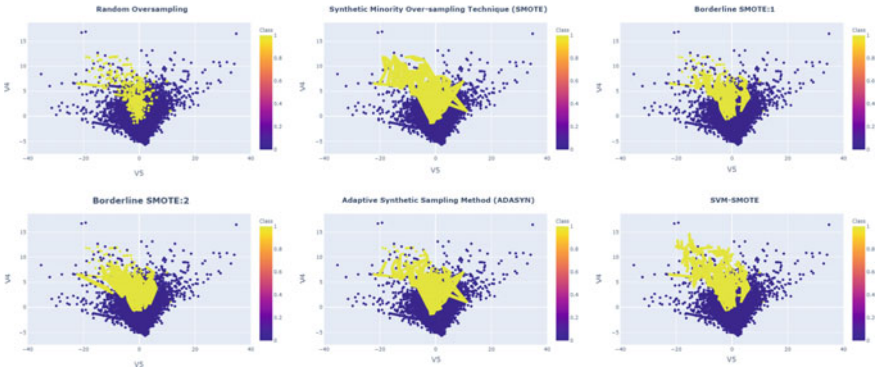


Fig. 4 Scatter plot of the credit card fraud data set after applying oversampling techniques

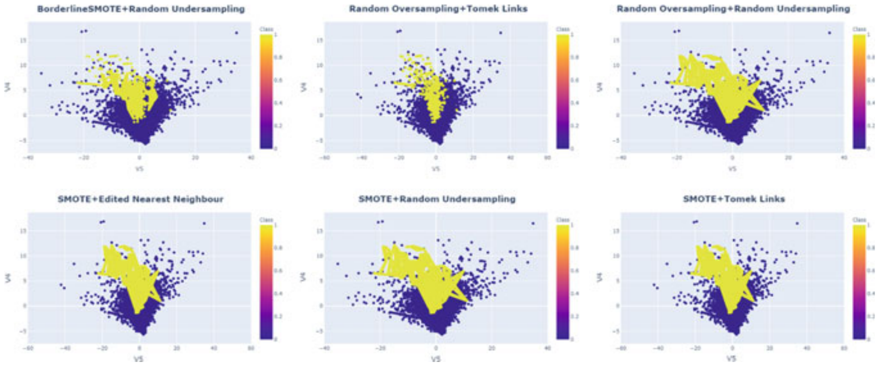


Fig. 5 Scatter plot of different combinations of oversampling and undersampling techniques

ods like 566,365 using SMOTE + ENN, 568,628 using SMOTE + Tomek links, and 568,630 using remaining methods. The pre-processed data using undersampling is then classified using LR, KNN, SVM, and DT. While the data is pre-processed using oversampling, the combination methods are classified using LR, KNN, and DT. As the literature has reported, most machine learning algorithms fail to perform on large data sets; though they show adequate performance, they consume more time and computation power to run the algorithm. SVM is one such algorithm that is not suitable for large data sets [24]. Hence, for all the oversampling techniques and combination methods, we dropped SVM.

As discussed in Sect. 3.2, GNB and DNN are not included in the study. The performance of the aforesaid machine learning algorithms after applying undersampling, oversampling, and the combination of both the techniques is presented in Tables 5, 6, and 7.

It is seen from the results that the employment of imbalanced data pre-processing methods improves the performance of machine learning algorithms. From Tables 6 and 7, we observe that for most of the methods using different evaluation metrics, we

Table 5 Performance of different machine learning methods with undersampling techniques

Undersampling methods	ML algorithm	AUROC	AUPR	$F1$-score	Precision	Recall
Random undersampling	LR	0.969	0.974	0.911	0.93	0.89
	KNN (5)	0.958	0.969	0.917	0.94	0.9
	SVM	0.969	0.975	0.915	0.93	0.9
	DT	0.917	0.94	0.918	0.92	0.92
Near miss-1 (Neighbour = 3)	LR	0.988	0.991	0.957	0.99	0.93
	KNN (5)	0.985	0.989	0.94	0.98	0.9
	SVM	0.981	0.988	0.953	0.99	0.92
	DT	0.966	0.977	0.967	0.98	0.96
Near miss-2 (Neighbour = 3)	LR	0.958	0.958	0.912	0.92	0.91
	KNN (5)	0.997	0.996	0.988	0.99	0.99
	SVM	0.949	0.957	0.9	0.91	0.89
	DT	0.963	0.972	0.964	0.96	0.97
Near miss-3 (Neighbour = 3)	LR	0.742	0.759	0.653	0.72	0.6
	KNN (5)	0.767	0.836	0.73	0.77	0.69
	SVM	0.743	0.767	0.637	0.75	0.55
	DT	0.696	0.777	0.693	0.71	0.67
Tomek links	LR	0.954	0.646	0.618	0.89	0.47
	KNN (5)	0.89	0.798	0.776	0.64	0.99
	SVM	0.939	0.618	0.566	0.86	0.42
	DT	0.811	0.618	0.617	0.61	0.62
Edited nearest neighbour (Neighbours = 3)	LR	0.955	0.691	0.672	0.89	0.54
	KNN (5)	0.906	0.854	0.815	0.99	0.69
	SVM	0.945	0.691	0.633	0.9	0.49
	DT	0.845	0.694	0.692	0.69	0.69
One-sided selection	LR	0.968	0.681	0.634	0.9	0.49
	KNN (5)	0.895	0.792	0.764	0.94	0.64
	SVM	0.948	0.689	0.635	0.89	0.49
	DT	0.834	0.672	0.671	0.67	0.67

were achieving 100% results. One must note that the credit card data set is extremely imbalanced, and the fraudulent samples have increased nearly to 50% from 0.172% after pre-processing. Naturally, with a balanced increase in both cases, any algorithm will tend to perform as accurately as possible.

Table 6 Performance of different machine learning methods with oversampling techniques

Oversampling methods	ML algorithm	AUROC	AUPR	$F1$-score	Precision	Recall
Random over-sampling	LR	0.964	0.97	0.915	0.94	0.89
	KNN(5)	1	1	0.999	1	1
	DT	1	1	1	1	1
SMOTE	LR	0.968	0.972	0.919	0.9	0.94
	KNN(5)	0.998	0.998	0.994	0.99	1
	DT	0.993	0.994	0.993	0.99	0.99
Borderline SMOTE-1	LR	0.991	0.989	0.967	0.96	0.98
	KNN(5)	1	1	0.99	1	1
	DT	0.99	0.99	0.99	1	1
Borderline SMOTE-2	LR	0.986	0.983	0.949	0.95	0.95
	KNN(5)	0.99	0.99	0.997	1	1
	DT	0.996	0.997	0.996	1	1
SVM-SMOTE	LR	0.995	0.994	0.968	0.97	0.97
	KNN(5)	1	1	0.99	1	1
	DT	0.999	0.999	0.999	1	1
ADASYN	LR	0.932	0.934	0.862	0.9	0.83
	KNN(5)	0.998	0.998	0.995	0.99	1
	DT	0.995	0.996	0.995	0.99	1

4 Conclusion and Future Work

Credit card fraud is a serious business problem that can cause a huge loss, to both individual and financial institutes. Henceforth, detection and prevention of fraudulent activities hold promise. The class imbalance problem poses a challenge in extracting fraud cases as the ratio of fraud to legitimate cases is usually small. In such cases, fraud transactions may be considered legitimate which poses a huge threat to financial institutions as well as individuals. The main aim of this paper was to compare different machine learning algorithms for the detection of fraudulent transactions dealing with the class imbalance problem. Hence, a comparison was made, and it was established that imbalanced data pre-processing improves the performance of traditional machine learning algorithms in detecting fraud cases. The performance was evaluated using different evaluation metrics, such as AUROC, AUPR, $F1$-score, recall, and precision. Feature selection has played a vital role in achieving significant results.

Future work would include investigation with different unsupervised methods like clustering and generative adversarial networks (GANs). Supervised methods like hybrid models and semi-supervised methods are part of ongoing research.

Table 7 Performance of different machine learning methods with the combination of undersampling and oversampling techniques

Methods	ML algorithm	AUROC	AUPR	$F1$-score	Precision	Recall
SMOTE + RU	LR	0.967	0.971	0.918	0.92	0.92
	KNN(5)	0.998	0.998	0.996	0.99	1
	DT	0.995	0.996	0.995	0.99	1
SMOTE + RU + Stratified fivefold	LR	0.9202	0.9717	0.9183	0.9406	0.897
	KNN(5)	0.9958	0.9983	0.9958	0.9917	1
	DT	0.9957	0.9964	0.9958	0.9942	0.9974
RO + RU	LR	0.964	0.97	0.915	0.94	0.89
	KNN(5)	1	1	1	1	1
	DT	1	1	1	1	1
RO +RU + Stratified fivefold	LR	0.9202	0.9717	0.9183	0.9406	0.897
	KNN(5)	0.9958	0.9983	0.9958	0.9917	1
	DT	0.9958	0.9965	0.9957	0.9943	0.9973
Borderline SMOTE + RU	LR	0.992	0.99	0.968	0.96	0.98
	KNN(5)	1	1	0.99	1	1
	DT	0.999	0.999	0.999	1	1
Borderline SMOTE + RU + Stratified fivefold	LR	0.9202	0.9717	0.9183	0.9406	0.897
	KNN(5)	0.9958	0.9983	0.9958	0.9917	1
	DT	0.9958	0.9965	0.9957	0.9943	0.9974
RO + Tomek links	LR	0.965	0.971	0.914	0.94	0.89
	KNN(5)	1	1	0.99	1	1
	DT	1	1	1	1	1
RO + Tomek links+ Stratified fivefold	LR	0.9202	0.9717	0.9183	0.9406	0.897
	KNN(5)	0.9958	0.9983	0.9958	0.9917	1
	DT	0.9957	0.9964	0.9958	0.9942	0.9973
SMOTE + Tomek links	LR	0.966	0.972	0.918	0.94	0.9
	KNN(5)	0.998	0.998	0.996	0.99	1
	DT	0.995	0.996	0.995	0.99	1
SMOTE + Tomek links+ Stratified fivefold	LR	0.9202	0.9717	0.9183	0.9406	0.897
	KNN(5)	0.9958	0.9983	0.9958	0.9917	1
	DT	0.9957	0.9964	0.9957	0.9941	0.9974
SMOTE + ENN	LR	0.967	0.972	0.919	0.94	0.9
	KNN(5)	1	1	0.99	1	1
	DT	0.996	0.997	0.996	0.99	1
SMOTE + ENN + Stratified fivefold	LR	0.9202	0.9717	0.9183	0.9406	0.897
	KNN(5)	0.9958	0.9983	0.9958	0.9917	1
	DT	0.9958	0.9964	0.9958	0.9942	0.9973

References

1. Richhariya P, Singh PK (2014) Evaluating and emerging payment card fraud challenges and resolution. Int J Comput Appl 107(14):5–10
2. Bhattacharyya S, Jha S, Tharakunnel K, Westland JC (2011) Data mining for credit card fraud: a comparative study. Decis Support Syst 50(3):602–613
3. Dal Pozzolo A, Caelen O, Borgne Y-AL, Waterschoot S, Bontempi G (2014) Learned lessons in credit card fraud detection from a practitioner perspective. Expert Syst Appl 41(10):4915–4928
4. Phua C, Alahakoon D, Lee V (2004) Minority report in fraud detection: classification of skewed data. ACM SIGKDD Explor Newslett 6(1):50–59
5. Sisodia DS, Reddy NK, Bhandari S (2017) Performance evaluation of class balancing techniques for credit card fraud detection. In: 2017 IEEE international conference on power, control, signals and instrumentation engineering (ICPCSI). IEEE, pp 2747–2752
6. Baabdullah T, Alzahrani A, Rawat DB (2020) On the comparative study of prediction accuracy for credit card fraud detection with imbalanced classifications. In: 2020 spring simulation conference (SpringSim). IEEE, pp 1–12
7. Al-Shabi MA (2019) Credit card fraud detection using autoencoder model in unbalanced datasets. J Adv Math Comput Sci 1–16
8. Pumsirirat A, Yan L (2018) Credit card fraud detection using deep learning based on autoencoder and restricted Boltzmann machine. Int J Adv Comput Sci Appl 9(1):18–25
9. Bühlmann P, Yu B (2002) Analyzing bagging. Ann Stat 30(4):927–961
10. Hido S, Kashima H, Takahashi Y (2009) Roughly balanced bagging for imbalanced data. Stat Anal Data Min ASA Data Sci J 2(5–6):412–426
11. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
12. Hastie T, Rosset S, Zhu J, Zou H (2009) Multi-class adaboost. Stat Interface 2(3):349–360
13. Natekin A, Knoll A (2013) Gradient boosting machines, a tutorial. Front Neurorobotics 7:21
14. Chen T, He T, Benesty M, Khotilovich V, Tang Y (2015) Xgboost: extreme gradient boosting. R package version 0.4-2, 1–4
15. Yap BW, Abd Rani K, Abd Rahman HA, Fong S, Khairudin Z, Abdullah NN (2014) An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In: Proceedings of the first international conference on advanced data and information engineering (DaEng-2013). Springer, Singapore, pp 13–22
16. Zhang J, Mani I (2003) KNN approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of the ICML'2003 workshop on learning from imbalanced datasets
17. Tomek I (1976) Two modifications of CNN. IEEE Trans Syst Man Commun 6:769–772
18. Laurikkala J (2001) Improving identification of difficult small classes by balancing class distribution. In: AIME'01: proceedings of the eighth conference on AI in medicine in Europe, pp 63–66
19. Kubat M, Matwin S (1997) Addressing the course of imbalanced training sets: onesided selection. In: ICML'97: proceeding of the fourteenth international conference on machine learning, pp 179–186
20. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority oversampling technique. J Artif Intell Res 16:321–357
21. Han H, Wang WY, Mao BH (2005) Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: International conference on intelligent computing. Springer, Berlin, pp 878–887
22. Tang Y, Zhang YQ, Chawla NV, Krasser S (2008) SVMs modeling for highly imbalanced classification. IEEE Trans Syst Man Cybern Part B (Cybernetics) 39(1):281–288
23. He H, Bai Y, Garcia EA, Li S (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). IEEE, pp 1322–1328
24. Cervantes J, Li X, Yu W, Li K (2008) Support vector machine classification for large data sets via minimum enclosing ball clustering. Neurocomputing 71(4–6):611–9

Outlier Detection Techniques: A Comparative Study

Chiranjit Das, Aditya Dubey, and Akhtar Rasool

Abstract With the recently rising technologies and numerous applications, the necessity of outlier detection is increasing drastically. Currently, a major variant of outlier detection techniques is witnessed. These techniques played a crucial role in the advancement of fields like medical health, MasterCard fraud, and intrusion detection. However, it is a significant work to spot abnormal behaviours or patterns out from sophisticated data. This paper provides a summary of the outlier detection strategies for the high-dimensional dataset and offers a comprehensive understanding of all basic techniques of outlier detection. This paper provides a comprehensive summary of the ongoing work on anomaly detection techniques, particularly with high-dimensional datasets and data with mixed attributes. The detection of outliers from the given dataset with anomalous data is meaningful work in the area of big data as the data is increasing exponentially every year. Specifically, this paper discusses the current advancement in the field of anomaly detection methods and simultaneously discusses the strengths and limitations of each outlier detection method.

Keywords Data mining · Machine learning · Outlier detection · Time-series data

1 Introduction

The term big data is a very vital topic [1] as the number of data is created exponentially every year. The concept of big data is particularly utilized within the industry and in terms of technology when an institution requires to manipulate a massive volume of data and a place to store them. The term is alleged to originate from web search companies that wanted to handle very largely distributed kinds of loosely structured data.

Anomaly analysis is remarkably interesting and a huge exploration field. It is also equally important in data processing and machine learning prospect [2]. The key objective is to interpret those sections from the dataset whose nature and design do not adjust to conventional results. The surprising nature of the data is instances

C. Das (✉) · A. Dubey · A. Rasool
Department of Computer Science and Engineering, MANIT, Bhopal, MP, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_42

whose characteristics are considered different from those of the rest of the given data and those that are irrelevant to that dataset, known as an anomaly or an outlier. However, there is no commonly valid or any formal definition of anomaly or outlier detection. From the past research and advancement of knowledge, an outlier can also be described as an associate outlier, an unrelatable data, and an unbeatable unit betting on specific application eventualities. Identifying fascinating patterns is extremely important to several domains, like higher reasoning processes, business automation, and data processing. For example, a comprehendible anomalous network communication could indicate that an automatic data processing system is confronted by some hacker, associate abnormalities dealing with ATM card could signify that some unauthorized usage and any sudden geological abnormalities in nature may be an indicator to any natural hazards like earthquake or tsunami, etc. Because of its real-time nature, anomaly detection techniques have a high range of applications, for example, various medical health facilities, ATM card scams, network-related issues, and public education-related issues. With the advancement of the latest technologies, data generated from real-world circumstances are transforming into large and large, not quality-wise but typically in higher dimensionality. Due to the large dimensionality of data objects, the data are nearly equidistant from each other.

Time-series data is any data that is linked to time (real time) [3]. For example, the stock of a company in a day may be real-time data or time-series data based on that day. Numerous applications like demand estimation and sales predicting may be characteristic time-series anomaly detection problems which might be solved by processes like Holt winters. Anomaly detection helps us in fixing future needs by estimating those concerning present data [4]. Once predicted that data value will be used to detect anomalies by relating them with actual. Table 1 presents different techniques with a brief description, pros and cons. There are mainly four primary categories of outlier's detection analysis and their comparison table is illustrated in Table 1:

1. Neighbourhood-based model
2. Subspace-based model
3. Ensemble-based model
4. Mixed-type model.

Currently, there are several powerful and effective detection algorithms developed to improve the scenario regarding these issues [5]. Moreover, these can be characterized into four major parts, that is, neighbourhood-based (e.g. K-nearest neighbour, local outlier factor (LOF)) [6], subspace-based (e.g. out-ranking), ensembled-based (e.g. feature bagging) [7, 8], and mixed-type methods (e.g. outlier detection for mixed attribute datasets (ODMAD), link-based outlier and anomaly detection (LOADED)) [9, 10]. The neighbour-based outlier detection approach primarily defines the neighbourhood of the provided itemset to check out the intensity of its neighbourhood and their corresponding distances. The subspace-based detection model deals with abnormality points by filtering the values in a subset of the dimensions that might be offered for anomaly analysis. Unlike the other analysis approaches, ensemble-based

Table 1 Comparisons between different detection models

Categories	Example	Strengths	Limitations
Neighbourhood-based model	Assign the anomaly score of the knowledge instance relative to their neighbours. Some of the examples are KNN, LOF, RBDA, COF, etc	1 The nature of the methods is unsupervised 2 These are completely data-driven	1 Suffers when it comes to high-dimensional data 2 Complexity in the testing phase is significantly high
Subspace-based model	The values in a subset of the dimensions might be offered for anomaly analysis. Some of the classic examples are SOD, OR, etc	1 Highly efficient 2 Go effective in many cases	1 Very hard to identify the relevant subspace for outlier detection
Ensembled-based model	This approach integrates the result of different existing approaches, compares them, and archives a consensus. Some typical examples are FB, HiSC, etc	1 Highly reliable 2 Less compassionate	1 Efficiency is very less 2 Difficult to opt for the correct metadata
Mixed-type model	This model is used to construct a cohesive model for distinctive data variants or take each type of data independently. It is represented by LOADED, ODMAD, etc	1 Capacity to handle mixed data types 2 Nearly accurate	1 Related structure or characteristic is difficult to gather 2 The complexity of the approach is high

models integrate the result of different existing approaches, compare between them, and archive a consensus.

2 Detection Techniques

2.1 Neighbourhood-Based Techniques

The main idea behind these detection techniques is to identify the anomalies by analysing the neighbourhood information of the data. Another approach is to define the LOF as the calculation of the corresponding outlier degree, where the outlier score is calculated by the relative distance between the neighbourhood. Based on techniques like LOP, every data object gives an outlier probability or commonly called a score, which can be easily interpreted and can also be equated over the whole dataset.

2.1.1 K-Nearest Neighbour (kNN)

In the field of outlier detection, the kNN algorithm is a very basic approach used to classify the data based on the nearest training examples within the feature space [11]. The kNN could be a sort of distance-based anomaly detection technique. Define the kNN outlier technique by a point O (a, b), which is the distance between its k-nearest neighbour $O_k(a_k, b_k)$. k-distance of O is represented as

$$D_k(O) = \sqrt{(b - b_k) + (a - a_k)} \tag{1}$$

Now, the maximum value of D_k can be defined as $\boldsymbol{n}$ data object, and the highest $\boldsymbol{n}$ points with the largest value of D_k magnitude are identified as the anomaly. The two approaches to calculate the anomaly are discussed below.

Unsupervised kNN Approach. The unsupervised kNN approach shown in Fig. 1 describes that in the first step datasets with (a, b) coordinates are given as an input, followed by defining the value of k [12]. A k-distance histogram is made by computing all the k-distance of every data object. Then the range of the k-distance is measured by selecting n largest k-distance. And finally followed by tracking all the relative data points as outliers.

Semi-supervised kNN Approach. The methodology shown in Fig. 2 describes that in the first step datasets with (a, b) coordinates are given as an input, followed by defining the value of k where one anomalous data point is given for training purposes. A k-distance histogram is made by computing all the k-distance of every data object. Apply the previous algorithm on the remaining part dataset, and finally followed by classifying all the relative outliers for each dataset.

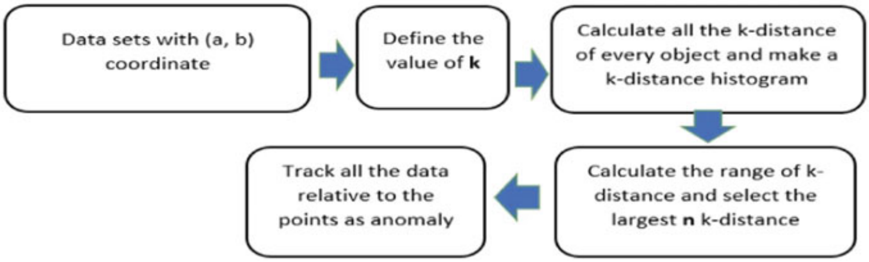

Data sets with (a, b) coordinate
Define the value of **k**
Calculate all the k-distance of every object and make a k-distance histogram
Calculate the range of k-distance and select the largest **n** k-distance
Track all the data relative to the points as anomaly

Fig. 1 Steps of the unsupervised kNN method

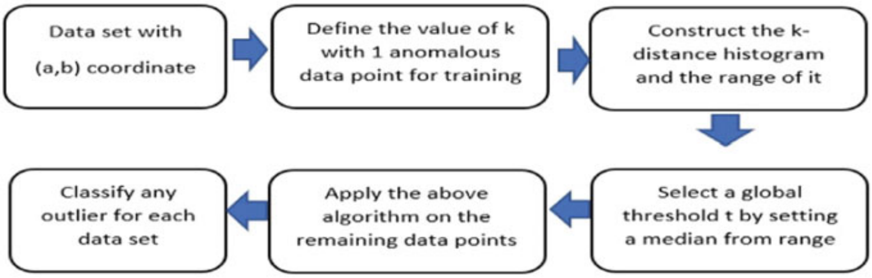

Data set with (a,b) coordinate
Define the value of k with 1 anomalous data point for training
Construct the k-distance histogram and the range of it
Select a global threshold t by setting a median from range
Apply the above algorithm on the remaining data points
Classify any outlier for each data set

Fig. 2 Steps of the semi-supervised kNN method

Typically, the computation time of kNN is given by $O(m \times n)$ if n is the number of training data and m is its dimensionality. Accuracy in terms of semi-supervised approach is higher than the unsupervised approach.

2.1.2 Local Outlier Factor (LOF)

A local anomaly is introduced based on every data point by measuring the degree of outlierness [13]. In this approach, the local outlier is set in the context of some restricted amount of neighbourhood. LOF is a neighbourhood-based outlier detection technique that compares the local distance variation of a data instance to its corresponding neighbours [6]. The local distance variation of a data object is inversely proportional to the mean distance to the k-nearest neighbourhood. A LOF score is set by calculating the ratio between the mean local distance variation of neighbours to the local distance variance of the data objects. This technique is nearly related to density-based clustering [14]. The term local reachability density (LRD) of a data point s is reciprocal of the mean reachability distance which depends upon the MinPts—the nearest neighbour of s [15]. Important parameter MinPts is needed by the LOF algorithm to show the number of closest neighbours which are used in describing the local neighbourhood of a data:

$$\mathrm{LRD}_{\mathrm{MinPts}}(s) = 1 \Bigg/ \frac{\left\{\sum_{p \in N_{\mathrm{MinPts}}(s)} \cdot \mathrm{reach} - \mathrm{distance}_{\mathrm{MinPts}}(s,\ p)\right\}}{N_{\mathrm{MinPts}}(s)} \tag{2}$$

where s and p denote the two data points. Let k-distance(s) be the distance of the data to the kth nearest neighbour. The reach-dist is required to calculate another new idea—the LRD. We denote the set of k nearest neighbours as $N_k(s)$. This distance is known as reachability distance and is denoted by

$$\mathrm{reach} - \mathrm{distance}_{\mathrm{MinPts}}(s, p) = \max\{k - \mathrm{distance}\,(p), (s, p)\} \\ \{\backslash\mathrm{displaystyle}\ N\_\{k\}(A)\} \tag{3}$$

The reachability distance of the data from s to p is the true distance between two data points. The LOF is defined by

$$\mathrm{LOF}_{\mathrm{MinPts}}(s) = \frac{\left\{\sum_{p \in N_{\mathrm{MinPts}}(s)} \cdot \frac{\mathrm{LRD}_{\mathrm{MinPts}(p)}}{\mathrm{LRD}_{\mathrm{MinPts}}(s)}\right\}}{|N_{\mathrm{MinPts}}(s)|} \tag{4}$$

The time complexity of this approach is an order of n^2, where n is the number of training data considered. This approach is more accurate for small datasets. With a slight increase in the size of the dataset, efficiency drops sharply.

2.1.3 Rank-Based Detection Algorithm (RBDA)

Rank-based detection algorithm (RBDA) is an algorithm that helps to track the outlier based on common neighbours adjacent to a data instance and its neighbours [16]. To know about mutual adjacency more precisely, let us assume a data instance $o \epsilon$ D and let $p \epsilon$ $\mathrm{N_k(o)}$. $N_k(o)$ denotes a set of data nearer to o. In other words, assume a p which is closer to instance o, as it is the nearest neighbour of o, and if the above assumption is true then it can be said that o and p are not outliers.

Algorithm 1: RBDA technique.

Step 1: For $o \epsilon$ D and $p \epsilon$ $\mathrm{N_k(o)}$, compute the rank of o with all its neighbourhood of p and assume the rank be r_p (o).
Step 2: Outlierness of o is given by $O_k(o)$ and can be represented as

$$O_k(o) = \frac{\sum_{p \in N_k(o)} r_p(o)}{|N_k(o)|} \tag{5}$$

And for o to be considered as an outlier $O_k(o)$ must be 'large'.
Step 3: To quantify the 'largeness', assume D1 $= \{o \in D|\mathrm{O_k(o)} \leq \mathrm{O_{max}}\}$. If D_1 is 75% of the D, then D_{max} is chosen as 'large'.

And to normalize the $O_k(o)$.

$$X_k(o) = \frac{1}{S_k}\left(O_k(o) - O_k'\right) \tag{6}$$

where

$$O_k' = \frac{1}{|D_1|}\sum_{o \in D} O_k(o) \tag{7}$$

$$S_k^2 = \frac{1}{|D_1| - 1}\sum_{o \in D}(O_k(o) - O_k')^2 \tag{8}$$

The data point o is said to be an outlier if the normalized value $X_k(o) \geq 2.5$. The time complexity of the approach is given by order of $(n + n\log(n))$ (where n is the number of training data) and this approach is highly accurate as it uses rank instead of the distance between the object.

2.2 Subspace-Based Model

Anomalous data typically shows abnormal behaviours based on one or more features. The high dimensional can be categorized into several low-dimensional subspaces. The low-dimensional subspaces of the original feature set are been used by high-dimensional anomaly analysis. Zimek et al. stated that data may have a high-dimensional feature set, out of which only some subsets of the features comprise useful data, while others are not that relatable to the datasets [17]. The existence of those irrelevant data may cause the inoperability of the algorithm. Thus, determining anomalies from a subspace of the original feature set seems to be a fascinating and logical task. In a distributed subspace-based approach, one or more low-dimensional and distributed kinds of subspaces are developed from a single high-dimensional feature space of an object [18]. Due to unusual lower density, the data objects fall into those distributed and low-dimensional subspaces. To overcome the time-consumable problem due to whole distributed subspace, Aggarwal et al. described a new algorithm where subspace development is done by considering subspace with the most negative variance coefficient.

2.2.1 Subspace Outlier Degree

Let $S(o)$ be a set of instances used as a reference; for instance,$o$and $p \in D$. Subspace outlier degree (SOD) of data instance o concerning $S(o)$ is given by [19]

$$\text{SOD}_{S(o)}(O) = \frac{\text{distance}(p, \mathcal{H}(\text{S(o)})}{\left|\nu^{S(o)}\right|} \tag{9}$$

Where

$$\text{distance}(p, \mathcal{H}(\text{s})) = \sqrt{\sum_{i=1}^{d} \upsilon_i^S \cdot \left(p_i, \mu_i^S\right)^2} \tag{10}$$

$\mathcal{H}$(s) is the subspace hyperplane. $\mathcal{H}$(S(o)) is extended by $S(o)$ which referred as a reference object for object $o \in D$ and $\nu^{S(o)}$ is denoted as a weighting vector. The mean μ^s and weighting vector ν^s define the subspace hyperplane $S(o)$. Thidistance value (computed through Eq. 9) near to 0 indicates the point o is an outlier. Given an anomaly o, by extending subspace defining the vector $v^{S(o)}$ it can identify a subspace where o is an outlier. This states that the subspace which is perpendicular to the subspace hyperplane of $S(o)$. Additionally, it is possible to derive an average value point in $S(o)$ in a given subspace. In the case of higher dimensionality, the local outlier detection model cannot be used because of high dimensions and the distance cannot be used to differ feature space neatly. Based on the number of common nearest neighbours in an approach, SNN is used to calculate the similarity point. Especially, two parameters are used in the SOD algorithm. The first one is 'k' which deals with the number of nearest neighbours which are considered to calculate the shared closest neighbour's similarities. This is often not a critical parameter as long as it is selected large enough to understand enough points from the equivalent generating mechanism. And the second one is 'l' which specifies the size of the reference set used. This also cannot be selected as small because of the same reason as described above. The computation time of this approach is given by order of (n^2) and its performance increases with increasing size of dimensionality.

2.2.2 Out-Ranking

In this method of outlier detection, a subspace grouping model is computed to calculate the outlier rank in the non-homogenous full-dimensional data. Out-ranking (OR) is an approach that can handle non-homogenous as well as full-dimensional data [20]. It computes the scoring functions to calculate outlier rank to assess the variation of one data point from the other data points, which is the result of subspace clustering analysis. If this model is compared with an outmoded full-dimensional clustering algorithm like DBSCAN, there may not exist any correct result [21]. Instead, the outcome is a set of overlapped clusters that is nearly enclosed to all the data points. In this paper, it is focused on defining the score function because transformation depends upon the subspace clusters. The rank of an outlier is given by computing a score that depends upon their deviation and then they are arranged in ascending order. A novel scoring function is adapted based on the result of the subspace clustering for every data in the database. One positive score value is assigned by the function for each data in the subspace. Scoring function analyses the data based on both the size of the attribute and the size of data objects. In the initial scoring function, the cluster range is $|R|$ and subspace dimensionality is $|D|$. Now the score data object o will be the summation of these two features and can be normalized based on maximal cluster size $\mathrm{c}_{\max}$ and the dimensionality $d_{\max}$.

Size and dimensionality scoring:

$$\mathrm{score}_1(o) = \sum_{o\epsilon(\mathrm{R,D})} \alpha\left(\frac{|R|}{c_{\max}}\right) + (1-\alpha)\left(\frac{|D|}{d_{\max}}\right) \tag{11}$$

And the second scoring function is density expectation scoring, which quantifies the data with high density. That data with high density must be normalized by expected density.

Density expectation scoring:

$$\text{score}_2(o) = \sum_{o\epsilon(\text{C,S})} \widetilde{F}(o) = \sum_{o\epsilon(\text{C,S})} \frac{\emptyset_\text{s}(o)}{E[\emptyset_\text{s}]} \tag{12}$$

where $\tilde{F}(o)$ is the factor by which a data exceeds expectation and that is a sufficient weight for a data score. For any determined density ØD with expectation E[Ø D], $\frac{\emptyset\text{D}}{\text{E}[\emptyset\text{D}]}$ is unbiased dimensionality.

2.2.3 Adaptive Outlierness in Subspace

Muller et al. proposed a method to detect the outlier from a given dataset by computing the rate of deviation based on the concept called the ranking of the data [22]. These rates of data deviation are calculated by the selection of relevant subspace RS(p) of an object p and given by the adaptive neighbourhood in the subspace. To derive a relevant outlier ranking depending upon the outlierness of a high-dimensional projection needs to define an outlierness measure *score* (p, S) which offers an adaptive outlierness degree as the general ranking of an object if the object represents the data point. That ranking mechanism is achieved by combining the data feature from a large feature subset $R \in R(p)$. This adaptive outlier is computed by calculating the adaptive density of data and the deviation of local data. The key issue in the approach is to determine the static neighbourhood. As distances are incremented with an increasing number of features, a static neighbourhood- $(p, R)\{o|dist_S\ (p, o) \leq \upsilon\}$ (where o and p are data points) comes null. Here, the adaptive object density, as well as the local object deviation, is discussed to explain the concept of adaptive outlierness.

Adaptive Object Density. To overcome this common issue of density evaluation in random subspaces, an alternative approach is proposed to compute adaptive outliers applying the concept of an adjustable neighbourhood. By incrementing the neighbourhood distance υ with an increasing number of features, sparse distances of objects are spontaneously adapted by computation of density. Therefore, a variety of subspaces turn out to be similar and outliers calculated by density determination can instinctively adapt the number of features. Based on the variable range $\upsilon(|R|)$ an adaptive neighbourhood is defined as

$$\mathcal{N}(\text{p, R})\{\text{p|dist}_\text{S}(\text{p, o}) \leq \text{u}|\text{R}|\} \tag{13}$$

The idea is to determine a variable range from the common output in subspace estimation.

Local Object Deviation. For quantifying outliers ranking depends on the density deviation of the data. The density of every data must be calculated and correlate them with local mean density (den and dev is density and deviation of the object p). By computing the density concerning mean and standard deviation, the deviation of the object is determined, termed as object deviation [22]. The object deviation is given by

$$\operatorname{dev}(p, R) = \frac{\mu - \operatorname{den}(p, R)}{2 \cdot \sigma} \tag{14}$$

Adaptive Outlierness. The anomalous nature of a data object p has to satisfy two major requirements. First, it should be adaptive to high-dimensional subspaces. Second, adaptive outliers need to deal with object deviation considering a statistical deviation from the mean. The density and variation of data in a subspace R are defined as outliers in that subspace.

$$\operatorname{Score}(p, R) = \begin{cases} \frac{\operatorname{dev}(p,R)}{\operatorname{den}(p,R)}, \text{ if } \operatorname{dev}(p,R) \geq 1 \\ 1, \text{ else} \end{cases} \tag{15}$$

The deviation of an object o concerning mean and standard deviation of the estimated density:

The outlierness is determined by comprising both features, i.e., density and the deviation of each instance. Low density and high deviation, both indicate the outlierness of the object.

2.3 *Ensembled-Based Model*

Ensemble learning is largely explored in the field of machine learning [23]. As it performs much better than other techniques available in this field, it is a furthermore often used technique for anomaly detection. Due to the complexity of data in a low-dimensional subspace, it is impossible to determine all the outliers. Thus, the hypothetical anomalies are determined by ensemble techniques, which are used to compute various machine learning techniques. There are three basic ensembled approaches for anomaly analysis, which are used to summarize the corresponding outlier scores.

2.3.1 Feature Bagging

There exist two different methods for combining these outlier detection approaches. In the general framework, the process for combining different outlier detection approaches yields a sequence of K iterations, and these iterations are also run correspondingly for faster implementation. In each iteration n, the outlier detection regulation is called and conferred with a unique set of characteristic F_i that is used to calculate the distance. The set of characteristics F_n is likely chosen from the actual dataset in such a way that characteristics in the F_n is randomly selected from d/2 and $(d - 1)$ range, where d represents the number of characteristics in the actual dataset [24]. With the selection of N_n features from F_n, the original dataset N_n features are selected arbitrarily. There arise some limitations of combining outlier score vectors which is conceptually somewhat like the idea of meta-search where separate rankings are resultant of separate search engines and then they are merged to construct the result that is highly associated with the given search string.

The output of each anomaly detection approach produces various types of anomaly score vector AS_n. This tends to the probability of every single data as an anomaly of dataset S. Suppose if $\mathrm{AS}_n(i) > \mathrm{AS}_n(j)$ then data x_i has a probability greater than the data point $\mathrm{y_j}$. At the end of K iterations, there exist K number of outlier score vectors where each one is associated with a single detection approach. Then a COMBINE function is responsible to summarize these K outlier vectors AS_n into a final vector $\mathrm{AS}_{\mathrm{final}}$. It has been used to allocate the last and final probability to each data point.

The most infamous approach variants of the combining framework are the Breadth-First approach. The second one is the cumulative sum approach. The general framework is shown below:

Algorithm 2**:** General framework to combine the score:

Step 1: A set S $(x_1, y_1)\ldots(x_i, y_j)$, $x_i \in X^d$ by labels $y_j \in Y^d = \{C, \mathrm{NC}\}$ is given $(x_n$, corresponds to anomalies C and NC, represents the normal class. Dimensionality is represented as d with vector X.
Step 2: Now go for the normalization of dataset S.
Step 3: For $n = 1,2,3,\ldots, K$.

i. Choose the capacity of the features $\subseteq N_t$, uniformly distributed between $\lfloor d - 2 \rfloor$ and $(d-1)$.
ii. Pick, form feature subset F_n, without replacement.
iii. O_n, an outlier detection approach by the feature subset F_n.
iv. AS_n is the output of outlier detection approach $\mathrm{O_n}$.
v. An outlier score factor AS_n is the result of the outlier detection approach.

Step 4: A final anomaly score factor comes after combing the AS_n and output.

2.3.2 High Contrast Subspace (HiCF)

This technique mainly focused on picking the highly dissimilar subspaces from the feature space to detect the outlier based on their density. With this approach, the primary goal is to determine the contrast of each attribute subspace. Thus, here with this approach we only determine the score for the high contrast subspaces. The concept behind the HiCS methodology is choosing subspace with high contrast statistically [25]. This procedure developed a sequence of statistical experiments. Contrast calculation is based on these two processes, i.e., statistical experiment and dependency comparison. It enables the means of quality raking of an outlier to provide the stage for choosing the subspace with high contrast. All the above-mentioned outlier score functions will be affected by the loss of contrast:

$$\text{score}(\vec{a}) \approx \text{score}(\vec{b})\,\forall\,\vec{a},\vec{b} \in \text{Db} \tag{16}$$

where database Db, consisting of objects, a real-valued object vector $\vec{a} = (\vec{a}_1 \ldots \vec{a}_D)$ with D-dimensionality, explains each object. The original data space of all presented features is defined by a set $S = \{1,\ldots, D\}$. A feature subset $s = \{s_1,\ldots, \text{s}_\text{d}\} \subseteq S$ will be known as a D-dimensional subspace projection. This weakness of outlier rankings of the subspace is tackled by evaluating the score calculated for lower-dimensional subspace estimates. They merely limit the gap computation to a picked subspace S. Thus, outlier ranking with a score(x) is often expressed as subspace score score_S ($\vec{a}$). The objective is to mix these score_S ($\vec{a}$) values with many subspaces. The concluding value of outlier ranking is calculated from the combination of those scores:

$$\text{score}(\vec{a}) = \frac{1}{|\text{RS}|}\sum_{S \in \text{RS}} \text{score}_S(\vec{a}) \tag{17}$$

This statistical way of searching method eliminates low contrast attribute subspaces, which create a clear view between the outlier and the normal data. If we summarize the whole approach, it can be noticed that the primary step is to find a high contrast subspace from a set of feature subsets. The biggest problem with this approach is that they are very expensive when it comes to computation.

2.3.3 An Unsupervised Approach to Combine Score of Outlier Detection

The broad techniques to combine the outcome of non-homogenous anomaly detection methods in any unsupervised learning environment are: (1) the Ensemble of Detectors with Correlated Votes (EDCV) and (2) the Ensemble of Detectors with Variability Votes (EDVV) [26]. Mainly EDCV and EDVV are distinguished by the calculation used to estimate the coefficients by comparing the different results of the methods. The similarity measure is calculated in the case of EDCV using the value called the correlation coefficient, while the dissimilarity measure is calculated in the case of EDVV using absolute variation between outputs (MAD). An alternative box plot method is used by both approaches to calculate the amount of outlierness votes that every observation receives from each algorithm. Now, two completely different procedures, i.e., EDCV uses correlation coefficient and EDVV uses absolute variation between outputs are computed to enhance the performance of the individual algorithms over a definite dataset. The similarity contrast measures allocate applicable weights to all of the algorithms of the ensemble, giving extra influence to those procedures whose results are similar to each other.

The correlation coefficient and MAD is used to determine the counts of similarity/dissimilarity for two different approaches in the case of numerical values. Then these counts are considered to compare the output of the different classifiers. The above discussed two approaches are used to combine the different outlier scores but the only difference is in the way they assign the weights to the different algorithms. The approaches are used to calculate the statistical similarity between all other output of outlier detection algorithms. The techniques also compare the input values and assign some results, i.e., assign value 1 for perfectly similar values, 0 to dissimilar values, and -1 for negatively similar values. It is also used to compute the absolute deviation of all outputs. In the case of EDVV, MAD assigns perfectly similar scores with the lowest values and in the case of EDCV, the correlation coefficient assigns similar scores with the highest values.

2.4 Mixed-Type Model

The majority of the algorithm discussed above is only capable of handling the numerical dataset which causes less versatility of the algorithms. The real-world dataset consists of both categorical and numerical values in one dataset. To handle those data the simple and easiest approach is to normalize the numerical and treat them like categorical values. The accuracy of the dataset may get deteriorated.

2.4.1 Link-Based Outlier and Anomaly Detection

It is a single-pass approach to detect the outlier in given datasets that comprise both continuous and discrete attributes. LOADED [27] is an adjustable-based technique, which will be able to compute the result for high accuracy to bound (e.g. interruption detection) the computation times. Tentative results prove the usefulness of this approach over real-world data. LOADED delivers high accuracy detection results and singleton reactor rates, which are much higher as compared to other approaches. LOADED identifies the anomaly-based predefined distance-based approaches. In the context of this approach, the primary focus is on the live real-world datasets.

To describe the approach, let us consider a dataset consisting of both numerical and categorical attributes. A pair of independent data points is said to be linked with one another if they are considered identical. A degree of linkage is determined for each pair of data points. If there exist any common attribute value between a pair of a data point, then they are linked in categorical attribute space. The strength of that link between a pair is given by the number of the shared attribute value. A score for an anomaly is generated by a scoring function that assigns the score to a data point that is inversely correlated to the summation of all the link strengths. The concept of frequent itemset mining is efficient enough to determine this score [28].

A pair of the point is said to be linked to each other in the mixed type of data space if they are also linked in the space of categorical data and if their progressive attributes are stick to the joint distribution shown in the similarity matrix. A data point that violates the above condition is termed as outliers. The strength of this approach is that it can work very well for categorical attributes.

2.4.2 Outlier Detection for Mixed Attribute Datasets

Outlier Detection for Mixed Attribute Datasets (ODMAD) is used, for retrieving abnormalities from data comprising both categorical and continuous attributes [28]. The approach initially determines the relevant outlier score for every data point thinking of irregularity between the explicit values, continual values, and also the associate relation between two ranges within the dataset. This approach is sub-divided into two phases. The first phase is the categorial phase where it has to quantify the irregularities in the categorical quantities and ensemble the values. The second phase is the numeric phase where the whole dataset is further divided into numeric subsets in the context of categorial quantities, to solve the issue of masking. In the initial phase, i.e., categorial phase, the first step is to reset the categorial score by 0 value for each result of the approach. Then the threshold is set which is the lowest acceptable frequency. In the second phase, the value identified in the first phase is kept aside and then it is tried to detect the outliers in the remaining data points. It is famous for producing highly accurate results.

3 Conclusion and Future Work

It has been discussed in this paper that for all application datasets, there is no single universally acceptable outlier detection approach. From the above-described techniques, a great variety of methods exist which cover the complete explanation of statistical, neural, and machine learning approaches for outlier detection techniques. It is nearly impossible to go through all the techniques of outlier detection in a single paper. This paper tries to provide the reader with an impression of covering the range of techniques available in the field of neighbourhood-based, subspace-based, ensembled-based, and mixed-type detection techniques. In outlier detection, the choice of relevant degree formula that is suitable for its dataset in the context of the appropriate distribution model, the correct attribute choice, the quantifiability, and the complexity are very critical. The anomaly detection approaches, which are generally suitable for a comparatively low-dimensional dataset, that help to build the technical foundation for numerous other strategies with high-dimensional data, have also been discussed. Also, outlier detection is a vastly developing field in data analysis and a lot of new methods can quickly be emerging shortly. Based on their emergence, it can be believed that the outlier detection techniques can play a more vital role in numerous sensible applications wherever they will be applied to.

In the near future, we will discuss some detection techniques based on networking. There are some broad research issues like methodologies being used in detection (e.g. distance-based or density-based method), choice of learning schema (e.g. supervised, semi-supervised, or unsupervised) and determination of size and dimensionality (e.g. large or small datasets and high or low dimensionality).

References

1. Dubey A, Rasool A (2021) Efficient technique of microarray missing data imputation using clustering and weighted nearest neighbour. Sci Rep 11: 1–12
2. Dubey A, Rasool A (2019) Data mining based handling missing data. In: Proceeding of the third international conference on I-SMAC (IoT in social, mobile, analytics and cloud) (I-SMAC). Palladam, India, pp 483–489
3. Dubey A, Rasool A (2020) Clustering-Based Hybrid Approach for Multivariate Missing Data Imputation. Int J Adv Comput Sci Appl 11(11): 710–714
4. Zhang J (2013) Advancements of outlier detection: a survey. ICST Trans Scalable Inform Syst 13(1):1–26
5. Xu X, Liu H, Yao M (2019) Recent progress of anomaly detection. Complexity
6. Upadhyaya S, Singh K (2012) Nearest neighbour-based outlier detection techniques. Int J Comput Trends Technol 3(2):299–303
7. Zimek A, Campello RJ, Sander J (2014) Ensembles for unsupervised outlier detection: challenges and research questions a position paper. ACM SIGKDD Explorat Newsl 15(1):11–22
8. Aggarwal C, Sathe S (2015) Theoretical foundations and algorithms for outlier ensembles. ACM SIGKDD Explorat Newsl 17(1):24–47
9. Do K, Tran T, Phung D, Venkatesh S (2016) Outlier detection on mixed-type data: an energy-based approach. In: Proceeding of the international conference on advanced data mining and applications. Springer, Cham, pp 111–125
10. Agrawal A (2009) Local subspace-based outlier detection. In: International conference on contemporary computing. Springer, Berlin, Heidelberg, pp 149–157
11. Dang TT, Ngan HY, Liu W (2015) Distance-based k-nearest neighbours outlier detection method in large-scale traffic data. In: Proceeding of the IEEE international conference on digital signal processing (DSP), pp 507–510
12. Shah P A critical survey on anomaly detection
13. Kriegel HP, Kröger P, Schubert E, Zimek A (2009) LoOP: local outlier probabilities. In: Proceedings of the 18th ACM conference on information and knowledge management, pp 1649–1652
14. Kriegel HP, Kröger P, Sander J, Zimek A (2011) Density-based clustering. Wiley Interdiscip Rev Data Mining Knowl Discov 1(3):231–240
15. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier
16. Huang H, Mehrotra K, Mohan CK (2013) Rank-based outlier detection. J Stat Comput Simul 83(3):518–531
17. Zimek A, Filzmoser P (2018) There and back again: outlier detection between statistical reasoning and data mining algorithms. Wiley Interdiscip Rev Data Mining Knowl Discov 8(6):1280
18. Zhang J, Yu X, Li Y, Zhang S, Xun Y, Qin X (2016) A relevant subspace-based contextual outlier mining algorithm. Knowl-Based Syst 99:1–9
19. Kriegel HP, Kröger P, Schubert E, Zimek A (2009) Outlier detection in axis-parallel subspaces of high dimensional data. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, Berlin, Heidelberg, pp 831–838

20. Muller E, Assent I, Steinhausen U, Seidl T (2008) Outrank ranking outliers in high dimensional data. In: Proceeding of the IEEE 24th international conference on data engineering workshop, pp 600–603
21. Chakraborty S, Nagwani NK Analysis and study of Incremental DBSCAN clustering algorithm. arXiv preprint arXiv,1406.4754.
22. Müller E, Schiffer M, Seidl T (2010) Adaptive outlierness for subspace outlier ranking. In: Proceedings of the 19th ACM international conference on information and knowledge management, pp 1629–1632
23. Zhou Z (2016) Machine learning. Tsinghua University Press, Beijing, pp 53–72
24. Lazarevic A, Kumar V (2005) Feature bagging for outlier detection. In: Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining, pp 157–166
25. Keller F, Muller E, Bohm K (2012) HiCS: high contrast subspaces for density-based outlier ranking. In: Proceeding of the IEEE 28th international conference on data engineering. Washington, DC, pp 1037–1048
26. Pasillas-Díaz JR, Ratté S (2016) An unsupervised approach for combining scores of outlier detection techniques, based on similarity measures. Electr Notes Theor Comput Sci 61(7):329
27. Ghoting A, Otey ME, Parthasarathy S (2004) Loaded: link-based outlier and anomaly detection in evolving data sets. In: Proceeding of the fourth IEEE international conference on data mining (ICDM'04), pp 387–390
28. Moens S, Aksehirli E, Goethals B (2013) Frequent itemset mining for big data. In: IEEE international conference on big data, pp 111–118

Energy-aware Application Scheduling on DVFS-Enabled Edge Computing with Mobile–Edge–Cloud Cooperation

Vishal Deka, Manojit Ghose, and Sukumar Nandi

Abstract With a significant increase in the compute-intensive mobile applications in recent times, the seamless integration of the cloud platform with mobile devices becomes essential. As a result, the mobile cloud computing (MCC) and mobile edge computing (MEC) paradigm become prominent in today's era. At the same time, scheduling user applications in these domains have also gained research attention. But a majority of scheduling policies for the MEC environment consider offloading and execution of tasks only between two computing platforms: (i) mobile and edge or (ii) edge and cloud. In this paper, we consider the scheduling of a set of user applications with partial offloading and execution of tasks in all the three layers of compute stack: the mobile device, edge, and the cloud, to minimize the energy consumption of the edge nodes. Our extensive simulation reveals that the proposed scheduling strategy achieves an energy reduction up to 30% with an execution speedup up to 6% while meeting the delay constraints of the applications.

Keywords Roadside cloud · Mobile edge computing · Energy efficient scheduling · Partial offloading · Host energy consumption

1 Introduction

The development of mobile applications involving compute-intensive tasks such as face recognition, natural language processing (NLP), augmented reality, etc., has increased the demand for processing power on mobile devices (MDs) such as smartphones, wearable devices. Autonomous vehicles with on-board computing units need to compute a number of tasks to ensure the safety of passengers and pedestrians which come with strict latency demands. User applications such as augmented reality are highly latency-sensitive and any delay causes a drop in quality of service. However,

V. Deka (✉) · M. Ghose · S. Nandi
Department of Computer Science and Engineering, Indian Institute of Information Technology Guwahati, Guwahati 781015, India

M. Ghose
e-mail: manojit@iiitg.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_43

powerful processing units come with an increasing energy consumption which is a major concern for MDs and autonomous vehicles being powered by batteries. One popular solution to this problem is Mobile Cloud Computing (MCC) [1, 2] which allows an MD to offload some or all of its compute-intensive tasks to the cloud so that battery power can be saved. This possibly even provides better response time given the more powerful computing units available remotely.

MCC, however, comes with an overhead resulting from the data transmission delay, i.e., uploading the task binaries, dependencies, and input data to the cloud and downloading results after the computation has finished, due to the large geographical distance between cloud data centers and the MD. It might be the case that the data transfer delay outweighs the benefit from computation speedup obtained from executing tasks in the cloud. This is especially the case when autonomous vehicles are involved where the slightest delay might have serious safety concerns. Mobile Edge Computing (MEC) has been introduced to tackle the problem of transmission delay by bringing computation closer to the user. MEC can be realized by installing small clusters of machines along roads that form a roadside cloud [3] that provides remote computing resources. Users are able to communicate with these machines via roadside units (RSUs) using wireless communication technologies like LTE. As these RSUs are closer to the user geographically, the communication latency is low. The low-data transfer delay, however, comes with an associated cost. Unlike cloud data centers, these roadside clouds are not as powerful. Therefore, the decision to offload computation tasks must incorporate the trade-off between computation and communications delays. Moreover, various other real-world problems are also being addressed by the MEC platform [4].

An application is made up of various components, termed as tasks. Depending on the characteristics of each task, it can be chosen to execute locally or remotely. For example, a task requiring input from the user or one that requires the use of a device sensor must be executed locally. On the other hand, a compute-intensive task such as image processing can be executed remotely on more powerful machines (either in the central cloud or roadside cloud or a combination of both), in that case, the user only needs to upload the code and input image to the remote platform and download the computation results. One can choose to offload all the tasks (full-offloading) or only some of them (partial offloading) depending on the nature of tasks and network constraints as data transmission delay can impose a significant overhead. Partial offloading can be achieved by partitioning the application into two sets of components: one to be executed locally and another to be executed remotely [5].

A major issue in edge computing is that the compute nodes in roadside clouds are resource-constrained. Being located outside data centers, their energy consumption requires more attention [6]. This calls for effective task scheduling strategies to minimize the energy consumption in these hosts. The other challenge is to find a way to run such tasks, which are written for mobile operating systems, in remote hosts with desktop operating systems (Linux, Windows, etc.). This can be achieved by means of emulation, virtualization, etc. A number of techniques involving remote procedure calls, virtual machines (VM) have been developed over the years [7]. Accordingly, a number of scheduling strategies are also proposed using VM-based virtualization [8, 9]. However, a full-blown traditional virtual machine comes with considerable overhead in terms of storage and time required to boot up a full operating system. It also incurs a significant amount of resource wastage if not managed properly [10]. Instead, container-based lightweight alternatives are becoming popular in recent time [11]. The proposed scheduling strategy in this work can easily be implemented in real system using these lightweight strategies. Here, we summarize the contributions of our paper as follows:

- We present a scheduling strategy to execute a set of user applications on a MEC environment to minimize the energy consumption of the edge hosts while ensuring their delay constraints.
- We consider all the three layers of the computation stack, that is, mobile device, edge nodes, and centralized cloud server together for partial or full-offloading and execution of user applications.
- We apply a polynomial-time algorithm to efficiently partition the user application graphs and apply DFVS technique to edge host to reduce energy consumption and to forward a few tasks to the central cloud system.

The rest of the paper is organized as follows. In Sect. 2, we present a brief literature review of related works. Section 3 explains the system model, problem formulation, and other essential components of our work. We present the scheduling strategy is Sect. 4. Performance evaluation is presented in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Related Work

The problem of task scheduling and resource allocation in edge computing has been addressed with various objectives in mind latency, the energy consumption of mobile devices/autonomous vehicles, energy consumption of the edge hosts, etc. Some authors have also considered the pricing and revenue of service providers. Quality of service is also considered by a couple of research. In addition to considering different scheduling objectives, task scheduling policies also consider different kinds of application and/or machine behaviors such as homogeneous edge nodes, heterogeneous edge nodes, independent task sets, etc. We briefly present a few important works that are relevant to our proposed work.

For instance, the authors of [12] focus on minimizing the execution delay of tasks after offloading. The cases for single and multiple users have been addressed. A branch and bound method is proposed for the single-user case and a heuristic algorithm for the multi-user case. Zhao et al. in [13] defines an utility maximization problem that takes into account the user's latency requirement and price of remote execution. Based on this, the proposed solution offloads tasks to the cloud or the edge, but only full-offloading is considered. Wang et al. [14] formulated the scheduling problem as a stochastic problem where they scheduled the network resources on C-RAN (cloud radio access network) and computing resources on MEC nodes to maximize the profit of the service provider. Authors in [15] designed a task distribution and computing policy for cache-enabled fog computing networks under low-latency and ultra-reliability constraints. They reported up to 91% reduction in computation latency with respect to the state-of-the-art work. Chiang et al. [16] designed a scheduling policy to minimize average co-task completion time by utilizing the resources efficiently.

A few research also target to jointly optimize latency and energy consumption of UEs. For instance, the authors of Liu and Wang [17] propose a Q-learning-based approach is to find an optimal offloading strategy. In Kuang and Li [18] and Tran and Pompilli [19], the problem of jointly minimizing both latency and user's energy consumption have been formulated as a weighted sum of the two. Iterative algorithms have been proposed in Kuang and Li [18] to solve the problem, while the authors of Tran and Pompilli [19] propose a combination of convex optimization techniques and a heuristic algorithm. These works however do not consider offloading of tasks to both the edge and cloud simultaneously. Dinh et al. [20] considered a scenario of offloading tasks from a single mobile device to multiple edge nodes to reduce energy consumption of MDs and task's execution time.

On the other hand, the following works address the issue of minimizing energy consumption of the MEC hosts. Ning et al. [6] proposes a heuristic algorithm wherein tasks are migrated to hosts where estimated energy consumption is below a threshold. The authors of Ning and Dong [21] propose a deep reinforcement learning-based approach to solve the problem. Authors of Guo et al. [22] consider all three layers of the computation stack to maximize quality-of-service (QoS) of end-users and presents a Lyapunov optimization-based approach. Liang et al. [23] designed a scheduling strategy for DVFS-enabled (dynamic voltage and frequency scaling) heterogeneous edge servers that works in two phases. In the first phase, tasks are offloaded and in the second phase, appropriate frequency is chosen.

In our work, we also target to minimize the energy consumption of the MEC nodes. However, our work is different from the existing literature mainly in the following ways: (i) we consider a set of applications generated by different users together so as to efficiently execute them and (ii) instead of considering full-offloading (as the majority of existing work in literature), we consider partial offloading and execution of user tasks at all three layers: mobile device, edge, cloud while reducing the energy consumption of the edge nodes.

3 System Model and Problem Formulation

Our system model consists of three layers: user, edge, and cloud. This is shown in Fig. 1. The user layer consists of a set of user equipment (UEs) that can be hand-held smart devices, wearable devices like smart glasses, or autonomous vehicles with onboard computation units. These devices are connected to their nearest RSU via

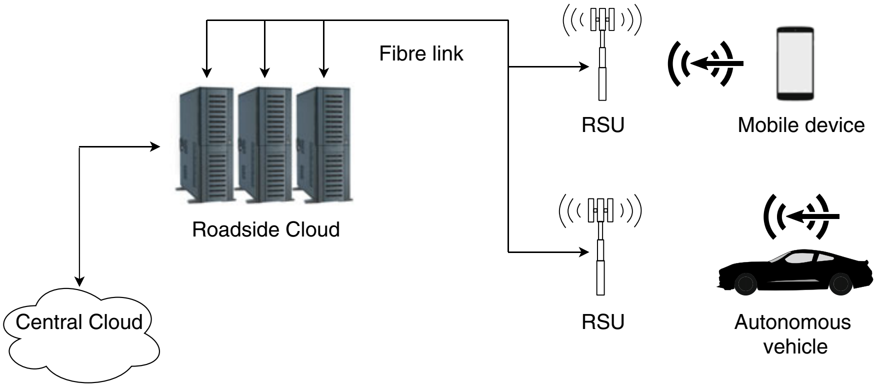


Fig. 1 The system architecture

a wireless network. A set of RSUs is connected to a computation facility by fiber links. This computation facility is termed as roadside cloud or MEC hosts or edge nodes/hosts. The MEC hosts are equipped with computing resources to provide lower computation latency, compared to MDs, to make up for the delay in data transfer over a wireless link. The entire user applications may be offloaded to the MEC node (known as full-offloading) or a part of them (known as partial-offloading) based on parameters like the current network condition, computation power of the MDs, permissible delay of the application, etc. If the offloaded tasks (that is, a portion of an application) cannot be run by the roadside cloud with sufficient performance improvement, the edge hosts may forward them to a central cloud. While forwarding the tasks to the central cloud, the scheduler placed at the edge node ensures the deadline constraints of the tasks. In our work, we assume that one user submits at most one application at a time.

3.1 Application Model

An application generated by a user (mobile device, autonomous vehicles, etc., as depicted in Fig. 1) can be represented as a directed acyclic graph (DAG), $G = (V, E)$. Here, V represents the set of vertices where each vertex is an indivisible task or module and E is the set of edges or links which represents the dependency among tasks that may be data dependency or functional dependency [24]. A module or task can be one of the many components or functional units of an application. We represent each task τ_i as $v_i = \{l_i, s_i\}$, where l_i is the number of CPU cycles required by that task and s_i is its dataset size. The dataset of a task includes the task code, dependencies, libraries, and static data required for its execution. For instance, in case of a video game application, the dataset may include 3D objects, textures, etc.

3.2 Scheduling Between Local Device and MEC Nodes

The applications generated by the mobile users can be partially or fully offloaded to the connected MEC nodes. Thus, the scheduler needs to decide which tasks of an application are to be executed locally and which are to be executed remotely (offloading) on the edge nodes so as to benefit from the offloading. The application graph is embedded with the cost of executing a task in its local device or at a remote location [5]. For a task v_i, the cost of local and remote execution is represented by $C^{\text{loc}}$and C^{rem}, respectively. Here, $C_i^{\text{loc}} = l_i/f_{\text{loc}}$ and $C_i^{\text{rem}} = l_i/f^{\text{rem}}$, where f^{loc} and f^{rem} are the local and remote processing speed. Each edge contains a weight $w(v_i, v_j)$ which represents the communication cost between two adjacent vertices (or tasks) v_i and v_j in the graph. This can be written as $w(v_i, v_j) = \text{inp}(v, v_j)/B$, where $\text{inp}(v_i, v_j)$ is the size of the input data from one task v_i to v_j and B is the available bandwidth. The edge weight becomes nil where two adjacent tasks are executed in the same platform (local or remote). Thus, a cost function for executing an application can be defined as follows.

$$C = \sum_{i=0}^{i=|V|} \left(X_i C_i^{loc} + (1 - X_i) C_i^{rem} \right) + \sum_{v_j, v_k \in E} X_{j,k}^{e} \, w(v_j, v_k) \tag{1}$$

where X_i is a binary variable such that its value is 1 if the task v_i is executed locally or 0 otherwise. Similarly, $X_{j,k}^{e} = 0$ if v_j and v_k are executed on the same platform and 1 otherwise.

To minimize C as expressed in Eq. (1), we apply a polynomial-time algorithm, minimum cut of phase (MCOP) [5]. The algorithm works by finding a minimum cut of the graph such that it partitions the graph into two sets of vertices: one set to execute locally and the other to execute remotely. Figure 2 shows an example of a

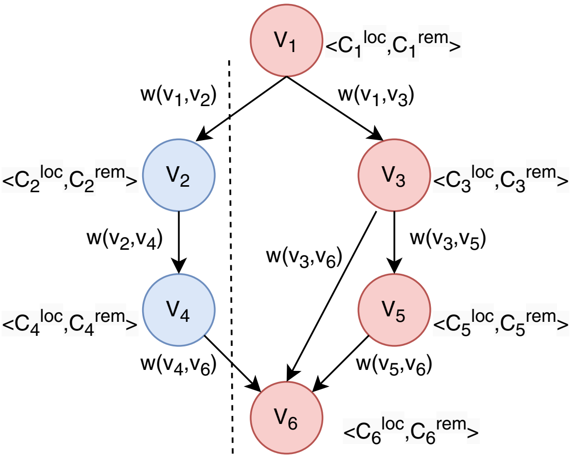


Fig. 2 An example of application graph partitioning

DAG partitioned into local and remote execution sets. The vertices in red are to be executed locally while those in blue are to be offloaded.

3.3 MEC Execution and Cost Model

After the applications are partitioned into local and remote sets, the tasks belonging to the remote set of different applications reach the MEC hosts for their execution. We consider discrete-time scheduling in our work. Let, at time instant t, n ready tasks, represented by $\tau = \{\tau_1, \tau_2, \ldots, \tau_n\}$, reach MEC nodes. A task becomes ready when all of its predecessor tasks are executed. In order to execute the set of n tasks, the edge nodes need to provide n VMs for them. We consider MEC compute nodes operate at discrete states where each state corresponds to a voltage level, CPU frequency [25]. Making use of dynamic voltage and frequency scaling (DVFS), the machine changes its operating state depending on the load on the CPU and optimize its energy consumption. The execution time of a task is given by

$$T_i = \frac{l_i}{f_i} \tag{2}$$

where f_i is the frequency at which the host runs the task τ_i.

The CPU energy consumption consists of a static and a dynamic component [26]. The energy consumed by a host for running n tasks is given by

$$E = \sum_{i=0}^{i=n} \left(P_{si} + Af_i^3\right)T_i \tag{3}$$

where P_{si} is the static power consumption of the host h_i and A is a system parameter which depends on the chip architecture.

3.4 Scheduling Between Edge and Cloud

The computation capacity of the MEC nodes is limited. When the number of tasks offloaded to a MEC node increases, its power consumption also increases along with the average execution time of the tasks. When a MEC host is not able to execute all the tasks with sufficient performance gain (i.e., crosses a predefined threshold), some of the tasks are forwarded to the central cloud system. The cloud hosts are considered to be more powerful than the edge hosts and can execute the tasks in lesser time. However, forwarding tasks to the central cloud results in transmission delay due to data transfer via the Internet. This delay depends on the amount of data that is required by a task for its execution which includes the size of the task binary, dependencies, and any other static data it requires. Let the bandwidth available for data transfer be B MB/s (megabyte per second) and the processing speed of the cloud host be f_{cc}. So, the total delay in executing task τ_i in the cloud is given by

$$D_i = \frac{l_i}{f_{cc}} + \frac{s_i}{B} \tag{4}$$

The partitioning procedure mentioned in Sect. 3.2 offloads tasks to only to the remote edge platform. Some offloaded tasks may benefit more from executing in the central cloud rather than in the edge nodes or vice versa. Once the tasks arrive at the edge, the edge host decides whether to forward some of the tasks to the cloud to minimize its power consumption. This offloading procedure is transparent to the user. The user is not aware of whether a task is running in the edge or the cloud once it is offloaded. If forwarding tasks to the cloud causes the user's perceived remote execution delay to increase, the user's offloading cost as represented in Eq. (1) leads to poor QoS. Therefore, the total delay of execution in the cloud must be less than or equal to the worst-case delay of execution at the edge host.

3.5 Problem Formulation

The main objective in this work is to schedule a set of user applications generated by mobile users on a MEC platform such that the energy consumption at the edge nodes is minimized, while the delay constraints of the applications are maintained. Let there be n tasks, $\tau = \{\tau_1, \tau_2, \ldots, \tau_n\}$, reach MEC nodes at time instant t after the applications are partitioned. For every task τ_i, we need to execute the task either on the MEC nodes or at the central cloud. Mathematically, the problem can be represented as follows:

$$\min_{i} \; Z(i) = \sum_{i=0}^{n} \Theta_i E_i \tag{5a}$$

$$subject\ to\ \Theta_i \in \{0,\ 1\}, \tag{5b}$$

$$f_i \leq F, \tag{5c}$$

$$D_i \leq T_i \tag{5d}$$

where Θi determines whether to forward the task τi to the cloud ($\Theta i = 0$) or not (Θi $=1$) and F is the maximum available processing speed in the MEC host. *Di* and *Ti* are calculated using Eqs. (4) and (2), respectively. As the task scheduling problem in MEC is shown to be NP-hard [19], we propose a heuristic approach to solve the problem.

Algorithm 1: Energy Aware Mobile-Edge-Cloud Orchestration Strategy

```
1  $G$ = Set of application graphs received;
2  Partition each application graph in $G$ using the procedure explained in
     Section 3.2 and obtain set of tasks to be scheduled on edge nodes;
3  $\tau$ = Set of $n$ tasks ready for execution at current time slot $t$;
4  $f$ = Minimum operating frequency of edge host;
5  Sort $\tau$ in increasing order of their data size;
6  $totalDelay = 0$;
7  for $i$=1 to $n$ do
8      $totalDelay \mathrel{+}= D_i^{MIN}$;
9      if $totalDelay \leq \frac{l_i}{f}$ then
10         Set deadline $\tau_i.d = \frac{l_i}{f}$;
11         Forward $\tau_i$ to cloud, set $\Theta_i = 0$;
12         Remove $\tau_i$ from $\tau$;
13     else
14         Set $\Theta_i = 1$
15 for each task $\tau_i$ in $\tau$ do
16     Find the VM with minimum energy consumption such that it
         satisfies the delay constraint of $\tau_i$ ;
17     Schedule the task $\tau_i$ on the selected VM and update the system
         resources of the corresponding edge node;
```

4 Proposed Scheduling Strategy

In this section, we present the Energy-aware Edge–Cloud Orchestration Strategy to execute a set of user applications with cooperation from the mobile device, edge nodes, and the central cloud system. The scheduler resides at the edge layer (as demonstrated in system architecture using Fig. 1) and this is invoked at a discrete-time interval. First, we obtain the user applications which are represented as DAG. They are partitioned using the procedure as explained in Sect. 3.2. After the applications are partitioned, they are scheduled either on the edge nodes or in the central cloud. We present the steps of our proposed strategy using Fig. 3.

The pseudo code of our proposed strategy is presented in Algorithm 1. Step 2 of the algorithm states the partitioning operation. The partitioning operation outputs a set of tasks that are to be executed at the edge nodes. Let τ be such a set of ready tasks at a time instant t. These tasks are sorted in the increasing order of their data size (line 5). We are assuming that each file is sent over the network sequentially and thus the individual delay of each task accumulates as shown in line 8 of the algorithm. Next, for each task, our policy checks whether the task's best-case cloud

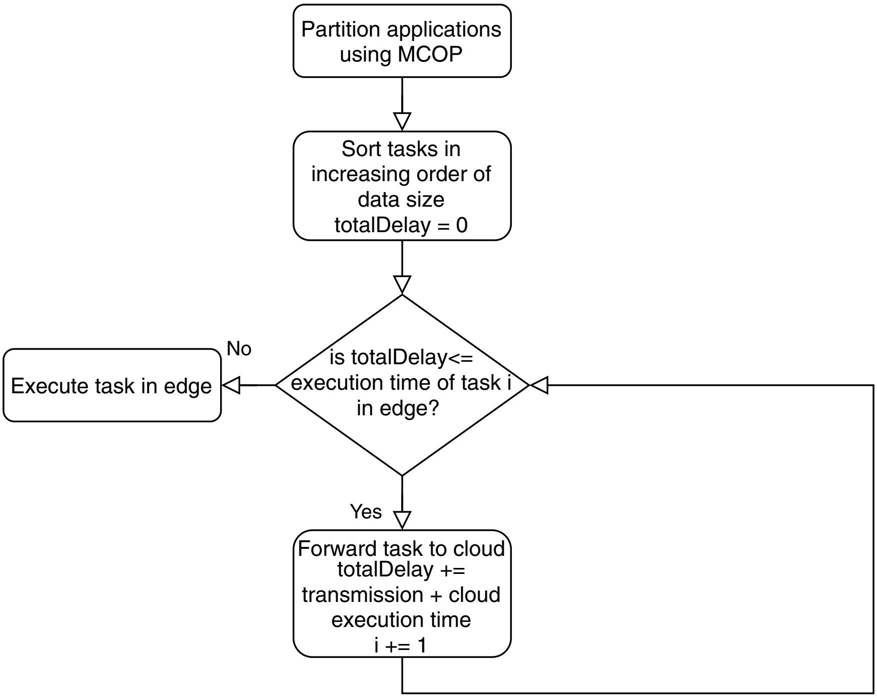

Partition applications using MCOP
Sort tasks in increasing order of data size totalDelay = 0
is totalDelay<= execution time of task i in edge?
No
Execute task in edge
Yes
Forward task to cloud totalDelay += transmission + cloud execution time i += 1

Fig. 3 Flowchart depicting the steps involved in the proposed approach

execution delay D_i^{MIN} is less than or equal to the worst-case execution time at the edge host $\left(T_i^{MAX} = l_i/f_i^{MIN}\right)$ (line 10). If this is the case, we set hard deadlines for the tasks to be offloaded to the central cloud, so that the cloud allocates the processing power accordingly and there is no more delay than expected $\left(T_i^{MAX}\right)$ (line 11). This ensures that the offloading device's perceived remote cost does not change regardless of what amount of processing power is allocated to the task by the edge host. The forwarded tasks are removed from the task queue of the edge host (line 12). Further, the remaining tasks are scheduled in an energy-efficient manner (line 15 to 17).

5 Performance Evaluation

In this section, we present the details about the simulation platform, experimental parameter setup and the results with their analysis.

Table 1 Simulation parameters

Parameter	Value
Task length	1×10^9–9×10^9 cycles
Task data size	1–50 MB
Number of cores in edge host	8,16
Bandwidth	10–100 MB/s
Set of frequencies	1.8–3.6 GHz

5.1 Simulation Environment

In order to perform our experiments, we create a discrete event-based simulation environment using Python toolkit. The simulation platform is integrated with a task generator and it accepts several system and application parameters, such as host computation frequency, network bandwidth, number of cores in a host, length of tasks, task data size, etc. We perform our experiments for a wide variety of parameters which are listed in Table 1. Each MEC host can have six operating frequencies (in GHz) [1.8, 2.3, 2.8, 3.0, 3.2, 3.6]. The cores are automatic DVFS enabled, i.e., they can independently change their frequencies based on the requirements. We consider the roadside cloud to be connected to the central cloud with optical fiber links having bandwidths ranging from 10 to 100 MB/s. The number of tasks is varied from 50 to 300.

We have taken the CloudFreq [27] scheduling policy as our baseline and the effectiveness of our proposed strategy is evaluated by comparing the results with and without the presence of our energy-saving policy. CloudFreq [27] is an energy-aware scheduling policy for executing a set of tasks on DVFS-enabled cloud hosts. They do not rely on prior knowledge of the tasks. Computation speed to the tasks is allocated according to a pre-defined parameter $\theta \in [0, 1]$. Lower values of θ ensure lower execution time at the cost of higher energy consumption and vice versa. Given a set of tasks along with their required number of CPU cycles, CloudFreq assigns each task to a core at a certain frequency so that the processor's overall energy consumption, task execution time, or combination of both is minimized depending on the value of θ. The value of θ for CloudFreq is set to 0.3 in our experiment.

5.2 Result Analysis

We measure the performance of the scheduling policies in terms of percent of energy savings and execution speedup with respect to the baseline energy-efficient scheduling policy *CloudFreq* [27]. Figure 4 shows the energy saving (in %) with respect to the number of tasks having other randomly generated characteristics. The percent of energy savings is obtained by $(E - E\text{new})/E \times 100$, where E is the energy consumption of the edge hosts without applying our proposed scheduling strategy,

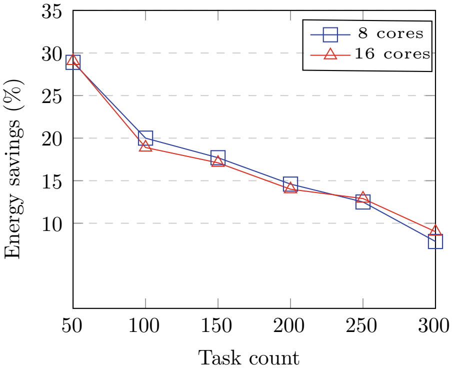


Fig. 4 Variation of % energy savings with number of tasks

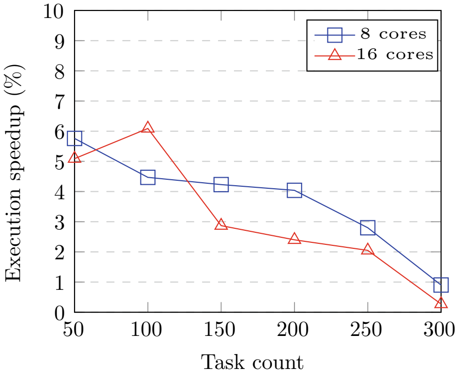


Fig. 5 Variation of % execution speedup with number of tasks

while E_{new} is the energy consumption of the edge hosts with our strategy. The execution speedup is calculated in a similar manner by taking the execution times with and without applying our proposed policy. Figure 5 shows the execution speedup with the number of tasks. We observed the energy savings for edge hosts are from 8 to 30%. Although the execution speedup is not very significant, we observed a maximum 6% improvement among the tasks running on the edge hosts after forwarding the other tasks to the cloud. In some rare cases, we observed a drop in the execution speedup of around 0.25% but we believe it is insignificant compared to the benefits.

As evident from Figs. 4 and 5 that with an increase in the number of tasks, the % of energy savings as well as execution speedup fall for both 8 and 16 core machines. This is expected as with an increasing number of tasks, the total amount of data to be sent to the cloud also increases. This results in higher transmission delay for each task which in turn causes a lesser number of tasks to be forwarded. Hence, it lowers the benefits from task forwarding to the cloud. In the case of energy savings, the results are very similar for both 8 and 16 core machines. However, in case of execution speedup, we noticed that the speedup for 8 core machine is more than that of a 16 core machine as the number of tasks increases. The reason behind this is that

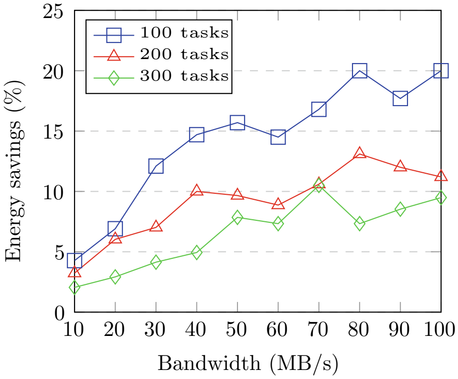


Fig. 6 Variation of % energy savings with network bandwidth

even with a greater load, the 16 core CPU is able to benefit from its larger number of processing elements, while 8 core machine suffers. Thus reducing the load on an 8 core machine by means of task forwarding can alleviate the problem resulting in better speedup.

In addition to the above, we also measure the % of energy savings with respect to the bandwidth for different number of tasks as shown using Fig. 6. It is observed that with an increase in bandwidth, the savings increase. It also backs the previous observations from Figs. 4 and 5 by showing the drop in gain with an increasing number of tasks. This shows that a large number of offloading among platforms is not very desirable especially when the network condition is poor. The occasional anomalies in the plots (such as the dip in gain at 90 MB/s for task count 100) are due to the randomness of the task characteristics.

6 Conclusion

In this paper, we have tackled the problem of scheduling a set of user applications generated by mobile users, smart devices, or autonomous vehicles in MEC platforms by means of effective cooperation among MDs, edge nodes, and the central cloud. While doing so, we first, efficiently partition the application graph into two sets and, subsequently, we use DFVS technique at the edge layer to decide forwarding of tasks to the central cloud. We performed extensive simulation experiments for a wide range of system and application parameters. The simulation results showed that our proposed scheduling strategy reduces the energy consumption of the edge nodes up to 30% with respect to the baseline policy. Our proposed policy also achieves a speedup of a maximum 6%.

In the near future, we plan on focusing more on a practical implementation of the work and also take into consideration other objectives like QoS and revenue of service providers.

References

1. Kumar K, Lu Y (2010) Cloud computing for mobile users: can offloading computation save energy? Computer 43(4), pp 51–56
2. Chen C-H, Lee C-R et al A mobile cloud framework for deep learning and its application to smart car camera. In: Internet of vehicles—technologies and services. Cham, pp 14–25
3. Yu R, Zhang Y, Gjessing S, Xia W, Yang K (2013) Toward cloud-based vehicular networks with efficient resource management. IEEE Netw 27(5):48–55
4. Roy S, Das AK, Chatterjee S, Kumar N, Chattopadhyay S, Rodrigues JJPC (2019) Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications. IEEE Trans Indus Inf 15(1):457–468
5. Wu H, Knottenbelt WJ, Wolter K (2019) An efficient application partitioning algorithm in mobile environments. IEEE Trans Parallel Distribut Syst 30(7), pp 1464–1480
6. Ning Z, Huang J et al (2019) Mobile edge computing-enabled internet of vehicles: toward energy-efficient scheduling. IEEE Netw 33(5):198–205
7. Paiker NR, Shan J et al (2020) Design and implementation of an overlay file system for cloud-assisted mobile apps. IEEE Trans Cloud Comput 8(1):97–111
8. Panda SK, Jana PK (2019) An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems. Clust Comput 22:509–527
9. Hilman MH, Rodriguez MA, Buyya R (2020) Multiple workflows scheduling in multi-tenant distributed systems: a taxonomy and future directions. ACM Comput Surv 53(1)
10. Prodan R, Torre E et al (2019) Dynamic multi-objective virtual machine placement in cloud data centers. In: 45th Euromicro conference on software engineering and advanced applications, pp 92–99
11. Portex container adoption survey. https://portworx.com/wp-content/uploads/2019/05/2019-container-adoption-survey.pdf
12. Ning Z, Dong P et al (2019) A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things. IEEE Internet Things J 6(3):4804–4814
13. Zhao J, Li Q et al (2019) Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks. IEEE Trans Vehicular Technol 68(8):7944–7956
14. Wang X, Wang K, Wu S, Di S, Jin H, Yang K, Ou S (2018) Dynamic resource scheduling in mobile edge cloud with cloud radio access network. IEEE Trans Parallel Distrib Syst 29(11):2429–2445
15. Elbamby MS, Bennis M, Saad W (2017) Proactive edge computing in latency- constrained fog networks. In 2017 European conference on networks and communications (EuCNC), pp 1–6
16. Chiang Y, Zhang T, Ji Y (2019) Joint cotask-aware offloading and scheduling in mobile edge computing systems. IEEE Access 7:105008–105018
17. Liu J, Wang S et al (2019) A task oriented computation offloading algorithm for intelligent vehicle network with mobile edge computing. IEEE Access 7:180491–180502
18. Kuang Z, Li L et al (2019) Partial offloading scheduling and power allocation for mobile edge computing systems. IEEE Internet Things J 6(4):6774–6785
19. Tran TX, Pompili D (2019) Joint task offloading and resource allocation for multi-server mobile-edge computing networks. IEEE Trans Vehicular Technol 68(1):856–868
20. Dinh TQ, Tang J, La QD, Quek TQS (2017) Offloading in mobile edge computing: task allocation and computational frequency scaling. IEEE Trans Commun 65(8):3571–3584
21. Ning Z, Dong P et al (2019) Deep reinforcement learning for intelligent internet of vehicles: an energy-efficient computational offloading scheme. IEEE Trans Cogn Commun Netw 5(4), pp 1060–1072
22. Guo M, Li L, Guan Q (2019) Energy-efficient and delay-guaranteed workload allocation in iot-edge-cloud computing systems. IEEE Access 7:78685–78697
23. Liang J, Liu C, Tan G, Yang L (2019) Joint offloading and frequency scaling technology for mobile edge computing. In: 2019 IEEE 21st international conference on high performance computing and communications, pp 2045–2052

24. Peng Q, Jiang H, Chen M, Liang J, Xia Y (2019) Reliability-aware and deadline-constrained workflow scheduling in mobile edge computing. In: IEEE 16th international conference on networking sensing and control (ICNSC), pp 236–241
25. Kim K H, Buyya R, Kim J (2007) Power aware scheduling of bag-of-tasks applica- tions with deadline constraints on dvs-enabled clusters. In: 7th IEEE International Symposium on CCGrid. IEEE, pp 541–548
26. Dayarathna M, Wen Y, Fan R (2016) Data center energy consumption modeling: a survey. IEEE Commun Surv Tutorials 18(1):732–794
27. Zhang Y, Wang Y, Hu C (2015) Cloudfreq: elastic energy-efficient bag-of-tasks scheduling in dvfs-enabled clouds. In: IEEE 21st international conference on parallel and distributed systems (ICPADS), pp. 585–592

Prediction of Heart Disease using LDL in Edge Computing Systems

K. Anitha Kumari, M. Ananyaa, P. S. Keshini, and M. Indusha

Abstract Elevated levels of Low-Density Lipoprotein (LDL), referred to as the "Bad cholesterol", result in the risk of heart disease. LDL causes the buildup of fatty deposits at the walls of arteries and blood vessels, increasing the rate of risk of heart failure or stroke. A device named the glucometer is used to derive the parameter Triglyceride (TGL). TGL composition is used to calculate the Very Low-Density Lipoprotein (VLDL) level, followed by computation of the LDL level using additional parameters such as the Total Cholesterol and High-Density Lipoprotein (HDL). If the LDL level is less than 100 mg/dL then it is optimal. If the LDL level is higher than 100 mh/dL, then the patient is under the risk of heart disease/stroke. Internet of Things (IoT) and Edge computing are two developing fields in the current web-based environment. The Smart Health care system plays a most predominant role in the current research for the integration of IoT devices with Edge Computing. For secure processing of patient data and test reports, two encryption schemes are applied, namely, Gorti's encryption scheme and Carmichael's encryption scheme. LDL values are computed using both the encryption schemes, analysis is performed, and the results are compared.

Keywords Low-density lipoprotein (LDL) · Very low-density lipoprotein (VLDL) · Internet of things (IoT) · Edge computing systems · Heart disease

1 Introduction

The glucometer is an instrument used for self-checking of blood sugar levels. With the accessibility of various glucometers, there is a tenacious endeavor to boost the exactness and therefore the accuracy of those glucometer readings, to coordinate the center estimations of glucose and other parameters. The accuracy of glucometer readings was tested in medical clinics with a lab reference strategy. An aggregate of 105 blood tests was gathered from in-tolerant and out-persistent from our tertiary

K. A. Kumari (✉) · M. Ananyaa · P. S. Keshini · M. Indusha
Department of IT, PSG College of Technology, Coimbatore, India
e-mail: kak.it@psgtech.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_44

consideration emergency clinic. Blood tests were gathered and minded six glucometers and the same biopsy was sent to the center for glucose estimation. The lab esteem was utilized as a source of perspective for correlation. The exactness was assessed by the ISO standards. The outcomes were assessed by Bland Altman charts, connection coefficients, disperse plots, and Clarke's blunder network examination. The strategies utilized for the estimation of glucose levels are comparable in both the emergency clinic-based and out-quiet glucometers. Diabetes is an infection of metabolic where the glucose level stays over the ordinary level for a significant period. The blood glucose observing framework uncovered an individual type of blood glucose and assists with recommending the best arranging of dinners, required activities, and drug times. An m-IoT arrangement technique for noninvasive glucose level estimating on genuine premise is proposed [1]. To measure the parameters of the glucometer, the corresponding test strip is selected and inserted in the glucometer. A drop of the blood sample is placed on the test strip, and the test results are obtained on the digital panel of the glucometer, where the parameters are displayed. The parameter used to calculate the Very Low-Density Lipoprotein (VLDL) is Triglyceride (TGL), which is obtained as an output parameter from the glucometer. Further, the computation of the LDL level is done using additional parameters such as the Total Cholesterol and High-Density Lipoprotein (HDL). This paper gives an outline of the coordination of smart Health care applications with edge computing into IoT; this includes an examination of the advantages coming about because of the coordination procedure and the usage challenges experienced in health care systems. IoT is for the most part characterized by certifiable little things, generally circulated, with constrained capacity and preparing limit, that includes concerns in regards to dependability, performance, security, and protection [2]. For secure storing and computational processes, the raw parameters are encrypted, and final LDL values are computed as encrypted values. After computation, the decryption process is carried out, followed by analysis, which provides the final test results, wherein a patient may be diagnosed with a risk of heart disease/stroke or the patient has healthy levels of LDL.

LDL is a small mass composed of an external edge of lipoprotein with a cholesterol community [3]. Sufficient LDL levels are fundamental to secure nerves and keep up sound cells and hormones. Nonetheless, unnecessary degrees of LDL can cause heart stroke. Perfect LDL levels are under 100 mg/dL. The close ideal LDL run is between 100 and 129 mg/dL. The marginal high LDL extent is between 130 and 159 mg/dL. The high LDL extent is between 160 and 189 mg/dL and high LDL ranges above 190 mg/dL. Sound cholesterol extends may rely upon the age, family ancestry, way of life, and other hazard factors. LDL decrease may predict disease progression and poor prognosis of COVID-19. A reduction in low-thickness lipoprotein (LDL) levels may foresee illness movement and helpless guess in patients with coronavirus ailment 2019, as indicated by study results distributed in Metabolism.COVID-19 has been related to an expected death pace of around 2.3% [4]. The objective of the current investigation was to survey the lipid pathophysiology in COVID-19. The review longitudinal investigation included 21 patients with research center affirmed COVID-19 admitted to Zhongnan Hospital of Wuhan University in Wuhan, China, between January 18 and February 8, 2020. Included patients had routine blood tests performed before confirmation for COVID-19 (between January 9 and 17, 2020), including an appraisal of the lipid profile. Control bunches without COVID-19 included 31 sound people and 21 patients with constant obstructive respiratory malady who had lipid tests acted in a similar medical clinic. When all is said in done, low LDL levels [less than 100 mg/dL] are useful for heart wellbeing.

2 Literature Survey

2.1 Carmichael's Hypothesis

Siddharth P. K, Pal. O, and Alam. B proposed a proficient arithmetical homomorphic encryption plot enthusiastic about Carmichael's hypothesis which works best for the numbers, and it underpins activities to be performed over the encoded information [5]. This plan unravels the problems regarding the traditional encryption plans. That is, in conventional encryption conspires the activities do not seem to be permitted to be performed on the scrambled information which prompts the safety problems with delicate information. The proposed calculation is employed for a few applications like electronic democratic, multiparty calculation, and then on. The activities related to the Carmichael's encryption plot is secluded number juggling in nature. Gorti's and Carmichael's are the two FHE plans for encryption and examining scrambled information to foresee whether the individual has the danger of getting a coronary illness or not.

2.2 Homomorphic Encryption

Al Mashhadi. H and Ala'a A. K proposed a good half and half homomorphic encryption procedure for picture encryption to ensure the sheltered trade of personal pictures within the open cloud enthusiastic about the square pixel position [6]. These three methods settle the potential issues identified with security and protection since cloud frameworks are normally in an open area when the clients transfer and offload the knowledge to the cloud utilizing customer gadgets. The proposed procedures requirement for Elgamal and EHC. The EHC strategy is extremely productive as far as security and time since it takes the nice attributes of Elgamal thus it gives awesome security and tiny run time executions. The disadvantage is that colossal asset and the further room is required. Khalil Hariss et al. broke down DGHV and BVBGV completely homomorphic encryption plans. The DGHV depends on registering over genuine whole numbers while the opposite one depends on Lattice-based Encryption (LWE).

2.3 Using Homomorphic Encryption

Naw Safrin Sattar et al. proposed a made sure about elevated photography utilizing homomorphic encryption by considering the problem because the specialists to encode the photographs and to transfer it on the cloud server that untrusted [7]. The server performs the calculation on the encoded information without the knowledge of the images utilizing NTRU conspire. After the link, the encoded result's conveyed to the concerned expert for decoding. The first preferred position is the utilization of a server for usage of perceiving samples of water, sky, then forth. The confinement is that the utilization of high circumspection for planning the zones of profoundly secret spots utilizing pictures in the cloud. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan proposed an easy fairly homomorphic" encryption plot which utilizes just rudimentary secluded math and utilize Gentry's procedures to alter over it into a homomorphic conspire. It simply utilizes expansion and increases over the entire numbers as against working with perfect grids over a polynomial ring. This system can improve the proficiency of the plan while saving the hardness of the estimated gcd issue which is that the principle little bit of leeway of this plan. Santhiya and Anitha Kumari presented work in analyzing DGHV and NTRU schemes [8]. The many disadvantages are that the safety of to a point homomorphic conspire is diminished while finding an inexact number gcd.

3 Proposed System

3.1 Gorti's Encryption Scheme

In 2013, Gorti et al. proposed EHC which is the Enhanced Homomorphic Cryptosystem [9]. The EHC is a completely homomorphic open key encryption plot. It utilizes expansion, increase, blended expansion, and blended duplication over the whole numbers. The private key will haphazardly be produced for every encryption procedure. A similar plain book does not produce the equivalent ciphertext, to keep the interloper from breaking the ciphertext considerably after it has a solid perception.

3.1.1 Key Generation

In a key generation, two large prime numbers 'p' and 'q' are chosen under the constraint $p > q$. Following this, the public key 'n' is computed using the formula [10],

$$p * q = n. \tag{1}$$

3.1.2 Encryption

The Input message for the encryption procedure ’m” is required to be a component of {1,0}. As the initial step of the encryption procedure, an irregular number ‘*r*’ is created to make this plan non-deterministic. What’s more, the created ‘*r*’ esteem is kept as a mystery [11]. Also, as the subsequent advance, the ciphertext is figured utilizing the formula,

$$m + \mathrm{r} * p^{q} \bmod n = C. \tag{2}$$

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead.

where,
p is a random integer which is kept secret,
q is a random integer which is kept secret,
n public key,
m message, and
r random parameter.

3.1.3 Decryption

In the unscrambling procedure, the first message is recovered from their separate ciphertexts with the assistance of their private keys and afterward named as [12],

$$c \bmod p = \mathrm{decrypt}\,(p, m). \tag{3}$$

3.2 Carmichael’s Encryption Scheme

Carmichael’s cryptosystem is the mathematical homomorphic encryption framework, which is most appropriate for positive whole numbers [13]. It is a completely homomorphic open key encryption plot. It underpins homomorphic activities like expansion and increases over the whole numbers. The activities associated with this plan are a particular number of juggling in nature. The plan is probabilistic, i.e., encryption of a similar message relies on some randomized number which produces distinctive code messages on each season of its encryption.

3.2.1 Key Generation

As the initial step of the key age process, two enormous prime numbers 'p' and 'q' are picked under the condition $p > q$. Furthermore, as the subsequent advance, the open key n is figured utilizing the equation,

$$p * q = n. \tag{2}$$

As the last advance the Carmichael's capacity,

$$1\,\mathrm{cm}\,((p-1)(q-1)) = \lambda(n). \tag{4}$$

3.2.2 Encryption

The input message for the encryption procedure 'm'; is required to be a component of $\{1{,}0\}$, and it ought to fulfill the condition $0 < m < n$. As the initial step of the encryption procedure, an arbitrary number r is produced to make this plan probabilistic. Also, the created 'r' esteem is kept a mystery. As the subsequent advance, the ciphertext is registered to utilize the equation [14],

$$M^{\mathrm{r} * \lambda(n) + 1} \operatorname{mod} n^2 = C, \tag{5}$$

where

m—message,
n—public key, and
r—a random integer that is kept secret.

3.2.3 Decryption

In the unscrambling procedure, the first message is recovered from their ciphertexts with the assistance of their private keys and afterward named as

$$c \operatorname{mod} n = \mathrm{decrypt}\,(p, m). \tag{6}$$

4 Case Study

4.1 Cholesterol

Cholesterol is a waxy, fat-like substance that is found in all the cells of a persons' body. Body needs some cholesterol to make hormones, nutrient D, and substances that assist with processing nourishments [15]. Body makes all the cholesterol it needs. Cholesterol is likewise found in nourishments from creature sources, for example, egg yolks, meat, and cheddar. On the off chance that if a person have an excess of cholesterol in blood, it can join with different substances in the blood to frame plaque. Plaque adheres to the dividers of courses. This development of plaque is known as atherosclerosis. It can prompt coronary vein illness, where the coronary conduits become restricted or even blocked.

4.1.1 High-density Lipoprotein

High-density lipoprotein (HDL) is one of the five significant gatherings of lipoproteins [16]. Lipoproteins are intricate particles made out of different proteins that transport every single fat atom (lipids) around the body inside the water outside cells. They are normally made out of 80–100 proteins for each molecule (composed by one, two, or three ApoA; more as the particles amplify getting and conveying progressively fat atoms) and shipping up to many fat particles per molecule.

4.1.2 Low-density Lipoprotein

Low-density thickness lipoprotein (LDL) is one of the five significant gatherings of lipoprotein that transports every fat particle around the body in the extracellular water [17]. Lipoproteins are unpredictable particles made out of different proteins, ordinarily 80–100 proteins for every molecule (sorted out by a solitary apolipoprotein B for LDL and the bigger particles). A solitary LDL molecule is around 220–275 angstroms in distance across, commonly moving 3,000–6,000 fat atoms for every molecule, and changing in size as per the number and blend of fat particles contained within. The lipids conveyed incorporate every single fat particle with cholesterol, phospholipids, and TGLs predominant; measures of each fluctuating impressively.

4.1.3 Triglyceride

Triglycerides are a kind of fat (lipid) found in blood [18]. When a person eat, body changes over any calories it does not have to utilize immediately into TGLs. The TGLs are put away in fat cells. Afterward, hormones discharge TGLs for vitality between meals. If a person normally eat a larger number of calories than adequate, especially from high-sugar nourishments, it may lead to high TGLs (hypertriglyceridemia).

4.1.4 Very Low-density Lipoprotein

Very low-thickness lipoprotein (VLDL) cholesterol is created in the liver and discharged into the circulation system to flexibly body tissues with a kind of fat (TGLs) [19]. There are a few kinds of cholesterol, each composed of lipoproteins and fats. Each kind of lipoprotein contains a blend of cholesterol, protein, and TGLs, yet in changing sums. About the portion of a VLDL molecule composed of TGLs. VLDL cholesterol is generally assessed as a level of a persons' TGL esteem. A raised VLDL cholesterol level is over 30 mg for every deciliter (0.77 millimoles/liter).

4.1.5 Heart Disease

Heart disease projects a range of conditions that deteriorate heart health. Blood vessel diseases such as coronary heart disease, heart rhythm problems (arrhythmias), and birth defects in heart (congenital heart defects) [20], are prominent. The LDL value of a person directly influences the degree of a heart defect in a person. It is calculated using the formula,

$$0.166 * \mathrm{TG} = \mathrm{VLDL} \tag{7}$$

$$(\mathrm{TOT} - C) - \mathrm{HDL} - \mathrm{VLDL} = \mathrm{LDL} \tag{8}$$

where

VLDL = Very low-density lipoprotein
TOT-C = Total Cholesterol
LDL = Low density lipoprotein
HDL = High-density lipoprotein

If the calculated LDL value is more than the LDL range "100 mg/dL", then the person has heart disease. Effects of Heart-Disease: Side effects will shift contingent upon the particular condition. A few conditions, for example, type 2 diabetes or hypertension, may at first reason no indications by any means [21]. Mild side effects of a fundamental cardiovascular issue may include: agony or weight in the chest, which may demonstrate angina, agony or inconvenience in the arms, left shoulder, elbows, jaw, or back windedness, queasiness and weakness, unsteadiness or tipsiness, cold sweats. Even though these are the most widely recognized ones, CVD can cause side effects anyplace in the body.

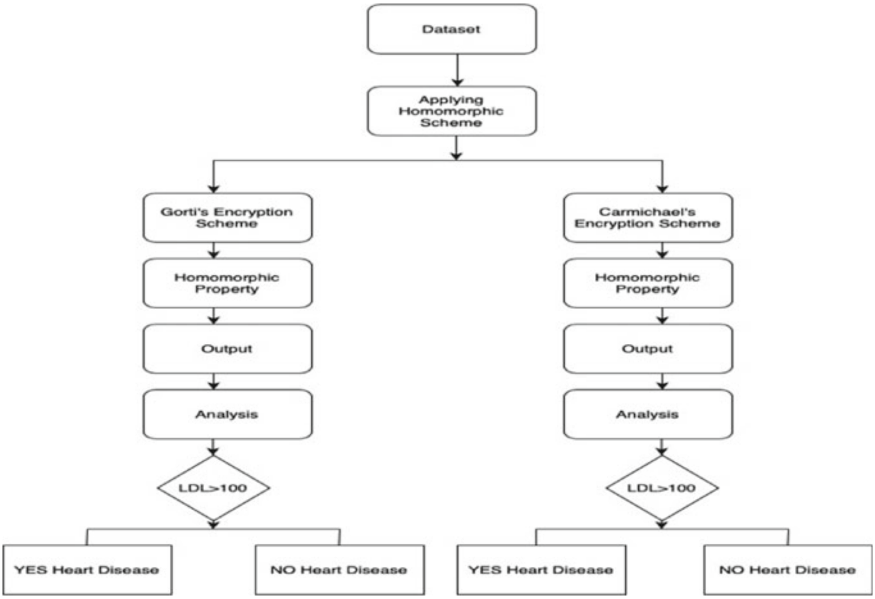


Fig. 1 Proposed system design

4.2 Proposed System Design

The work process of the proposed framework configuration has been portrayed in Fig. 1. It incorporates the accompanying stages:

- Gathering the datasets for the plans.
- Appling EHC and Carmichael's homomorphic encryption components for giving protection and security of the information.
- Performing procedure on encrypted information.
- A relative investigation is made dependent on the exhibition measurements of EHC and Carmichael's homomorphic encryption.

4.3 Data Collection

The cholesterol dataset is gathered from the Kaggle site. With the guide of homomorphic encryption upon the gathered information, we accomplish secure calculation. Qualities considered are HDL [mg/dL], TG [mg/dL], and TC [mg/dL]. The cholesterol dataset collected from [22] Kaggle is shown in Table 1.

Table 1 Data collection

ID	HDL	TG	TC
1	58	64	168
2	61	57	154
3	55	57	225
4	78	93	202
5	72	77	238
6	57	67	162
7	75	78	161
8	43	49	195
9	61	62	210
10	58	64	165
11	51	47	206
12	44	39	142

4.4 *Carmichael's Encryption and Decryption*

Gorti's homomorphic encryption scheme is implemented for the computation of test results in an encrypted format. The EHC encrypted dataset is shown in Table 2.

After computation, the encrypted data is decrypted using corresponding keys, and the EHC decrypted dataset is shown in Table 3.

Table 2 EHC encrypted data

ID	HDL	TG	TC	LDL
121241	121298	121304	121408	−121140
63218	63277	63273	63370	−63132
110310	110362	110364	110532	−110146
126173	126247	126262	126371	−126060
58958	59025	59030	59191	−58799
109263	109314	109324	109419	−109163
161131	161199	161202	161285	−161051
189610	189645	189651	189797	−189458
130297	130349	130350	130498	−130149
10724	10772	10778	10879	−10617
101648	101688	101684	101843	−101489
34942	34974	34969	35072	−34838

Table 3 EHC decrypted data

ID	HDL	TG	TC	LDL
1	58	64	168	100
2	61	57	154	84
3	55	57	225	161
4	78	93	202	109
5	72	77	238	154
6	57	67	162	94
7	75	78	161	73
8	43	49	195	144
9	61	62	210	139
10	58	64	165	97
11	51	47	206	148
12	44	39	142	92

4.5 *Carmichael's Encryption and Decryption*

Carmichael's encrypted data is shown in Table 4.

After computation, the encrypted data is decrypted using corresponding keys, and Carmichael's decrypted dataset is shown in Table 5.

Table 4 Carmichael encrypted data

ID	HDL	TG	TC	LDL
6560988	8015031	6798575	342073	−255429
576279	5525685	1818772	14321002	−8533572
4281171	1539406	7821204	26143277	1145604
22220110	44728363	50342498	40754281	16487893
31299655	1.18E+08	40292853	71206919	57662026
51023337	61217274	1.18E+08	2.08E+08	1.17E+08
11435443	4046205	10090881	4377338	629471
11309324	20877167	10070618	9282948	−7603317
49869656	10563471	33647847	28787349	−2.3E+07
2986711	65453777	78712046	94417072	−8739556
2.01E+08	1.51E+08	3.2E+08	46123996	2.32E+08
6697761	547092	489503	10214104	−2270877

Table 5 Carmichael decrypted data

ID	HDL	TG	TC	LDL
1	58	64	168	100
2	61	57	154	84
3	55	57	225	161
4	78	93	202	109
5	72	77	238	154
6	57	67	162	94
7	75	78	161	73
8	43	49	195	144
9	61	62	210	139
10	58	64	165	97
11	51	47	206	148
12	44	39	142	92

4.6 Analysis of Heart-Disease

The LDL equation is applied over the encoded HDL, TG, and TC (Total Cholesterol). The calculated LDL value is checked against the encrypted value of "100" and if the encrypted LDL value is greater than the encrypted value of "100", then the person is diagnosed with heart disease, and the output is added to the database. Carmichael's analysis on the encrypted computed data is shown in Table 6.

Gorti's analysis on the encrypted computed data [23] is shown in Table 7.

Table 6 EHC Analysis on encrypted data

ID	HDL	TG	TC	LDL	Heart disease
1	58	64	168	100	NO
2	61	57	154	84	NO
3	55	57	225	161	YES
4	78	93	202	109	YES
5	72	77	238	154	YES
6	57	67	162	94	NO
7	75	78	161	73	NO
8	43	49	195	144	YES
9	61	62	210	139	YES
10	58	64	165	97	NO
11	51	47	206	148	YES
12	44	39	142	92	NO

Table 7 Carmichael analysis on encrypted data

ID	HDL	TG	TC	LDL	Heart disease
6560988	8015031	6798575	342073	−255429	NO
576279	5525685	1818772	14321002	−8533572	NO
4281171	1539406	7821204	26143277	1145604	YES
22220110	44728363	50342498	40754281	16487893	YES
31299655	1.18E+08	40292853	71206919	57662026	YES
51023337	61217274	1.18E+08	2.08E+08	1.17E+08	NO
11435443	4046205	10090881	4377338	629471	NO
11309324	20877167	10070618	9282948	−7603317	YES
49869656	10563471	33647847	28787349	−2.3E+07	YES
2986711	65453777	78712046	94417072	−8739556	NO
2.01E+08	1.51E+08	3.2E+08	46123996	2.32E+08	YES
6697761	547092	489503	10214104	−2270877	NO

4.7 *Computational Time Analysis*

The computational time is calculated in nanoseconds [24]. The computational time analysis is shown in Table 8. The Carmichael's encryption scheme takes more time to encrypt than Gorti's encryption scheme. Therefore, Gorti's encryption scheme proves to be more efficient than Carmichael's encryption scheme.

The computational time comparison, taken in nanoseconds, of each row encryption time between Gorti's scheme and Carmichael's scheme, is shown in Fig. 2.

The computational time comparison, taken in nanoseconds, of each row decryption time between Gorti's scheme and Carmichael's scheme, is shown in Fig. 3.

The computational time comparison, taken in nanoseconds, of each row analysis time on the encrypted dataset between Gorti's scheme and Carmichael's scheme is shown in Fig. 4.

Table 8 Computational time analysis

Computational time (nanoseconds)	Gorti's encryption scheme	Carmichael's encryption scheme
Total encrypted time	46425600	959505800
Total decrypted time	23646501	23964500
Total analysis time on encrypted data	107000	1090500
Total analysis time on decrypted data	229200	546100

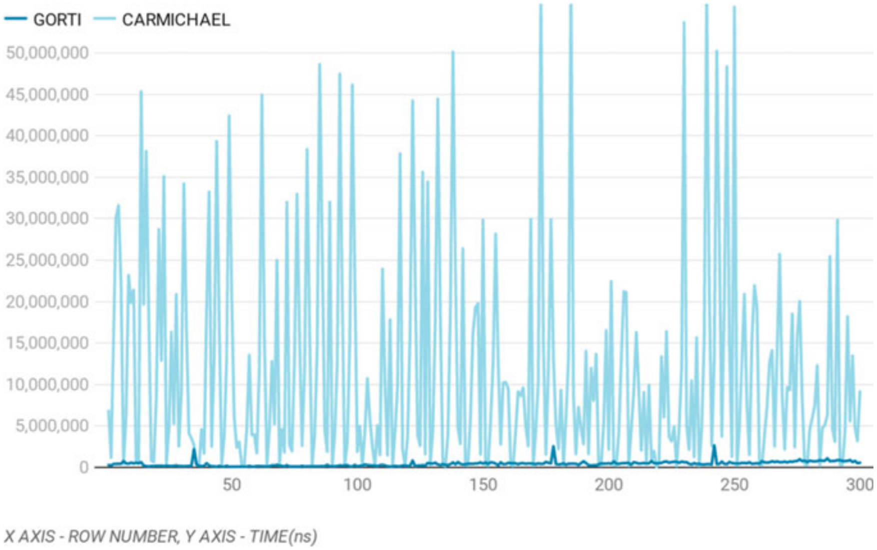


Fig. 2 Comparison of each row encryption time between Gorti's scheme and Carmichael's scheme

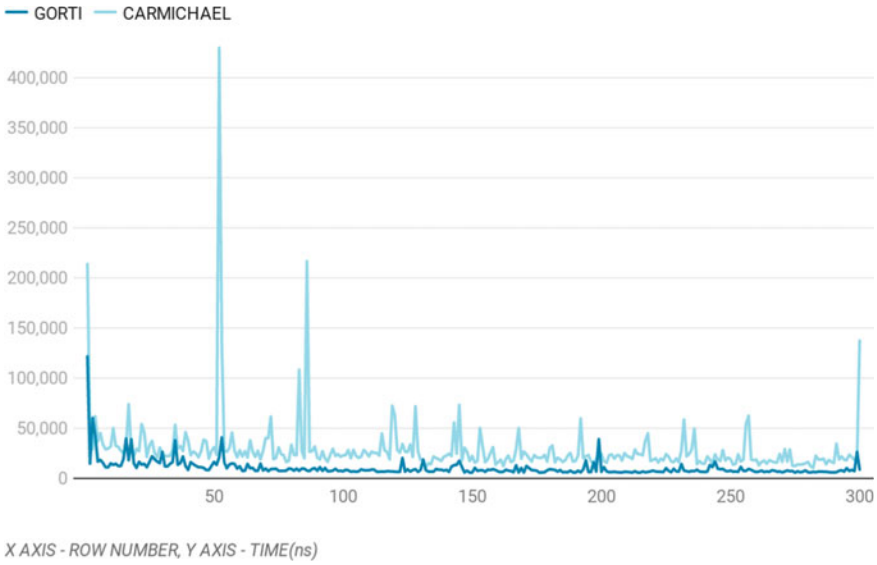


Fig. 3 Comparison of each row decryption time between Gorti's scheme and carmichael's scheme

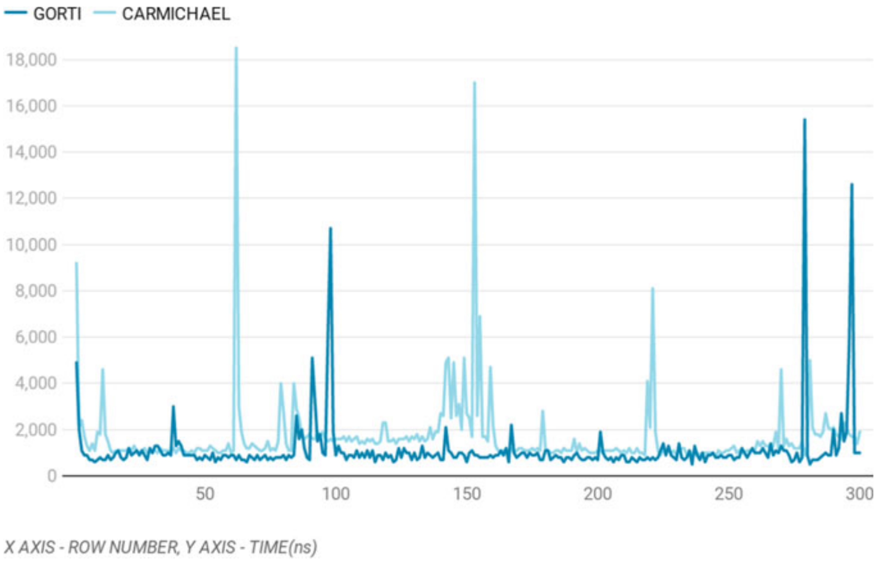


Fig. 4 Comparison of each row analysis time on the encrypted dataset between Gorti's scheme and Carmichael's scheme

5 Attack Analysis

User security serves as a major threat during data processing and analysis. Adoption of homomorphic encryption schemes such as the Gorti encryption scheme and Carmichael's encryption scheme eliminates this threat, by processing data and computation in encrypted values. Client's security stays a significant test, as the specialist organization can without much of a stretch access the client's information. It has been indicated that completely homomorphic encryption plans may be the ideal arrangement, as it permits one gathering to process the client's information homomorphically, without the need of knowing the corresponding mystery keys. The encryption conspire is made sure against assaults, for example, plaintext recuperation assault like picked plaintext assault and known-plaintext assault and Indistinguishability against Chosen Ciphertext Attack (IND–CCA) [25, 26].

6 Conclusion and Future Enhancements

With the growing technological influences, it becomes essential for the secure processing of data. Therefore, homomorphic encryption schemes are employed. Homomorphic encryption schemes allow operations to be performed on the encrypted data as on plaintext. The computation is carried out on the encrypted data, hence without knowing anything of its real value. Finally, the values are decrypted, and the analysis is carried out. This enhances the security of data [25]. From the analysis shown above, Gorti's encryption scheme proves more effective than Carmichael's encryption scheme.

Acknowledgement The authors are indebted to the All India Council for Technical Education (AICTE), New Delhi for supporting this research work under Research Promotion Scheme (RPS) Grant-F.No.8-38/RIFD/RPS/Policy-1/2017-2018.

References

1. Elazhary H (2018) Internet of things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: disambiguation and research directions. J Netw Comput Appl 128:108–140
2. Bhadoriya R, Chattopadhyay MK, Dandekar PW (2016) Low-cost IoT for laboratory environment. Symposium on colossal data analysis networking (CDAN), 978-1-5090-0669-4/16.2016 IEEE
3. https://www.webmd.com/heart-disease/ldl-cholesterol-the-bad-cholesterol#1.
4. https://ourworldindata.org/coronavirus
5. Siddharth PK, Pal O, Alam B (2016) A homomorphic encryption scheme over integers based on Carmichael's theorem. In: 2016 international conference on electrical, electronics, communication, computer and optimization techniques (ICEECCOT). IEEE, pp. 17–20, December 2016
6. Al-Mashhadi M, Khalf AA (2018) Hybrid homomorphic cryptosystem for secure transfer of color image on public cloud haider. IJCSNS Int J Comput Sci Netw Security 18(3), March 2018
7. Sattar NS, Adnan MA, Kali MB (2017) Secured aerial photography using homomorphic encryption. NSysS 2017, pp 107–114
8. Santhiya B, Anitha Kumari K (2020) Analysis on DGHV and NTRU fully homomorphic encryption schemes. In: Proceedings of international conference on artificial intelligence, smart grid and smart city applications (AISGSC 2019). Springer Nature, Switzerland, pp 669 – 678
9. Rao G, Subba VNKV, Uma G (2013) An efficient secure message transmission in mobile ad hoc networks using enhanced homomorphic encryption scheme. GJCST-E: Network, Web & Security 13.9
10. https://www.lri.fr/~fmartignon/documenti/systemesecurite/6-PublicKey.pdf
11. Rivest S, Adleman A (1978) Method for obtaining digital signatures and public-key cryptosystems. CACM 21(2), 120–126
12. Rivest R, Adleman L, Dertouzos M (1978) On data banks and privacy homomorphisms. In: Foundations of secure computation. Academic Press, London, pp 169–177
13. https://doc.sagemath.org/pdf/en/reference/cryptography/cryptography.pdf
14. https://en.wikipedia.org/wiki/Carmichael_function
15. Law MR, Wald NJ, Wu T, Hackshaw A, Bailey A (1994) Systematic underestimation of association between serum cholesterol concentration and ischaemic heart disease in observational studies: data from the BUPA study. BMJ 308(6925):363–366
16. Gordon DJ, Probstfield JL, Garrison RJ et al (1989) High-density lipoprotein cholesterol and cardiovascular disease four prospective american studies. Circulation 79:8–15
17. Friedewald WT, Levy RI, Fredrickson DS (1972) Estimation of the concentration of low-density lipoprotein cholesterol in plasma without the use of the preparative ultracentrifuge. Clin Chem 18:499–502
18. NordestgaardBG Varbo A (2014) Triglycerides, and cardiovascular disease. Lancet 384:626–635

19. Eck MV, Oost J, Goudriaan JR, Hoekstra M, Hildebrand RB, Bos IS, van Dijk KW, Van Berkel TJ (2005) Role of the macrophage very-low-density lipoprotein receptor in atherosclerotic lesion development. Atherosclerosis 183:230–237
20. Ritchey MD, Wall HK, George MG, Wright JS (2019) US trends in premature heart disease mortality over the past 50 years: where do we go from here?external icon (published online ahead of print, 27 Sept 2019). Trends Cardiovasc Med. S1050-1738(19)30134-3
21. WHO (2011) Global atlas on cardiovascular disease prevention and control. World Health Organization
22. https://www.kaggle.com/cdc/national-health-and-nutrition-examination-survey?select=labs.csv
23. http://ijirt.org/Article?manuscript=149587
24. https://www.journals.elsevier.com/computational-statistics-and-data-analysis
25. Vishwanath S (2013) Implementing RSA encryption algorithm to enhance the data security of cloud in cloud computing. Int J Pure Appl Res Engineering Technol 1(8), 220–227. ISSN-2319-507X IJPRET
26. Alexi W, Chor B, Goldreich O, Schnorr P (1988) Bit security of RSA and rabin functions. SIAM J Comput 17(2):194–209

Edge Computing as an Architectural Solution: An Umbrella Review

Ajay Bandi and Julio Ariel Hurtado

Abstract Cloud computing architecture and cloud service applications follow a centralized architecture with bottlenecks in the cloud infrastructure. This infrastructure is significantly affected when services respond to many heterogeneous end devices because of the limitations of bandwidth and the servers' workload; consequently, it introduces a high latency. The advantages of using content delivery networks are to speed up web performance by caching web content on edge nodes near the user. However, there are challenges with streaming data. Researchers create an intermediary infrastructure to store, secure, and compute end devices' services became a new concept called edge computing. Edge computing can leverage applications that are sensitive to latency. However, other issues appear, such as security and deployability. This paper reviewed the literature to analyze edge computing as an architectural solution and identify the underlying architectural quality attributes, tactics, and strategies. The performance quality attribute drives the edge architecture, mainly to reduce the latency and jitter concerns. The quality requirements are addressed by caching, migration, and virtualization strategies. However, the solution introduces other quality attribute concerns such as security, deployment, and scalability. This paper is a first approach for unveiling the rationale behind edge computation from an architectural viewpoint.

Keywords Edge computing · Fog computing · Multi-access edge computing · Software architecture · Data streaming

A. Bandi (✉)
Northwest Missouri State University, Maryville, MO 64468, USA
e-mail: ajay@nwmissouri.edu

J. A. Hurtado
Universida del Cauca, Popayán, Colombia
e-mail: ahurtado@unicauca.edu.co

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_45

1 Introduction

In recent days, due to massive increase in the usage of mobile devices around the world. Several businesses rely on mobile applications to serve their customers and develop free apps. Mobile applications are prevalent in dating, e-commerce, education, medical, health care, recreation, transportation, social media, research, entertainment, mission-critical systems, among others. The emergence for connecting different objects through the Internet, termed as the Internet of Things (IoT), allows the machine-to-machine communication with embedded sensors. These devices collect vast amounts of data for appropriate decision making and reduce the workload on the automation. Therefore, the exponential growth of data needs to be stored and retrieved efficiently [2].

These applications are heterogeneous and require diverse resources. Cloud computing is a desirable solution with advanced computing and communication network technologies [5]. Cloud computing is a model for gathering physically distributed resources such as processors, memory, bandwidth, and storage capacity to deliver on-demand services to users [5, 13]. A cloud can provide infrastructure, platform, and software as a service to users. Over the past decade, cloud computing deals with large-scale storing and computing data in data centers. These data centers usually connect with other data centers to form a data center network and provide end-user services as a single resource. However, due to an increase in smart devices, massive growth using IoT applications and augmented virtual reality requires real-time and quick responses based on context-awareness and location data [5]. Applications that use streaming data has the high round-trip transmission time to and from the cloud. Examples of streaming data are sensors that collect the continuous data of industrial equipment, oversee the pieces of equipment's performance, identify the defects in advance, and automatically order a spare part. A dating app tracks the users' geographic location and provides partners' recommendations based on their profile. Similarly, online gaming apps track player–game interactions and provide relevant promotional offers to the player.

Latency is the delay between the users' request and the applications' response to that request, usually measured in milliseconds. Bandwidth is the amount of data transmitted through the network at a given time. Throughput is the amount of the data transferred over a certain period. Both bandwidth and throughput are measured in bits/second. For efficient network communication, the latency must be low and bandwidth should be high. Even though the communication within the data center network is efficient with low latency, communication between the cloud and end-users is challenging. Also, it lacks the context-awareness and location-awareness of the users. The Business Insider Web site [15] estimated that the usage of IoT devices would be more than 41 billion by 2027, up from about 8 billion in 2019. Thus, increase the need for a 5G network [8] to companies to transmit the big data generated by these devices to reduce the network traffic. Due to these challenges, the communication between the cloud and end-users degrades the quality of the service and experience.

The communications industry introduces [8, 22] edge computing to overcome the challenges of centralized cloud computing. Satayanarayanan et al. [18, 19] defined edge computing as computing and storing resources at the Internet's edge near IoT devices. These resources are referred to as cloudlets and fog nodes (microdata centers). The Internet's edge would decentralize the storing and processing of data from the cloud and add a middle tier between end devices and the original cloud datacenter. Researchers published literature reviews [1, 4, 9, 11, 12, 16, 21, 23], in machine learning, deep learning, IoT apps, security, and communication integrating with edge computing. In this paper, we presented an umbrella review of edge computing from an architectural perspective.

System architects operate with both tactics and patterns for making decisions addressing certain quality concerns. The architecture pattern is a well-organized strategy representing the system's high-level structure and behavior for its requirements. Similarly, a tactic is a design decision or specific ideas to accomplish the quality attributes. A group of tactics could be organized as an architectural strategy or pattern. In this paper, we draw the quality attributes of edge computing to architectural tactics and strategies. Edge computing is a decentralized architecture with robust storage and computing resources at the internet edge nearer to the end devices. Edge computing is introduced mainly to achieve quality attributes such as performance (latency, bandwidth, handling concurrent requests, accuracy), along with scalability, deployability, security, and portability. Because edge computing as architecture solutions lacks of a unified knowledge about the underlying rationale necessary for analyzing and developing edge applications, servers, and services [17], we integrate the architecture patterns and tactics [6] related with edge computing in order to codify knowledge for supporting making decisions during architectural design.

The organization of the remainder of this paper is as follows. Section 2 explains edge computing implementations and architecture tactics. Section 3 illustrates the umbrella review process. Section 4 presents the detailed analysis of the results and conclusions in the final section.

2 Related Work

This section explains the edge computing implementations of cloudlets, fog computing, and multi-access edge computing. We discussed the background of integrating edge computing with architecture patterns and tactics.

2.1 Cloudlet

A cloudlet is a decentralized, datacenter in a box closer to the end devices connected via the Internet. Cloudlets add a middle tier to the cloud and mobile devices and form a three-tier architecture to provide the computing and storage resources to the end

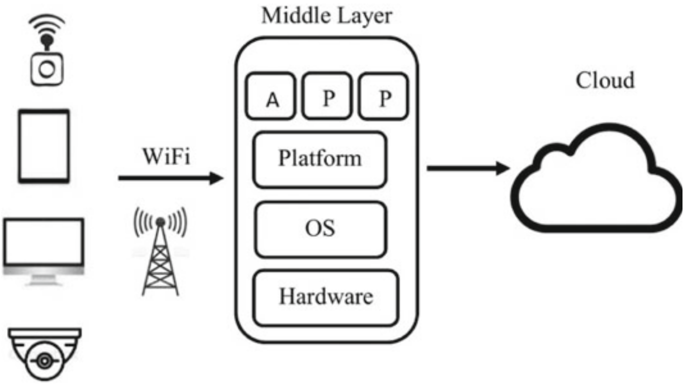


Fig. 1 Cloudlet

devices within one wireless hop [18, 19]. Cloudlets are rich in resources and focus on latency-sensitive data and services that require high bandwidth sharing between a few users. These cloudlets are self-managed, and professional attention is not required. A cloudlet has a soft state, which means it has cache data or the data stored in another place. An end mobile device acts as a client to the nearby cloudlet for fast end-to-end response time. If there is no cloudlet within the closer proximity, the mobile device may degrade and use cloud services and gain the performance after identifying another cloudlet. However, cloudlet placement is challenging [18]. Zhao et al. [30] proposed a ranking-based near-optimal algorithm placing a cloudlet to minimize the access delay of IoT applications in a software-defined network (SDN). This ranking-based algorithm outperforms the K-median clustering algorithm in the access delay.

The usage of cloudlets provides low latency, high bandwidth, and low jitter responsive end-to-end services for the nearest mobile devices [18]. The applications using AR/VR, speech recognition, machine learning, computer vision, natural language processing offload the intensive computation and storage resources to the cloudlet. The cloudlets help perform essential services during the failure of clouds due to cyber-attacks, network jamming in cyber-wars, and physical destruction of the network infrastructure due to natural disasters [18] (Fig. 1).

2.2 Fog Computing

Fog computing is a decentralized, geographically distributed computing with fog nodes between the end devices and the cloud. A fog node is powerful in storage resources and processing capability with any equipment including but not limited to routers to base stations, switches, and access points to IoT devices. Fog nodes are multiple data centers distributed geographically with the closer proximity of end devices to offload services at the edge. For a given business, the number of fog

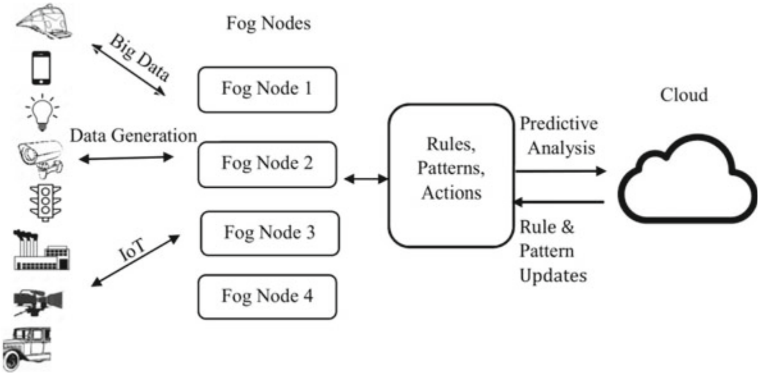


Fig. 2 Fog edge computing

nodes varies from ten to hundreds. Yi et al. [27] proposed a three-layered fog node with infrastructure, platforms and applications with appropriate Application Program Interfaces (API).

Fog computing helps analyze the latency-sensitive data at the edge rather than sending it to the cloud to achieve comprehensive data privacy and security with low operational costs. Cisco introduced the first fog node called IoX [22] to host multiple applications and analyze the data generated by various end devices. Cisco IoX fog node is heterogeneous with the business networking operating system and Linux [22]. Figure 2 shows the CISCO's fog architecture.

2.3 Multi-access Edge Computing

Edge computing in mobile devices and networks was referred to as mobile edge computing (MEC) by the European Telecommunications Standards Institute (ETSI) [8]. By moving the data-intensive storage and computation power nearer to the edge, network operators can overcome the network traffic challenges. This decentralized architecture aggregates communication technologies for IoT devices and 5G systems to IT services to the end devices. In 2016, ETSI dropped the word mobile from MEC and renamed it as multi-access edge computing [14], also abbreviated as MEC.

MEC has many use case scenarios such as vehicle to any device (V2X) communication, health care, retail business, AR/VR, accelerated video, caching services, and IoT applications. MEC provides services to various stakeholders, including Over-The-Top (OTT) players, software vendors, mobile network operators, etc. [23]. With the massive and exponential growth in IoT applications and the data generated by mobile devices, Verizon uses MEC and 5G as critical technologies for edge computing. With MEC's help, users of mobile end devices can effectively use the

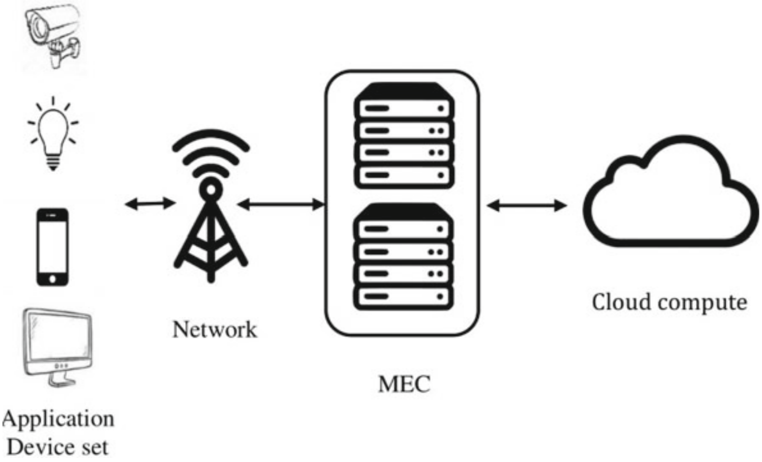


Fig. 3 Multi-access edge computing

computing and storage resources. Massive bandwidth, reduced latency, reduced computing power on the device, and minimization of the network traffic are the advantages of the MEC. Figure 3 is the MEC architecture of Verizon's edge computing.

2.4 Architecture Patterns and Tactics in Edge Computing

Khan et al. [9] discussed the significance of edge computing in real-world scenarios. They presented a new taxonomy of edge computing for cloudlets, fog nodes, and MEC. The work compares 23 primary studies to reduce latency, maximize resource utilization, ensure optimized privacy, strengthen security, and collect real-time data insights. Due to the increase in the need for 5G networks and MEC, Mao et al. [12] in their literature review focused on joint and radio computational resource management and Mach et al. [11] on computation offloading in MEC. Ni et al. [16], in their literature review, focused on securing fog nodes for IoT applications. Taleb et al. [23] focused on the 5G networks, orchestration, and deployment of edge computing. Chen and Ran [4] presented of a detailed and exhaustive deep learning techniques review to improve edge computing performance. Sittón-Candanedo and Casado-Vara [21] discussed the computing consortium reference architectures and presented a proposal for tier architecture. In addition, several researchers [1, 4, 9, 11, 12, 16, 21, 23, 29] discussed open research problems, opportunities, and challenges in edge computing. However, these studies lack the knowledge from the architecture perspective of the edge computing systems. Architecture knowledge [6] constitutes quality attributes, tactics, and strategies. We focus on integrating edge computing with architecture knowledge proposed by Harrison and Avgeriou [6].

- *Quality attributes*: are features that the system has such as reliability, security (authenticity, confidentiality, data integrity), scalability, deployability, usability, maintainability, and performance. Quality attributes satisfaction is along a scale, always viewed within the specific quality scenario where there is a required output to reach within particular boundaries of a system (at a specific state) facing an input [3].
- *A tactic* is a design decision or the sequence of actions to accomplish a design concern. Tactics are steps taken to enhance quality characteristics [3]; for instance, a tactic is defined common and abstract services in order to improve the maintainability.
- *A pattern* groups several tactics positively or negatively impacting to several quality attributes. Architecture patterns commonly describe the decomposition of modules of the system at a high and abstract level [6]. Layers, tiers, publish/subscribe are examples of architectural patterns.

3 Methodology

Our study follows a theoretical review [24] focusing on a model based on the conceptual model of Harrison and Avgeriou [6], which includes tactics, quality attributes, and architectural patterns (strategies). Given this knowledge is scattered in the industry, this paper aims to organize the architectural level to edge computing as an emergent technology.

3.1 Research Goal and Research Questions

This research's main goal is to unveil architectural concerns from secondary studies (survey, reviews, or systematic reviews) on edge computing solutions. The main research questions are oriented to identify architectural concerns:

- What quality attributes and related concerns are analyzed in these edge computing solutions?
- What tactics are unveiled from these edge computing solutions?
- What strategies are used as part of these edge computing solutions?

3.2 Research Execution

First, the primary studies selected from relevant sources were used IEEE, Springer, Elsevier, and Scopus. The research string used and adapted to each repository was "edge computing" AND architecture* AND (tactic OR "quality attribute" OR pattern) AND (survey or review). Once the papers were recovered (17), the abstract of each one was read, and duplicated works were removed, the papers were filtered, getting nine paper after using the inclusion and exclusion criteria as follow. These nine studies are annotated as "[Primary Study]" in the references.

- Inclusion criteria: secondary studies, edge computing, works analyzing some architectural concerns such as quality attributes and architectural tactics, patterns, or strategies.
- Exclusion criteria: primary studies, proposal, architectural ideas without any real assessments.

Finally, we analyzed the nine papers to answer each research question; data collected was analyzed, and the results were organized and analyzed. Each architectural aspect was evaluated considering the knowledge body conformed by Bass et al. [3], Harrison and Avgeriou [6], and Osses et al. [17].

4 Results and Analysis

4.1 *Quality Attributes in Edge Computing*

The first question related to quality attributes in edge computing shows that the main driver is the performance. The main concern is to resolve the delay in transmission and latency problems, particularly for latency-sensitive applications. Edge computing takes advantage of cloud computing but physically closer it to the end-user to achieve a quicker than cloud concerning the response time [9, 18]. However, adding an intermediary element between the devices and the cloud has consequences to other

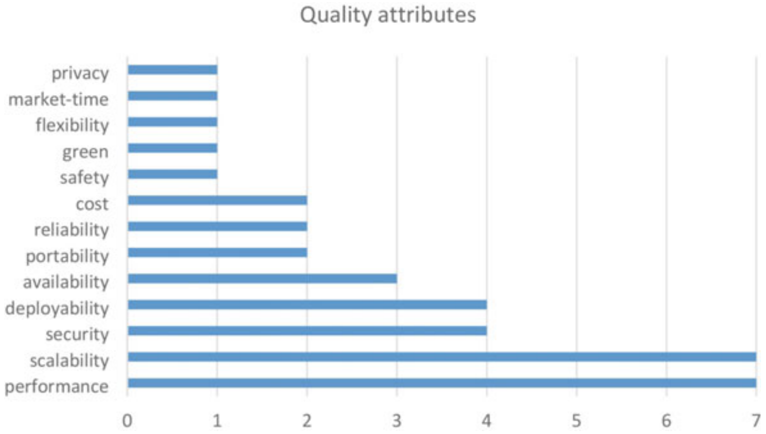


Fig. 4 Quality attributes identified

quality attributes. Thus, edge computing has other limitations for availability, security, and resource management regarding mobility, context-awareness, and location-awareness. Khan et al. [9] discussed the key specifications required to use edge computing to address these limitations. Techniques such as dynamic billing mechanism of multi-vendor systems with network parameters such as latency, bandwidth, real-time application support, resource management, the common business design for deployment and management, scalability to accommodate various IoT applications execute consistently on the heterogeneous resources, anticipate robust and resilient edge computing systems over network outrages, and security with advanced cryptography schemes [9]. The increased frequency of quality attributes taken into account in the works is shown in Fig. 4.

Other frequent quality attributes in edge computing are scalability and deployability. It is essential to consider designing a scalable architecture to accommodate various IoT applications to execute consistently using heterogeneous resources. The resource virtualization, trust-enabled technologies, and edge orchestration can achieve scalability [9]. The usage of containers over virtual machines in MEC allows portable run time services for mobile users [23]. Also, containers provide techniques for quick and secure packaging and deployment to various apps across the platform [23].

4.2 Architectural Tactics in Edge Computing

The more frequently used tactics in the literature were caching, resource management, resource allocation, and temporary storage, as shown in Fig. 5. For instance, in a MEC, one fundamental problem is how to balance the trade-off between the massive database and finite storage capacity when data caching is realized by a single

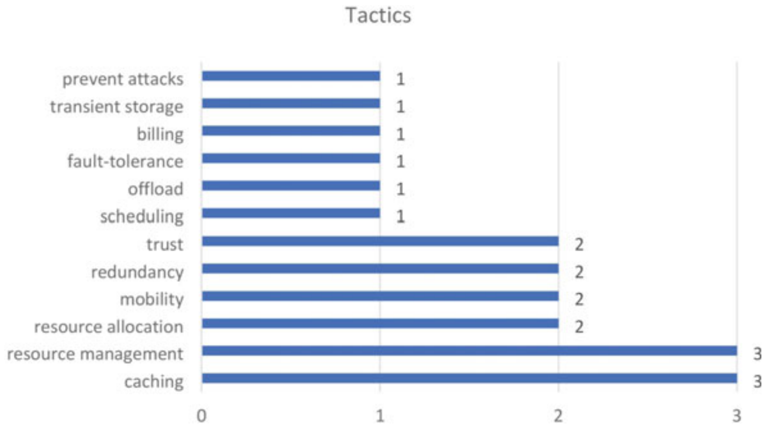


Fig. 5 Tactics identified

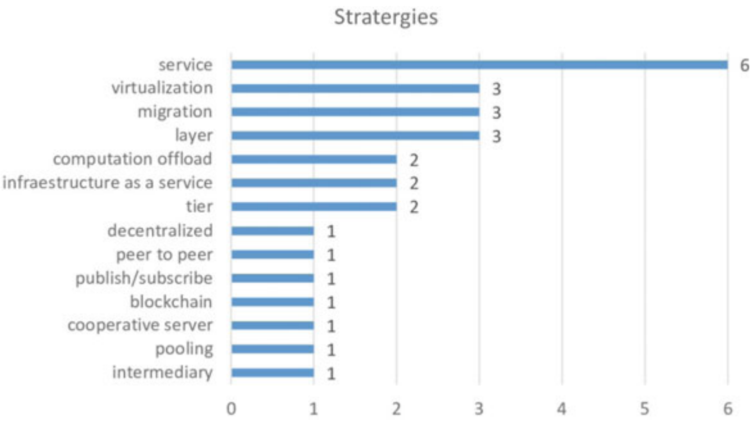


Fig. 6 Strategies identified

edge server, so temporary or transient storage appears as a general design decision. The allocation of computing resources is significantly associated with the offloaded application that allows parallelization or partitioning distributed to various computing nodes. Edge computing can offload part of the workload of the cloud. Traditionally, only the data is cached in intermediate servers, but the computations applied to the data are cached at the edge servers in the edge computing paradigm (Fig. 6).

4.3 Architectural Strategies in Edge Computing

The most common strategies found were services (software as a service, infrastructure as a service, security as a service, etc.), virtualization, process migration, intermediary, tiers, and layers. Computations as services is a strategy for migrating and executing computations offered by a well-defined service layer, an essential strategy for achieving the advantages of offloading tactic. The desired strategy was using an intermediary since edge self works as an intermediary between local devices and the cloud. This intermediary is designed typically using a tiered architecture: data center tier (cloud), large/medium edge tier, small edge tier (for instance cloudlet), near layer(for instance, fog), and the device layer. Dolui et al. [5] presents a three-tiered architecture, including an upper tier, intermediary tier, and low tier.

4.4 Tactics Achieving Quality Attributes in Edge Computing

Tactics are abstract architectural decisions for resolving a specific quality attribute. Table 1 shows as the identified tactics allow to achieve some quality attributes. The

Table 1 Quality attribute versus tactics

Attributes versus tactics	Caching	Resource management	Resource allocation	Transient storage	Computation offloading
Performance	+++	++	++	++	+/–
Scalability	+	++	+++	++	–
Deployability	o	–	––	––	+++
Security	––	o	o	––	––
Availability	+++	++	++	+	++

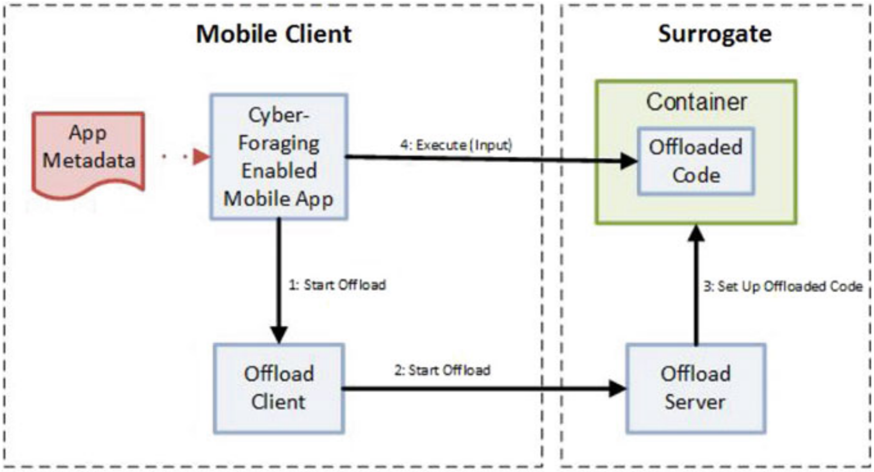


Fig. 7 Offloading tactic taken from [10]

plus sign represents the tactic that positively affects the quality attribute. The minus sign represents the corresponding tactic that negatively affects the quality attribute. The number of plus or minus varies the intensity of that particular tactic to achieve the quality attribute. The symbol 'o' represents that a particular tactic neither positively nor negatively affects the quality attribute. Consider the *Offloading computations* tactic scenario applying computation offload from a mobile device to a proxy [10]. When the end-user runs the search-enabled applications on their mobile devices, the mobile application offloads the nearest proxy's computation. Figure 7 shows the essential components of the offloading tactic with a sequence of operations. Offloading benefits are less disruption to the end-user and reduce delays and improve user experience quality. Offloading has limitations as data and computations are cloned across different processors and heterogeneous storage devices; it is a security and privacy risk [26].

Table 2 Tactics verses strategies

Tactics/strategies	Services	Migration	Virtualization	Intermediary	Layered	Tiered
Caching	x			x	x	x
Resource management	x		x			x
Resource allocation	x		x	x		x
Transient storage		x		x		
Computation offloading		x	x			

4.5 Strategies Grouping Tactics in Edge Computing

An architectural strategy can embody several tactics. Table 2 shows the identified strategies and some identified tactics; it is not a complete view about tactics and strategies related to edge computing, but the intention is that those can be taken as a start point for organizing the architecture knowledge.

For instance, the migration strategy packages the computation offloading and virtualization tactics. Yousafzai et al. [28] established a case study for processing computational offloading for IoT mobile edge computing. They developed a framework for seamlessly migrating a mobile device with limited resources toward a computing infrastructure with more available resources. The framework includes components at the user side (app migration coordinator and migration preference manager) and components at the edge side (migration manager and admission control), as depicted in Fig. 8. Yousafzai et al. [28] assessed the performance of this framework using standard benchmarks and specific workloads considering eight distinct intensity levels in order to find some association between time, workload, and power savings. This showed a significant and positive impact on the framework performance.

Computational offloading is an architectural strategy described by Sheng et al. [20], for answering questions such as: When a specific computation requires to be offloaded?, especially considering several closed servers satisfying the offloading conditions; How to select the server for executing the computation, including the virtual machine to execute the computation once the server had been executed? To address these questions, the authors propose a computation offloading strategy breaking into three steps:

1. Offloading decision making is about comparing the local computation with the offloading computation. The offloading choice is in accordance with the computation concerns like performance and energy consumption.
2. Selecting an adequate server by balancing concerns such as performance, energy consumption, and server CPU resources.
3. Schedule and execute the computations on the suitable virtual machine so that the MEC servers can perform more computation at the same time interval.

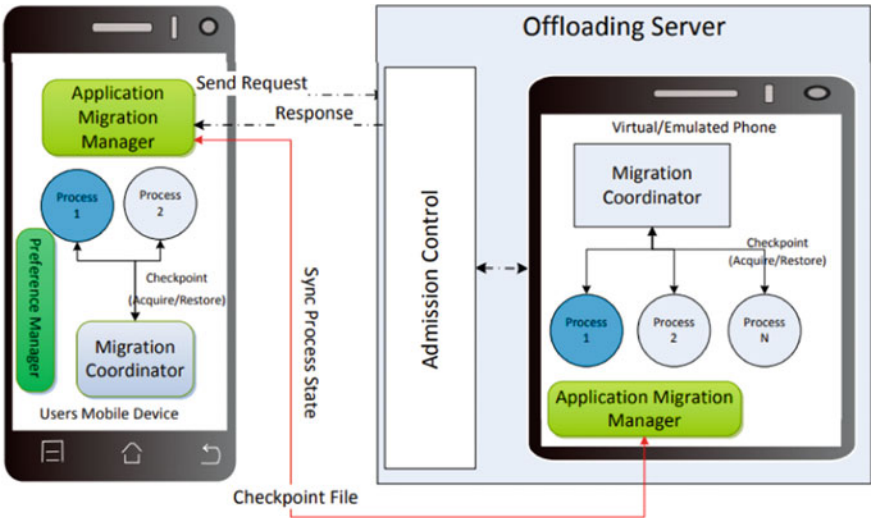


Fig. 8 Migration strategy proposed by Yousafzai et al. [28]

4.6 *Synthesis and Discussion*

We used Harrison's conceptual framework [6] to synthesize and discuss edge computing as an architecture solution following a pattern style for its description.

1. *Name*: Edge Computing as Architectural Solution
2. *Problem*: Cloud computing architecture and cloud service applications follow a centralized architecture with bottlenecks in the cloud infrastructure. This infrastructure is significantly affected when services respond to many heterogeneous end devices because of the limitations of bandwidth and the servers' workload; consequently, it introduces a high latency.
3. *General Solution*: Edge computing brings computation and storage resources to the nearer location to intensify response times and save bandwidth. Edge computing architecture is a service-based and n-tiered solution where typically is identified as an intermediary tier composed by edges nodes (edge tier) and defining specifics capabilities for each tier, including the upper tier (cloud layer), the edge tier, and lower-tier (edge devices tier).
4. *Participants*:

   (a) Cloud tier: A centralized data center and cloud computing infrastructure considering requirements for working with the edge tier (caching, computation migration, virtualization).

   (b) Edge tier: This tier contains devices that could be used as edge servers as general-purpose servers, platforms specifically deployed for the satisfy edge needs (caching, computation migration, virtualization) or specific domains (traffic, for example)

   (c) Device tier: This tier contains every end device that generates huge data while executing on the end-user application. The edge devices include IoT apps on vehicles, computers, and mobile phones.

5. *Tactics and strategies used*

   (a) Resource management and provisioning: service and provisioning (on-demand resources and migration) and placement (VM and services)
   (b) Computations offloading and task scheduling
   (c) Security and infrastructure as a service (prevents attacks and on-demand redundancy resources)

6. *Consequences*:

   (a) Ultra-low latency closing services to end-users and the smart devices, and migrating the computations to more capable virtual machines.
   (b) Edge computing requires to minimize the service downtime and guarantee high availability of edge services.
   (c) New security issues must be addressed, such as vulnerabilities related to authorization and authentication, distributed denial of service attacks, malware injection attacks, and side-channel attacks.
   (d) Scalability issues because increasing the capacity to an existing edge require increasing the inter-edge working bandwidth to avoid congestion and reduce the system capacity.

7. *Known uses*: Edge Computing Reference Architecture 2.0 from Edge Computing Consortium (ECC) and Alliance of Industrial Internet (AII). Nokia launched its own MEC platform based on the Cellular Vehicle-to-Everything (C-V2X) technology protocol to achieve a latency below 20 ms, along with a flexible deployment model [7]. IMS and Dell deployed an IoT-connected edge computing platform to obtain and maintain data from their refrigeration systems and build management systems to automatically adjust the temperature of the food, maintaining the quality standards, lessen waste and optimize refrigeration costs [25].

5 Conclusions, Limitations, and Future Work

Edge computing is a novel technology moving the services from the cloud to the most closer device. Edge computing defines a boundary between the cloud and the device tiers to resolve services' latency problems. This paper presented, in a preliminary way, edge computing as an architectural solution. We used a small set of secondary studies to extract and analyze the architectural information related to quality attributes, architectural tactics, and edge computing strategies.

The main contribution is codifying edge computing knowledge as an architectural solution for software architects' decisions. The problems can be viewed as a set of tactics and strategies and the positive and negative consequences on different quality attributes. The main key quality attribute addressed by edge architectural solution is the performance, mainly the latency concern addressed by caching, migration, and virtualization strategies. However, it introduces requirements on other quality attribute concerns such as security, deployment, and scalability. This edge tier implements the strategies but drag requirements on the other traditional tiers; for instance, offloading competitions require that services communicate the network among edge servers, edge devices.

Although this paper unveils the rationale behind edge computing from the architectural viewpoint, in the future, more extended studies are needed to achieve a more in-depth knowledge of edge computing, architectural quality attributes' tactics, and strategies.

References

1. Ahmed E, Rehmani MH (2017) Mobile edge computing: opportunities, solutions, and challenges [Primary Study]
2. Bandi A, Fellah A (2017) Design issues for converting websites to mobile sites and apps: a case study. In: 2017 international conference on computing methodologies and communication (ICCMC), pp 652–656. https://doi.org/10.1109/ICCMC.2017.8282547
3. Bass L, Clements P, Kazman R (2012) Software architecture in practice, 3rd edn. Addison-Wesley Professional
4. Chen J, Ran X (2019) Deep learning with edge computing: a review. Proc IEEE 107(8):1655–1674 [Primary Study]
5. Dolui K, Datta SK (2017) Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing. In: 2017 global internet of things summit (GIoTS), pp 1–6
6. Harrison NB, Avgeriou P (2010) How do architecture patterns and tactics interact? A model and annotation. J Syst Softw 83(10):1735–1758. https://doi.org/10.1016/j.jss.2010.04.067
7. Hilt V, Sparks K (2019) Future edge clouds. Bell Labs Tech J 24:1–17
8. Hu YC, Patel M, Sabella D, Sprecher N, Young V (2015) Mobile edge computing-a key technology towards 5G. ETSI White Paper 11(11):1–16
9. Khan WZ, Ahmed E, Hakak S, Yaqoob I, Ahmed A (2019) Edge computing: a survey. Future Gener Comput Syst 97:219–235. https://doi.org/10.1016/j.future.2019.02.050
10. Lewis G, Lago P (2015) A catalog of architectural tactics for cyber-foraging. In: 2015 11th international ACM SIGSOFT conference on quality of software architectures (QoSA), pp 53–62
11. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surv Tutor 19(3):1628–1656 [Primary Study]
12. Mao Y, You C, Zhang J, Huang K, Letaief KB (2017) A survey on mobile edge computing: the communication perspective. IEEE Commun Surv Tutor 19(4):2322–2358 [Primary Study]
13. Marapareddy R, Bandi A, Tirumala SS (2012) Cloud computing architectures: a retrospective study. J Innov Comput Sci Eng 2(1):1–5
14. Morris I (2016) ETSI drops "mobile" from MEC. Light Reading
15. Newman P. The internet of things 2020: here's what over 400 IoT decision-makers say about the future of enterprise connectivity and how IoT companies can use it to grow revenue. https://www.businessinsider.com/internet-of-things-report?r=US&IR=T
16. Ni J, Zhang K, Lin X, Shen XS (2017) Securing fog computing for internet of things applications: Challenges and solutions. IEEE Commun Surv Tutor 20(1):601–628 [Primary Study]
17. Osses F, Márquez G, Astudillo H (2018) Exploration of academic and industrial evidence about architectural tactics and patterns in microservices. In: Proceedings of the 40th international conference on software engineering: companion proceedings. ICSE '18, Association for

Computing Machinery, New York, NY, USA, pp 256–257. https://doi.org/10.1145/3183440.3194958

18. Satyanarayanan M (2017) The emergence of edge computing. Computer 50(1):30–39 [Primary Study]
19. Satyanarayanan M, Bahl P, Caceres R, Davies N (2009) The case for VM-based cloudlets in mobile computing. IEEE Pervas Comput 8(4):14–23
20. Sheng J, Hu J, Teng X, Wang B, Pan X (2019) Computation offloading strategy in mobile edge computing. Information 10(6):191. https://doi.org/10.3390/info10060191
21. Sittón-Candanedo I, Alonso RS, Corchado JM, Rodríguez-González S, Casado-Vara R (2019) A review of edge computing reference architectures and a new global edge proposal. Future Gener Comput Syst 99:278–294 [Primary Study]
22. Solutions CFC (2015) Unleash the power of the internet of things. Cisco Systems Inc
23. Taleb T, Samdanis K, Mada B, Flinck H, Dutta S, Sabella D (2017) On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Commun Surv Tutor 19(3):1657–1681 [Primary Study]
24. Turner JR, Baker R, Kellner F (2018) Theoretical literature review: tracing the life cycle of a theory and its verified and falsified statements. Hum Resour Dev Rev 17(1):34–61. https://doi.org/10.1177/1534484317749680
25. Ud Din I, Guizani M, Hassan S, Kim B, Khurram Khan M, Atiquzzaman M, Ahmed SH (2019) The internet of things: a review of enabled technologies and future challenges. IEEE Access 7:7606–7640
26. Wang F, Diao B, Sun T, Xu Y (2020) Data security and privacy challenges of computing offloading in fins. IEEE Netw 34(2):14–20
27. Yi S, Hao Z, Qin Z, Li Q (2015) Fog computing: platform and applications. In: 2015 third IEEE workshop on hot topics in web systems and technologies (HotWeb). IEEE, pp 73–78
28. Yousafzai A, Yaqoob I, Imran M, Gani A, Noor RM (2020) Process migration-based computational offloading framework for IoT-supported mobile edge/cloud computing. IEEE Internet Things J 7(5):4171–4182. https://doi.org/10.1109/JIOT.2019.2943176
29. Yousefpour A, Fung C, Nguyen T, Kadiyala K, Jalali F, Niakanlahiji A, Kong J, Jue JP (2019) All one needs to know about fog computing and related edge computing paradigms: a complete survey. J Syst Archit 98:289–330
30. Zhao L, Sun W, Shi Y, Liu J (2018) Optimal placement of cloudlets for access delay minimization in SDN-based internet of things networks. IEEE Internet Things J 5(2):1334–1344

Content-Based Recommender System for Similar Products in E-Commerce

Abhijnyan Chandra, Arif Ahmed, Sandeep Kumar, Prateek Chand, Malaya Dutta Borah, and Zakir Hussain

Abstract The recommendation system aims to provide the best suggestion to the user by analysing the buyer's interest. Most of the time, we can get good product as the recommender(s) already has/have much knowledge about the product. So, this work is about the development of a recommender system for similar products in E-commerce with two particular cases of book recommendation and movie recommendation. This work is based on content-based recommendation technique. We have experimented the content-based techniques using book data set, namely BookCrossing.csv and movie review data set IMDB_Top250Engmovies2 _OMDB_Detailed.csv. For book recommendation, we have used pivot table for finding correlation coefficient. For movie recommendation, we have implemented TF-IDF-based recommender system using CountVectorizer function of scikit-learn and cosine similarity for getting good recommendation. Our system recommends most similar books by calculating the correlation coefficient for the book entered by the user and recommends top ten most similar movies using cosine similarity for a movie entered by user. This system can be helpful for people to find books and movies of their interest. This system can reduce the effort to put on physically for getting recommendation from people as it is an automatic system for recommendation.

Keywords Content-based recommender · Recommender in E-commerce · Book recommendation · Movie recommendation

1 Introduction

The recommendation system is basically used to suggest products to the users. They are basically software agents that analyse interest and preferences of individual cus-

A. Chandra · A. Ahmed · S. Kumar · P. Chand · M. D. Borah · Z. Hussain (✉)
Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam, India
e-mail: zakir_rs@cse.nits.ac.in

M. D. Borah
e-mail: malayaduttaborah@cse.nits.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_46

tomers and make recommendation keeping it in mind. They use various techniques to find similarity among products. The quality and the content are taken into consideration by employing content filtering for recommendation. For book recommendation, an intelligent algorithm reduces the overhead of the people. This provides benefit to both the seller and the consumer creating the win-win situation. The E-commerce site to network security all demands the need for the recommended system to increase their revenue rate. Decision-making techniques employed help buyers by the strong recommendations as there are various books, as buyers sometimes cannot find the item they search for. For movies, recommendation system uses various functions simply works by using the cosine similarity matrix which we have generated by algorithm. It checks the movies which have the highest cosine similarity to the movie entered by the user and returns top ten most similar movies to it.

1.1 Motivation

Joseph Pine said 'The age of standard, universal, mass products is over. Various customers, various demands, heterogenous (personalized) products are needed'. Jeff Bezos (Amazon, CEO): '*If I have got 2 million customer I have to have 2 million shops on the web*'. Issue of selection: There are a huge number of accessible items; hence, the client needs to locate the best items among them (data over-burden). Consumers of a specific item can approach different choices for comparative items with ease. Boss goal of recommender systems: The potential things must be separated/sorted. The client sees just relevant items; therefore, personalized proposals are required. To create a framework where shoppers can make learned/taught buy on the off chance that they are given enough data of comparative items on other storefronts. When purchasers are given choices, they will be bound to pick the quality item among them.

2 Literature Review

In the work [1], the authors used probabilistic methods for the system. Naive Bayes can be used as a classifier to determine the similar items. For relevance feedback, Rocchio's algorithm can be used by taking into consideration the feedback received. Most of the recommendation systems in different domains share a common mean for the representation of user profiles and items. The major drawback of this method is that recommendation of items can be done only for a higher score when matched with user profiles, and hence, the user might only get recommendations for items that are similar to items that are already rated before.

In the work [2], a recommender system for brand new items is generated, and features from transactions are extracted. Clustering of users is done based on similarity in interests. Recent items in users clusters are considered for recommendation.

It relies on underlying common characteristics and users–items relationship. Rather than items being considered for recommendation, items are automatically matched against user's interests and are actively recommended. Users may assign false ratings that are not their real opinions and may result in undesirable outcomes.

The work in [3] involves analysis of content, learning of profiles, recommendation and analysis of emotion. Comparison of emotion as provided by the user is done to the emotion from the text as detected by the system, and accordingly recommendation is done. In work [4], content-based recommender system algorithm and privacy-preserving algorithm are applied. It uses homomorphic encryption to guard customer's private data. Cryptographic techniques cost more computationally. The work in [5] consists of Outbrain's recipe of recommendations based on content, content understanding and profile aggregation matching. Content discovery is personalized, and the recommendations express the profile of the user. When the user is new, then the recommender system does not work properly.

In the paper [6], the author used clustering algorithm. It uses the characteristics of the items and also the ratings on items that can be also considered as data for the recommendation. It requires products to be rated before recommendation. In work [7], a content-based recommendation algorithm is based on convolutional neural networks for learning resources. To obtain the latent factors from the text, the convolutional neural network can be used. In order to training the convolutional neural network, the input and output solution must be obtained first. The major benefit of the recommender system is that recommendations are personalized, and as a result, superior outcome is obtained. The proposed recommendation algorithm is based on convolutional neural networks which makes it computationally expensive. The method described in [8] involves selection of feature module, the term frequency and inverse document frequency (TF-IDF), chi square selection of features and softmax regression module. Recommendation is based on the common properties underlying and relation between items and users. Rather than items being found, they can be automatically matched with users, and actively recommendations can be done to those users. In work [9], serendipity is induced in the content-based recommender system. It consists of analysis of content, learning of profiles and recommendation. This method induces exploration in the recommender system. The idea of inducing serendipity is based on chance and hence in some cases may result in undesirable results.

In work [10], classification and regression trees are used. It involves generation of negative samples from the implicit feedback data and comparison of different learning models like random forests and gradient boosted trees. The work in [11] involves two-layer graph model, direct retrieval, the association rule mining and high degree association retrieval. This recommendation approach results in better performance than that of collaborative approach. Three different combinations of customer, transaction and product information are needed to use this method. The work in [12] involves collection of implicit feedback, construction of matrix for pseudo rating, computation of neighbours and the recommendation of items. Users may assign false ratings that are not their real opinions and may result in undesirable outcomes. For some cases, asking users for ratings is not possible. The proposed model in [13] is based on the keyword-based vector space model. Recommendations based on user queries are also available. Contents are required for content-based techniques, and thus content accessibility plays a factor.

3 Methodology

From the literature survey, it has been observed that a typical method of recommendation technique involves the following steps:

- Acquiring preference from customer's input data: There can be various types of input data like demographic data of the user, production data, transaction data, rating score and pattern data that depends on user's behaviour, etc. Analysing and processing of these data will help to build a strong user profile based on individual user preferences.
- Recommendation using proper techniques: This step includes the following procedures:
  Content analyser: The primary duty of the procedure is to analyse the content of the items. It uses the feature extraction method to extract the useful feature or information from the item.
  Profile learner: This procedure will gather information about the user inclinations and preferences and try to sum up the collected data together to develop the user profile.
  Filtering component: This procedure involves matching the features of the items with the features of the user profile and then recommending suitable and similar items for each specific user.

The techniques that can be used to develop our recommender system are as follows: TF/IDF measure, KNN algorithm, clustering methods, the artificial neural network (ANN) and correlation methods (Fig. 1).

3.1 Roadmap of the Work

Our work roadmap is shown in Fig. 2. After the analysis of various existing techniques, we are proposing a model that improves upon the most accurate existing model.

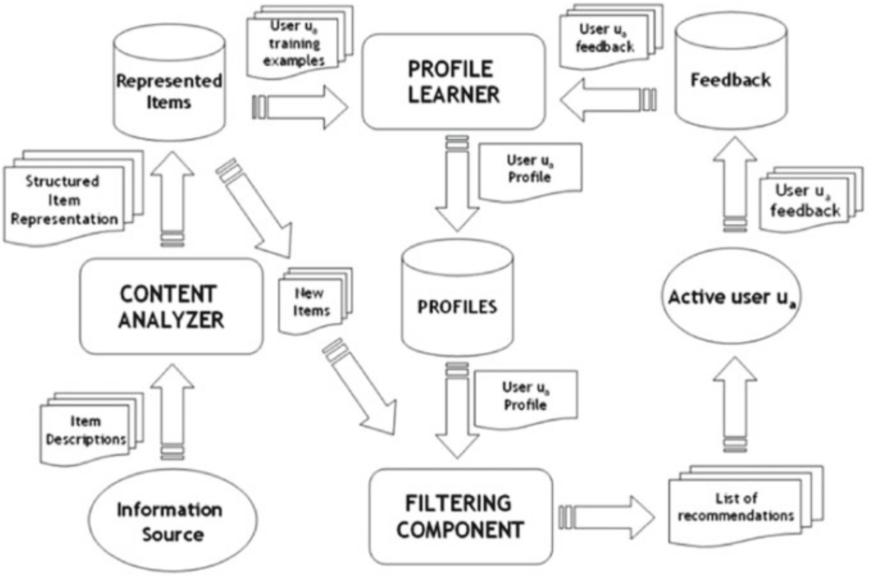


Fig. 1 Block diagram representation of content-based recommender system [1]

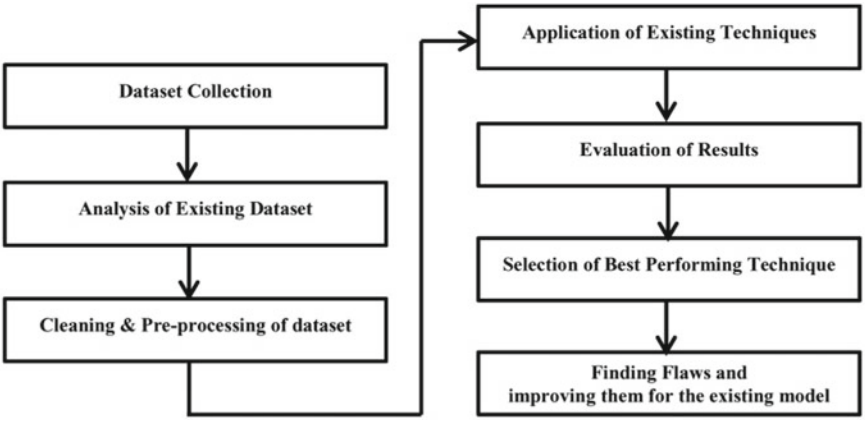

Dataset Collection
Analysis of Existing Dataset
Cleaning & Pre-processing of dataset
Application of Existing Techniques
Evaluation of Results
Selection of Best Performing Technique
Finding Flaws and improving them for the existing model

Fig. 2 Roadmap of the proposed work

3.2 Data Set Selection and Analysis

We have worked on ‘BookCrossing’ data set (Book review data set for similar book recommendation) and ‘IMDB_top250Engmovies_OMDB_detailed_Dataset’ (Movie review data set for similar movie recommendation).

The book-crossing data set contains three tables:

- BX-Users: Contains user descriptions. In this, userID column has been rounded and mapped onto integer. Data like age or location are provided if available, otherwise Null is assigned to them.
- BX-Books: ISBN values are used to identify books in data set. Invalid ISBN values are removed. This table contains information about book title, book author, year of it being published and name of publisher.
- BX-Book-Ratings: Contains information regarding different ratings given to books. Ratings given are either explicit that is given on scale of 1–10 or implicit 0.

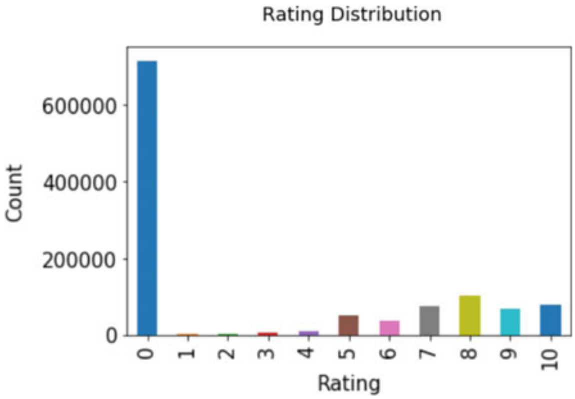
Pre-processing (cleaning, selection, integration, transformation) of book-crossing data set has been done on each table. It involves the following steps:

- In 'Books-Ratings' table, ratings are unevenly distributed, and a large number of samples contain implicit rating (0-rating) which makes the sparsity high. So, we removed those samples.
- In 'Books' table, unnecessary image URL columns are removed.
- Non-numerical values of 'yearOfPublication' column have been replaced with their actual value.
- Null values of 'publisher' attribute are replaced by 'others':
- In the 'Users' table, age distribution analysis is done, and most active users are in their 20–30s. Age below 5 and more than 100 does not make much sense. Those discrepant age values are replaced by attribute mean (Figs. 3, 4 and 5).

The movie data set 'IMDB_top250Engmovies_OMDB_detailed_Dataset' contains 25 variables and 5000 observations, in which we required only five variables and 250 observations for our model. We have performed some pre-processing of data that majorly including these steps:

- Discarded commas between the actors' full names and getting only the first three names.
- Putting the genres in a list of words.

Fig. 3 Count plot of ratings



- Merging together the first and last name for each actor and director, so it is considered as one word, and there is no mix up between people sharing a first name.
- Generated bag of words.

4 Results and Discussions

Our recommender system finds the correlation between every pair of the user and the book. We have formed a matrix using the pivot table, which helps in finding the correlation.

For our work, we have used numpy.corrcoef(x, y, rowvarr, bias, ddoff) function in Python, which returns Pearson product-moment correlation coefficient. Let 'R' be a correlation coefficient matrix and 'COV' be a covariance matrix. The relation between them is $R_{ij} = \frac{\mathrm{COV}_{ij}}{\sqrt{\mathrm{COV}_{ii} \times \mathrm{COV}_{jj}}}$; where $-1 \leq R \leq 1$.

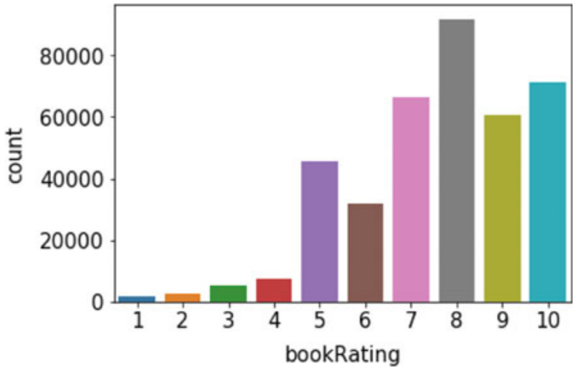


Fig. 4 Count plot after removal of 0-ratings

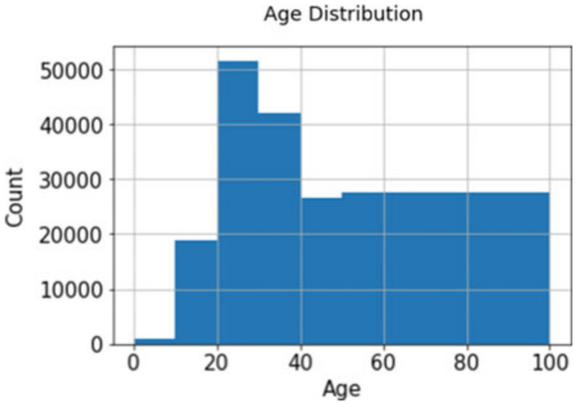


Fig. 5 Age distribution: most active users are in their 20–30s

4.1 Book Recommendation Engine

The data had several tables which were cleaned and then formatted properly in three tables. One table was of books, and others were of users and its explicit rating. We have joined these three tables and created a single table of all the fields which are of our interest. From the data frame in Appendix, we have created a pivot table of UserID, BookTitle and BookRatings. This pivot table will show information that which user has assigned what rating to which book, and then the correlation coefficient is calculated against the pivot table. Then simply for any book the user enters to get the recommendation, we find its correlation with other users' book and recommend him the new book based on the values of correlation which we have already computed. One example of getting a recommendation against one book is available in Appendix.

4.2 Movie Recommendation

We have implemented a TF-IDF-based recommender system using the CountVectorizer function of scikit-learn. After that, we used the cosine similarity measure for getting good recommendations. Scikit-learn's CountVectorizer is utilized to change corpora of text to a vector of term/token tallies. It additionally gives the capacity to pre-process the text information before producing the vector form making it a profoundly adaptable feature portrayal module for text. Likewise, it helps in pre-processing of the text information suitably just as separating extra features from the text data set.

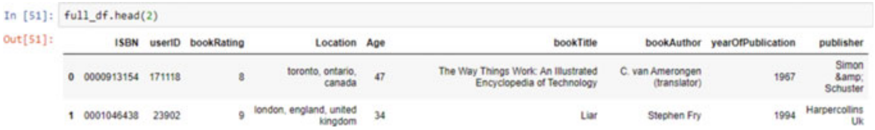
After properly pre-processing the data, we have applied CountVectorizer on the bag of words column which has all the words, and then it generated a vector against every row which in our case is movies. We have then computed the cosine similarity between all the movies to check which movies are related to or similar to each other. After creating the cosine similarity matrix, we have created a simple recommendation function which takes the title of a movie entered by a user and recommends him the top ten most similar movies to it. This recommendation function simply works by using the cosine similarity matrix which we have generated previously. It checks the movies which have the highest cosine similarity to the movie entered by the user and returns the top ten most similar movies to it.

5 Conclusion and Future Work

For this work, we are using the content-based technique to develop a recommendation system for books and movies. From the literature survey, we discovered some of the models used presently and also their drawbacks. We acquired the data set ‘Book-Crossing data set’ for books and ‘IMDB_Top250Engmovies2_OMDB_Detailed.csv’ data set for movies. We have implemented a model that yields the most accurate recommendation for books and movies, respectively. We obtained the data set required for training and testing the recommender system and performed pre-processing on the data set. For book recommendation, we have merged three tables of the data set and created a single table of all the fields to form a pivot table. Then the correlation coefficient is calculated against the pivot table. Based on the values of correlation which we have already computed, we found its correlation with other books and recommend him similar books. For movie recommendation, we used CountVectorizer on the bag of words column on pre-processed data to compute the cosine similarity between all the movies for generating the recommendation. We have used the cosine similarity matrix which we have generated previously. It checks the movies which have the highest cosine similarity to the movie entered by the user and returns top ten most similar movies to it.

6 Appendix

See Figs. 6, 7, 8, 9, 10, 11 and 12.

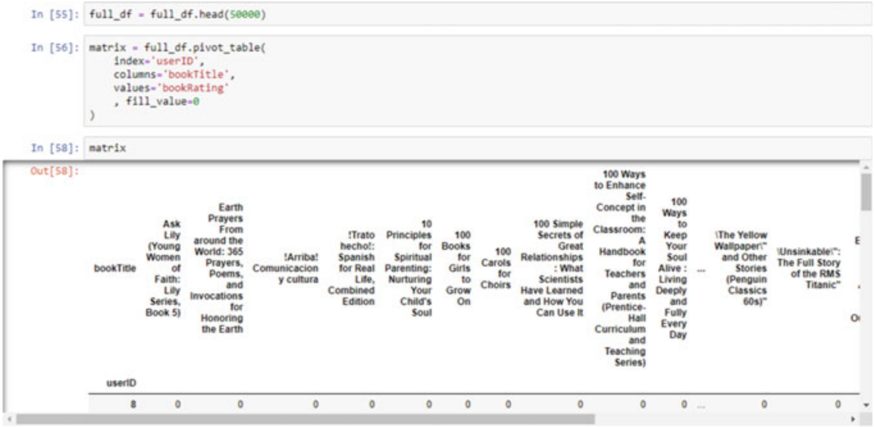


```
In [51]: full_df.head(2)
```

Out[51]:

	ISBN	userID	bookRating	Location	Age	bookTitle	bookAuthor	yearOfPublication	publisher
0	0000913154	171118	8	toronto, ontario, canada	47	The Way Things Work: An Illustrated Encyclopedia of Technology	C. van Amerongen (translator)	1967	Simon &amp; Schuster
1	0001046438	23902	9	london, england, united kingdom	34	Liar	Stephen Fry	1994	Harpercollins Uk

Fig. 6 Visualization of data frame

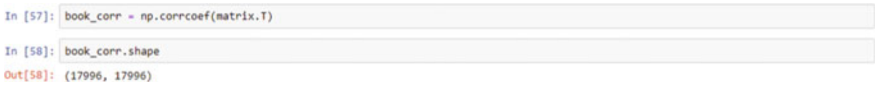


```
In [55]: full_df = full_df.head(50000)

In [56]: matrix = full_df.pivot_table(
             index='userID',
             columns='bookTitle',
             values='bookRating'
             , fill_value=0
         )

In [58]: matrix
```

Out[58]:

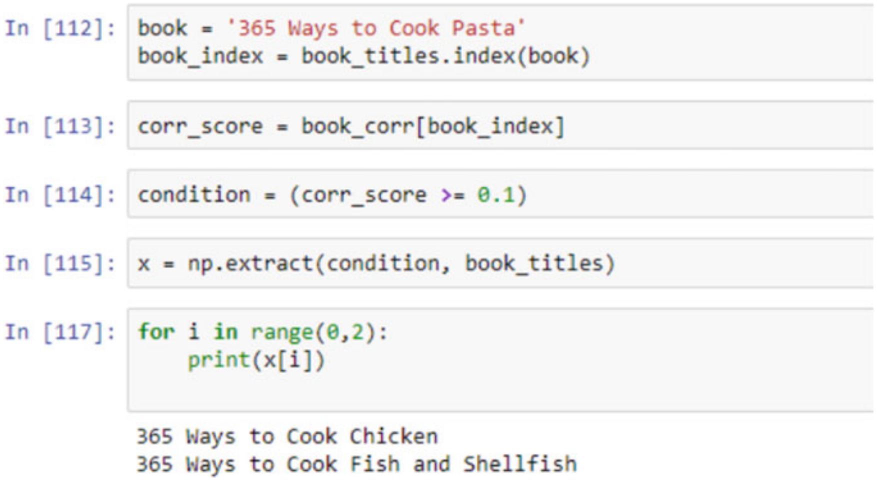
bookTitle	Ask Lily (Young Women of Faith: Lily Series, Book 5)	Earth Prayers From around the World: 365 Prayers, Poems, and Invocations for Honoring the Earth	!Arriba! Comunicacion y cultura	!Trato hecho!: Spanish for Real Life, Combined Edition	10 Principles for Spiritual Parenting: Nurturing Your Child's Soul	100 Books for Girls to Grow On	100 Carols for Choirs	100 Simple Secrets of Great Relationships : What Scientists Have Learned and How You Can Use It	100 Ways to Enhance Self-Concept in the Classroom: A Handbook for Teachers and Parents (Prentice-Hall Curriculum and Teaching Series)	100 Ways to Keep Your Soul Alive : Living Deeply and Fully Every Day	...	\The Yellow Wallpaper\" and Other Stories (Penguin Classics 60s)"	\Unsinkable\": The Full Story of the RMS Titanic"
userID													
8	0	0	0	0	0	0	0	0	0	0	...	0	0

Fig. 7 Matrix formed using pivot table



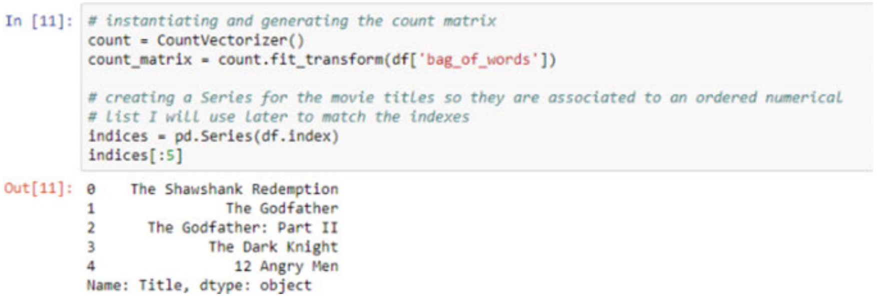
```
In [57]: book_corr = np.corrcoef(matrix.T)

In [58]: book_corr.shape
Out[58]: (17996, 17996)
```

Fig. 8 Use of correlation coefficient



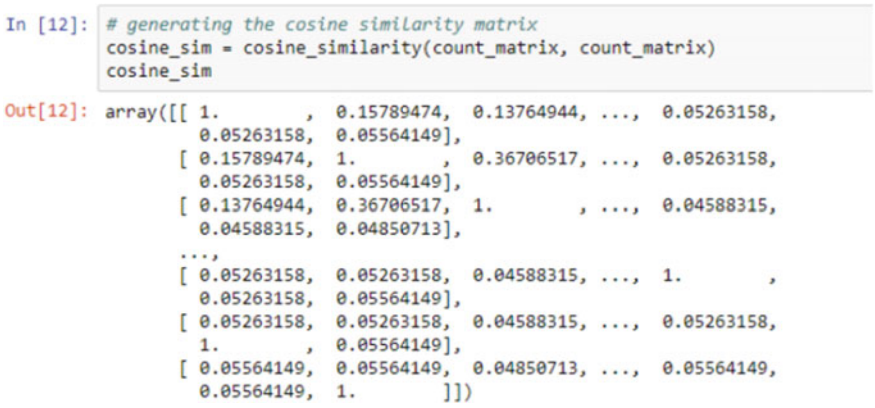
```
In [112]: book = '365 Ways to Cook Pasta'
          book_index = book_titles.index(book)

In [113]: corr_score = book_corr[book_index]

In [114]: condition = (corr_score >= 0.1)

In [115]: x = np.extract(condition, book_titles)

In [117]: for i in range(0,2):
              print(x[i])

          365 Ways to Cook Chicken
          365 Ways to Cook Fish and Shellfish
```

Fig. 9 Example of book recommendation



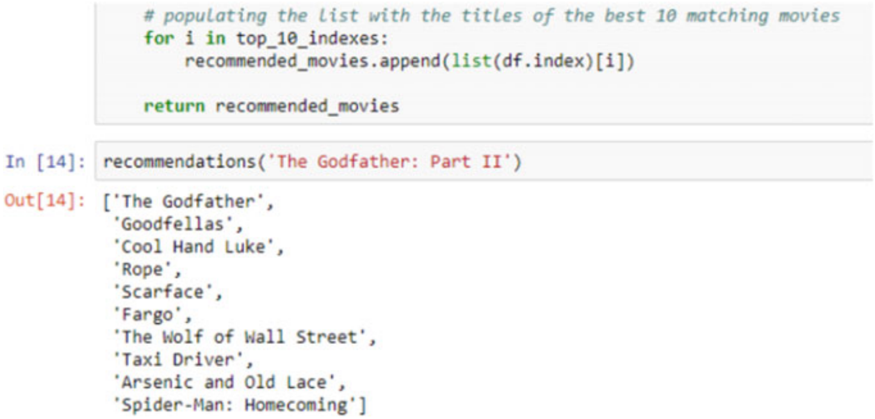
```
In [11]: # instantiating and generating the count matrix
         count = CountVectorizer()
         count_matrix = count.fit_transform(df['bag_of_words'])

         # creating a Series for the movie titles so they are associated to an ordered numerical
         # list I will use later to match the indexes
         indices = pd.Series(df.index)
         indices[:5]
Out[11]: 0    The Shawshank Redemption
         1                The Godfather
         2       The Godfather: Part II
         3              The Dark Knight
         4                 12 Angry Men
         Name: Title, dtype: object
```

Fig. 10 Finding TF-IDF weight using CountVectorizer



```
In [12]: # generating the cosine similarity matrix
         cosine_sim = cosine_similarity(count_matrix, count_matrix)
         cosine_sim

Out[12]: array([[ 1.        ,  0.15789474,  0.13764944, ...,  0.05263158,
                  0.05263158,  0.05564149],
                [ 0.15789474,  1.        ,  0.36706517, ...,  0.05263158,
                  0.05263158,  0.05564149],
                [ 0.13764944,  0.36706517,  1.        , ...,  0.04588315,
                  0.04588315,  0.04850713],
                ...,
                [ 0.05263158,  0.05263158,  0.04588315, ...,  1.        ,
                  0.05263158,  0.05564149],
                [ 0.05263158,  0.05263158,  0.04588315, ...,  0.05263158,
                  1.        ,  0.05564149],
                [ 0.05564149,  0.05564149,  0.04850713, ...,  0.05564149,
                  0.05564149,  1.        ]])
```

Fig. 11 Generating the cosine similarity matrix



```
    # populating the list with the titles of the best 10 matching movies
    for i in top_10_indexes:
        recommended_movies.append(list(df.index)[i])

    return recommended_movies

In [14]: recommendations('The Godfather: Part II')

Out[14]: ['The Godfather',
          'Goodfellas',
          'Cool Hand Luke',
          'Rope',
          'Scarface',
          'Fargo',
          'The Wolf of Wall Street',
          'Taxi Driver',
          'Arsenic and Old Lace',
          'Spider-Man: Homecoming']
```

Fig. 12 Getting recommendation against one movie

References

1. Lops P, de Gemmis M, Semeraro S (2011) Content-based recommender systems: state of the art and trends. Springer, Boston
2. Jian C, Jian Y, Jin H (2005) Automatic content-based recommendation in e-commerce. In: 2005 IEEE international conference on e-Technology, e-Commerce and e-Service, pp 748–753
3. Narducci F, de Gemmis M, Lops P (2015) A general architecture for an emotion-aware content-based recommender system. In: Proceedings of the 3rd workshop on emotions and personality in personalized systems 2015. EMPIRE'15. ACM, New York, pp 3–6. https://doi.org/10.1145/2809643.2809648
4. Erkin Z, Beye M, Veugen T, Lagendijk RL (2012) Privacy-preserving content-based recommender system. In: Proceedings of the on multimedia and security (MM&Sec'12). ACM, New York, pp 77–84
5. Tamir I, Bass R, Kobrinsky G, Brutman B, Lempel R, Dayagi Y (2016) Powering content discovery through scalable, realtime profiling of users' content preferences. In: Proceedings of the 10th ACM conference on recommender systems. RecSys'16. ACM, New York, pp 399–400. https://doi.org/10.1145/2959100.2959111
6. Xu B, Zhang M, Pan Z, Yang H (2005) Content-based recommendation in e-commerce. In: Computational science and its applications—ICCSA 2005. Springer, Berlin, pp 946–955
7. Shu J, Shen X, Liu H, Yi B, Zhang Z (2018) A content-based recommendation algorithm for learning resources. Multimedia Syst 141–144
8. Wang D, Liang Y, Xu D, Feng X, Guan R (2018) A content-based recommender system for computer science publications. Knowl Based Syst 157:1–9
9. Iaquinta L, Gemmis MD, Lops P, Semeraro G, Filannino M, Molino P (2008) Introducing serendipity in a content-based recommender system. IEEE Computer Society, Washington, DC
10. Zhao Q, Zhang Y, Friedman D, Tan F (2015) E-commerce recommendation with personalized promotion. In: Proceedings of the 9th ACM conference on recommender systems. RecSys'15. ACM, New York, pp 219–226. https://doi.org/10.1145/2792838.2800178
11. Huang Z, Chung W, Chen H (2004) A graph model for E-commerce recommender systems. J Am Soc Inf Sci 55
12. Lee TQ, Park Y, Park YT (2008) A time-based approach to effective recommender systems using implicit feedback. Expert Syst Appl 34(4):3055–3062. https://doi.org/10.1016/j.eswa.2007.06.031
13. Philip S, Shola PB, John AO (2014) Application of content-based approach in research paper recommendation system for a digital library

Extractive Summarization of Indian Legal Documents

M. N. Satwick Gupta, N. L. Siva Narayana, V. Sai Charan, Kunam Balaram Reddy, Malaya Dutta Borah, and Deepali Jain

Abstract A legal document is generally very long and possess hierarchical structure which makes it very difficult to get a quick understanding of these documents. Legal practitioners generally engage legal experts for making summaries of lengthy documents which is a very costly and time-consuming process. Recently, automatic summarization systems have been proposed to deal with this problem, which has the potential to be much more efficient at simplifying complicated legal documents. In this work, web scrapping is used for extraction of legal documents. Then, a comparative analysis is done on these documents using various extractive summarization approaches. From the experimental results, it has been found that the graph-based approaches performed well in general. Another important finding is that, in addition to the frequency of words, other information such as context information, lexical information is also equally important for summarization and can improve the performance of summarization systems.

Keywords Extractive summarization · Legal document summarization · Comparative analysis

1 Introduction

In today's era where lots of data is present on the web, it becomes very difficult to go through the whole data and get the useful information from it. So, if short-hand versions are available of those data, then it will be very helpful and also save lot of time. The process of removing unimportant sentences from a document while keeping the important sentences so as to shorten the length of document is called text summarization.

M. N. Satwick Gupta (✉) · N. L. S. Narayana · V. S. Charan · K. B. Reddy · M. D. Borah · D. Jain
National Institute of Technology Silchar, Assam, India 788010
e-mail: malayaduttaborah@cse.nits.ac.in

D. Jain
e-mail: deepali_rs@cse.nits.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_47

Text summarization becomes very important in the field of law where documents are easily available through online sources such as The Judgement Information System (JUDIS) [1], Indian Kanoon [2]. These documents are very lengthy and very difficult to understand even for legal practitioners. Therefore, text summarization tools can be very helpful for legal people and also for common citizens where they can easily access summary of any case. In this work, Indian legal judgement documents, which are present online, are extracted using web scrapping because there is no Indian legal summarization dataset publicly available. Several algorithms are applied on the extracted dataset, and their Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scores are compared. Finally, a comparative analysis is done to investigate into the results of these classical algorithms. Text summarization is an innovative technique to help legal practitioners to have the quick understanding of complicated legal documents.

Based on output of summary, text summarization can be of two types: (1) extractive summarization and (2) abstractive summarization. Summaries which are formed out of important sentences from a document are called extractive summarization while when the human experts write summary of a document in their own words is called abstractive summarization. Several research works had been done on extractive summarization which are based on analysis of frequency of words [3–7]. Many graph-based approaches are also proposed for extractive summarization which perform better than frequency-based approaches [8–11]. One of the approach based on vector space model (VSM) is also proposed for extractive summarization [12]. There have been several neural network-based approaches for extractive summarization [13, 14]. There have been also several works on abstractive summarization which are based on recurrent neural networks (RNN) deep learning (DL) approaches. Some of them include [15–18].

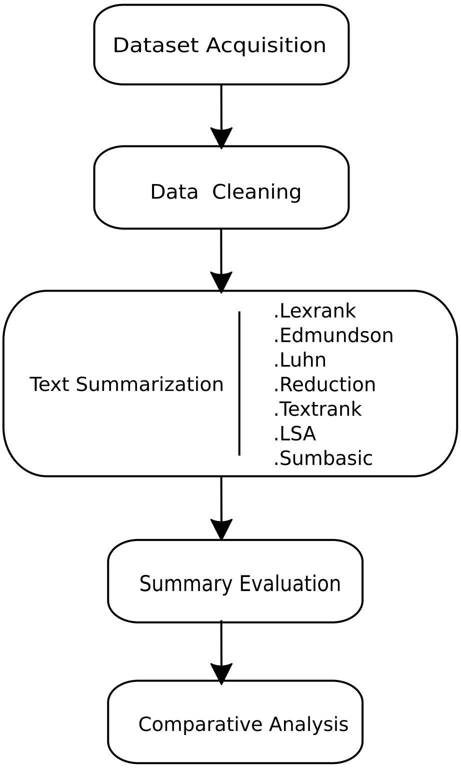
However, abstractive summarization techniques do not perform well on legal documents due to their complex and lengthy structure, and they also require a large training time [19]. Several works have been done in the past on legal document summarization such as [20–23]. Several legal tools have also been proposed such as LetSum [24], Kaftie [25], HAUSS [26], and CaseSummarizer [27].

The paper is organized in the following manner: Methodology of the work is described in Sect. 2 in detail. Experimental results are given in Sect. 3 which are then discussed in Sect. 4. The conclusion of the paper is given in Sect. 5 with the findings of this work along with the future work directions.

2 Methodology

The methodology consists of five major steps which are data acquisition, in which data is collected. Then, data cleaning is performed. After that, several classical approaches have been applied on the collected dataset which is then evaluated using ROUGE score with respect to reference summaries. Finally, based on scores,

Fig. 1 Flowchart of comparative analysis on legal dataset



comparison of techniques has been done. The detailed description of the methodology is given in the following sub-sections. The flowchart of this work in order to perform summarization and comparative analysis is shown by Fig. 1

2.1 Dataset Description

Scrapping is done in order to extract data from web. For scrapping purpose, Python requests and Python BeautifulSoup (bs4) library are used. For extracting summaries, three websites have been used which are law brief [28], Cyber Blog India [29], and Law Times Journal [30]. Indian Kanoon [2] website has been used for extracting the documents corresponding to those extracted summaries. This is done in order to make labelled dataset and so that evaluation can be done for the generated summaries. After extraction, the dataset is stored in the *.txt* format. Then, preprocessing of the dataset is done. Several preprocessing steps such as removing unwanted space, unwanted text, applying regular expressions, removal of unwanted headers, and make sure all the characters are in UTF-8 encoding are done.

2.2 Techniques

Various classical extractive approaches are applied. Some graph-based algorithms like TextRank, LexRank, and reduction and some heuristic-based approaches like Luhn and Edmundson, statistical-based approach such as SumBasic and latent semantic analysis (LSA) are used for performing extractive summarization.

- **TextRank**: in the TextRank algorithm [9], a text graph is formed, where sentences are the vertices and weights are the edges of this graph. Similarity score between two sentences helps in determining the weights. Then, ranking of the sentence is done based on the final score attached with each vertex.
- **LexRank**: another unsupervised approach known as LexRank [11] finds the importance of sentence based on eigenvector centrality, thereby ranking of sentences is done.
- **Reduction**: an another graph-based approach is reduction technique [8] in which unwanted phrases are removed from the tree of a sentence, and this forms a summary. The whole idea is based on the "graph reduction". Information such as lexical context is taken into consideration for determining the important phrase in a sentence.
- **Luhn**: the idea of word frequency and its position is used by HP Luhn [3] for text summarization.
- **Edmundson**: Edmundson [4] used other information such as cue words, title, location in addition to word frequency for finding the important sentences and thus forms a summary.
- **Latent Semantic Analysis (LSA)**: the idea of semantic relations is used by LSA technique [5] which is based on singular value decompositon (SVD).
- **SumBasic**: SumBasic[6] works on the idea that most frequent words are important for summarization than less frequent words.

2.3 Summary Evaluation

The performance of the generated summaries are evaluated with respect to the reference summaries using ROUGE which are the recall-based metric [31]. In this work, three variants of ROUGE have been considered. Overlapping of unigrams is ROUGE-1, while ROUGE-2 is the overlapping of bigrams. ROUGE-L considers the longest subsequence occurring in n-grams. Equation 1 shows the definition of ROUGE-N.

$$\text{ROUGE-N} = \frac{\sum_{S \in RS_{set}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in RS_{set}} \sum_{gram_n \in S} Count(gram_n)} \tag{1}$$

where S is a sentence, RS_{set} is the reference summaries $gram_n$, $count_{match}$ is the maximum n-grams matched in a candidate summary and a set of reference summaries, and n is n-gram length. Here, N stands for n-gram's length.

Finally, all the techniques are compared in order to analyse their performance on the extracted legal dataset.

3 Experimental Results and Analysis

The discussed summarization techniques are applied on the legal judgement documents that have been collected using scraping. The experiments were performed on Windows 10 × 64 machine with the help of freely available Python-based NLP packages: Gensim [32] and Sumy [33]. The experimental results using the algorithms are shown in Tables 1, 2, and 3, respectively. Figures 2, 3, and 4 depict the pictorial representation of the results on the law brief, Law Times Journal and law case datasets.

Figure 2 shows the results on law brief dataset. From the figure, it has been shown that LexRank achieves the best 0.447 ROUGE-1 score, Edmundson achieves the best 0.254 ROUGE-2 score, and Luhn achieves the best 0.379 ROUGE-L score .

Figure 3 shows the results on Law Times Journal dataset. The results shows that LexRank achieves the best ROUGE scores on Law Times Journal dataset with ROUGE-1, ROUGE-2, and ROUGE-3 of 0.422, 0.213, and 0.347, respectively.

Table 1 Different extractive text summarization performance (law brief cases)

Technique	ROUGE-1	ROUGE-2	ROUGE-L
LexRank	**0.447**	0.253	0.370
Edmundson	0.434	**0.254**	0.372
Luhn	0.421	0.243	**0.379**
Reduction	0.418	0.205	0.367
TextRank	0.410	0.218	0.368
LSA	0.400	0.182	0.353
SumBasic	0.341	0.142	0.329

Bold refers to the highest score

Table 2 Different extractive text summarization performance (Law Times Journal)

Technique	ROUGE-1	ROUGE-2	ROUGE-L
LexRank	**0.422**	**0.213**	**0.347**
Edmundson	0.387	0.197	0.325
Luhn	0.383	0.192	0.320
Reduction	0.387	0.202	0.329
TextRank	0.401	0.206	0.337
LSA	0.411	0.198	0.325
SumBasic	0.390	0.158	0.307

Bold refers to the highest score

Table 3 Different extractive text summarization performance (cyber cases)

Technique	ROUGE-1	ROUGE-2	ROUGE-L
LexRank	**0.339**	**0.144**	**0.289**
Edmundson	0.247	0.128	0.253
Luhn	0.257	0.124	0.250
Reduction	0.257	0.131	0.260
TextRank	0.278	0.137	0.272
LSA	0.279	0.126	0.247
SumBasic	0.231	0.124	0.223

Bold refers to the highest score

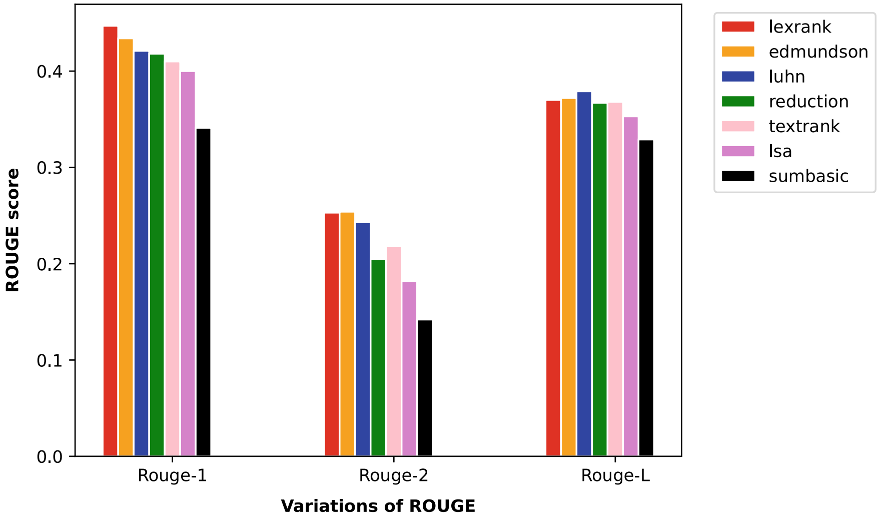


Fig. 2 ROUGE scores on law briefs dataset

Figure 4 shows the results on cyber cases. From the results, it has been shown that LexRank achieves the best ROUGE-1 score of 0.339, TextRank achieves the best ROUGE-2 and ROUGE-L score of 0.144 and 0.289, respectively, on case law dataset.

The results have shown that LexRank which is a graph-based algorithm has performed better in almost all three cases. The reason for its good performance is that eigenvector centrality is used by this algorithm for finding important sentences. It has also been shown that Edmundson has also performed good in terms of ROUGE-2. The reason for its good performance is that the algorithm considers other information also such as cue words, location, title in addition to keyword frequency. Luhn has also performed good in terms of ROUGE-L because ROUGE-L is a longest common subsequence-based metric for which the position- and context-related information is important which luhn considers for finding important words and, hence, sentences for summarization.

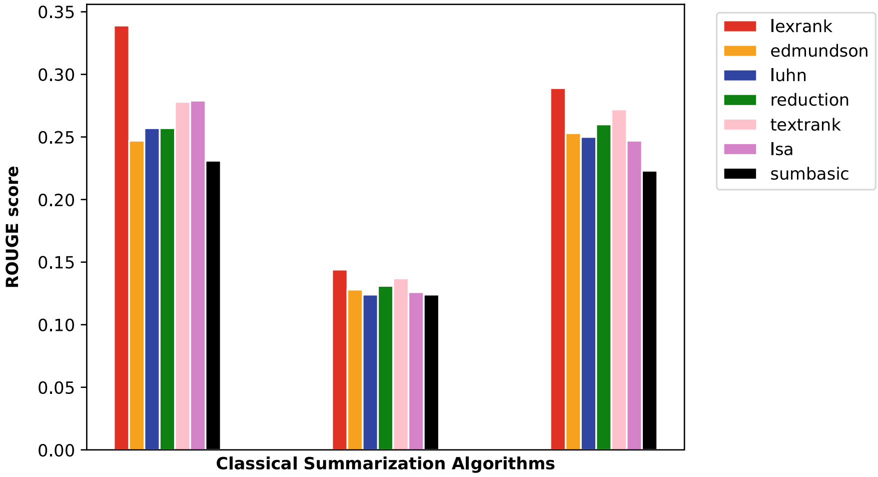


Fig. 3 ROUGE scores on law briefs dataset

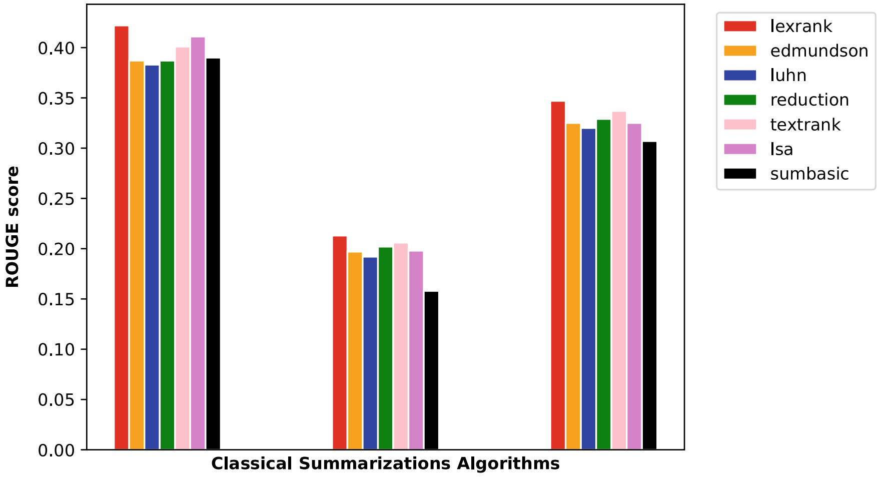


Fig. 4 ROUGE scores on law briefs dataset

4 Discussion

From the results, it has been shown that LSA and SumBasic are not able to perform as well as other algorithms. The reason is that they utilize frequency-based information which is such a limited information in order to perform summarization. For example SumBasic assigns a probability distribution to each words, assigns weight to the sentences, and ends in picking up the highest scoring sentence as per highest probability distribution of words in a sentence. LSA involves formation of document-term matrix which utilizes the information from words. So, these techniques use the word distribution in a unigram manner and do not consider other important information like context with other words. Therefore, this is the reason for their low performance as compared to algorithms like Luhn, Edmundson which also takes into account information from other words in its context and also the information like cue words, location, title in addition to word frequency, in case of Edmundson. So, it is obvious for Edmundson to perform better as compared to Luhn. These algorithms make use of concept like eigenvector centrality which makes these algorithms extremely powerful for ranking of sentences.

It has also been shown from the results that graph-based approaches like LexRank and TextRank perform better as compared to frequency-based approaches. This is due to the fact that these algorithms also consider sentence similarity and not just depend upon words only. One of the algorithms called as reduction algorithm also performed better than frequency-based approach. The key idea of this algorithm is to remove unnecessary information from a sentence which is being considered. So, based on the several information being collected at the leaf node, it removes sub-trees from a tree, which do not have any useful information and, hence, are not useful for summary. The algorithm makes use of syntactic knowledge, context, and probability information for removing unwanted phrases from a sentence.

The two main findings of this comparative analysis work where the exploration of several classical algorithm is done on the collected Indian legal judgement documents can be summarized as follows:

- It has been shown that graph-based algorithms perform well as compared to frequency-based algorithms in general.
- Only frequency-based information is not sufficient for effective summarization, since, in that case, it can ignore many useful information like context information, syntactic knowledge, lexical information.

This work is only limited to classical extractive summarization algorithms applied in the field of law. However, the recent approaches, such as neural network-based extractive and abstractive summarization, need further exploration in this legal domain, where supervised signals can be very helpful for summarization, which are typically provided in the benchmark datasets. Also, the performance of abstractive summarization goes down with the increase in the length of text, which is the case with legal documents. This underexplored area needs to be investigated further.

5 Conclusion and Future Work

A comparative analysis of several classical algorithms is done in this work on the dataset that has been collected via web scrapping. From the results, it has been found that graph-based approaches performed well in general. Also, considering only frequency of words may not so effective for summarization. It would be better if other information like context information, syntactic information, and lexical information is also considered for performing summarization, which may really help in enhancing the performance of summarization systems and, hence, help legal practitioners.

The supervision-based summarization techniques need to be further explored in this legal domain, where the techniques make use of gold summaries for making better prediction models for summarization. The study of such type of machine learning-based techniques on this collected dataset will be a part of future work.

References

1. The judgement information system. Last accessed on 20 Sept 2020. http://164.100.79.153/judis/
2. Indian Kanoon. Last accessed on 7 July 2020. https://indiankanoon.org/
3. Luhn HP (1958) The automatic creation of literature abstracts. IBM J Res Dev 2(2):159–165
4. Edmundson HP (1969) New methods in automatic extracting. J ACM (JACM) 16(2):264–285
5. Steinberger J, Jezek K (2004) Using latent semantic analysis in text summarization and summary evaluation. Proc ISIM 4:93–100
6. Nenkova A, Vanderwende L (2005) The impact of frequency on summarization. Microsoft Research, Redmond, Washington, Tech Rep MSR-TR-2005, 101
7. Haghighi A, Vanderwende L (2009) Exploring content models for multi-document summarization. In: Proceedings of human language technologies: the 2009 annual conference of the North American chapter of the association for computational linguistics, pp 362–370 (June)
8. Jing H (2000) Sentence reduction for automatic text summarization. In: Sixth applied natural language processing conference, pp 310–315 (Apr)
9. Mihalcea R, Tarau P (2004) Textrank: bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing, pp 404–411) (July)
10. Erkan G, Radev D (2004) Lexpagerank: prestige in multi-document text summarization. In: Proceedings of the 2004 conference on empirical methods in natural language processing, pp 365–371 (July)
11. Erkan G, Radev DR (2004) Lexrank: graph-based lexical centrality as salience in text summarization. J Artif Intell Res 22:457–479
12. Kågebäck M, Mogren O, Tahmasebi N, Dubhashi D (2014) Extractive summarization using continuous vector space models. In: Proceedings of the 2nd workshop on continuous vector space models and their compositionality (CVSC), pp 31–39 (Apr)
13. Narayan S, Cohen SB, Lapata M (2018) Ranking sentences for extractive summarization with reinforcement learning. arXiv preprint arXiv:1802.08636
14. Nallapati R, Zhai F, Zhou B (2017) Summarunner: a recurrent neural network based sequence model for extractive summarization of documents. In: Thirty-first AAAI conference on artificial intelligence (Feb)
15. Rush AM, Chopra S, Weston J (2015) A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685
16. Paulus R, Xiong C, Socher R (2017) A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304
17. Gehrmann S, Deng Y, Rush AM (2018) Bottom-up abstractive summarization. arXiv preprint. arXiv:1808.10792
18. Zhang J, Zhao Y, Saleh M, Liu PJ (2019) Pegasus: pre-training with extracted gap-sentences for abstractive summarization. arXiv preprint arXiv:1912.08777
19. Bhattacharya P, Hiware K, Rajgaria S, Pochhi N, Ghosh K, Ghosh S (2019) A comparative study of summarization algorithms applied to legal case judgments. In: European conference on information retrieval. Springer, Cham, pp 413–428 (Apr)

20. Saravanan M, Ravindran B, Raman S (2006) Improving legal document summarization using graphical models. Front Artif Intell Appl 152:51
21. Saravanan M, Ravindran B, Raman S (2008) Automatic identification of rhetorical roles using conditional random fields for legal document summarization. In: Proceedings of the third international joint conference on natural language processing: volume-I
22. Galgani F, Compton P, Hoffmann A (2012) Citation based summarisation of legal texts. In: Pacific Rim international conference on artificial intelligence. Springer, Berlin, Heidelberg, pp 40–52 (Sept)
23. Galgani F, Compton P, Hoffmann A (2012) Combining different summarization techniques for legal text. In: Proceedings of the workshop on innovative hybrid approaches to the processing of textual data, pp 115–123 (Apr)
24. Farzindar A, Lapalme G (2004) Letsum, an automatic legal text summarizing system. Legal knowledge and information systems. JURIX, pp 11–18
25. Pham SB, Hoffmann A (2004) Incremental knowledge acquisition for building sophisticated information extraction systems with Kaftie. In: International conference on practical aspects of knowledge management. Springer, Berlin, Heidelberg, pp 292–306 (Dec)
26. Galgani F, Compton P, Hoffmann A (2014) HAUSS: incrementally building a summarizer combining multiple techniques. Int J Human-Comput stud 72(7):584–605
27. Polsley S, Jhunjhunwala P, Huang R (2016) Casesummarizer: a system for automated summarization of legal texts. In: Proceedings of COLING 2016, the 26th international conference on computational linguistics: system demonstrations, pp 258–262 (Dec)
28. Lawbriefs. Last accessed on 20 Sept 2020. https://lawbriefs.in/
29. Cyber India Blog. Last accessed on 20 Sept 2020. https://cyberblogindia.in/category/case-summary/
30. Law Times Journal. Last accessed on 20 Sept 2020. https://lawtimesjournal.in/category/case-summary/
31. Lin CY (2004) Rouge: a package for automatic evaluation of summaries ACL. In: Proceedings of workshop on text summarization branches out post conference workshop of ACL, pp 2017–05
32. Rehurek R, Sojka P (2010) Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks
33. Sumy: automatic text summarizer. Last accessed on 20 Sept 2020. https://github.com/miso-belica/sumy

Privacy Enhanced Registered Devices for Fine-Grained Access Control

Puneet Bakshi and Sukumar Nandi

Abstract Aadhaar is one of the largest biometric identity systems in the world which aims to assign a unique digital identity to each resident of India. A resident can authenticate himself using his Aadhaar number and a biometric. Biometric is a very sensitive data, and registered devices were introduced to eliminate the use of stored biometrics. Each registered device is supposed to have a unique identity and to follow suggested best practices to ensure security. With the proliferation of digital services, the use of registered devices is expected to grow more in near future. Although registered device is a good initiative, attribute-based access and communication may not be very efficient at present. In the present world of ubiquitous computing, IoT and 5G, a device may need to provide an assurance that it possesses a certain set of requisite attributes rather than some serial or a model number. Furthermore, owner of the device may not want to reveal device identity to protect his privacy since device identity may be correlated with owner's identity. In this paper, we present an efficient scheme of privacy enhanced attribute-based registered devices for fine-grained access control using attribute-based signature.

Keywords Registered devices · Aadhaar · Privacy

1 Introduction

In year 2009, Government of India entrusted Unique Identity Authority of India (UIDAI) [1] with a mission to assign a unique 12-digit identity number to each resident of India. This number is called *Aadhaar* [2]. So far, UIDAI has assigned Aadhaar number to more than 80% of the country population [3]. To receive an Aadhaar number, a resident needs to register himself with UIDAI, for which he needs to provide personal and demographic information including biometrics (fingerprints

P. Bakshi (✉) · S. Nandi
Indian Institute of Technology, Guwahati, Assam, India
e-mail: b.puneet@iitg.ac.in

S. Nandi
e-mail: sukumar@iitg.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_48

and iris scan), mobile number and email address. After verification of the details, the resident is issued a 12-digit Aadhaar number. Aadhaar-based authentication is being used by various nationwide online services such as eKYC [4], DigiLocker [5] and eSign [6].

Each time a resident needs to use an Aadhaar-based service, he needs to authenticate himself by providing his biometric (or a one time password (OTP) for low-risk activity). Biometric is a sensitive data, and an utmost care should be taken to ensure security of devices used to store and transmit biometric. UIDAI introduced *registered devices* [7] with three major requirements. First is that every device must have a unique identifier for traceability, analytics and fraud management. Second is that the device uses its private key to sign biometric within the device. This is to eliminate the use of stored biometrics. Third is that the service provided by the device provider must be certified by UIDAI. UIDAI acknowledges *public devices* also but mandates that necessary security measures must be taken to ensure security of devices. Registered devices are categorized in two levels (L0 and L1) based on their compliance level. In L0 compliance devices, signing and encryption of biometric are done within the software in host operating system. In this case, software should ensure the security of private keys from other users and applications in the system. In L1 compliance devices, signing and encryption of biometric are done within the secure device storage area. In this case, the key is secured from other users and applications. An L0 device is identified by $\texttt{idHash} = \texttt{SHA256(DeviceSerialNo)}$, and an L1 device is identified by $\texttt{idHash} = \texttt{DeviceSerialNo}||\{\texttt{DeviceSerialNo}; \texttt{Timestamp}\}_{\texttt{CI}_\texttt{k}}$, where $\texttt{CI}_\texttt{k}$ is the Chip Identity Certificate stored in secure storage area of the device. Each device provider has a unique key called device provider private key, and each device has a unique key called device private key. The corresponding public keys are signed by UIDAI and the device provider, respectively.

Each device provider provides a registered device service which provides two APIs, namely `capture` and `device_info`. When an application needs to capture biometric of a person, the device captures required biometric records of the person using `capture` API and signs the same to obtain $\texttt{B}_{\texttt{Si}} = \{\texttt{SHA256(biorecord}_\texttt{i}\texttt{)|timestamp||UniqueDe viceCode}\}_{\texttt{D}_\texttt{PRK}}$, where $\texttt{D}_\texttt{PRK}$ is the device private key and `i` ranges from one to number of biometric records. Now, a Personal Identity Data (PID) block [8] is created which includes device identity (`idHash`), biometric records ($\texttt{B}_{\texttt{Si}}$), device provider identifier, registered device service version and device model identifier. `device_info` is used to obtain device-specific information. Device encrypts the PID block using a dynamic session key, which is further encrypted with UIDAI public key. The encrypted PID block is sent to the application.

At present, registered devices are supposed to be connected locally to the system and are primarily designed to handle biometric data. Although at present, this model may be suffice, with the proliferation of connected devices and online services, registered devices may soon become ubiquitous, required to operate remotely and to process other sensitive personal data as well. In a ubiquitous world of registered devices, an application may want to query and use a valid registered device having a specific set of attributes rather than a registered device having a specific random string of serial number or a model number. Since identity of the device may be correlated with identify of its owner, owner of the device may not want to disclose identity of the device to protect his privacy. Owner may just want to let the device be recognized as a valid registered device having a certain set of attributes. Since present model of registered devices is based on PKI infrastructure, it has some inherent limitations such as it attests device identity to a message and not to the device attributes. Furthermore, the present model of registered devices can be improved in providing attribute-based discovery and usage of the device while still maintaining the device privacy.

In attribute-based signature [9], signer is represented by a set of attributes rather than his identity, and the signature assures that the signer holds a specific set of attributes. Although attribute-based signature seems a natural choice here, the scheme is still not used widely, and a careful and efficient construction is still one of the major issues. Moreover, during usage of the device, multiple authorities may participate in assigning attributes to the device. For example, device attributes may be assigned by manufacturer, operational attributes may be assigned by hosting agency, context attributes may be assigned by hosting service, usage attributes may be assigned by operations team and the user himself, etc.

This paper presents a scheme to implement privacy enhanced fine-grained access control devices in which multiple authorities may participate to arrive at an attribute-based token which can be used to assure the validity and the possession of a specific set of attributes. The token can be reused till it expires and is collusion resistant.

Rest of this paper is organized as follows. Section 2 presents some of the related work, Sect. 3 presents a brief on some of the required preliminaries, Sect. 4 presents our proposed model, Sect. 5 presents an informal security analysis, and Sect. 7 presents the conclusion and the future work.

2 Related Work

Public key infrastructure (PKI) was introduced by Diffie and Hellman in 1977 [10]. Most of the traditional secure systems are built using PKI. In PKI, every subject has a private key and a corresponding public key. A trusted entity called Certificate Authority certifies public key of the subject and issues him a Digital Signature Certificate (DSC). PKI has an additional overhead of management of DSCs.

Identity-based encryption (IBE) was introduced by Shamir in 1984 [11] at a broad level without details on its construction. In year 2001, Boneh and Franklin [12] introduced a possible construction of the same using bilinear pairing. In IBE, each subject has a well-defined identity. A trusted entity known as Private Key Generator (PKG) hosts a master public key and generates a private key for a given identity. The corresponding public key can be derived from the identity and the master public key. IBE has a benefit over PKI in that there is no overhead of certificate management, and public key of a subject can be derived directly from identity of the subject.

Attribute-based encryption (ABE) can be divided into two categories: key policy attribute-based encryption (KP-ABE) and ciphertext policy attribute-based encryption (CP-ABE). The first was introduced by Sahai et al. [13], and the second was introduced by Benthencourt et al. [14]. In KP-ABE, private key is linked with the access policy, and ciphertext is linked with a set of attributes. A receiver can decrypt a ciphertext only if access policy in his private key satisfies attributes in the ciphertext. In CP-ABE, private key is linked with a set of attributes, and ciphertext is linked with an access policy. A receiver can decrypt a ciphertext only if attributes in his private key satisfy access policy in the ciphertext.

In attribute-based signature (ABS), a signature is based on signer's attributes and implies possession of certain attributes by the signer. ABS facilitates the signer to sign a document proving possession of certain attributes without even revealing his attributes. Guo et al. [15] proposed an initial ABS scheme in which they used strong extended Diffie–Hellman assumption to prove their claims. Later, Tan et al. [16] presented a weakness in Guo's scheme and explained that the scheme is weak for partial key replacement attacks. Later, Maji et al. [17] proposed a scheme which can use different kinds of gates such as AND gates, OR gates or threshold gates.

3 Preliminaries

In this section, some of the prerequisites are presented. The prerequisites include bilinear pairing, decisional bilinear Diffie–Hellman (DBDH) assumption, strong extended Diffie–Hellman assumption and access Structures.

3.1 Bilinear Pairings

In a pairing-based cryptography [18], a bilinear map is $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of prime order p such that discrete log problem is hard on them. Let g_1 be generator of $\mathbb{G}_1$ and g_2 be generator of $\mathbb{G}_2$, a group element P belongs to first group, $P \in \mathbb{G}_1$, and a group element Q belongs to second group $Q \in \mathbb{G}_2$. A bilinear pairing should have three properties

1. The map should be bilinear: $e(P^a, Q^b) = e(P, Q)^{ab}$.
2. The map should not degenerate: $e(g_1, g_2) \neq 1$.
3. The map $e(P, Q)$ should be efficiently computable.

3.2 Decisional Bilinear Diffie–Hellman (DBDH) Assumption

Let $\mathbb{G}$, $\mathbb{G}_T$ be two p-order cyclic groups such that $p > 2^\lambda$, where $\lambda \in \mathbb{N}$, g is the generator of $\mathbb{G}$, $e\colon \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is bilinear pairing map and $a, b, c, z \in \mathbb{Z}_p$ are random numbers. According to DBDH assumption [19], $(g, g^a, g^b, g^c, e(g, g)^{abc})$ cannot be efficiently differentiated from $(g, g^a, g^b, g^c, e(g, g)^z)$ in a polynomial time.

3.3 Strong Extended Diffie–Hellman (S-EDH) Assumption

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two p-order cyclic groups such that $p > 2^\lambda$ where $\lambda \in \mathbb{N}$, g_1 and g_2 are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, $\mathbb{R}_{x,y}$ is an oracle which takes input $k \in \mathbb{Z}_p^*$ and outputs $\langle g_1^r, g_2^{1/(x+r)}, g_2^{1/(k+r), g_2^{yr}} \rangle$ for a random $r \in \mathbb{Z}_p^*$. Let $\mathcal{A}$ be a probabilistic polynomial time adversary and B be the set of queries it can make to $\mathbb{R}_{x,y}$. S-EDH assumption [20] states that for all $\mathcal{A}$, all $v, c \in \mathbb{Z}_p^*$ and all $a \in \mathbb{G}_1$ such that $a \neq 1, \Pr[x \overset{R}{\leftarrow} \mathbb{Z}_p : \mathcal{A}^{O_{xy}}(g, g^x, g_2, g_2^y) = (k, a, a^x, a^r, g_2^{1/(x+r)}, g_2^{1/(k+r)}, g^{yr}) | k \notin B] \prec 1/\text{poly}\,(k)$.

3.4 Access Structure

Let $P_1, P_2, ..., P_n$ be the set of entities. For a subset $S_1 \subseteq 2^{P_1, P_2, ..., P_n}$ to be a monotone subset, it is required that if $S_2 \in S_1$ and $S_2 \subseteq S_3$, then $S_3 \in S_1$. A monotone set S_1 of $\{P_1, P_2, .., P_n\}$ is an access structure [21], which implies $S_1 \subseteq 2^{P_1, P_2, ..., P_n} \setminus \{\phi\}$. Authorized sets are the sets present in S_1, and unauthorized sets are the sets not present in S_1. A monotone access structure constructed as an access tree is generally used to represent an access policy.

In an access tree $\mathcal{T}$, a non-leaf node x is described by the number of its child nodes num_x and a threshold value k_x. For example, for an OR gate, $k_x = 1$, and for an AND gate, $k_x = \text{num}_x$. A leaf node x in an access tree $\mathcal{T}$ is described by an attribute, and threshold value for a leaf node is one, $k_x = 1$. $\text{parent}(x)$ represents parent of the node x, and $\text{attr}(x)$ represents attribute of a leaf node x and is defined only for the leaf nodes. $\text{index}(x)$ represents ordering of the node x among its sibling nodes.

A subtree of $\mathcal{T}$ which is rooted at node x is represented by $\mathcal{T}_x$. If a set of attributes λ satisfies a subtree $\mathcal{T}_x$, it is represented as $\mathcal{T}_x(\lambda) = 1$. For every non-leaf node x, $\mathcal{T}(y_i)$ is computed for every child node y_i of x. If at least k_x child nodes return 1, the parent node x returns $\mathcal{T}_x(\lambda) = 1$. For every leaf node x, $\mathcal{T}_x(\lambda) = 1$ if $\text{attr}(x) \in \lambda$.

4 Our Construction

In this section, we describe the proposed construction of privacy enhanced registered devices for fine-grained access control using attribute-based signature.

An attribute is represented by a descriptive string and has an associated private key and a corresponding public key. A private key can be any integer, and a public key is a point on the chosen group. The proposed scheme introduces two entities. First is *Attribute Management Authority of India (AMAI)* which manages the whole set of device attributes and assigns a range of attributes to individual agencies to manage the range further. Second is *Attribute Service Provider (ATSP)*, which manages its assigned range of attributes by choosing private keys for each attribute in the range. Since during usage of the device, attributes to a device can be assigned by multiple entities, there can be multiple ATSPs such as device manufacturer, device firmware provider, host software, host agencies and user itself. However, the scheme assumes only one AMAI.

4.1 Attribute Management Authority of India (AMAI)

AMAI executes a `setup(k)` procedure to initialize its parameters. k is used to choose two suitable cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order p where p is a prime number. The groups are chosen such that the discrete logarithm problem is hard on them. AMAI chooses generator elements from each group. Let g_1 and g_2 be the generators of $\mathbb{G}_1$ and g_2, respectively. Now, the AMAI chooses a bilinear map $e\colon \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ such that the bilinear Diffie–Hellman problem is hard on it. Now, from the universe of attributes $\mathbb{U} = \{1, 2, ...n\}$, AMAI delegates management of a specific subset of attributes to a specific category of ATSPs. For example, AMAI can assign attributes $1-50$ to itself, $51-100$ to manufacturers, $101-150$ to host agencies, etc. These attributes are represented by $\mathbb{A}_{\text{AMAI}}$, $\mathbb{A}_{\text{ATSP}_1}$ and $\mathbb{A}_{\text{ATSP}_2}$ and so on. User is also given an ownership of some of the attributes. These attributes may represent this consent, the intended purpose of consuming his data, the expected user of his data, etc.

AMAI generates random numbers $\gamma \in_R \mathbb{Z}_p$ and $t_i \in_r \mathbb{Z}_{\text{p}}$ for each attribute $i \in \mathbb{A}_{\text{AMAI}}$, computes its private key SK and derives the public key MPK and a master public key MPK.

$$\begin{aligned} \text{SK} &= \{\gamma, \{t_i\}_{\forall i \in \mathbb{A}_{\text{AMAI}}}\} \\ \text{PK} &= \{g^{\gamma}, \{g^{\gamma}\}_{\text{PVTA}}, \{T_i, \{T_i\}_{\text{PVTA}}\}_{\forall i \in \mathbb{A}_{\text{AMAI}}}\} \\ \text{MPK} &= \{\mathbb{A}_{\text{AMAI}}, \mathbb{A}_{\text{ATSP}_1}, \mathbb{A}_{\text{ATSP}_2}, \mathbb{A}_{\text{ATSP}_i}, \mathbb{G}_1, \mathbb{G}_2, g_1, g_2, p\} \end{aligned} \tag{1}$$

where $T_i = g^{t_i}$ and $\{T_i\}_{\text{PVTA}}$ are T_i signed by another private key PVTA of AMAI.

4.2 Attribute Service Providers (ATSP)

All ATSPs need to register with AMAI. ATSP$_i$ generates random numbers $\alpha \in_R \mathbb{Z}_p$ and $t_i \in_R \mathbb{Z}_p$ for each attribute $i \in \mathbb{A}_{\text{ATSP}_i}$, derives public keys $T_i = g_i^t$ and sends them to AMAI for attestation. AMAI verifies validity of the ATSP$_i$, signs the received public keys using its private key PVTA and sends them back to ATSP$_i$. ATSP$_i$ now has its private key ASK$_i$ and corresponding public key APK$_i$.

$$\begin{aligned} \text{ASK}_i &= \{\alpha, \{t_i\}_{\forall i \in \mathbb{A}_{\text{ATSP}_i}}\} \\ \text{ASK}_i &= \{g^{\alpha}, \{g^{\alpha}\}_{\text{PVTAA}}, \{T_i, \{T_i\}_{\text{PVTAA}}\}_{\forall i \in \mathbb{A}_{\text{ATSP}_i}}\} \end{aligned} \tag{2}$$

4.3 Attribute-based Private Key

During usage of the device, a device is assigned attributes from multiple ATSPs such as the manufacturer, the firmware agency, the host agency and the user. Each ATSP creates an access subtree representing device attributes it has assigned to the device. These access subtrees are combined to form a common access tree. All attributes from a single ATSP are assumed to be in one access subtree. Refer Fig. 1 for an illustration of an access tree $\mathcal{T}$.

A typical procedure for a device with identifier ID$_i$ to generate its attribute-based private key against an access tree $\mathcal{T}_j$ is illustrated in Algorithm 1. IDT$_{ij}$ represents ID$_i$ and $\mathcal{T}_j$ collectively. The device calls PullKeyAll(IDT$_{\texttt{ij}}$, K) API (refer Algorithm 3) of AMAI to retrieve attribute-based private key components from each participating ATSP by calling PullKeyAll(IDT$_{\texttt{ij}}$, K) API (refer Algorithm 2) of each participating ATSP. Two helper functions GetATSP($\mathcal{T}$) and genParitalKey(IDT$_{\texttt{ij}}$, K, r) are used in these algorithms. GetATSP($\mathcal{T}$) returns a set of ATSPs which contributed in assigning some attributes in $\mathcal{T}$. $L(\mathcal{T}) = \text{leaves}(T) \cap \mathbb{A}_{\text{ATSP}_i}$.

ATSPs use genParitalKey(IDT$_{\texttt{ij}}$, K, α) to compute their part of attribute-based private key. This procedure works as follows. Let λ represent a set of attributes assigned by ATSP$_i$ to device with identity ID$_i$. For root node R, a polynomial q_R is chosen with degree $d_R = k_R - 1$, where k_R is the threshold value of the root node. A random number α is assigned to $q_R(0)$ such that $q_R(0) = \alpha$, and rest of the d_R points are chosen randomly to define the polynomial q_R completely. Let k_x represent the threshold value of node x. Now, for each child node x of root node, a polynomial q_x

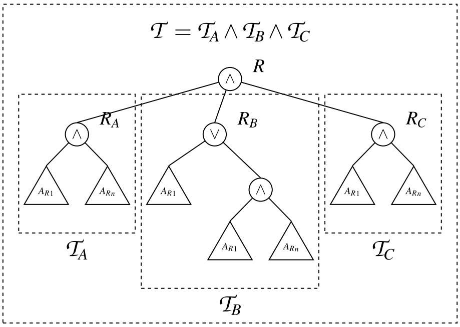


Fig. 1 Example of an access tree

with degree d_x which is equal to $k_x - 1$ is chosen. $q_{\text{parent}}(x)(\text{index}(x))$ is assigned to $q_x(0)$ such that $q_x(0) = q_{\text{parent}(x)}(\text{index}(x))$ and other d_x points are chosen randomly to define the polynomial q_x completely. The same process is used to generate polynomial for each node (including the leaves). Now, for each leaf node x, $M_{1_x} = K^{q_x(0)/t_i}$ is computed where $i = \text{att}(x)$, $x \in \lambda$ and K is a group element. Attribute-based private key assigned by ATSP$_i$ to the device is $\langle M_{1_x} = K^{q_x(0)/t_i}, M_2 = K^{\alpha} \rangle$. Each participating ATSP provides its part of attribute-based private key in a similar way.

Algorithm 1 Device : GenPvtKey

```
Require: $\langle \mathrm{IDT_{ij}} \rangle$
  $\mathrm{r} \in_{\mathrm{R}} \mathbb{Z}_{\mathrm{p}}$
  $\mathrm{K} \leftarrow \mathrm{g^r}$
  $\langle \mathrm{M_1, M_2} \rangle \leftarrow$ AMAI : PullKeyAll($\mathrm{IDT_{ij}}$, K)
  $\mathrm{IDT_{ij}} =$
    $\mathrm{ID_{ij}} : \mathrm{ID_i}, \mathcal{T}_j$
    $\mathrm{M_1} : \mathrm{M_{1_{AMAI}}} \cup \mathrm{M_{1_{ATSP_1}}} \cup \mathrm{M_{1_{ATSP_2}}} \cup \ldots$
    $\mathrm{M_2} : \mathrm{K}^{\alpha}, \mathrm{K}^{\beta}, \mathrm{K}^{\gamma}, \ldots$
  return
```

Algorithm 2 ATSP : PullKey

```
Require: $\langle \mathrm{IDT_{ij}}, \mathrm{K} \rangle$
  $\alpha \in_{\mathrm{R}} \mathbb{Z}_{\mathrm{p}}$
  $\langle \mathrm{M_1, M_2} \rangle \leftarrow$ genParitalKey($\mathrm{IDT_{ij}}$, K, $\alpha$)
    $\mathrm{M_{1_x}} = \mathrm{K}^{\mathrm{q_x(0)/t_i}} \quad \forall \mathrm{x} \in \mathrm{L}(\mathrm{IDT_{ij}} \rightarrow \mathrm{T})$
    $\mathrm{M_2} = \mathrm{K}^{\alpha}$
  return $\langle \mathrm{M_1, M_2} \rangle$
```

Algorithm 3 AMAI : PullKeyAll

```
Require: $\langle \mathrm{IDT_{ij}}, \mathrm{K} \rangle$
  $\mathrm{M_1} = \mathrm{M_2} = \phi$
  ATSP $\leftarrow$ GetATSP($\mathrm{IDT_{ij}} \rightarrow \mathrm{T}$)
  while ATSP $\neq$ empty do
    $\mathrm{ATSP_i} \leftarrow$ DEQUEUE(ATSP)
    Call API of $\mathrm{ATSP_i}$ API :
      $\langle \mathrm{M_1}\prime, \mathrm{M_2}\prime \rangle \leftarrow$ PullKey($\mathrm{IDT_{ij}}$, K)
    $\mathrm{M_1} = \mathrm{M_1} \cup \mathrm{M_1}\prime$
    $\mathrm{M_2} = \mathrm{M_2}, \mathrm{M_2}\prime$
  end while
  return $\langle \mathrm{M_1, M_2} \rangle$
```

4.4 Token Generation

Generation of private key is a costly operation since it involves computation and retrieval from multiple participating agencies and hence doing so for each request may not be very efficient. For better efficiency, a reusable token can be used which contains in it the private key and other parameters such as token expiry date from all participating ATSP $\in$ GetATSP($\mathcal{T}$) where GetATSP($\mathcal{T}$) represents all ATSPs which has some contribution of attributes in $\mathcal{T}$. For requests with same access tree, same token can be reused till it expires. A token is generated by mutual collaboration of all participating ATSPs in arriving at a common group element $K \in \mathbb{G}_1$. ATSP pulls an updated value of K using `PullK(IDT_ij, K)` and pushes the updated value of K using `PushK(IDT_ij, K)`. `GenCommonK(IDT_ij, K)` lets AMAI facilitates arrive at a common group element K. Device initiates token generation using `GenTok(IDT_ij, K)` API. Details of these functions are explained in Algorithms [4–7]. At the end, a token ABT_{ij} is generated in the device.

$$\mathrm{ABT}_{ij} = \begin{cases} \mathrm{IDT}_{ij} & = \mathrm{ID}_i, \mathcal{T}_j \\ M_1 & = \bigcup\limits_{\forall k} M_{1_{\mathrm{ATSP}_k}} \quad \mid \mathrm{ATSP}_k \in \mathrm{GetATSP}(\mathcal{T}_\mathrm{j}) \\ M_2 & = K^{\alpha},\, K^{\beta},\, K^{\gamma},\, \ldots \mid \alpha,\beta,\gamma,\ldots \in \text{Secrets with } \mathrm{ATSP}_k \end{cases} \tag{3}$$

Algorithm 4 ATSP : PullK

```
Require: $\langle \mathrm{IDT_{ij}, K} \rangle$
  $\mathrm{r} \in_\mathrm{R} \mathbb{Z}_\mathrm{p}$
  Store mapping : $\mathrm{IDT_{ij}} \leftrightarrow \mathrm{r}$
  return $\mathrm{K^r}$
```

Algorithm 5 AMAI : GenCommonK

```
Require: $\langle \mathrm{IDT_{ij}, K} \rangle$
  ATSP $\leftarrow$ GetATSP($\mathrm{IDT_{ij}} \rightarrow$ T)
  for all elements $\mathrm{ATSP_i}$ in ATSP do
    K $\leftarrow$ $\mathrm{ATSP_i}$.PullK($\mathrm{IDT_{ij}}$, K)
  end for
  ATSP $\leftarrow$ GetATSP($\mathrm{IDT_{ij}} \rightarrow$ T)
  for all elements $\mathrm{ATSP_i}$ in ATSP do
    $\mathrm{ATSP_i}$.PushK($\mathrm{IDT_{ij}}$, K)
  end for
  return K
```

Algorithm 6 ATSP : PushK

```
Require: $\langle \mathrm{IDT_{ij}}, \mathrm{K} \rangle$
  $\mathrm{r} \in_\mathrm{R} \mathbb{Z}_\mathrm{p}$
  Update mapping : $\mathrm{IDT_{ij}} \leftrightarrow \mathrm{K}$
  return
```

Algorithm 7 Device : GenTok

```
Require: $\langle \mathrm{IDT_{ij}} \rangle$
  $\mathrm{r} \in_\mathrm{R} \mathbb{Z}_\mathrm{p}$
  $\mathrm{K} \leftarrow \mathrm{g^r}$
  $\mathrm{K} \leftarrow$ AMAI : GenCommonK($\mathrm{IDT_{ij}}$, K)
  Store mapping : $\mathrm{IDT_{ij}} \leftrightarrow \mathrm{K}$
  $\langle \mathrm{M_1}, \mathrm{M_2} \rangle \leftarrow$ AMAI : PullKeyAll($\mathrm{IDT_{ij}}$, K)
  $\mathrm{IDT_{ij}} =$
    $\mathrm{ID_{ij}} : \mathrm{ID_i}, \mathcal{T}_\mathrm{j}$
    $\mathrm{M_1} : \mathrm{M1_{ATSP_1}} \cup \mathrm{M1_{ATSP_2}} \cup ...$
    $\mathrm{M_2} : \mathrm{K}^\alpha, \mathrm{K}^\beta, \mathrm{K}^\gamma, ...$
  return
```

4.5 Privacy Enhanced Token-based Device Signature

When device needs to sign a message m, it uses a random number $r_4 \in_R \mathbb{Z}_p$, one way secure hash $H(m)$ of message and the token ABT_{ij} to compute signature σ_{ij} as below. This attribute-based signature is given to the consumer application.

$$\sigma_{ij} = \left\{ \begin{array}{l} A = g^{r_4} \\ C = g^{\frac{1}{r_4 + H(m)}} \\ D = \{K^\alpha\}^{r_4} \cdot \{K^\beta\}^{r_4} \cdot \{K^\gamma\}^{r_4} \ldots \\ \quad = g^{r_4 (\prod_{\forall k} r_k)(\sum_{\forall k} ASK_k)} \\ E_i = {M_1}^{r_4} = g^{r_4 (\prod_{\forall k} r_k)(\frac{q_x(0)}{t_i})} \end{array} \right\} \forall\ k \mid \mathrm{ATSP_k} \in \mathrm{GetATSP}(\mathcal{T}) \tag{4}$$

4.6 Signature Verification

Consumer application can use an offline procedure Verify$(M, \sigma, \mathrm{MPK})$ for verification. This procedure uses the function $\mathrm{VerN}(T_i, E_i, i)$, where first parameter is the public key of the attribute i, second parameter is the corresponding private key, and third parameter is the attribute of the node $i = \mathrm{attr}\,(x)$. The function is defined as below.

$$\mathrm{VerN}(T_i, E_i, x) = \begin{cases} e(T_x, E_x) & \text{if attr}(x) \in \gamma \\ \perp & \text{otherwise} \end{cases} \tag{5}$$

$\mathrm{VerN}(T_z, E_z, z)$ is called for every child node z of non-leaf node x, and the result is stored in L_z. Let k_x represent node x threshold value, and the set of child nodes z of node x is represented by a k_x size set V_x such that $L_z \neq \perp$. If V_x does not exist, then the function returns $\perp$ implying the node is not satisfied. If V_x exists, L_x is computed as below.

$$\begin{aligned} L_x &= \prod_{z \in V_x} L_z \Delta_{i,V_x'}(0) \quad \text{where } i = \text{index}(z),\ V_x' = \{\text{index}(z)\colon z \in V_x\} \\ &= \prod_{z \in V_x} L_z \Delta_{i,V_x'}(0) \\ &= \prod_{z \in V_x} (e(g,g) r r_4 q_z(0))^{\Delta_{i,V_x'}(0)} \\ &= \prod_{z \in V_x} (e(g,g) r r_4 q_{\text{parent}(z)}(\text{index}(z))) \Delta_{i,V_x'}(0) \\ &= \prod_{z \in V_x} e(g,g) r r_4 q_x(i) \Delta_{i,V_x'}(0) \\ &= e(g,g) r r_4 q_x(0) \text{ using polynomial interpolation} \end{aligned} \tag{6}$$

It can be verified that R, $\mathrm{VerN}(T_R, E_R, R) = e(g,g)^{r_4(\prod_{\forall k} r_k)(\Sigma_{\forall k} \mathrm{ASK}_k)}$ if signature satisfies the access tree $\mathcal{T}_R$.

To ensure that the signer holds necessary attributes, the verifier verifies whether following equalities hold valid, and if they are, the signature is considered valid.

$$\begin{aligned} e(g, D) &\overset{?}{=} L_R \\ e(g^m.A, C) &\overset{?}{=} e(g,g) \end{aligned} \tag{7}$$

5 Security Analysis

This section presents an informal security analysis of the proposed model. Traditional security requirements such as confidentiality, data integrity and mutual authentication are assumed to be present and are intentionally kept out of scope for this paper.

5.1 *Privacy*

Neither the signed document nor the signature contains any information about identity of the signer. At the receiver side, verification of a signature does not require signer identity to be known. Hence, signer privacy is maintained.

5.2 *Unforgeability*

It can be deduced that with the strong extended Diffie–Hellman assumption, the proposed model is existential-unforgeable under chosen message (eu-cma) attack. Using proof by contradiction method, if the model is forgeable with a non-negligible advantage ϵ, then the S-EDH assumption can also be broken with the same advantage ϵ. Furthermore, since private key of the device is always kept in a secure storage such as trusted execution environment (TEE), partial key replacement attack is also not possible.

6 Performance Analysis

Present model of registered devices is based on PKI and not on attribute-based schemes; hence, the two models may not be compared efficiently. In this analysis, a number of signing, exponent and pairing operations are computed for each phase of the model. Functions $\mathrm{NL}(\mathcal{T})$ and $L(\mathcal{T})$ compute number of non-leaf nodes and leaf nodes, respectively, in a given access tree $\mathcal{T}$. N number of ATSPs are assumed to contribute an average of $\mathcal{A}_{\mathrm{ATSP}}$ attributes in access tree $\mathcal{T}$. The model consists of five phases, setup, registration, token generation, signature and verification. Most of these phases are one time activities except signature which is invoked for every request. As can be seen from Table 1, the signature cost includes linear number of exponent operations and grows linear to the number of attributes.

$$\mathrm{AmortizedCost}_{\mathrm{DeviceSign}} = \mathcal{O}(L(T)) * \mathrm{Cost}_{\mathrm{exponent}} \tag{8}$$

7 Conclusion and Future Work

With proliferation of Aadhaar-based services, ubiquitous computing devices, IoT and 5G, the number and use of registered devices are expected to grow in terms of volume and sensitivity of data they carry. Two foreseeable requirements in this direction are attribute-based verification and owner and hence device identity privacy.

Table 1 Performance assessment: number of operations

		Signing	Exponent	Pairing
Setup	AMAI	$\lvert\mathcal{A}_{\text{AMAI}}\rvert + 1$	$\lvert\mathcal{A}_{\text{AMAI}}\rvert + 1$	
	ATSP			
	Device			
ATSP Registration	AMAI	$N *$ $(\lvert\mathcal{A}_{\text{ATSP}}\rvert + 1)$		
	ATSP Device		$N *$ $(\lvert\mathcal{A}_{\text{ATSP}}\rvert + 1)$	
Token Generation	AMAI			
	ATSP		$N *$ $(\lvert L(\mathcal{T}_{\text{ATSP}})\rvert + 1)$	
	Device		1	
Attribute-based Signature	AMAI			
	ATSP			
	Device		$\lvert L(\mathcal{T})\rvert + 5$	
eSignverification	Any		$\lvert NL(\mathcal{T})\rvert$	$\lvert L(\mathcal{T})\rvert + 2$

This paper presented a model to extend present model of registered devices to achieve these two requirements. Other mechanisms such as oblivious transfer [22] and secure multi-party computation [23] can also be used to improve device privacy.

References

1. UIDAI, About UIDAI (2009) https://uidai.gov.in/about-uidai.html
2. Banerjee S (2016) Aadhaar: Digital inclusion and public services in India. World Development Report, pp 81–92
3. UIDAI, Annual Report 2018-19 (2019) https://uidai.gov.in/images/AADHAR_AR_2018_19_ENG_approved.pdf
4. UIDAI, Aadhaar eKYC API Specification, v2.1 (2017) https://uidai.gov.in/images/resource/aadhaar_ekyc_api_2_1.pdf
5. MeitY, Digital Locker Technical Specification, v2.3 (2015) https://img1.digitallocker.gov.in/assets/img/technical-specifications-dlts-ver-2.3.pdf
6. CCA, eSign API Specifications v3.2 (2019) http://cca.gov.in/sites/files/pdf/esign/eSign-APIv3.2.pdf
7. UIDAI, Aadhaar Registered Devices Technical Specification v2.0 (2019) https://uidai.gov.in/images/resource/Aadhaar_Registered_Devices_2_0_4.pdf
8. UIDAI, Aadhaar Authentication API Specification v2.0 (2017) https://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_2_0.pdf
9. Shanqing G, Yingpei Z (2008) Attribute-based signature scheme. In: International conference on information security and assurance
10. Whitfield D, Hellman M (1976) New directions in cryptography. In: IEEE transactions on information theory, pp 644–654

11. Shamir A (1984) Identity-based cryptosystems and signature schemes. In: Theory and application of cryptographic techniques workshop, pp 47–53
12. Boneh D, Franklin M (2001) Identity-based encryption from the Weil pairing. Int Cryptology Conf 213–229
13. Goyal V et al (2006) Attribute-based encryption for fine-grained access control of encrypted data. In: 13th ACM conference proceedings on computer and communications security, pp 89-98
14. Bethencourt J et al (2007) Ciphertext-policy attribute-based encryption. In: Symposium on security and privacy, pp 321–334
15. Shanqing G, Yingpei Z (2008) Attribute-based signature scheme. In: International conference on information security and assurance, pp 509–511
16. Tan S et al (2009) On the security of an attribute-based signature scheme. In: International conference on U-and E-service, science and technology, pp 161–168
17. Maji H et al (2011) Attribute-based signatures, Cryptographers' track at the RSA conference, pp 376–392
18. Zhang F et al (2004) An efficient signature scheme from bilinear pairings and its applications. In: International workshop on public key cryptography, pp 277–290
19. Yacobi Y (2002) A note on the bilinear Diffie-Hellman assumption. IACR Cryptology ePrint Archive
20. Ateniese G et al (2005) Practical group signatures without random oracles. IACR Cryptology ePrint Archive
21. Beimel A et al (1996) Secure schemes for secret sharing and key distribution. Technion-Israel Institute of technology, Faculty of computer science
22. Rabin Michael O (2005) How to exchange secrets with oblivious transfer. IACR Cryptology ePrint Archive
23. Goldreich O (1998) Secure multi-party computation, manuscript, preliminary version 78

Stratification of the Lesions in Color Fundus Images of Diabetic Retinopathy Patients Using Deep Learning Models and Machine Learning Classifiers

Avnish Panwar, Geeta Semwal, Silky Goel, and Siddharth Gupta

Abstract Diabetic Retinopathy (DR) is a frequently occurring eye disease that is diagnosed in people suffering from diabetes for a long period of time. Patients who live with diabetes for a long duration of time exhibit mild to severe symptoms of DR. This pathos flourishes gradually and leads to complete blindness over time. The diagnosis of DR is a time-consuming and error-prone process for ophthalmologists due to the pictorial complexities of the images. Therefore, a method based on Machine Learning (ML) and Deep Learning (DL) is proposed to classify the retinal fundus images of the patients into classes based on the severity level of the disease. We employ a CNN architecture armed with the power of deep learning and pre-trained with Transfer Learning to accomplish the task. Outsmarting the already existing approaches, the proposed model functions via extracting a feature vector from the test set of the images which on feeding to classifier models can classify the new images with high accuracy. In our work, we apply various CNN models to extract the features from several diabetic fundus images. The extracted features are provided as the input to various classifiers which as a result classify several lesions accurately. The results show that using deep learning along with transfer learning can accurately classify the fundus images into the right category of lesions.

Keywords Transfer learning · Diabetic retinopathy · Image classification

1 Introduction

The fluctuation of sugar level in the human body sometimes majorly affects several human body organs such as the kidney, brain, and eyes. When sugar level becomes too high it results in a chronic disease called diabetes or Diabetic Mellitus (DM) [1].

A. Panwar
Graphic Era Hill University, Dehradun, India

G. Semwal · S. Gupta (✉)
Graphic Era Deemed to be University, Dehradun, India

S. Goel
University of Petroleum and Energy Studies, Dehradun, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_49

A situation arises where the blood vessels to the retina become blocked therefore DM results in vascular damage which is known as Diabetic Retinopathy (DR) [2]. Each year about one to two million people become blind and around 45 million people are blind across the globe due to DR [3]. This majorly becomes the leading cause of vision loss. According to the survey conducted by the World Health Organization (WHO), the expected number of patients suffering from DR will become 80 million by the end of the year 2030 in India [4]. Due to the very poor ophthalmologist-to-patient, ratio in India, the patients suffering from DR are exponentially increasing each year [5]. The symptoms of DR are not observed at the early stages. Therefore, the dilated examination of eyes at frequent intervals of time should be done. The other ways are Fundus Fluorescence Angiography (FFA) [6] and Optical Coherence Tomography (OCT) [6]. The DR treatment involves both a doctor and an ophthalmologist. The doctor helps you to control the level of insulin in the body and fluctuation of sugar level that directly or indirectly affects the body organs such as the kidney, brain, and heart function. Apart from this when leakage of fluid in the retina becomes severe, an ophthalmologist uses surgical procedure or laser treatment to prevent or cure the damage caused by it. Laser photocoagulation is a laser treatment done to cure DR [6]. When the problem becomes too severe, the ophthalmologist creates small, painless retinal burns that seal off the leaking fluid in the blood vessels and also reduces the swelling. However, if the laser treatment does not sufficiently halt the growth of new vessels or the hemorrhages occurred are not clear, the doctors recommend vitrectomy surgery. This surgery is done when a patient feels retinal detachment [7]. DR is broadly categorized into five categories based on the systems: No disease, Mild DR, Moderate DR, Severe DR, and Proliferative DR [8]. There are no signs of DR at the early stages but with the passing time, the symptoms like Microaneurysms (ME), leaking blood vessels in the retina, swelling of the retinal vessel, abnormal growth of blood vessels, and damaged nerve tissues [9] start arising. Therefore detection of DR is essential. However, in countries like India, the equipment available is less according to a vast population. Also, the screening of DR is not possible with the available resources [9].

These days using contemporary techniques like image classification, pattern recognition, Machine Learning (ML), and Transfer Learning (TL) creates a satisfactory impact on the medical field [10]. Several features like Microaneurysms (ME), Hard Exudates (EX), Soft Exudates (SE), and Hemorrhages (HE) are extracted from the retinal fundus images using the classification techniques. Convolutional Neural Network (CNN) architecture comprises of Convolutional layer, Pooling layer, Rectified Linear Unit (ReLU) layer, Fully connected layer, and the Loss layer. Several input images are fed to the core CNN layer [11]. It consists of several filters that are also called kernels to distinguish several features and then forward to the other layer. Pooling layer is used to reduce the size of representation (parameters) and computations in the network. Max-pooling is the most common approach used in pooling. The ReLU is an activation function that is used to set all negatives values to zero. The output of the pooling layer is fed as an input to a fully connected layer. Finally, the deviation between predicted output and the true labels is shown by the loss layer [11]. Transfer Learning (TL) is a technique used in DL where a model

Table 1 International clinical DR and DME disease severity scales [13]

Stage	Ophthalmologist observations	Severity
Stage I	No observable abnormalities	No DR
Stage II	Microaneurysms—primary changes not visible by naked eyes	Mild non-proliferative DR
Stage III	• Microaneurysms • Retinal dot • Cotton wool spots No signs of severe non-proliferative DR	Moderate non-proliferative DR
Stage IV	• 20 or more intra-retinal hemorrhages in one of the 4 quadrants • Venous bleeding in 2 or more quadrants • Prominent Intra-retinal Microvascular Abnormality (IRMA) in 1 or quadrants No signs of proliferative DR	Severe non-proliferative DR
Stage V	• Neovascularization • Vitreous/pre-retinal hemorrhage	Proliferative DR

is trained for the base network and this experience is reused for the new network. However, if a classifier is trained with small or fewer samples it may lead to the risk of over-fitting the new data. Dropout and data augmentation are methods used to prevent neural networks from over-fitting. In this work, TL is used with CNN to fit for small size datasets used in medical image classification without over-fitting [12]. Table 1 includes the several observations made by ophthalmologist about the severity of DR at different stages. Stage I is the initial stage where it will not be possible to observe the symptoms of DR with the naked eye. Stage II is mild non-proliferative DR where a cluster of red small round dots is observed. They do not affect the vision of the eye. Stage III is moderate non-proliferative DR where the capillaries near retina swell up due to which the flow of blood is not possible sometimes. These physical changes in the retina may lead to DME. Stage IV is a severe non-proliferative DR where the blood vessel that provides nourishment to the retina is blocked. Signals are sent to the brain from the retinal area to grow new blood vessels for providing the nourishment to the retina. The last and final stage V is called proliferative DR or neovascularization, this is the most severe stage of DR where the abnormal growth of blood vessels takes place, and these new blood vessels sometimes start bleeding.

In Fig. 1, the first image shows the healthy retina where no symptoms of DR are observed. The next image b shows the presence of several ME. Image c is the fundoscopic image that shows the existence of hard exudates. Image d in Fig. 1 shows the IRMA which is the sign of abnormality in blood vessels and may lead to severe NPDR. Image e shows the neovascularization where the rate of growth of blood vessels in retina becomes abnormal and may bleed. Finally, image f shows the pre-retinal hemorrhages where sometimes the detachment of the retina takes place and this is the type of proliferative DR.

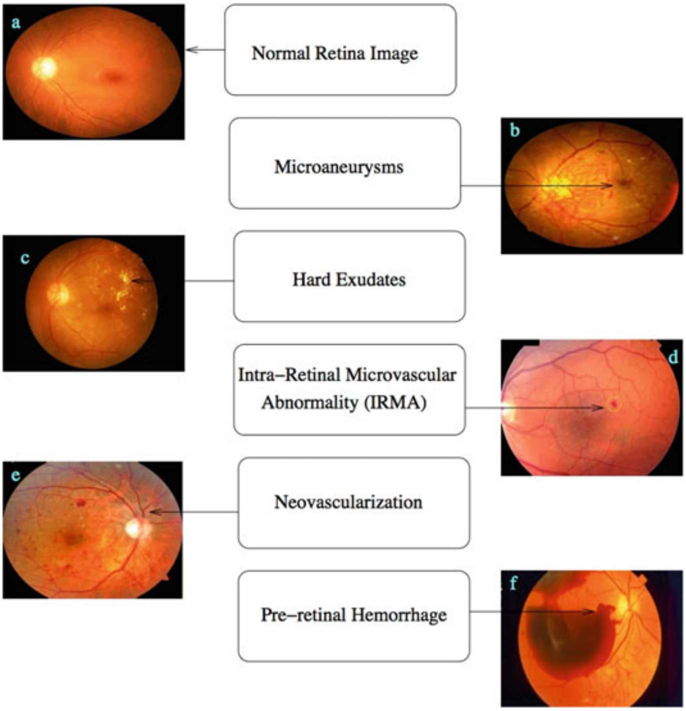


Fig. 1 Fundus images differentiating several stages of diabetic retinopathy. **a** STAGE I: no DR, **b** STAGE II: mild non-proliferative DR, **c** STAGE III: moderate non-proliferative DR, **d** and **e** STAGE IV: severe non-proliferative DR, and **f** STAGE V: proliferative DR

In this article, we explored various DL CNN architectures that are used to differentiate various lesions from retinal fundus images based on the severity of the stage. The results are carried out by taking 122 fundus images of lesions from the IDRID dataset which is publically available. We find that the VGG16 model along with Logistic Regression (LR) classifier gives the best accuracy values as compared with the VGG19 model applied by Gupta et al. in [14] on the same dataset as reported in [14].

2 Related Work

Li et al. [15] used DRI and Messidor dataset that comprises of 1014 and 1200 fundus images and uses it for classification using CNN based transfer learning. The results obtained after applying several methods and classifiers were eye-catching. Applying CNN based TL on small datasets gives satisfactory results. Zhang et al. [16] proposed a model called deepDR which detects the presence and severity of DR in fundus images using TL. It used the CNN and deep neural network. The proposed model obtained sensitivity of 97.5%, specificity of 97.7% and AUC of 97.7%. This work helps ophthalmologist to detect the severity of DR more accurately and precisely. Wan et al. [17] described the major three challenges faced in the implementation of CNN: classification, segmentation, and detection. Christopher et al. [18] used a large database comprises of 14,822 fundus photographs. Several deep learning architectures are applied to detect Glaucomatous Optic Neuropathy (GON) in fundus images. The obtained sensitivity and specificity are 88% and 95%. Also, the AUC is 91%. This research helps clinical doctors for making decision support system in automatic detection and screening of DR. Wang et al. [13] used several cameras that are used to capture more than 35 thousand fundoscopic images from the Kaggle dataset. Out of these 35 thousand images after removing noise and sub-optimal lighting conditions, only 166 high-quality images are selected as input. Several CNN architectures such as AlexNet, VGG16, Inception V3 are deployed for DR stage classification. The results obtained show that the InceptionV3 net algorithm performs exceptionally well. Gulshan et al. [19] used an eye PACS-1 dataset consisting of 9963 images and a Messidor-2 dataset with 1784 images. Several deep CNN algorithms used for detecting Referable Diabetic Retinopathy (RDR) give 0.991 AUC for the eye PACS-1 and 0.990 AUC for Messidor-2 dataset. The sensitivity and specificity obtained for eye PACS-1 dataset are 90.3% and 98.1% and for the Messidor-2 dataset are 87.0% and 98.5%. Gupta et al. [14] used the VGG19 model to classify several lesions such as MA, EX, SE, and HE from retinal fundus images. The dataset used is the IDRID dataset of the Indian population. The features extracted from the VGG19 model are fed to several classifiers such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest (RF), LR, and AdaBoost. The accuracies obtained for classifying MA, SE, EX, and HE are 96.7%, 95.1%, 91.8%, and 90.2%.

3 Method

CNNs are a class of neural network that plays a major role in the field of Computer Vision (Image Recognition and Image Classification). Convolution performs by extracting the features from the given input images. Once the images are convoluted, the next operation performed is ReLU which is a non-linear operation that replaces all the negative pixels by zero. The next operation performed is pooling which reduces the size of the representation. After applying all the aforementioned operations, finally, a classification neural network is used for classification by adding the output features to it [11].

In this paper, we deploy the CNNs architectures: VGG16 and VGG19 for retinal fundus image classification. We will see the detailed description of these state-of-art CNN architectures in the next sub-section.

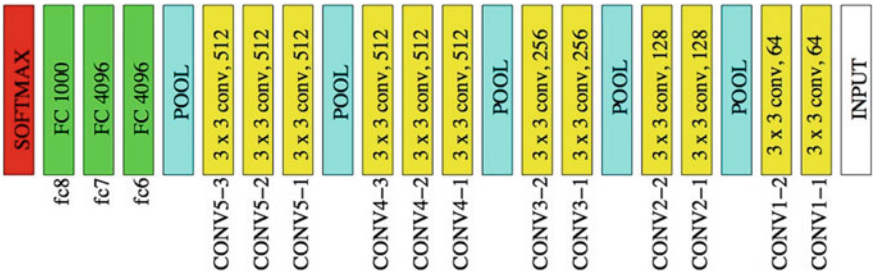


Fig. 2 VGG16 architecture

3.1 VGG16

Karen Simonyan and Andrew Zisserman invented VGG16 in the year 2014. Image of dimension (224, 224, 3) is given as an input to VGG16 architecture. The first two layers comprised of 64 channels with a kernel size of (3 × 3) and stride 1 and the same padding. After this, the pooling layer is applied in which a special version of pooling called max-pooling is used which comprises of (2 × 2) filters with stride 2. In addition to this, two convolution layers of 256 filters are implemented and the size of each filter is (3 × 3). Convolution filter increase with the power of 2: it starts with 64 and becomes 512 with filter size (3 × 3). Then there are 2 sets of 3 convolution layers and a max-pooling layer as shown in Fig. 2. The detailed description of VGG16 can be extracted from [20]. In this work, we have taken images from the IDRID dataset and after pre-processing the images are fed to the VGG16 model for feature extraction. These extracted images are finally passed to several ML classifiers for classifying the IDRID images as MA, SE, EX, and HE.

3.2 VGG19

There are various variants of VGG such as VGG11, VGG16, and VGG19. The VGG19 comprises 19 layers (16 convolutional layers, 3 fully connected layers, 5 Max Pool layers, and 1 Softmax layer). This architecture consists of 64 channels each having kernel size of (3 × 3). In VGG19 architecture, the filters increase with the power of two: it starts with 64 filters, 128 filters, 256 filters, and 512 filters as shown in Fig. 3. Also, there are a total of 5 sets of a convolutional layer, out of which the first set having 64 filters, the next set having 128 filters, followed by another set of 4 convolutional layers with 256 filters, and finally, 2 sets of 4 convolutional layers with 512 filters. The max-pooling layer comprises of (2 × 2) filter size with stride 2. The detailed information about VGG19 architecture can be extracted from [21]. Figure 3 represents the detailed VGG19 architecture. Authors in [14] used VGG19 architecture on the IDRID dataset, after pre-processing the images are passed to the

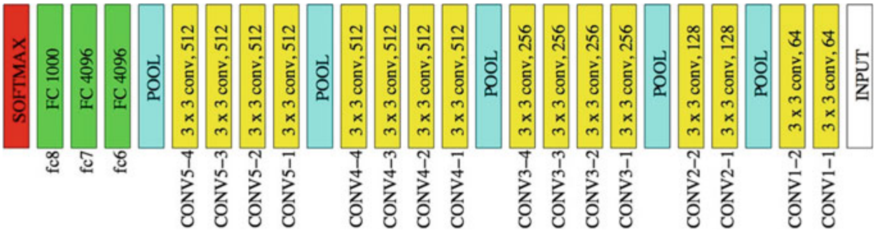


Fig. 3 VGG19 architecture

VGG19 model for feature extraction and finally extracted images are fed to various ML classifiers for the classification of images as MA, SE, EX, and HE.

3.3 Inception V3

The inception V3 model comprises 48 layers. VGGNet comprises simple architecture but the computational cost of this model is higher as compared to the inception architecture of GoogleNet in terms of memory and computation. The authors achieved state-of-art performance from inception architecture by utilizing the channel concatenation. The channels obtained from 1 × 1 convolution, 3 × 3 convolution, and 5 × 5 convolution along with the pooling is concatenated. The width and height of channels remain unchanged. Also, as compared with other architecture like AlexNet that uses 60 million parameters, the GoogleNet employed only 5 million parameters. Due to the low computational cost of Inception than VGGNet, it is feasible to utilize the inception network in big data where a huge amount of data needs to be processed [22].

3.4 Transfer Learning

Transfer learning and machine learning algorithms make use of the neural network to solve future data problems using statistical models that are already trained on problems that are being solved. The layers used in previous solved models are included in the new models to be trained. To ensure the proper implementation of TL on the given dataset, the images of datasets are resized for various CNN architectures: VGG16 and VGG19 [12].

4 Experimental Results and Discussions

4.1 Dataset Collection and Pre-processing

An eye clinic located in Nanded, Maharashtra, India, gathered the fundus images of the IDRID dataset [23]. The images were captured using a non-invasive fundus camera with a xenon flash lamp. The distance between the examined eye and camera lens is 39 mm. The resolution of the image is 4288 × 2848 pixels, and the format used to store is .jpg file format. The size of each image is 800 KB. All the fundus images were extracted from several DR patients between the duration of 2009–2017 [23]. Table 2 shows the training and testing images of the IDRID dataset.

IDRID dataset comprises 516 images of Indian patients. This dataset is broadly divided into two categories: (1) retinal images with the sign of DR or DME and (2) retinal images with no sign of DR or DME (normal images). Ground truth images are included in this dataset with the sign of DR or DME and normal retinal images which are described as:

- ***Pixel level annotated data*** [23]

  This data includes 81 color images with the sign of DR. It includes the color fundus images with the sign of DR. The color fundus images in .jpg files and binary mask lesions (.tif files). 81 binary mask images are of ME lesions, 81 binary mask images are of EX lesions, 80 binary mask images are of HE, and 40 images are of SE lesions.
- ***Image level disease grading*** [23]

  For image-level grading, it comprises of total 516 images with conditions for DR and DME. The file format for all images is .csv file format. Based on the standards set by the international clinical DR scale, the diabetic retinal images are divided into different groups.

However, the images present in the dataset are of different shapes and sizes. Also, due to variable resolutions that are varied width and height of several images, the classification of images becomes challenging. To overcome this problem the images are pre-processed by using several techniques such as image cropping and image resizing. As a result, the images with the same resolution are extracted and used for the purpose of classification.

Table 2 IDRID dataset description

Type	Dataset		Total images
	Training set images	Testing set images	
Microaneurysms (MA)	54	27	81
Soft exudates (SE)	26	14	40
Hard exudates (EX)	54	27	81
Hemorrhages (HE)	53	27	80

4.2 Evaluation Metrics

Accuracy is the measure to define how accurately our procedure classifies the samples correctly versus all the given samples of a particular class. In order to compare the performance of several classifiers, accuracy has opted as the parameter [13]. Formally accuracy can be calculated by Eq. (1), mentioned below.

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n} xi \tag{1}$$

where

$xi = 1$ if predicted label is equal to true label.
$xi = 0$ for others.

4.3 Classifiers and Energy Function Used

After training the VGG16 architecture on a given dataset, several classifiers are used to detect the presence of lesions like ME, SE, EX, and HE in retinal fundus images. The classifiers used are Neural Network (NN), SVM, RF, LR, and AdaBoost. At last several parameters like precision, recall [24], $F1$ score, accuracy, and AUC are used to measure the performance of the aforementioned classifiers [25]. The formulas used to calculate the value of several parameters are described in (2), (3), (4), and (5).

$$\text{Precision} = \text{TP}/\text{TP} + \text{FP}, \tag{2}$$

$$\text{Recall} = \text{TP}/\text{TP} + \text{FN}, \tag{3}$$

$$F1\,\text{Score} = 2 * (\text{Recall} * \text{Precision})/\text{Recall} + \text{Precision}, \tag{4}$$

$$\text{Accuracy} = \text{TP} + \text{TN}/\text{TP} + \text{FP} + \text{FN} + \text{TN}, \tag{5}$$

where TP is True Positive, FP is False Positive, TN is True Negative, and FN is False Negative.

4.4 Performance Evaluation

The work carried out in this paper comprises the use of several CNN architectures. The used architecture (VGG16) has been trained on 122 lesion images from the

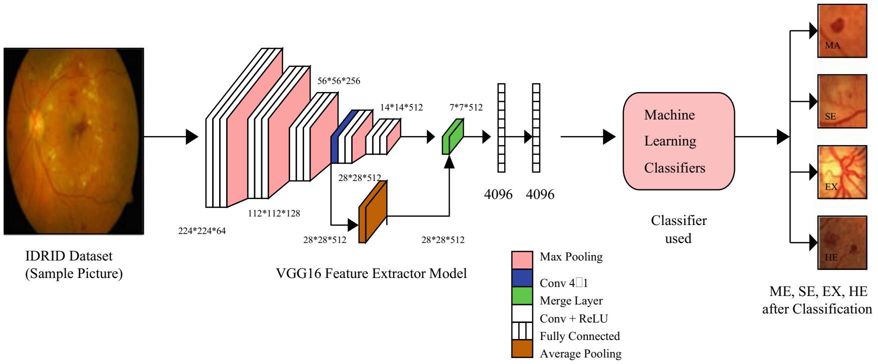


Fig. 4 Architecture for VGG16 model used in classification of several lesions

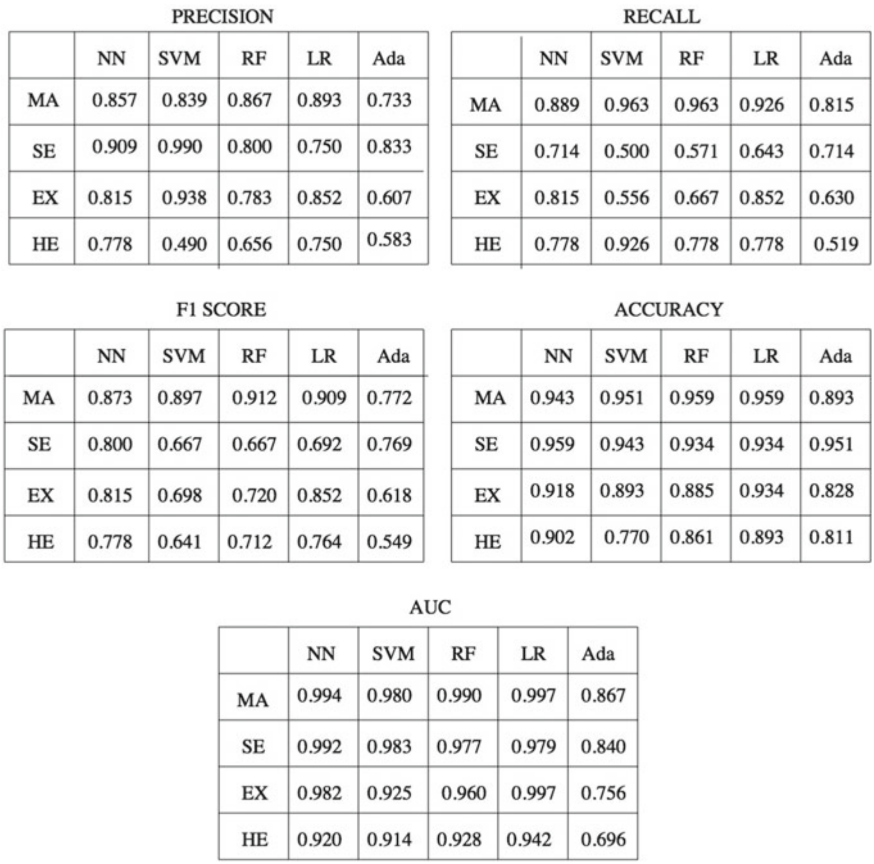
IDRID dataset (which is publically available). These 122 images contain several lesions images: 27 images of MA, 14 images of SE, 27 images of EX, and 27 images of HE. These images are pre-processed by using several pre-processing techniques like image resizing and image cropping. After this these images are fed to the VGG16 model that is used as the state-of-art model to classify the lesions. The extracted images are finally passed to several ML classifiers such as NN, SVM, RF, LR, and AdaBoost for images classification based on the severity and type of lesions. Figure 4 shows the architecture of VGG16 model for classification of several lesions. After classification, several lesions such as MA, SE, EX, and HE are obtained.

Several parameters are used to judge the performance of CNN models and used classifiers. Figure 5 shows the values obtained by several classifiers after lesion classification by applying the VGG16 model. These values are obtained by considering several parameters such as precision, recall, $F1$ score, accuracy, and AUC. Out of these parameters we are considering the accuracy and AUC values [13].

The AUC and accuracies obtained for classification of several lesions using the VGG16 model overcomes the values obtained by Gupta et al. in [14] by using the same IDRID dataset. Tables 3 and 4 show a comparison of VGG16 and VGG19 model on the AUC and accuracies values for MA, SE, EX, and HE using NN, SVM, RF, LR, and AdaBoost classifiers. Table 3a contains the AUC values for the VGG19 model as stated in [14]. Table 3b represents the AUC values for the VGG16 model [26].

In Table 4a, the accuracy values for VGG19 model are described as stated in [14] and Table 4b represents the accuracy values for the VGG16 model. Only for classification of MA the exception case is present and for rest of the lesions VGG16 overcome the VGG19 model.

To evaluate the efficiency of the training model, the ROC [27] curve for the VGG16 model is depicted in Fig. 6.



PRECISION

	NN	SVM	RF	LR	Ada
MA	0.857	0.839	0.867	0.893	0.733
SE	0.909	0.990	0.800	0.750	0.833
EX	0.815	0.938	0.783	0.852	0.607
HE	0.778	0.490	0.656	0.750	0.583

RECALL

	NN	SVM	RF	LR	Ada
MA	0.889	0.963	0.963	0.926	0.815
SE	0.714	0.500	0.571	0.643	0.714
EX	0.815	0.556	0.667	0.852	0.630
HE	0.778	0.926	0.778	0.778	0.519

F1 SCORE

	NN	SVM	RF	LR	Ada
MA	0.873	0.897	0.912	0.909	0.772
SE	0.800	0.667	0.667	0.692	0.769
EX	0.815	0.698	0.720	0.852	0.618
HE	0.778	0.641	0.712	0.764	0.549

ACCURACY

	NN	SVM	RF	LR	Ada
MA	0.943	0.951	0.959	0.959	0.893
SE	0.959	0.943	0.934	0.934	0.951
EX	0.918	0.893	0.885	0.934	0.828
HE	0.902	0.770	0.861	0.893	0.811

AUC

	NN	SVM	RF	LR	Ada
MA	0.994	0.980	0.990	0.997	0.867
SE	0.992	0.983	0.977	0.979	0.840
EX	0.982	0.925	0.960	0.997	0.756
HE	0.920	0.914	0.928	0.942	0.696

Fig. 5 Precision, recall, $F1$ score, accuracy, and AUC values for classification of Microaneurysms (MA), Soft Exudates (SE), Hard Exudates (EX), and Hemorrhages (HE) using NN, SVM, RF, LR, and AdaBoost classifiers

Table 3 AUC comparison of CNN models: (a) VGG19 model [14]. (b) VGG16 model

	(a)					(b)				
	NN	SVM	RF	LR	Ada	NN	SVM	RF	LR	Ada
MA	0.969	0.980	**0.987**	**0.987**	0.840	0.994	0.980	0.990	**0.997**	0.867
SE	**0.992**	0.963	0.958	0.988	0.773	**0.992**	0.983	0.977	0.979	0.840
EX	**0.963**	0.877	0.947	0.955	0.691	0.982	0.925	0.960	**0.997**	0.756
HE	0.897	0.871	0.904	**0.906**	0.683	0.920	0.914	0.928	**0.942**	0.696

Table 4 Accuracy comparison of CNN models: (a) VGG19 model [14]. (b) VGG16 model

	(a)					(b)				
	NN	SVM	RF	LR	Ada	NN	SVM	RF	LR	Ada
MA	0.934	0.943	0.951	0.967	0.902	0.943	0.951	0.959	0.959	0.893
SE	0.934	0.910	**0.951**	**0.951**	0.910	**0.959**	0.943	0.934	0.934	0.951
EX	0.910	0.836	**0.918**	**0.918**	0.779	0.918	0.893	0.885	**0.934**	0.828
HE	0.861	0.738	0.885	**0.902**	0.803	**0.902**	0.770	0.861	0.893	0.811

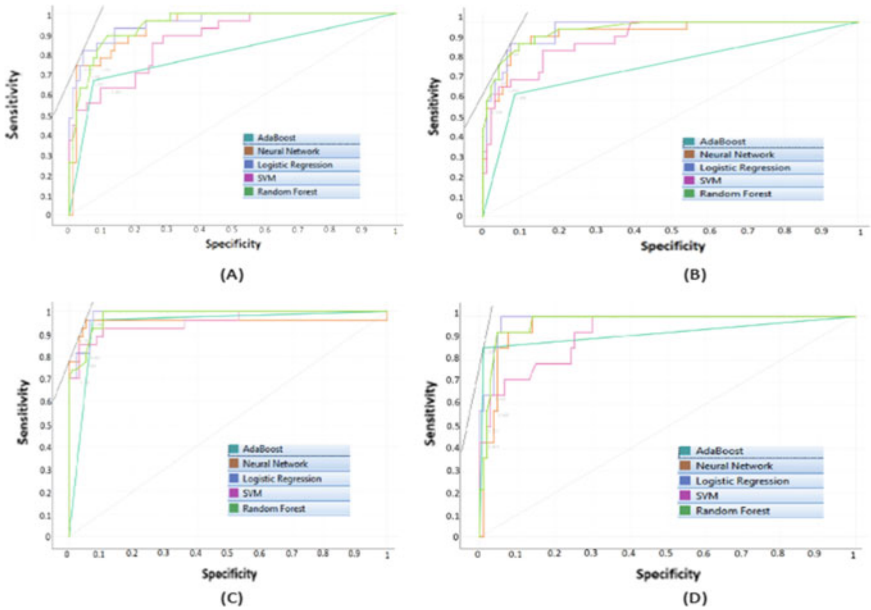


Fig. 6 ROC-AUC analysis for VGG16 model by respective classifiers: **a** for Hemorrhages (HE), **b** for Hard Exudates (EX), **c** for Microaneurysms (MA), and **d** for Soft Exudates (SE)

5 Conclusion

On the basis of population, India's ranking is number two in the list of the countries across the world. With such a huge population, the ratio of DR patient-to-ophthalmologist is very uncertain due to which large numbers of patients across the country face the problem of vision loss. Therefore early detection of DR becomes the utmost important. In this work, we have deployed the CNN VGG16 architecture along with various classifiers such as NN, SVM, RF, LR, and AdaBoost to classify lesions in different stages based on the level of severity. The accuracy obtained by the VGG16 model along with the LR classifier is 95.9% as compared to the VGG19 model with the LR classifier as reported in [14]. Classification of lesions for DR by using the fundus images in the IDRID dataset is not an easy task even for the expert ophthalmologists. This framework helps the ophthalmologist to detect the severity of the stage of DR at right time. In this work, the dataset used comprises a 66% training dataset and 34% testing dataset images. We have evaluated the VGG16 model on several parameters such as precision, recall, F1 score, accuracy, and AUC values. The accuracies obtained to classify MA images is 95.9% using LR classifier, SE is 95.9% by using NN classifiers, EX is 93.4% by using LR classifiers, and HE is 90.2% by using the NN classifier which is enhanced as compared to the reported results in [14] expect in case of MA. In future work, the results obtained can be enhanced by using more powerful CNN architecture or by increasing the number of fundus images in the dataset.

References

1. Vujosevic S, Toma C, Villani E, Gatti V, Brambilla M, Muraca A, Ponziani MC, Aimaretti G, Nuzzo A, Nucci P, De Cilla S (2019) Early detection of microvascular changes in patients with diabetes mellitus without and with diabetic retinopathy: comparison between different swept-source OCT-A instruments. J Diab Res vol 2019, Article ID 2547216, pp 12. https://doi.org/10.1155/2019/2547216
2. Faust O, Acharya R, Ng EY-K, Ng K-H, Suri JS (2012) Algorithms for the automated detection of diabetic retinopathy using digital fundus images: a review. J Med Syst 36(1):145–157
3. Prokofyeva E, Zrenner E (2012) Epidemiology of major eye diseases leading to blindness in Europe: a literature review. Ophthalmic Res 47(4):171–188
4. Wild S, Roglic G, Green A, Sicree R, King H (2004) Global prevalence of diabetes: estimates for the year 2000 and projections for 2030. Diab Care 27(5):1047–1053
5. Sharma T, Raman R, John S, Badrinath SS (2012) Telescreening for diabetic retinopathy in South India. Digital Teleretinal 137–142
6. Masood S, Luthra T, Sundriyal H, Ahmed M (2017) Identification of diabetic retinopathy in eye images using transfer learning. In: International conference on computing, communication and automation (ICCCA), pp 1183–1187
7. Rizzo S, Genovesi-Ebert F, Di Bartolo E, Vento A, Miniaci S, Williams G (2008) Injection of intravitreal bevacizumab (Avastin) as a preoperative adjunct before vitrectomy surgery in the treatment of severe proliferative diabetic retinopathy (PDR). Graefes Arch Clin Exp Ophthalmol 246(6):837–842
8. Wilkinson CP, Ferris III FL, Klein RE, Lee PP, Agardh CA, Davis M, Dills D et al (2003) Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. Ophthalmology 110(9):1677–1682
9. Doshi D, Shenoy A, Sidhpura D, Gharpure P (2016) Diabetic retinopathy detection using deep convolutional neural networks. In: International conference on computing, analytics and security trends (CAST), pp 261–266
10. Dai L, Fang R, Li H, Hou X, Sheng B, Wu Q, Jia W (2018) Clinical report guided retinal microaneurysm detection with multi-sieving deep learning. IEEE Trans Med Imaging 37(5):1149–1161
11. Pratt H, Coenen F, Broadbent DM, Harding SP, Zheng Y (2016) Convolutional neural networks for diabetic retinopathy. Procedia Comput Sci 90:200–205
12. Benson J, Carrillo H, Wigdahl W, Nemeth S, Maynard J, Zamora G, Barriga S, Estrada T, Soliz P (2018) Transfer learning for diabetic retinopathy. Med Imaging 2018 Image Process 10574
13. Wang X, Lu Y, Wang Y, Chen W-B (2018) Diabetic retinopathy stage classification using convolutional neural networks. In: IEEE international conference on information reuse and integration (IRI), pp 465–471

14. Gupta S, Panwar A, Goel S, Mittal A, Nijhawan R, Singh AK (2019) Classification of lesions in retinal fundus images for diabetic retinopathy using transfer learning. In: International conference on information technology (ICIT), pp 342–347
15. Li X, Pang T, Xiong B, Liu W, Liang P, Wang T (2017) Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification. In: International congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI), pp 1–11
16. Zhang W, Zhong J, Yang S, Gao Z, Hu J, Chen Y, Yi Z (2019) Automated identification and grading system of diabetic retinopathy using deep neural networks. Knowl Based Syst 12–25
17. Wan S, Liang Y, Zhang Y (2018) Deep convolutional neural networks for diabetic retinopathy detection by image classification. Comput Electr Eng 274–282
18. Christopher M, Belghith A, Bowd C, Proudfoot JA, Goldbaum MH, Weinreb RN, Girkin CA, Liebmann JM, Zangwill LM (2018) Performance of deep learning architectures and transfer learning for detecting glaucomatous optic neuropathy in fundus photographs. Sci Rep 1:1–13
19. Gulshan V, Peng L, Coram M, Stumpe MC, Derek W, Narayanaswamy A, Venugopalan S et al (2016) Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA 316(22):2402–2410
20. Nguyen QH, Muthuraman R, Singh L, Sen G, Tran AC, Nguyen BP, Chua M (2020) Diabetic retinopathy detection using deep learning. In: Proceedings of the 4th international conference on machine learning and soft computing, pp 103–107
21. Mateen M, Wen J, Song S, Huang Z (2019) Fundus image classification using VGG-19 architecture with PCA and SVD. Symmetry 11(1)
22. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826
23. Porwal P, Pachade S, Kamble R, Kokare M, Deshmukh G, Sahasrabuddhe V, Meriaudeau F (2018) Indian diabetic retinopathy image dataset (IDRiD): a database for diabetic retinopathy screening research. Data 3(3)
24. Davis J, Goadrich M (2006) The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd international conference on machine learning, pp 233–240
25. Gupta S, Panwar A, Goel S (2019) Classification among microaneurysms, exudates, and lesion free retinal regions in the eye images using transfer learning CNNs. Int J Eng Adv Technol (IJEAT) 5508–5512
26. Ricci E, Perfetti R (2007) Retinal blood vessel segmentation using line operators and support vector classification. IEEE Trans Med Imaging 26(10):1357–1365
27. Geetha Ramani R, Balasubramanian L, Jacob SG (2003) ROC analysis of classifiers in automatic detection of diabetic retinopathy using shape features of fundus images. In: International conference on advances in computing, communications and informatics (ICACCI), pp 66–72

BER Analysis of Massive MIMO in Heterogeneous Cellular Network

Gitimayee Sahu and Sanjay S. Pawar

Abstract In fifth-generation (5G) technology, massive multiple input and multiple output (MIMO) plays a very significant role in spectral efficiency enhancement and interference cancellation. It also does beam forming using maximal ratio combining (MRC) to obtain high throughput and good quality of service (QoS). All these are achievable by using higher-order modulation technique, transmission power control, and proper usage of radio resources in RF channel with minimizing interference. As MIMO technology uses multiple transmitting and receiving antennas, the spectral resources can be shared between multiple UEs simultaneously. Further, the 5G network is multi-tier and ultra-dense heterogeneous cellular network (HetNet) that accomplishes efficient spectrum usage through carrier aggregation. For improving spectrum resource efficiency (SRE) and RF channel utilization, source bits must be accurately transmitted to the destination. This can be quantified with high symbol efficiency and minimum Bit Error Rate (BER). In this paper, BER for various RF channels was investigated with AWGN noise, Rayleigh fading, and MIMO capacity for multiple UEs. The experimentation is done using MATLAB14b, and the iteration is carried out till convergence is obtained.

Keywords MIMO · BER · 5G · Small cell · AWGN · Rayleigh fading · Heterogeneous cellular network

1 Introduction

MIMO technology was proposed by Kailath et al. [1], where multiple input antennas are connected to transmitting antenna and multiple output antennas connected to the receiving antenna in order to have better spectral efficiency and high throughput. The eNodeB with MIMO antenna at the transmitting side communicates with multiple users using the equivalent time–frequency resource block (RB) known as multi user MIMO (MU-MIMO) [2]. Massive MIMO is a pattern of MU-MIMO where the base station uses an antenna array with hundreds of active elements to handle hundreds

G. Sahu (✉) · S. S. Pawar
Department of ExTC, UMIT, SNDT Women's University, Juhu, Mumbai, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_50

of active users (UEs) using the same RB [3, 4]. It increases the transmitting and receiving diversity gain [5] and the multiplexing gain to enhance the reliability.

In the present scenario, the wireless communication industry is vigorously advancing in a variety of permissive technologies, like massive MIMO, HetNet, Ultra-Dense Network (UDN), and millimeter wave to expedite the growing of 5G standardization and pre-commercial trial process.

The next generation wireless communication technology offers large degree of freedom (DoF) and simultaneously drives large number of users with guaranteed QoS and high security. Key Performance Indicators (KPI) are spectrum efficiency (SE), energy efficiency (EE), peak data rate, traffic density, device to device (DtoD) communication, reduced latency, and reliability for accomplishing more extensive service provision and improved user experience.

The 5G usage scenarios can broadly classified as enhanced mobile broadband service (eMBB), Ultra reliable low-latency communication (URLLC), and massive machine type communications (mMTC). eMBB supports high data rate across a wide coverage area, mMTC connects huge number of devices, while URLLC provides ultra low latency of nearly 1 ms. URLLC is essential for remote surgery, autonomous vehicles, and tactile internet. The 5G enhanced performance parameter requirements as per IMT advanced are as shown in Fig. 1 and specified in Table 1.

The applications of 5G includes smart city, self driving car, industrial automation, remote surgery, augmented and virtual reality (AR/VR), eHealth services, mission critical applications, smart home building, cloud computing, internet of things (IoT), and mobile internet. These applications can be obtained by several

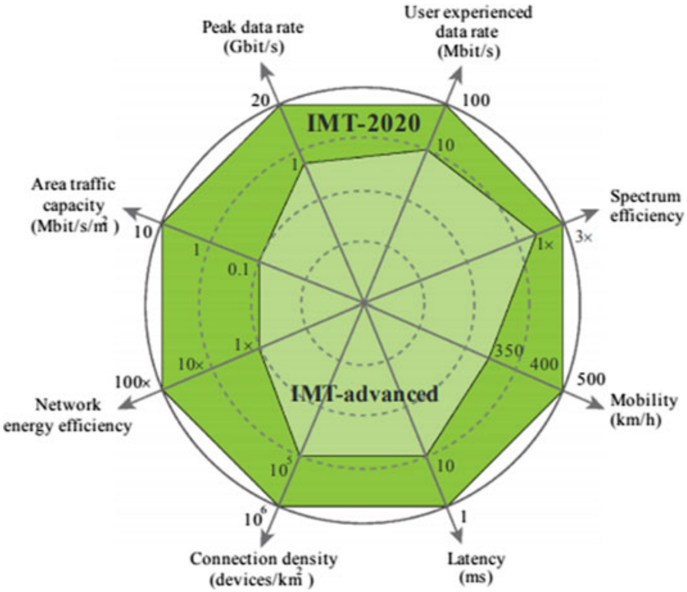


Fig. 1 Enhancement of key capabilities from IMT advanced to IMT 2020 [6]

Table1 5G performance requirements

Parameters	Value
Peak data rate	20 Gbit/s
Area traffic capacity	10 Mbit/s/m^2
Network energy efficiency	>100 Times
Connection density	10^6 devices/km^2
Mobility	500 km/h
Spectrum efficiency	>3 times
User experienced data rate	100 Mbit/s
Service density	10 Gbps/km^2
End to end latency	Order of millisecond

advancements in signal processing and multiplexing in air interfaces like 5G New Radio (NR), universal filtered multicarrier (UFMC), filter bank multicarrier (FBMC), generalized frequency division multiplexing (GFDM), Non orthogonal multiple access (NOMA), massive MIMO, millimeter wave communication (mm wave), beam forming, HetNet, and UDN. MIMO–OFDM uses massive MIMO antenna combined with OFDM multi carrier transmission method. The transmission modes of MIMO are transmission diversity and spatial multiplexing. Transmission diversity enhances the reliability and link quality while spatial multiplexing increases the data rate and spectral efficiency.

MIMO technique is used to enhance the system SE while mm wave frequency band used to increase the system bandwidth. Further to improve the geographic frequency reuse, HetNet, and UDN architecture were deployed in 5G network.

1.1 Femto Cell Networks

As the contemporary cellular network uses data hungry wireless devices such as, smart phone, laptop, tablets, and various sensors including environmental and body area network (BAN). Much more traffic is generated indoor as compared to the outdoor scenario. This leads to non-uniform traffic distribution over the unified coverage area of the network.

The traditional wireless network is designed to provide better coverage in a broad area and assures guaranteed services of uniformly distributed traffic. But it faces the challenge of handling non-uniform traffic from different spatial geographical regions. Therefore to increase the capacity of wireless network effectively reduction of the coverage area of the macrocell known as smallcell. Due to deployment of smallcell the network becomes ultra dense which increases spatial multiplexing of the frequency resources. As the transmission distance between the BS and the UE decreases, transmission loss is reduced and throughput can be improved with power efficiency.

The heterogeneous architecture of 5G is multi-tier which includes micro cell, pico cell, femto cell, and DtoD communication under the coverage of macrocell. Micro, femto, and pico are termed as small cell with different power levels, coverage area and user serving capacity [7]. Small cell plays very significant role in the future wireless communication, since the crucial requirement of technology revolution is high throughput, guaranteed QoS, and minimum end to end latency. All this requirements are fulfilled only when the distance between the user and the base station reduces and restricted number of user admissions to the serving cell.

The femtocell is an indoor solution known as Home-eNodeB (HeNB) uses licensed spectrum and serves limited number of UEs with guaranteed QoS. As a solution over and above, small cells with different power levels were installed at highways, malls, offices, and residential homes to get adequate user experience. It encompasses transformation over the conventional wireless network where the scope of self-organized, self-optimized, self-discovered, self-healing, and intelligent cells are overlaid under the coverage of macrocell.

As the macrocell gets congested during peak hours, the users at the edge of the cell can be biased to offload to the nearby small cell. Due to offloading to the small cell, the SE of small cell enhances through spatial reuse and EE of macrocell increases due to decrease in power consumption. The users get better QoS and high throughput which improves the overall performance of the cellular network.

The challenges faced by the small cells are inter channel and intra channel interference due to sharing of resources and backhaul network. To avoid interference, dynamic power control (DPC), co-ordinated multipoint (CoMP), or cluster can be formed by adjacent smallcells. In the backhaul network, centralized or distributed scenario can be used where the smallcells forward the traffic to the co-ordinator/cluster head that collectively forward to the core network [8, 9]. Optical fiber or millimeter wave can be used as backhaul network for efficient operation with broad bandwidth. Figure 2 shows the wireless backhaul network using massive

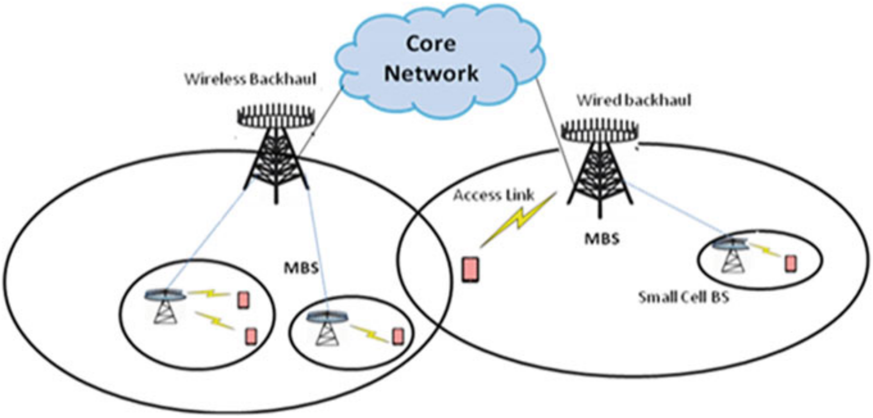


Fig. 2 Backhaul network of massive MIMO in HetNet

MIMO of HetNet and the core network.

The large number of antenna elements in the base station uses linear pre-coding, e.g., maximal ratio transmission (MRT) or zero forcing for obtaining better channel capacity [10, 11]. It is low complex, supports transmission diversity, and does beam forming toward the user equipment. By increasing the transmission power to the receiver through sharp focusing beams the radiated power is reduced, interference is minimized and at the same time SE and EE can be improved. The challenges faced by massive MIMO are accuracy in channel estimation and acquisition that can be obtained through high gain and low BER. The error occurred due to pilot contamination and usage of same pilot in neighbor cells for channel estimation purpose.

The structure of the research paper is as follows. Section 2 represents the system model, Sect. 2.1 calculation of SNR, Sect. 2.2 capacity of MIMO wireless System, Sect. 2.3 BER calculation in wireless communication system, Sect. 2.4 Beam Forming. Section 3: Results and discussion, and in Sect. 4 conclusion is given.

2 System Model

HetNet is one of the emerging techniques to break the capacity crunch by exploring and exploiting both frequency and spatial diversities [12]. In HetNet, multiple low-power smallcells overlaid in the coverage of a high-power macrocell. The femtocell or home e-NodeB (HeNB) is a type of smallcell for indoor application. The system model consists of 3tier HetNet having macrocell, picocell, and femtocell in 1 km^2 area. The simulation is carried out in MATLAB14b using homogeneous Poisson point process (PPP) and voronoi tessellation. The smallcells with density λ_{sc} and UEs (more than ten times of the smallcells) with density λ_{UE} were spatially distributed in the Euclidean plane. The path loss is assumed to be Rayleigh fading [13] (Fig. 3).

The small cell with massive antenna array is able to affirm the azimuth and elevation angles to enhance the coverage in house, office, residential buildings, and hotspots. The cell radius is varied using CRE technique and the MUEs offloaded from macro cell to small cell for load balancing as shown in Fig. 4.

The relationship between the transmitted and received resource elements on different antennas is expressed by a system of linear equations. Figure 5 shows the MIMO channels between ‘M’ transmitting antennas and ‘N’ receiving antennas. The vector of resource elements on received antenna is obtained by multiplying the MIMO channel matrix and the vector of transmitted elements. The MIMO system equations used to find the best estimation of the transmitted resource element of a given subcarrier. It consists of resource elements and the channel response [or the channel state information (CSI)] which connects each pair of transmitting and receiving antennas.

The different attenuation factors are Rayleigh fading and large scale path loss. The signal to interference ratio (SIR) at the BS is given by,

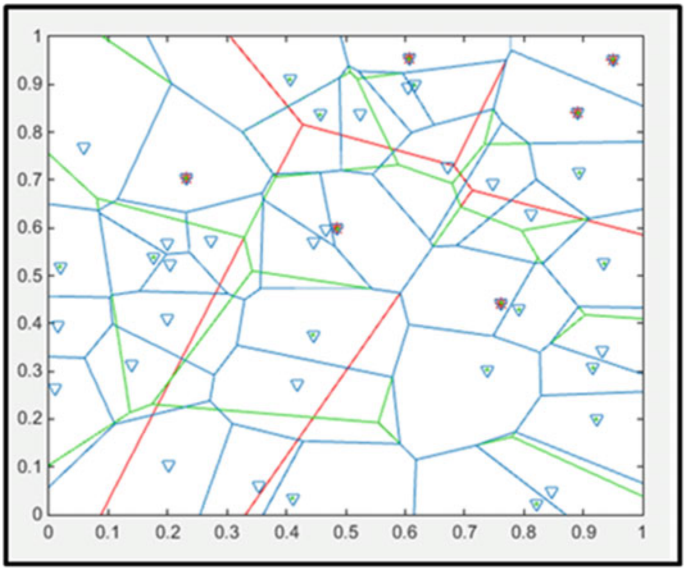


Fig. 3 Three-tier HetNet including macrocell, picocell, and femtocell

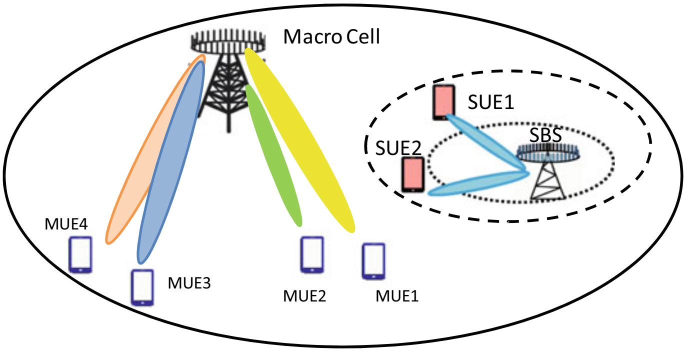


Fig.4 HetNet with cell range expansion (CRE) of small cell

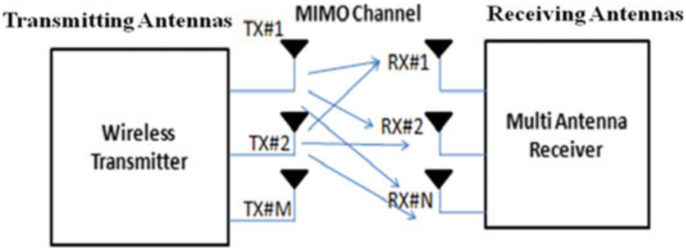


Fig. 5 Schematic diagram of MIMO system using ‘M’ transmitting and ‘N’ receiving antennas

$$\mathrm{SIR} = \frac{\|h_{lmn}\|^2 d_{lmn}^{-\alpha}}{\sum_{k\in\Phi_{\mathrm{BS}}} d_{lkn}^{-\alpha}\|h_{lkn}\|^2}, \tag{1}$$

where Φ_{BS} denotes the set of smallcell base stations (BSs). The numerator presents the signal received by the UE from the associated BS and the denominator presents the interference received by the UE from other BSs. h_{lmn} is the channel gain of UE 'n' in cell 'l' and base station 'm', α is the path loss coefficient, and d_{lmn} is the distance of UE 'n' from the base station 'm' in cell 'l'. In the denominator d_{lkn} represents the interference received by the UE 'n' in cell 'l' from base station 'k' and h_{lkn} denotes the channel gain by the UE from the interfering BS.

2.1 Calculation of SNR

Consider the wireline system with AWGN channel. The transmission of BPSK symbols is given by $x(k) = \pm\sqrt{P}$ along a Rayleigh fading wireless channel. The baseband wireless system can be modeled as

$$y(k) = hx(k) + n(k). \tag{2}$$

The difference between the wireless channel and wireline channel is fading coefficient 'h', and the received baseband signal is given by '$h \cdot x(k)$'.

The received signal power in the wireless channel is given by

$$E\{|hx(k)|^2\} = |h|^2 E\{|x(k)|^2\} = |h|^2 P = a^2 P, \tag{3}$$

where $\|h\|^2 = a^2$. The received power at the wireless receiver depends on the amplitude 'a' of the fading coefficient 'h'. The instantaneous SNR is $a^2\frac{P}{\sigma_n^2}$.

The average SNR of wireless system can be expressed as

$$E\left\{a^2\frac{P}{\sigma_n^2}\right\} = E\{a^2\}\frac{P}{\sigma_n^2} = \frac{P}{\sigma_n^2}. \tag{4}$$

Since $E\{a^2\} = E\{|h|^2\} = 1$,
the average SNR of the wireless system is equal to $\frac{P}{\sigma_n^2}$.

2.2 Capacity of MIMO Wireless System

The MIMO channel capacity between transmitter and receiver is given by

$$C = \max_{P_i \sum_i P_i \leq P} \sum_i B \log_2\left(1 + \frac{P_i \gamma_i}{\sigma_n^2}\right) \tag{5}$$

The power allocation to different channels is carried out using water-filling algorithm.

The total MIMO system capacity is given by

$$C = \min(M_t, M_r) \cdot B \log_2(1 + \rho). \tag{6}$$

Hence capacity grows linearly with the size of the antenna array in massive MIMO. The singular value decomposition (SVD) can be deployed into three channels between the transmitter and receiver.

The output of ith channel is given by

$$\tilde{y}_i = \sigma_i \tilde{x}_i + \tilde{w}_i \tag{7}$$

where x_i is the transmitted symbol. σ_i is the gain of the ith channel.

The signal power of the transmitted symbol x_i is given by

$$E\{|x_i|^2\} = P_i. \tag{8}$$

The signal to noise power ratio of the ith channel is given by

$$\mathrm{SNR}_i = \frac{\sigma_i^2 P_i}{\sigma^2}. \tag{9}$$

The capacity of the channel is given by using Shannon's law,

$$C_i = B_i \log_2\left(1 + \frac{\sigma_i^2 P_i}{\sigma^2}\right). \tag{10}$$

For 'M' number of transmitted channels the total capacity is given by

$$C = \sum_{i=1}^{M} \log_2(1 + \mathrm{SNR}_i). \tag{11}$$

The MIMO channel capacity (C) is given by the maximum sum rate with total power as constraint.

$$C = \max \sum_{i=1}^{M} \log_2\left(1 + \frac{\sigma_i^2 P_i}{\sigma^2}\right)$$

$$\text{Subject to } \sum_{i=1}^{M} P_i \leq P \tag{12}$$

By using Lagrange multiplier (λ), the above equation can be solved as

$$\sum_{i=1}^{M} \log_2\left(1+\frac{\sigma_i^2 P_i}{\sigma^2}\right)+\lambda\left(P-\sum_{i=1}^{t} P_i\right). \tag{13}$$

It can be solved by differentiating w.r.t. power

$$P_i=\begin{cases} \frac{1}{\lambda}-\frac{\sigma^2}{\sigma_i^2}, \text{ for}\left(\frac{1}{\lambda}-\frac{\sigma^2}{\sigma_i^2}\right)\geq 0 \\ 0, \qquad\qquad \text{otherwise}\end{cases} \tag{14}$$

where P_i maximizes the sum rate which achieves the capacity of the MIMO communication system and is given by

$$P_i=\left(\frac{1}{\lambda}-\frac{\sigma^2}{\sigma_i^2}\right)^{+}. \tag{15}$$

The Lagrange multiplier (λ) can be found from

$$\sum_{i=1}^{t} P_i=P, \tag{16}$$

$$\Rightarrow \sum_{i=1}^{t}\left(\frac{1}{\lambda}-\frac{\sigma^2}{\sigma_i^2}\right)=P. \tag{17}$$

The optimal algorithm to achieve the capacity of maximum sum rate of the MIMO system is given by water filling algorithm. The terminals in each cell transmit pilot sequences; by receiving pilots, the terminal of base station de-spreads the signal. Due to pilot reuse the resultant signal will be superposition of channel matrices of all the cells that share identical pilot sequences. Within lth cell the resultant signal will be

$$Y'_{pl}=\sqrt{\tau_p \rho_{ul}}\sum_{l'\in p_l} G^{l}_{l'}+W'_{pl}, \tag{18}$$

where W'_{pl} represents noise whose elements are independent identical distributed (i.i.d) noise lies in between 0 and 1.

In terms of component, Eq. (16) can be written as

$$[Y'_{pl}]_{mk}=\sqrt{\tau_p \rho_{ul}}\sum_{l'\in p_l} g^{lm}_{l'}+[W'_{pl}]_{mk}. \tag{19}$$

To implement preceding and decoding [14], the femto cell requires only an estimate of its own channel matrix $[G_l^l]$. Performance calculation and power control algorithm depend on the mean-square channel estimates of all the cells.

The MMSE estimation of $g_{l'k}^{lm}$ is

$$\hat{g}_{l'k}^{lm} = \frac{\sqrt{\tau_\rho \rho_{ul}}\beta_{l'k}^{l}}{1 + \tau_\rho \rho_{ul} \sum_{l'' \in p_l} \beta_{l''k}^{l}} [Y'_{pl}]_{mk}, \quad l' \in P_l \tag{20}$$

The estimates obtained by the small cell for different values of l', but with the same terminal index, 'k', are perfectly correlated, which is the essence of pilot contamination. The mean-square channel estimate is denoted by $\gamma_{l'k}^{l}$, where

$$\gamma_{l'k}^{l} = E\left\{\left|\hat{g}_{l'k}^{lm}\right|^2\right\} \tag{21}$$

$$= \frac{\tau_p \rho_{ul} (\beta_{l'k}^{l})^2}{1 + \tau_p \rho_{ul} \sum_{l'' \in p_l} \beta_{l''k}^{l}} [Y'_{pl}]_{mk}, \quad l' \in P_l \tag{22}$$

It is clear that $\gamma_{l'k}^{l} \leq \beta_{l'k}^{l}$. As a consequence of pilot contamination, multi-cell channel estimation is considerably noisier than the single-cell counterpart.

$$\tilde{g}_{l'k}^{lm} = \hat{g}_{l'k}^{lm} - g_{l'k}^{lm}. \tag{23}$$

Equation 23 represents the error in channel estimation.

The mean-square estimation error is independent of m and is given by

$$E\left\{\left|\tilde{g}_{l'k}^{lm}\right|^2\right\} = \beta_{l'k}^{l} - \gamma_{l'k}^{l}, \quad l' \in p_l \tag{24}$$

The channel estimation in matrix form is given by

$$\hat{G}_{l'}^{l} = Z^l D_{\gamma_{l'}^{l}}^{1/2}, \quad l' \in p_l \tag{25}$$

where $[\hat{G}_{l'}^{l}]$ is a matrix of channel estimation, whose (m, k)th element equals to $\hat{g}_{l'k}^{lm}$ $\gamma_{l'}^{l} = [\gamma_{l'1}^{l}, \ldots, \gamma_{l'k}^{l}]^{\mathrm{T}}$, and the elements of Z^l is the i.i.d. noise. It is important that Z^l is independent of 'l''.

2.3 BER Calculation in Wireless Communication System

BER is the measure of the system performance. It is the ratio of number of erroneous bits received to the total number of bits. The BER performance matrix is used to describe the realization of wireless communication system. The transmitted information $+\sqrt{P}$ can be decoded as 1 and erroneously as 0 and vice versa.

Consider a transmitted signal 'S_0' at time 't' can be represented by

$$S_0(t) = a_0 \exp(j\omega_0 + \phi_0), \tag{26}$$

where a_0 is the amplitude of the transmitted signal, $\omega_0 = 2\pi f_c$, f_c frequency of the carrier signal, and ϕ_0 is the initial radian phase of the transmitted signal.

The received signal at time 't' from path 'i' including multipath effect can be written as

$$S_i(t) = a_i \exp[j\left(\phi_i + \frac{2\pi}{\lambda} vt\cos\theta_i\right)]\exp[j(\omega_0 + \phi_0)], \tag{27}$$

where

ϕ_i is the phase change due to the time dispersion of the multipath.
v is the speed of the mobile station,
λ is the wavelength of the transmitted signal,
θ_i is the signal arriving angle, and
$\frac{2\pi}{\lambda} vt\cos\theta_i$ is the phase change introduced by Doppler shift in frequency.

Assume that total number of paths is 'N', the received signal 'S' at the receiver is the combination of the signals from 'N' number of different paths:

$$S(t) = \sum_{i=1}^{N} S_i(t), \tag{28}$$

$$S(t) = \left[\sum_{i=1}^{N} a_i \cos\left(\phi_i + \frac{2\pi}{\lambda} vt\cos\theta_i\right) + j\sum_{i=1}^{N} a_i \sin\left(\varphi_i + \frac{2\pi}{\lambda} vt\cos\theta_i\right)\right] \times \exp[j(\omega_0 + \varphi_0)] \tag{29}$$

Multipath effect in the wireless environment produces a change in the real part and imaginary part of the transmitted signal. Let x_i designate the real part and y_i designate the imaginary part, i.e.,

$$x_i = a_i \cos\left(\phi_i + \frac{2\pi}{\lambda} vt\cos\theta_i\right), \tag{30}$$

$$y_i = a_i \cos\left(\phi_i + \frac{2\pi}{\lambda} vt\cos\theta_i\right), \tag{31}$$

where $S(t)$ can be expressed as

$$S(t) = (x + jy)\exp[j(\omega_0 + \phi_0)], \tag{32}$$

and

$$x = \sum_{i=1}^{N} x_i \text{ and } y = \sum_{i=1}^{N} y_i \tag{33}$$

The probability of BER in BPSK of AWGN channel can be calculated using Q functions as

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right). \tag{34}$$

QPSK (Quadrature phase shift keying, 4-PSK, or 4-QAM) having four points on the constellation diagram which are spaced equally around a circle. QPSK can encode two bits per symbol. By using gray coding BER can be minimized and is given by

$$\text{BER} = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right). \tag{35}$$

2.4 Beam Forming

Beam forming sends the same symbol over each transmitting antenna with different scale factor. At the receiver, all received signals are coherently combined using various scale factors. This constitutes transmit/receive diversity system, whose SNR can be maximized by optimizing the scale factors of MRC. Beam forming leads to much higher SNR than the individual channels in the parallel channel decomposition [15]. It transforms MIMO systems into SISO system with transmit and receive diversity. The different types of equalization techniques are

1. Zero forcing (ZF) Equalizer,
2. Minimum mean square error (MMSE) estimator.

ZF equalizer is a linear equalizer uses the inverse frequency response of the channel. It brings the inter symbol interference (ISI) to zero in a noise free case. However, it is not significant since channel impulse response is of finite length. MMSE estimator minimizes the mean square error (MSE) and measures the estimator quality. It minimizes noise and inter-symbol interference (ISI) in the total power.

MRC represents theoretical optimal combiner over fading channel as diversity scheme. Theoretically multiple copies of the same signal were combined to maximize the instantaneous SINR at the output. It enhances the correct reception and reduces the ISI. Here each signal branch is multiplied by a weight factor which is proportional to the signal amplitude. Thus, strong signals were amplified and weak signals get attenuated. MRC is a process of diversity combining in which signals from each channel were added together. The gain of each channel is proportional to the RMS signal level and inversely proportional to the mean square noise level. In this case different proportionality constants were used for each individual channel.

3 Results and Discussion

The BER of AWGN channel is calculated numerically, and simulations are done using MATLAB 2014b. AWGN channel noise is considered for wired channel. As shown in Fig. 6 the BER value decreases with increase in SNR. The theoretical and the simulated values are almost close to each other.

The BER for Rayleigh fading channel is calculated numerically. Rayleigh fading channel noise was considered for wireless channel. The BER value decreases with increase in SNR. The theoretical and the simulated values match to each other almost 94% as shown in Fig. 7.

The BER for MRC is calculated. The BER value decreases with increase in SNR. The theoretical and the simulated values match almost 91% as shown in Fig. 8.

The MIMO capacity was plotted with transmitted power in dB as shown in Fig. 9. With increase in power the MIMO capacity increases. With transmitting power of 30 dB the MIMO capacity reaches up to 1200.

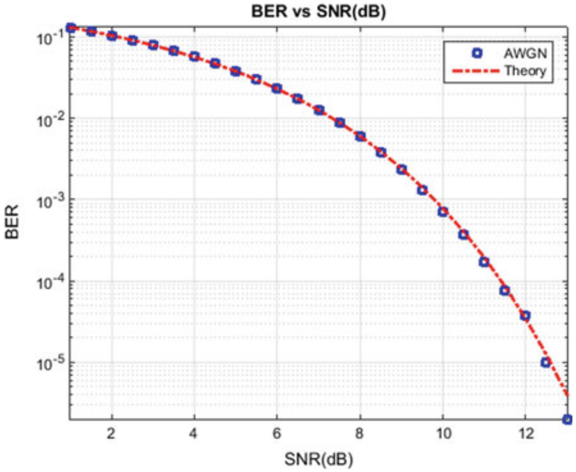


Fig. 6 Theoretical and simulation representation of SNR (dB) versus BER of AWGN channel

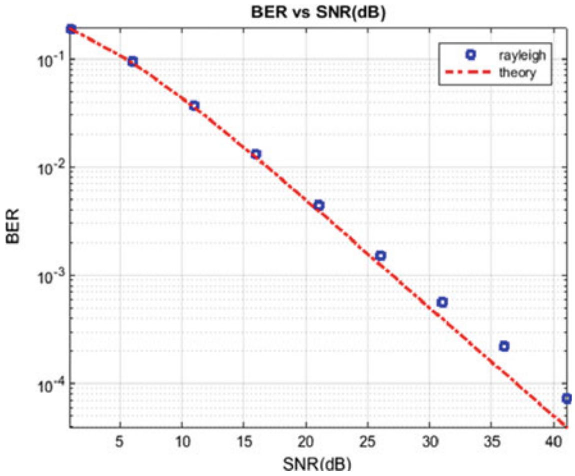


Fig. 7 Theoretical and simulation representation of SNR (dB) versus BER of Rayleigh fading channel

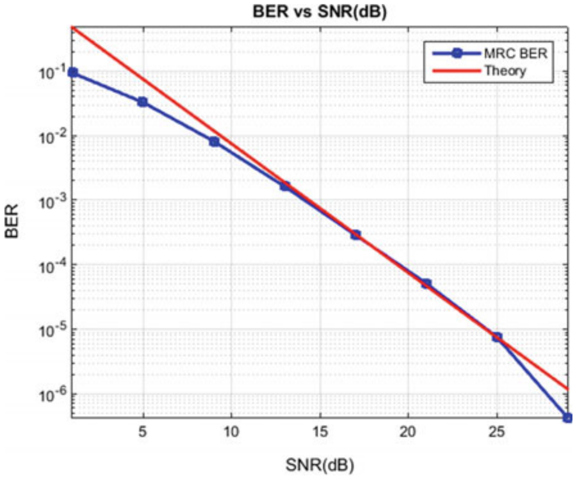


Fig. 8 Theoretical and simulation representation of SNR (dB) versus BER of MRC fading channel

The MIMO capacity is plotted with number of iterations till convergence is obtained. As shown in Fig. 10 after 140 iterations the MIMO capacity reaches to steady state with the value = 5.2 and for further iterations the capacity does not changes.

Figure 11 shows MIMO capacity for user 1 and user 2. The capacity increases linearly with 100 iterations and then it is converged and maintains steady value until 200 iterations. With one user the achievable MIMO capacity is 3.1 and with two users the MIMO capacity increases and reaches till 3.8.

Figure 12 shows BER versus SNR in dB for single user. As shown in the figure with increase in SNR the BER value decreases. For SNR of 21 dB, the calculated BER equals to 10^{-8}.

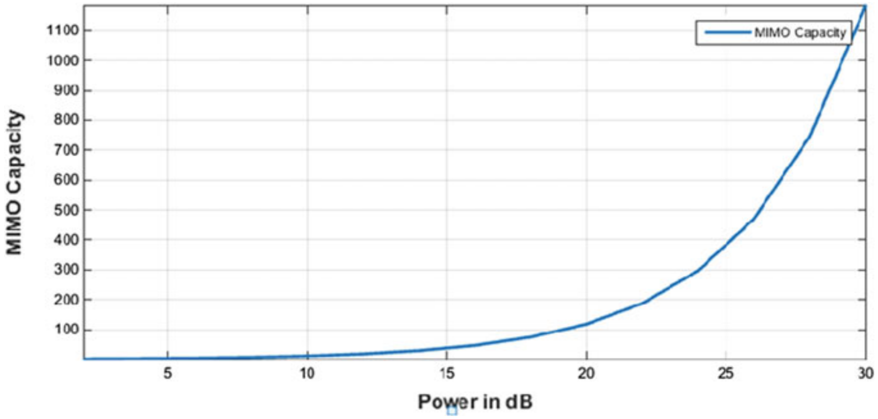


Fig. 9 Average MIMO capacity versus power in dB for single user

Fig. 10 Game theoretic MIMO capacity for user 1

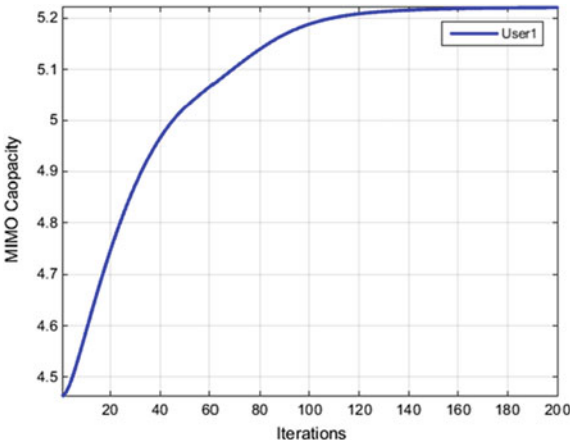


Fig. 11 MIMO capacity versus power for user 1 and user 2 using game theory

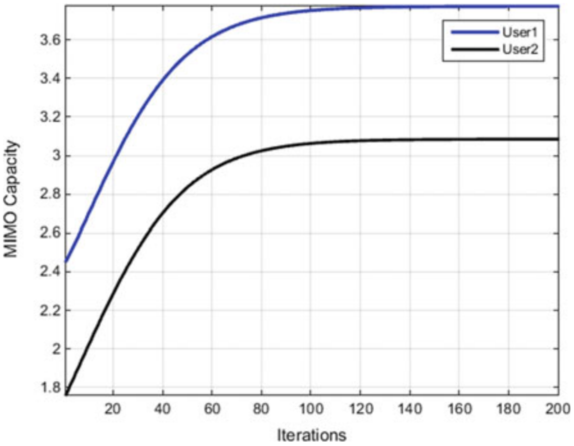


Fig. 12 BER versus SNR (in dB) transmitting beam forming for single user

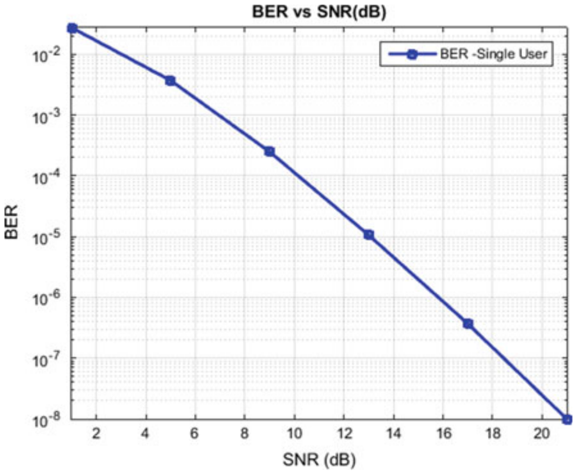


Figure 13 shows BER verses SNR for multiple users using zero forcing. With increase in SNR the BER decreases upto 10^{-7}.

Figure 14 shows convergence of SNR with increase in number of iterations. After 1000 iterations it is found that the SNR obtains a steady value. User 1 get SNR of 3.3 dB and user 2 gets steady value of 8 dB.

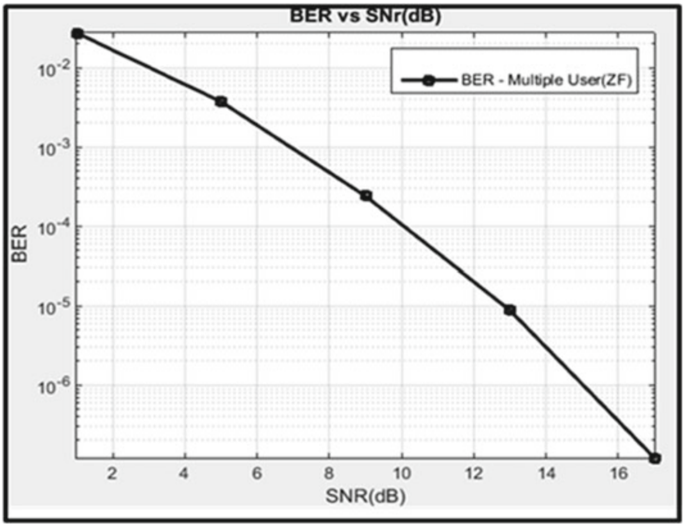


Fig. 13 BER versus SNR (dB) for multiple user using game theoretic beam forming

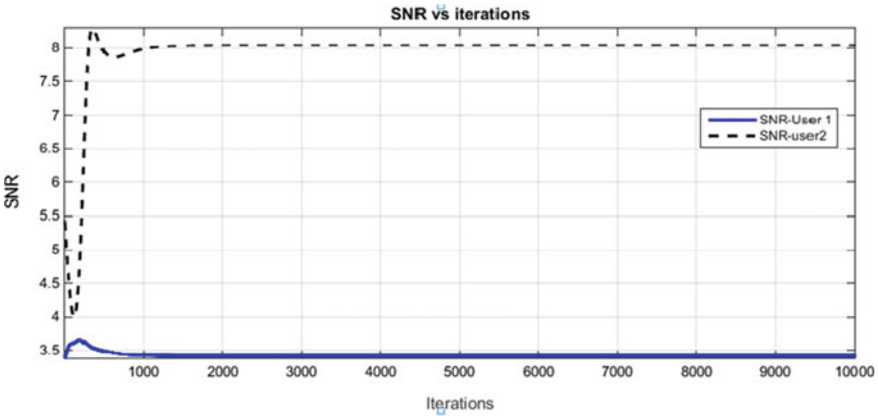


Fig. 14 Game theoretic beam forming for user 1 and user 2

4 Conclusion

From this research work it is found that BER value decreases with increase in SNR. The BER decreased to a minimum of 10^{-8} which is almost negligible. Minimum BER indicates better channel quality with high SNR and spectral efficiency. This leads to better channel capacity, good throughput, high QoS, and ultimately user satisfaction. BER value is investigated for both wireline channel with AWGN noise and wireless channel with Rayleigh fading. The theoretical and simulated results match almost 98% for AWGN channel and 93% for Rayleigh fading channel. MIMO capacity is also experimented for single user and multiple users. It is observed with increase in number of users MIMO capacity increases. The capacity reaches a steady value as the number of iteration progresses.

Acknowledgements This work was supported in part by Visvesvaraya Ph.D. Scheme/DIC/MeitY of Govt. of India under ESDM/5G communications research scheme with reference number **VISPHD-MEITY-779**.

References

1. Kailath T, Paulraj AJ (1994) Increasing capacity in wireless broadcast systems using distributed transmission/directional reception (DTDR). US Patent 5,345,599
2. Mehmood Y, Afzal W, Ahmad F, Younas U, Rashid I, Mehmood I (2013) Large scaled multi-user MIMO system so called massive MIMO systems for future wireless communication networks. In: 19th IEEE international conference on automation and computing, 13–14 Sept 2013, London, UK. Electronic ISBN: 978-1-908549-08-2
3. Garcia N, Wymeersch H, Larsson EG, Haimovich AM, Coulon M (2017) Direct localization for massive MIMO. IEEE Trans Sig Process 65(10):2475–2487

4. Chataut R, Akl R (2018) Efficient and low complex uplink detection for 5G massive MIMO systems. In: 2018 IEEE 19th wireless and microwave technology conference (WAMICON), pp 1–6
5. Lozano A, Jindal N (2010) Transmit diversity vs. spatial multiplexing in modern MIMO systems. IEEE Trans Wirel Commun 9(1)
6. ITU-R M.2083-0 (2015) IMT vision—framework and overall objectives of the future development of IMT for 2020 and beyond
7. Sahu G¸ Pawar SS (2018) Coverage probability of HetNet using stochastic geometry and Poisson point process. Int J Comput Appl 4(8) (2250-1797)
8. Nishimori K (2018) Novel technologies using massive MIMO transmission toward 5G and its beyond systems. In: 2018 international symposium on antennas and propagation (ISAP), pp 1–2
9. Kashima T, Qiu J, Shen H, Tang C, Tian T, Wang X, Hou X, Jiang H, Benjebbour A, Saito Y, Kishiyama Y (2016) Large scale massive MIMO field trial for 5G mobile communications system. In: 2016 international symposium on antennas and propagation (ISAP), pp 602–603
10. Kibona L, Jian L, Yingzhuang L (2020) BER analysis using zero-forcing linear pre-coding scheme for massive MIMO under imperfect channel state information. Int J Electron 107(6)
11. Zhao L, Zheng K, Long H, Zhao H (2014) Performance analysis for downlink massive MIMO system with ZF pre-coding. Trans Emerg Telecommun Technol 25:1219–1230
12. Song X, Fettweis G (2015) On spatial multiplexing of strong line-of-sight MIMO with 3D antenna arrangements. IEEE Wirel Commun Lett 4(4)
13. Eduru S, Rangaswamy N (2018) BER analysis of massive MIMO systems under correlated Rayleigh fading channel. In: 2018 9th international conference on computing, communication and networking technologies (ICCCNT), Bangalore, India
14. Kageyama T, Muta O (2019) Bit Error rate analysis of MRC precoded massive MIMO-OFDM systems with peak cancellation. In: 2019 IEEE 90th vehicular technology conference (VTC2019-Fall), Honolulu, USA
15. Fernandes D, Cercas F, Dinis R (2020) Analytical performance evaluation of massive MIMO techniques for SC-FDE modulations. Electronics 9(3)

ALO-SBD: A Hybrid Shot Boundary Detection Technique for Video Surveillance System

Saptarshi Chakraborty, Dalton Meitei Thounaujam, Alok Singh, and Gautam Pal

Abstract A novel shot boundary detection (SBD) method is proposed using nature-inspired algorithm called ant lion optimizer (ALO). Here, ALO is used to optimize the weights of the feed-forward neural network (FNN) which in turn enhances the performance of the system. A hybrid technique is formulated using the aid of continuity matrix (ϕ) and an outlier to fetch possible set of transition frames. Using a threshold δ_1, the actual transition frames are drawn out from all the possible transition frames. For the experimentation, some challenging videos from TRECVid 2001 and 2007 datasets are selected. ALO-SBD generates better performance in terms of F1 score and outperforms recent state-of-the-art techniques in the field of SBD.

Keywords ALO · PSO · Abrupt transition · Cut · FNN

1 Introduction

Video surveillance is a process of observing scenes and looking for specific behaviors that are inappropriate or that may show the existence of doubtful behavior. The video surveillance systems are mainly used at public places, in events like sports events, public transportation (for security and automation purposes), and in the places which are under high security for regulating the entry, exit, and abnormal behavior of the crowd.

The video surveillance process includes recognition of concern areas and selection of cameras or groups of cameras with high specifications that may able to view spotted areas. If the surveillance system is automatically able to successfully spot the security breach or the possibilities of its occurrence, it will be very helpful for society. It will also help in reducing the manual interaction of the person in viewing the video time to time.

S. Chakraborty (✉) · D. M. Thounaujam · A. Singh
Computer Vision Laboratory, Computer Science and Engineering Department, National Institute of Technology, Silchar, India

G. Pal
Tripura Institute of Technology, Agartala, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_51

For extracting the contents and useful information from the video, a video should be segmented into a set of similar frames called shots. A shot includes a set of frames taken uninterruptedly and have similar content. The action of spotting the boundary in the middle of two continuous shots is known as SBD. Broadly, the boundaries are categorized into two classes: abrupt and gradual transition (GT) [1]. Since in an abrupt transition, frame's content get changed suddenly; that is why, it is termed as a cut transition; also, while the GT occurs due to editing effects. The GT is further categorized into a dissolve, wipes, and fade. In these types of transitions, more than one frame is involved. The fade transitions mainly occur to due manipulation with the illumination of the frame. Further, fade-in and fade-out are the two main types of transition. When the visual entities in the frame slowly appear from a dark black frame, the process is known as fade-in, and when the lighting effect of the frame gradually gets blank, it is termed as a fade-out. When the content of two continuous shots gets mixed for some frames, that phenomenon is termed as dissolve transition and when transition takes place with some animated effect, that transition is classified as wipes transition. There are various application of an efficient SBD algorithm; for example, it can be used as a plug and play module for selecting informative frames for effective video captioning [2], video action recognition [3], and other video-related tasks.

The TRECVid videos are considered as standard dataset. Using these videos, lots of research work have been done in the field of boundary detection [4, 5]. Most of the work carried out in the field of SBD based on the nature of features employed for boundary detection can be categorized into two domains: compressed and uncompressed domains. Most of the SBD algorithm in compressed domain focuses on DCT co-efficient features, macroblock types, and motion vector-based features. [6–10]. While in an uncompressed domain, the pixel-based, block-based, and histogram-based are employed [11, 12]. The features extraction process is categorized into global and local feature extraction. The feature extraction process when the feature is extracted from the whole frame is global feature extraction, whereas in the local feature extraction process, features are extracted by considering a region or block of the frame. The popular local informative descriptors mostly used for SBD are SIFT, MSER, and SURF [13, 14].

For boundary detection, various machine learning-based approaches are also explored. For the classification of all the transition except wipes transition, a fuzzy logic-based approach is proposed in [15]. Further, in [16], the fuzzy logic-based approach is combined with a genetic algorithm for fine-tuning the fuzzy membership and detecting the boundaries. A deep learning-based approach is proposed in [17] from boundary detection. Using rough sets and fuzzy c-means clustering, an SBD algorithm is proposed in [18].

In [19], ignoring the thresholding-based approach and using discontinuity signal's cumulative moving average, a boundary detection algorithm is proposed. In this approach, the discontinuity signal categorized the frames into non-transition frames and possible transition frames.

For reducing the effect of illumination and motion to a certain limit, edge-based features have shown promising results [1]. When a large change in the position or number of edges appearing or disappearing is noticed, then a transition is declared at that location [10, 13]. Computation of motion strength using a block matching algorithm is also an intuitive idea for removing motion effects [20]. Using the edge, color, motion strength, and texture features collectively for getting a feature descriptor of each frame [21] proposed an SBD approach. A Walsh-Hadamard transform (WHT) kernel and matrix are employed in this approach.

In the proposed approach, features like CIEDE2000 color difference, normalized 3D Euclidean standard deviation, and 3D color difference are passed as input to FNN. For the weights optimization of FNN, an ALO algorithm is employed. Further, in the next stage, the possible transition frames are analyzed for classifying the frame into abrupt transitions.

Further, the paper is organized as follows: Sect. 2 includes a brief background knowledge of the features used in the proposed approach. In Sects. 3 and 4, details of weights optimization process and proposed approach are given, respectively. A detailed discussion on experimental results and parameter settings is given in Sect. 5, followed by conclusion and future work in Sect. 6.

2 Background Knowledge of Feature Extraction

In this section, a brief discussion on feature extraction approach used in the proposed system is done.

2.1 Color Histogram Difference

Color histogram difference (HD) is used as a feature in the proposed method. This feature is very simple and have an advantage of non-sensitivity to motion [15, 22]. It is also used extensively as a image feature for SBD. The normalized color histogram difference between ith and $(i + 1)$th frames of a video is evaluated using Eq. 1.

$$\mathrm{HD}_i = 1 - \left(\frac{1}{3n}\right)\left[\sum_{j=1}^{256}\min(K_{rj}^{i}, K_{rj}^{i+1}) + \sum_{j=1}^{256}\min(K_{gj}^{i}, K_{gj}^{i+1}) + \sum_{j=1}^{256}\min(K_{bj}^{i}, K_{bj}^{i+1})\right] \tag{1}$$

where n is the number of pixels in the frame. K_{xj}^{i} is the number of pixels in the jth bin of ith frame where $x = (r, g, b)$. The output of Eq. 1 remains in between [0, 1], and when the value of HD_i is close to 0, then both frames are classified as similar, but if this is close to 1, then both frames are classified as a dissimilar frames.

2.2 Normalized: 3D Euclidean Standard Deviation (SD)

To measure each pixel's difference of deviation from the mean between each plan of ith and $i + 1$th frames, SD is employed. The mathematical expression of SD is given by Eq. 2.

$$\mathrm{SD}_i = \frac{\sqrt{(\sigma_R^i - \sigma_R^{i+1})^2 + (\sigma_G^i - \sigma_G^{i+1})^2 + (\sigma_B^i - \sigma_B^{i+1})^2}}{M \times N} \tag{2}$$

where σ_X^i and σ_X^{i+1} $(X = R, G, B)$ represent standard deviation of ith and $(i + 1)$th frames, respectively. If the value of SD_i is close to 0, the similarity increases, and if SD_i is close to 1, the dissimilarity increases.

2.3 Color Difference: CIEDE 2000

The mathematical expression for CIEDE2000 color difference is derived from CIELab color space [19]. The CIEDE200 color difference between the a pair of color values in the CIELab space is evaluated using Eq. 3.

$$\Delta E_{00} = \Delta E_{00}(L_1^*, a_1^*, b_1^*, L_2^*, a_2^*, b_2^*)$$
$$= \sqrt{\left(\frac{\Delta L'}{K_L S_L}\right)^2 + \left(\frac{\Delta C'}{K_C S_C}\right)^2 + \left(\frac{\Delta H'}{K_H S_H}\right)^2 + R_T \left(\frac{\Delta C'}{K_C S_C}\right)^2 + \left(\frac{\Delta H'}{K_H S_H}\right)^2} \tag{3}$$

The hue, chroma, and lightness difference between the pairs of samples is given by $\Delta H'$, $\Delta C'$, and $\Delta L'$, respectively, in CIEDE2000, and R_T is a rotation function. In terms of correction to CIELab, five terms have been introduced in CIEDE2000 that are weighting functions, rotation term, chroma (S_C), lightness (S_L), and the hue (S_H) as shown in Eq. 3. For measuring the similarity between the sequence of frames in a video, the CIEDE 2000 color difference is employed.

3 Weights Optimization of FNN

3.1 Ant Lion Optimization

In the ALO algorithm, the initialization of ant matrices and ant lion is done using a random function. Each ant position with respect to an ant lion in every iteration is selected by the roulette wheel operator, and the elite gets updated using function B. For accomplishing the updated position, two random walks around the selected ant lion and elite are done. After accomplishing the random walk by all ants, a fitness function is used to evaluate, if any ant becomes fitter than other ant lions, then the position of ant lions for the next iteration is updated by the position of that ant. During optimization, a comparison between best ant and best ant lion is done and perform substitution if it is required. These steps continue until false is returned by function C.

4 Proposed Method

In this section, the details of proposed framework using hybrid FNN are discussed. Figure 1 shows the steps in the proposed framework.

4.1 Features Extraction

In the proposed framework, initially, the extraction of features takes place; it involves the extraction HD, ΔE, and SD between the consecutive frames.

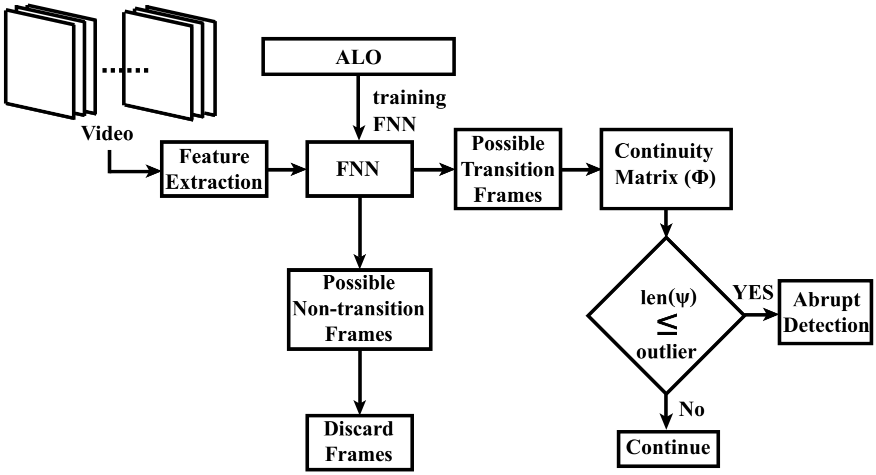


Fig. 1 Pictorial representation of the framework

4.2 Recognition of Possible Transition Frames

After feature extraction, classification of frames into non-transition and possible transition frames is performed using hybrid FNN. The optimization algorithm optimized both biases and weights of the FNN. Here, the frames which are classified as non-transition frames are rejected directly which helps in improving the computational efficiency of the proposed approach by discarding unwanted frames and processing only the required frames for the subsequent stages. During training, non-transition and transition frames are labeled with 0 and 1, respectively.

4.3 Continuity Matrix Generation (ϕ)

With the help of all the possible transition frames, ϕ is generated by assuming that there is no gap in the sequential order of the frames. If any gap exists in sequence, then that frame is added to the next row of ϕ. As the lengths of rows in ϕ vary so for consistency, each row is appended with zeros which are removed in upcoming stages of the algorithm.

4.4 Identification of Shot Transitions

In this stage, for the identification of transitions ϕ, and a value, Outlier are used. Based on the number of frames (or length) involved in the GT, the value of Outlier is set. Based on the previous observations from [23], the length of GT lies in between [6] frames, the value of Outlier is set to 6. To represent the rows of ϕ, a parameter ψ_i is used where i implies the ith row of ϕ. Based on the number of non-zero elements present in the ψ_i, the frames in the ψ_i is examined for abrupt transition using the condition shown in Eq. 4 where count of non-zero values in ψ is return by len(ψ).

$$\text{function}_{\text{call}} = \begin{cases} \text{Abrupt\_sec}(), & \text{if len}(\psi) \leq \text{ Outlier} \\ \text{continue}, & \text{otherwise} \end{cases} \tag{4}$$

Finally, for the declaration of the abrupt transitions, HD and ΔE are used. Algorithm 1 shows the pseudocode of proposed framework.

Abrupt Transition Detection For the declaration of abrupt transition frames the HD_i' and $\Delta E_i'$ are calculated for the frames present in the ψ_i using Eqs. 5 and 6.

$$\text{HD}_i' = \max\left(\text{HD}_{\psi_i} - \text{HD}_{\psi_{i-1}}, \text{HD}_{\psi_i} - \text{HD}_{\psi_{i+1}}, \text{HD}_{\psi_i} - \text{HD}_{\psi_{i-2}}, \text{HD}_{\psi_i} - \text{HD}_{\psi_{i+2}}\right) \tag{5}$$

Algorithm 1 Proposed transition detection algorithm

```
Input : $Input\ Video\ Stream,\ V$ Output : $Cut$
procedure BOUNDARY_DETECTION(V)
    $[HD, \Delta E, SD] \leftarrow$ Input Features (V);
    $PTF \leftarrow$ Hybrid_FNN(V);                    ▷ to find Possible Transition Frames (PTF)
    $\phi \leftarrow$ Continuity_Matrix(PTF);
    $Outlier = 6,\ \delta_1 = 0.17,$                   ▷ Experimentally calculated values
    $[m, n] = size(\phi)$;
    for p=1 to m do
        for q=1 to length($\psi_p$) do              ▷ $\psi_p$ represents the row of $\phi$
            $l = length(nnz(\psi_p))$
            if $l \leq Outlier$ then                  ▷ $Outlier\ is\ set\ at\ 6$
                ABRUPT_SEC($\psi, l$);
function ABRUPT_SEC($\psi, l$)
    $c = 1$;
    for i=1 to l do
        $HD_i' = max\left(HD_{\psi_i} - HD_{\psi_{i-1}}, HD_{\psi_i} - HD_{\psi_{i+1}}, HD_{\psi_i} - HD_{\psi_{i-2}}, HD_{\psi_i} - HD_{\psi_{i+2}}\right)$;
        $\Delta E_i' = max\left(\Delta E_{\psi_i} - \Delta E_{\psi_{i-1}}, \Delta E_{\psi_i} - \Delta E_{\psi_{i+1}}, \Delta E_{\psi_i} - \Delta E_{\psi_{i-2}}, \Delta E_{\psi_i} - \Delta E_{\psi_{i+2}}\right)$;
    for i=1 to l do
        if $HD_i' > \delta_1$ && $\Delta E_i' > \delta_1$ then      ▷ $\delta_1$ is a threshold
            $Cut(c) = \psi_i$;
            $c+ = 1$;
```

$$\Delta E_i' = \max\left(\Delta E_{\psi_i} - \Delta E_{\psi_{i-1}}, \Delta E_{\psi_i} - \Delta E_{\psi_{i+1}}, \Delta E_{\psi_i} - \Delta E_{\psi_{i-2}}, \Delta E_{\psi_i} - \Delta E_{\psi_{i+2}}\right) \tag{6}$$

Equation 7 is employed for the declaration of an abrupt transition between the consecutive frames ψ_ith and ψ_{i+1}th. The threshold δ_1 is used for the confirmation of abrupt transition which is set after experimentation. Table 3 in Sect. 5.3 is used for the selection of optima threshold, and it also shows the performance of the proposed approach with other threshold settings.

$$\text{Abrupt}_i = \begin{cases} \text{True}, & \text{if } \text{HD}_i' > \delta_1 \ \&\& \ \Delta E_i' > \delta_1 \\ \text{False}, & \text{otherwise} \end{cases} \tag{7}$$

5 Experimental Results and Discussion

5.1 Dataset

The prove the superiority, the proposed approach has tested over **TRECVid 2001** and **TRECVid 2007** datasets. Table 1 presents the details of all videos in the datasets.

Table 1 Statistic of both TRECVid 2001 and 2007 datasets

Video	Frame #	Transition			Sources
		Abrupt	Gradual	Total	
D2	16586	42	31	73	TRECVid 2001
D3	12304	39	64	103	
D4	31389	98	55	153	
D5	12510	45	26	71	
D6	13648	40	45	85	
BG_3027	49815	126	1	127	TRECVid 2007
BG_16336	2462	20	–	20	
BG_36136	29426	88	12	100	
BG_37309	9639	11	8	19	
BG_37770	15836	8	27	35	

5.2 Performance Evaluation

To evaluate the performance of the existing approaches and the proposed approach, evaluation parameters recall (Rec), precision (Pre) and F1 Score (F1) are employed as shown in Eqs. 8, 9, and 10, respectively. The symbols N_C, N_M, and N_F used in Eqs. 8, 9 and 10 are number of correctly, missed, and falsely detected transitions, respectively.

$$\text{Rec} = \frac{N_C}{N_C + N_M} \tag{8}$$

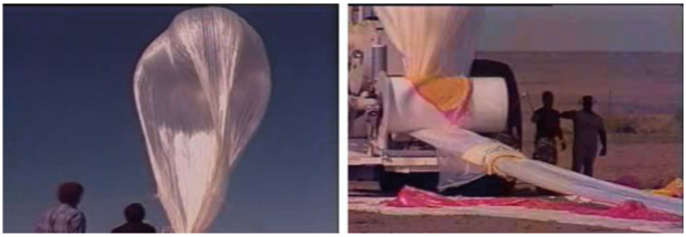
$$\text{Pre} = \frac{N_C}{N_C + N_F} \tag{9}$$

$$\text{F1} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{10}$$

An example of an abrupt transition is shown in Fig. 2, and Table 2 reports the performance of the proposed framework.

5.3 Threshold Selection

Since the content and behavior of the video vary from video to video, the selection of a single threshold that will work for all videos by resulting in a high F-score is very



(a) 468 (b)469

Fig. 2 Showing abrupt transition

Table 2 Performance of the proposed framework on both datasets

Videos	Parameter measure			Computation time in seconds (approx.)
	R	P	F1	
D2	92.9	90.7	91.8	1250
D3	97.4	100.0	98.7	732
D4	94.9	96.9	95.9	2500
D5	100.0	97.8	98.9	940
D6	100.0	95.2	97.6	960
BG_3027	100.0	95.5	97.7	3690
BG_3314	100.0	88.0	93.6	840
BG_16336	95.0	100.0	97.4	174
BG_37309	90.9	100.0	95.2	690
BG_37770	100.0	88.9	94.1	1147
Average	97.1	95.3	96.1	1292.3

Table 3 Variation in the F1 Score of TRECVid videos 2001 at different thresholds

Approach	Thresholds (δ_1)	Videos					Average
		D2	D3	D4	D5	D6	
ALO	0.15	82.6	96.2	94.0	96.8	92.0	92.3
	0.17	91.8	98.7	95.9	98.9	97.6	**96.6**
	0.20	90.5	96.0	94.8	97.8	96.3	95.1
	0.22	90.5	94.6	94.8	95.6	97.5	94.6

Bold-face signifies the highest score obtained by the algorithm

crucial. In this section, we analyzed the performance of the model with the different values of the threshold δ_1. From the experimentation, it is observed that when the value of δ_1 is set to 0.17, the model reported the highest average F1 Score. So, for all the videos, the value of δ_1 is set to 0.17 (Table 3).

5.4 Comparison

For the comparison, some state-of-the-art techniques are considered—gradient-oriented feature distance (GOFD) [25], WHT-SBD [24], fast framework [20], PSO-GSA [23], stationary wavelet transform (SWT) [26], ST-CNN [27], LBP-HF-based SBD [28], an adaptive low-rank and svd-updating approach in [29] and SBD using color histogram [30]. The comparison of the proposed approach with other recent methods is reported in Table 4.

Table 4 Performance analysis of proposed approach with recent methods

Methods metrics	Evaluation	Videos				Average
		D2	D3	D4	D6	
Proposed	R	92.9	**97.4**	94.9	**100.0**	96.3
	P	90.7	**100.0**	96.9	97.6	96.3
	F1	91.8	**98.7**	**95.9**	97.6	**96.0**
[24]	R	**97.0**	82.0	88.0	95.0	90.5
	P	85.0	86.0	90.0	97.0	90.0
	F1	91.0	84.0	89.0	96.0	90.0
[25]	R	80.0	82.0	78.0	92.0	83.0
	P	94.0	**100.0**	96.0	84.0	94.0
	F1	87.0	90.0	86.0	88.0	88.0
[26]	R	**97.0**	97.0	93.0	**100.0**	**97.0**
	P	6.0	8.0	7.0	8.0	7.0
	F1	12.0	16.0	13.0	16.0	14.0
[27]	R	57.0	46.0	75.0	89.0	67.0
	P	**100.0**	**100.0**	98.0	**100.0**	99.6
	F1	72.0	63.0	85.0	94.0	79.0
[23]	R	**97.0**	92.0	**100.0**	**100.0**	**97.0**
	P	82.0	**100.0**	89.0	**100.0**	92.0
	F1	89.0	96.0	94.0	**100.0**	94.0
[28]	R	89.0	92.0	85.0	87.0	88.0
	P	87.0	**100.0**	**100.0**	**100.0**	96.0
	F1	88.0	96.0	92.0	93.0	93.0
[29]	R	90.0	89.0	89.0	92.0	90.0
	P	**100.0**	**100.0**	94.0	97.0	98.0
	F1	**95.0**	94.0	92.0	94.0	94.0
[30]	R	92.8	92.3	91.8	100.0	94.2
	P	90.6	**100.0**	93.7	**100.0**	96.0
	F1	91.6	95.9	93.2	100.0	95.2
[30]	R	88.1	**97.4**	87.8	97.5	92.7
	P	94.9	88.4	91.5	92.9	92.0
	F1	91.4	92.7	89.6	95.1	92.2

Bold-face signifies the highest score obtained by the algorithm

6 Conclusion and Future Work

In this paper, a shot boundary detection approach is proposed for improving the performance of video surveillance systems. For an effective and computationally efficient surveillance system, we are required to select key event frames from the video, employing a automatic shot boundary detection algorithm as an initial stage is an intuitive approach. The proposed boundary detection approach takes the advantages of a nature-inspired algorithm ALO whose weights are optimized using the FNN. For the boundary detection, a continuity matrix (ϕ) is generated, and by employing a threshold δ_1, the boundary is declared. The overall performance of the proposed approach is comparable to recent SBD approaches in abrupt boundary detection.

The future work will be to extend the proposed boundary detection approach for detecting a GT in the video which will increase the effectiveness and robustness of surveillance systems.

References

1. Koprinska I, Carrato S (2001) Temporal video segmentation: a survey. Signal Process: Image Commun 16(5):477–500
2. Singh A, Singh TD, Bandyopadhyay S (2020) NITS-VC system for VATEX video captioning challenge 2020. arXiv preprint arXiv:2006.04058
3. Lin T, Zhao X, Su H, Wang C, Yang M (2018) BSN: boundary sensitive network for temporal action proposal generation. In: Proceedings of the European conference on computer vision (ECCV), pp 3–19
4. Thounaojam DM, Trivedi A, Manglem Singh K, Roy S (2014) A survey on video segmentation. Springer, New Delhi, pp 903–912
5. Smeaton AF, Over P, Doherty AR (2010) Video shot boundary detection: seven years of TRECVid activity. Comput Vis Image Underst 114(4):411–418. Special issue on Image and Video Retrieval Evaluation
6. Nagasaka A, Tanaka Y (1992) Automatic video indexing and full-video search for object appearances. In: Proceedings of the IFIP TC2/WG 2.6 second working conference on visual database systems II, Amsterdam, The Netherlands. North-Holland Publishing Co., pp 113–127
7. Arman F, Depommier R, Hsu A, Chiu My (1994) Content-based browsing of video sequences. In: ACM multimedia, pp 97–103
8. Arman F, Hsu A, Chiu MY (1993) Image processing on compressed data for large video databases. International conference on multimedia, MULTIMEDIA ’93. New York, NY, USA. ACM, pp 267–272
9. Nakajima Y (1994) A video browsing using fast scene cut detection for an efficient networked video database access (special issue on networked reality). IEICE Trans Inf Syst 77:1355–1364
10. Zhang H, Low CY, Smoliar SW (1995) Video parsing and browsing using compressed data. Multimed Tools Appl 1:89–111
11. Pei S-C, Chou Y-Z (1999) Efficient mpeg compressed video analysis using macroblock type information. IEEE Trans Multimed 1(4):321–331
12. Zhang H, Kankanhalli A, Smoliar SW (1993) Automatic partitioning of full-motion video. Multimed Syst 1(1):10–28
13. Anjulan A, Canagarajah N (2007) Object based video retrieval with local region tracking. Signal Process: Image Commun 22(7):607–621

14. Junaid B, Nitin A, Shin'ichi S (2013) A framework for video segmentation using global and local features. Int J Pattern Recognit Artif Intell 27(05):1355007
15. Fang H, Jiang J, Feng Y (2006) A fuzzy logic approach for detection of video shot boundaries. Pattern Recognit 39(11):2092–2100
16. Thounaojam DM, Khelchandra T, Singh KM, Roy S (2016) A genetic algorithm and fuzzy logic approach for video shot boundary detection. Comput Intell Neurosci 2016
17. Xu J, Song L, Xie R (2016) Shot boundary detection using convolutional neural networks. In: 2016 visual communications and image processing (VCIP), pp 1–4
18. Gao Xb, Han B, Ji Hb (2005) A shot boundary detection method for news video based on rough sets and fuzzy clustering. In: Proceedings of the second international conference on image analysis and recognition, ICIAR'05, Berlin, Heidelberg. Springer, pp 231–238
19. Gaurav S, Wencheng W, Datal EN (2005) The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations. Color Res Appl 30(1):21–30
20. Li Y, Lu Z, Niu X (2009) Fast video shot boundary detection framework employing preprocessing techniques. IET Image Process 3(3):121–134
21. Thounaojam DM, Bhadouria VS, Roy S, Singh KM (2017) Shot boundary detection using perceptual and semantic information. Int J Multimed Inf Retr 6:167–174
22. Jadon R, Chaudhury S, Biswas K (2001) A fuzzy theoretic approach for video segmentation using syntactic features. Pattern Recognit Lett 22(13):1359–1369
23. Chakraborty S, Thounaojam DM (2019) A novel shot boundary detection system using hybrid optimization technique. Appl Intell 49:3207–3220
24. Lakshmi Priya GG, Domni S (2014) Walsh-Hadamard transform kernel-based feature vector for shot boundary detection. IEEE Trans Image Process 23(12):5187–5197
25. Kar T, Kanungo P (2017) A motion and illumination resilient framework for automatic shot boundary detection. Signal Image Video Process 11(7):1237–1244
26. Warhade KK, Merchant SN, Desai UB (2013) Shot boundary detection in the presence of illumination and motion. Signal Image Video Process 7:581–592
27. Hassanien A, Elgharib MA, Selim A, Hefeeda M, Matusik W (2017) Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks. CoRR, abs/1705.03281
28. Singh A, Thounaojam DM, Chakraborty S (2019) A novel automatic shot boundary detection algorithm: robust to illumination and motion effect. Signal Image Video Process 1–9
29. Youssef B, Fedwa E, Driss A, Ahmed S (2017) Shot boundary detection via adaptive low rank and SVD-updating. Comput Vis Image Underst 161:20–28
30. Thounaojam DM, Thongam K, Jayshree T, Roy S, Singh KM (2019) Colour histogram and modified multi-layer perceptron neural network based video shot boundary detection. Int Arab J Inf Technol 16:686–693

Fair Trading of Crops in a Trusted and Transparent Manner using Smart Contracts

Vikas Chouhan, Sachi Pandey, Naman Sharma, and Naveen Prajapati

Abstract In the past few years, Indian farmers face problems due to middlemen's excessive exploitation in several ways. Intermediaries play a vital role as a bridge between farmers and consumers, due to the existence of several layers of commission agents or mediators(middlemen) who are buying crops from farmers at least possible cost and sell those at extravagant prices which monetary abuse both the farmers and consumers. The government has already tried plentiful various schemes, yojanas, and online solutions to clear up the obstacles, occur in the process of buying and selling produce. However, every technique to unfold the root problems was either short term or not up to the mark. Hence, it is necessary to provide a trusted, transparent, and decentralized solution. Therefore, this paper proposed a Blockchain-based framework that eliminated the agriculture sector's intermediates, specifically for crop trade. Additionally, we integrated Blockchain with the Android application to provide an easy interface and maximize user satisfaction. The proposed strategy makes it easy for farmers to sell their products to nationwide buyers, which provides them a reasonable allowance according to the input cost and the inflation rates, respectively, and it also creates a trusted wide marketplace to sell their crops. Through this, by exhibiting just a single layer in trade, the social and monetary welfare of farmers, consumers and government are targeted. We created several Smart Contracts and deployed to a Blockchain Network of Hyperledger Fabric Platform and then measured the performance using Hyperledger Caliper benchmark.

Keywords Agriculture · Crop · Blockchain · Smart contract · Hyperledger fabric · Hyperledger caliper

V. Chouhan (✉)
Department of Computer Science and Engineering, Indian Institute of Technology, Roorkee, India
e-mail: vchouhan@cs.iitr.ac.in

S. Pandey · N. Sharma · N. Prajapati
Department of Computer Scienceand Engineering, SRM Institute of Science and Technology, Delhi-NCR Campus, Ghaziabad, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_52

1 Introduction

In India, Agriculture is the foremost occupation and plays the most crucial segment of the economy. It contributes around 18% of India's gross domestic product (GDP) and employs over 50% of entire employment, which is a massive number [7]. With the huge number, there are many loopholes as it has multiple levels of work; if we put lights on farmers and consumers, we can notice a big misconstruction due to a level called mediators or middlemen by which two levels (farmers and consumers) are badly exploited. The most prominent reason for this problem is unawareness and lack of farmers' knowledge, leading to misuse of their outputs. Various private and government organizations are directly and indirectly involved in agriculture to somehow make this sector better, but after implementing tons of schemes, funds, and yojanas, there is no fantastic success noticed yet. In this modern era, the industry is still using traditional tools instead of using new technologies and machines to make their work (harvesting, plow, seeds sowing, etc.) much easier.

The main problem arises when middlemen buy crops from farmers at lower prices and sell them in the market at higher prices that exploit farmers and consumers, farmers do not have any options because the government's Minimum Support Prices (MSP) are comparatively low to the inflation rates at that price. They cannot even repay the expenditure on the cultivation of products, and due to lack of feasible options, they sell their crops to the middlemen at a lower price. Later, mediators sell those crops to local vendors (sabzi walas) at a higher price. Those vendors also add their profit and sell those in the market to consumers; finally, when any vegetable or crop reaches the consumer, its price abruptly rises.

In a report, CRISIL has found that "While, the average annual growth (in MSP) between the agriculture year 2009 and 2013 was 19.3%, it was only 3.6% between 2014 and 2017. India has seen an inflation of 5–7% in different regions. A government employee gets up to 15% inflation allowance". In fact, due to exploitation done by middlemen, private landlords, and loan providers with high-interest rates, farmers get caught into debt traps, which results in suicide. As per the National Crime Records Bureau (NCRB) [9, 16], the number of self-destructions by farmers and farmworkers raised to 12,360 in 2014 against 11,772 in the past year. Of these self-destructions, 5650 were farmer's suicides. The main cause of self-destructions is middlemen's [11]. Due to several levels, corruption and profiteering are drastically increased, producers and consumers are exploited at each level, so we got an approach which is quite helpful in this problematic scenario.

In the proposed approach, we made an android application which supports the farmer in service providing and good selling in every possible way. The main highlight of this application is that there is an online product-selling portal controlled by the government. Due to this, a farmer can sell their products nationwide with more profits, and the buyer has to request the products on this app, and the pricing(bid) proposed by the buyer to sell should be higher than the MSP given by the government. This policy supports the farmers in monetary terms. Along with that app is having several core useful features. NEARBY FACILITIES is a feature by which users can locate nearby banks, marketplaces, warehouses, and much more useful stuff using GPS. CROP DEMAND is the feature that provides the details of live market by which farmer can monitor that which crop is in demand or which crop should be sowed and yield with maximum profit, and this feature can balance the demand and supply chain and obstruct the variable pricing of crops. In farming, farmers can get the Ripe Track feature by which they can track the crop's ripeness level, which can help them to sell their crops at reasonable prices. Finance and Loan, this feature enables the farmers to know about nearby banks, their interest rates, or outstanding loans provided for farmers, and in this section, the user will get to know about government loan and finance schemes for farmers. By this feature, farmers are not supposed to be caught in the debt trap of local money lenders who lend their capital at higher interest rates.

Blockchain can be viewed as a decentralized database in which information can be stored, removing all access to a single entity, eliminating third parties, and providing trustworthy trade [1]. Therefore, we innovatively adapted this technology to maximize user satisfaction. We integrated Blockchain Smart Contracts with Android application to provide trust and transparency between the Blockchain network participants. Every transaction in a Blockchain Network is captured into the shared, immutable ledger visible to all participants. Ledger provides tamper-proof evidence that renders fraud-tolerance and enables trust between untrusted participants. All the participants collectively maintain the ledger record, typically through a P2P network [17]. The network must verify a new record before appending it to the blockchain. Direct communication between producers and consumers will be the cause behind the elimination of mediators. These techniques will be governed by the government to deliver minimal deprivations with maximum agricultural market returns.

2 Related Work

Among the various surveys and feedback have done across the country [3, 18], the root problem arises in the agricultural sector is mediators (middlemen), which exploits farmers as well as the market functionaries at each stage in exchange of goods. Due to fewer MSPs, farmers end up dealing with mediators and get tricked later, which contrives the farmers, and as the chain goes ahead, each level gets affected, which ends up with the crops getting sold at much higher prices. As the producers (farmers) get low value for their crops with fewer profits, the production for the next cycle of crops gets extravagant. Farmers borrow money from banks and private money lenders with high-interest rates, getting backstabbed and resulting in suicides.

Nalinipriya et al. [8] discussed the enhancement of farmer's lifestyles that include some interactive features like bidding and crop demand are immensely useful for farmers to get better prices and monitoring availability and lack of stocks of different crops in the market. Crop Biding delivers the user better choices for its investment in crops when buying and selling. However, the Crop Demand feature in the application can help farmers predict the demand and supply of different crops. A crop price forecasting engine is proposed in [2] based on Autoregressive Integrated Moving Average (ARIMA).

Recently, Blockchain technology is increasingly bringing significant attractions in the agricultural sector with its diverse applications in the ecosystem for agricultural products [4, 6]. It will be positively impactful in eliminating intermediaries; this technology transforms the way trust is achieved - instead of trusting mediators, trust is gained through the cryptography techniques and P2P architecture. Thus, it helps in restoring the trust between sellers and buyers, which can minimize the transaction expenses in the agriculture business [15]. It can track food provenance and thus help create trustworthy food supply chains. It supports revolutionizing the Agricultural and Food (Agri-food) supply chain, which is presented [12] by enabling the transparency toward food supply, monitoring product quality, traceability, and food safety throughout the forecast period for smart farming using a blockchain network.

Lin et al. [5] present the blockchain-based food safety and traceability mechanism to manage the food data. It is essential to monitor accurately, share, store, and trace the specific data within the whole food supply chain, including production, processing, distributing, transportation, retail, and consumption.

Several apps are developed and used by farmers for smart agriculture [10]. All of these apps have different usage as per its functionalities and being utilized for different kind of features regarding the farming activities like cropping and irrigation knowledge, pesticides, fertilizer, seed, selling and buying of the crop, prediction of crop production, estimation of demand, and supply of different crops, weather knowledge and information regarding the best practices of farming [10]. We found that most of the apps are static and centralized. Instead of that, dynamic and decentralized apps will be better to use. Also, if all such listed functionalities are bundled into one single application and in the native language, it is easy to utilize it.

Our application has been proposed with multiple functionalities, including selling and buying crops, weather info, finance and loan, agriculture schemes, and crop demand-android application benefits to connecting farmers with retailers and excluding mediators. Moreover, delivering a straightforward way to trade. An application such as [13] with basic features provides communication between farmers and the agricultural market.

3 Proposed Framework

In this section, we present a Blockchain-based framework along with an android application and its features. We created several Smart Contracts (SC) and deployed them into the Blockchain Network (BCN). Specific SC is automatically invoked when the requisite event is triggered; it recorded entire transactions into the distributed ledger. The proposed crop-based framework is illustrated in Fig. 1.

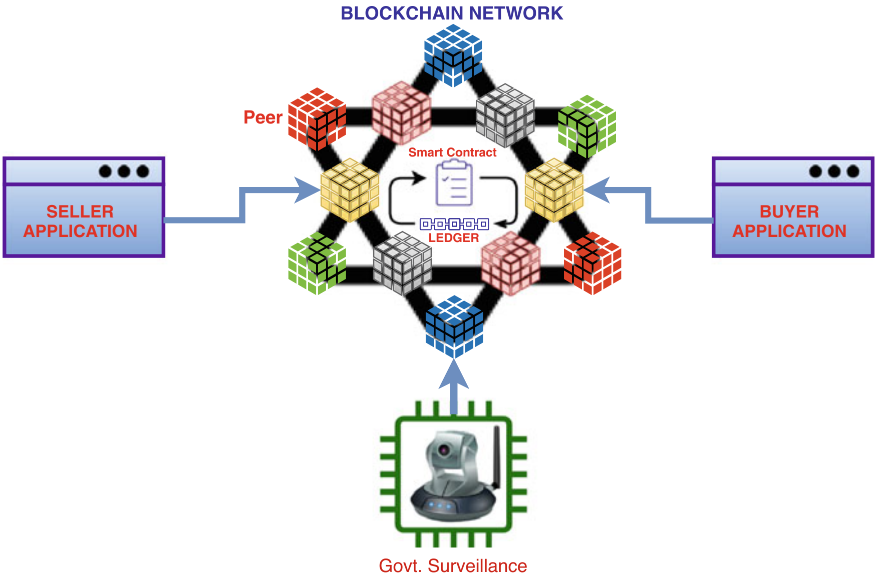


Fig. 1 Proposed framework

This framework primarily consists of seller, buyer, and government surveillance applications that are executing on top of decentralized BCN to enable trust and transparency between participating entities. Further, the application has six functionalities that are: (i) SELLING and BUYING of produce between buyers and sellers, (ii) FINANCE and LOAN to provide updates about better interest rates, (iii) AGRICULTURAL SCHEMES feature to provide information about various government and private schemes, (iv) CROP DEMAND for the current demand of a particular crop, (v) RIPE TRACK for tracking the crops, and (vi) LOCATION tracking feature for nearby banks, farming facilities, and warehouses available in the region. Further, the APP screenshots for adding products and buying crops are displayed in Fig. 2.

3.1 Smart Contracts

A smart contract (SC) is a set of executable codes that is automatically executed when a particular condition is triggered according to the contract's predefined terms [14]. It allows the performance of credible transactions without third parties (middlemen); this implementation will surely create a sense of trust toward modern technologies and will motivate people to use the latest technology instead of old traditional ways of trade in agriculture. We created and deployed several SCs in BCN, such as inserting

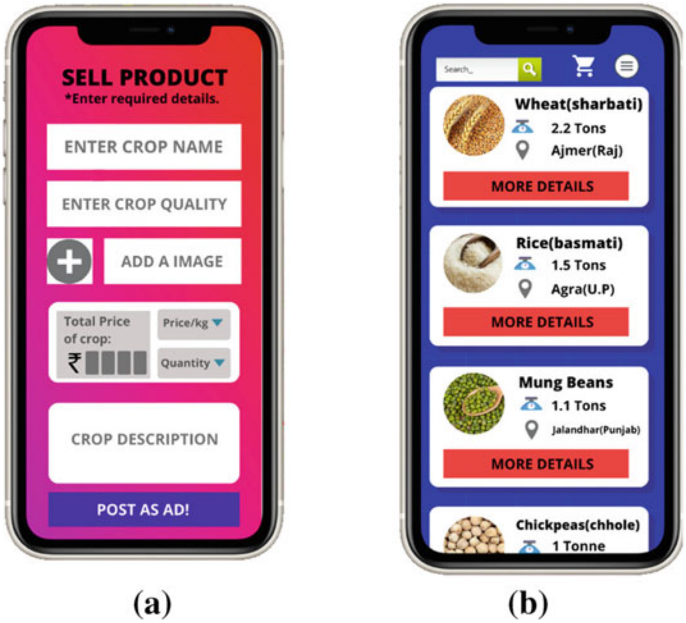


Fig. 2 **a** Adding product detail in the application. **b** Buying new crop and comparing prices.

and viewing sellers, buyers, crops information to/from blockchain, and crop transactions from the seller to buyer. Specifically, we present three SCs for crop module, i.e., Insert crop details, crop transaction, read crop.

The ‘Insert crop details’ contract inserts the crop information such as crop name, quantity, quality, image, and details into the blockchain described in Algorithm 1. It takes the stub and arguments list as an input and returns a corresponding response. This contract first checks the required number of input arguments. Then it initialized all the arguments in the form of structure for inserting the crop record into the blockchain. The ‘CropDetails’ structure contains crop information, and the ‘InsertCropInstance’ structure is used for inserting crop records into the decentralized platform. The crop instance structure contains Crop Identity (CID), CropDetails, Price, Seller Identity (SID), and Availability information (Avail). Then it checks whether the crop record exists, returns an error if the crop record already exists. Later, it marshals the crop record and stores them into the block corresponding to the CID by executing the function PutState. Finally, it returns a successful response containing a message ‘crop record has been stored’.

The ‘read crop’ smart contract presented in Algorithm 2 describes the procedure of viewing the BCN’s crop record. It takes the stub and arguments list as an input and returns the retrieved crop record corresponding to the CID as a response. First, it invokes the GetState method of the blockchain’s world state database to retrieve stored information as a Bytes form if information exists; otherwise, it failed to get

Algorithm 1: Smart contract for inserting crop details

```
Input  : $stub\ ChaincodeStubInterface,\ args\ [\ ]string$
Output: $Response$
begin
    if $len(args)! = 8$ then
        $return\ Error($“$Incorrect\ number\ of\ arguments.\ Expecting\ 8$”$)$
    end
    /* Initializing all the structures                                   */
    $var\ crop\ CropDetails$
    $crop.CropName \longleftarrow InitStructValue(args[1])$
    $crop.ImageData \longleftarrow InitStructValue(GetImageData(args[2]))$
    $crop.RemainQuantity \longleftarrow InitStructValue(args[3])$
    $crop.Quality \longleftarrow InitStructValue(args[4])$
    $crop.Details \longleftarrow InitStructValue(args[5])$
    $var\ insert\ InsertCropInstance$
    $insert.CID \longleftarrow InitStructValue(args[0])$
    $insert.CropDetail \longleftarrow crop$
    $insert.Price \longleftarrow InitStructValue(args[6])$
    $insert.SID \longleftarrow InitStructValue(args[7])$
    $insert.Avail \longleftarrow true$
    /* Check if crop record already exists                               */
    if $IsCropRecordExist(stub, insert.CID)$ then
        $return\ Error($“$This\ crop\ record\ $”$+ insert.CID\ +$ “$is\ already\ exists.$”$)$
    end
    /* Putting state in block                                            */
    $PutState(insert.CID,\ Marshal(insert))$
    /* crop record has been entered                                      */
    $return\ Success($“$crop\ record\ has\ been\ stored$”$)$
end
```

state and return an error response. Then the ‘croprecord’ object is created to stores the unmarshal information of record Bytes. Finally, it returns a success response, including retrieved bytes information.

Algorithm 2: Smart contract for view stored crop record

```
Input  : $stub\ ChaincodeStubInterface,\ args\ [\ ]string$
Output: $Response$
begin
    $cid\ :\longleftarrow\ args[0]$
    $recordAsBytes, err\ :\longleftarrow\ stub.GetState(cid)$
    if $err\ ! =\ nil$ then
        $jsonResp\ :\longleftarrow$ “{\“$Error$\”: \“$Failed\ to\ get\ state\ for$” $+ cid\ +$ “\”}”
        $return\ Error(jsonResp)$
    end
    $var\ croprecord\ InsertCropInstance$
    $Unmarshal(recordAsBytes, \&croprecord)$
    $return\ Success(recordAsBytes)$
end
```

The ‘crop transaction’ smart contract is defined in Algorithm 3 that describes the crop transaction procedure from seller to buyer. It takes the stub and arguments list as an input and returns a corresponding success or failure response. The following steps are involved in this contract.

Algorithm 3: Smart contract for creating crop transaction

```
Input  : $stub\ ChaincodeStubInterface,\ args\ [\ ]string$
Output: $Response$
begin
    if $len(args)! = 5$ then
        return Error(“Incorrect number of arguments. Expecting 5”)
    end
    $cid, sid, bid, pid, reqquant :\longleftarrow InitializeValues(args)$
    if $reqquant <= 0$ then
        return Error(“This buyer didn't insert enough quantity− ” + reqquant)
    end
    $seller, err :\longleftarrow GetSeller(stub, sid)$
    if $err! = nil \,||\, seller.Enabled == false$ then
        return Error(“This seller does not exist or is disabled−”+ sid)
    end
    $buyer, err :\longleftarrow GetBuyer(stub, bid)$
    if $err! = nil \,||\, buyer.Enabled == false$ then
        return Error(“This buyer does not exist or is disabled− ”+ bid)
    end
    $cropInstance, err :\longleftarrow GetCropInstance(stub, cid)$
    if $err! = nil \,||\, cropInstance.Avail == false$ then
        return Error(“This cropInstance does not exist or is unavailable− ”+ cid)
    end
    $tR :\longleftarrow cropInstance.CropDetail.RemainQuantity$
    if $tR >= reqquant \,\&\&\, tR > 0$ then
        $tR \longleftarrow tR - reqquant$
        $cropInstance.CropDetail.RemainQuantity \longleftarrow tR$
        Print(“The crop's remaining quantity are” + tR)
    end
    else if $tR > 0 \,\&\&\, reqquant > tR$ then
        return Error(“Not enough quantity available. Your maximum required quantity
        of crops is : − ” + tR)
    end
    if $tR <= 0$ then
        Print(“The crop with cid ” + cid + “is unavailable”)
        $cropInstance.Avail \longleftarrow false$
    end
    /* Putting state in block                                          */
    PutState(cropInstance.CID, Marshal(cropInstance))
    /* crop record has been entered                                    */
    var purchase PurchaseInstance
    $purchase.PID \longleftarrow pid$
    $purchase.CropDetail.CropName \longleftarrow cropInstance.CropDetail.CropName$
    $purchase.CropDetail.ImageData \longleftarrow cropInstance.CropDetail.ImageData$
    $purchase.CropDetail.Quality \longleftarrow cropInstance.CropDetail.Quality$
    $purchase.CropDetail.Details \longleftarrow cropInstance.CropDetail.Details$
    $purchase.SID \longleftarrow sid$
    $purchase.BID \longleftarrow bid$
    $purchase.SellQuantity \longleftarrow reqquant$
    /* Putting state in block                                          */
    PutState(purchase.PID, Marshal(purchase))
    /* purchase record has been entered                                */
    return Success(“purchase record has been stored”)
end
```

- This contract first checks the required number of input arguments. Then it calls the function InitializeValues to set all the required values, i.e., identities of the crop, seller, buyer, purchase, and required quantities. It returns with error response if the required quantities for purchase are less than equal to zero.
- It gets the seller object corresponding to the seller identity (sid) by calling the function GetSeller. It returns with an error response if the seller does not exist or its license is disabled.
- It gets the buyer object corresponding to the buyer identity (bid) by calling the function GetBuyer. It returns with an error response if the buyer does not exist or its license is disabled.
- It gets the 'cropInstance' object corresponding to the crop identity (cid) by calling the function GetCropInstance. It returns with error response if the cropInstance does not exist or is unavailable.
- Total remaining quantity is assigned to the variable tR.
- In the case, quantity of the remaining crop (tR) is greater than equal to the required quantities and the existing tR greater than zero, it set the tR values by remaining quantities (tR-reqquant) and then updates the remaining quantities of the crop instance by tR. It also prints the remaining crop quantity.
- Otherwise, it returns an error response if the remaining tR is greater than zero and the required quantities are greater than the remaining quantities.
- It makes crop is unavailable if the remaining tR quantities are less than equal to zero.
- Later, it marshals the 'cropInstance' and stores them into the block corresponding to the CID by executing the function PutState.
- The 'purchase' object is then created for 'PurchaseInstance' structure to set all the values of purchase instance, and then it marshals the purchase object and stores them into the block corresponding to the purchase identity (PID) by executing the function PutState.
- Finally, it returns a successful response containing a message 'purchase record has been stored'.

4 Experiments and Results

4.1 Implementation Details

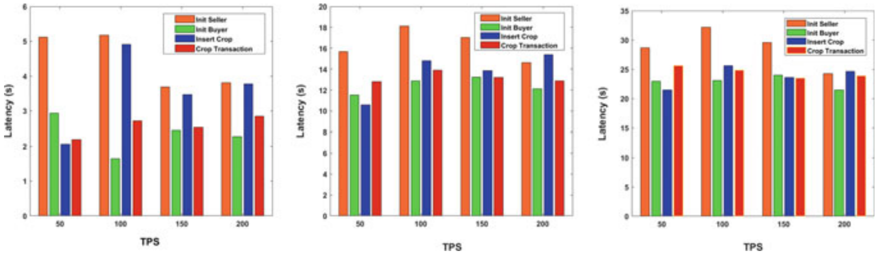
The application is developed in Android IDE to create an interactive graphical user interface with a better layout and smooth performance using various supporting APIs. Also, integrating with the Hyperledger Fabric Blockchain Platform, where the android app will communicate with BCN using REST API. Further, the performance is captured using Hyperledger Caliper Benchmark with the predefined configuration of sending rate between 50 and 200 transactions per second (tps) for all the Smart Contracts.

The application is developed for two languages, English and Hindi, with some more enhanced features to provide a feasible approach for farmers to gain direct access to the market. The application has been pilot-tested at many levels as well as by some farmers and worked well, with satisfactory results and positive feedback.

4.2 Performance Analysis

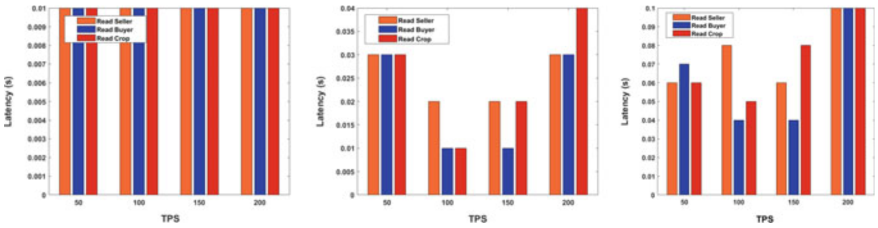
We conducted several experiments at different transaction rates to observe the performance of deployed smart contracts that are: (i) Register and initialize seller into the BCN, (ii) Register and initialize buyer into the BCN, (iii) View seller from BCN, (iv) View Buyer from BCN, (v) Insert crop details, (vi) Read stored crop information and (vii) Crop transaction from the seller to the buyer. The performance is measured using the Caliper benchmark tool based on transaction latencies and throughputs. The performance metrics of the Caliper benchmark is captured for {50, 100, 150, 200} transactions per seconds.

We grouped the performance of latency operations into three sets, i.e., minimum, average, and maximum, for analyzing the performance behavior of smart contracts. Initialization and transaction operations latencies of smart contracts are shown in



(a) min **(b)** avg **(c)** max

Fig. 3 Initialization and transaction operations latencies of smart contracts.



(a) min **(b)** avg **(c)** max

Fig. 4 Reading operations latencies of smart contracts.

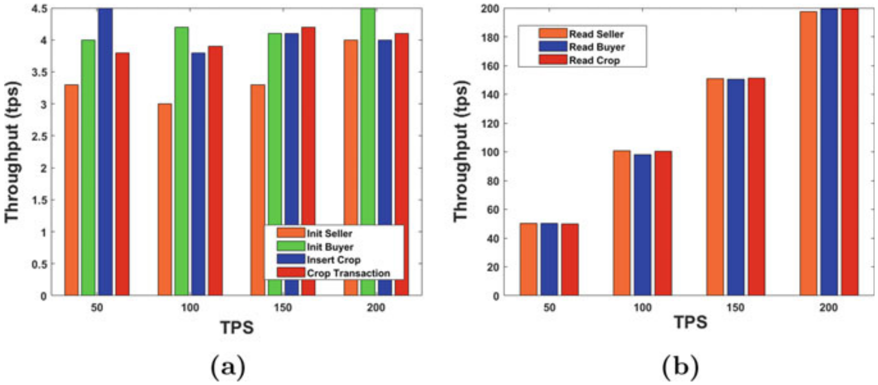


Fig. 5 Smart contracts throughput's **a** throughput of initialization and crop transaction operations, **b** throughput of reading operations.

Fig. 3. As shown from the graph, the optimal latencies take the least 1.64s, an average 10.57s, and the highest 21.43s. Then the reading operations latencies of smart contracts are presented in Fig. 4. The optimal latencies of all reading operations are consistent, ranging between 0.01 and 0.06s. Latencies of reading operations are more economical than other operations because it does not carry other complicated operations.

Further, Fig. 5 represents the throughputs of the smart contracts. The throughputs are grouped into two sets: (a) initialization and crop transaction operations, (b) reading operations for analyzing the performance behavior of smart contracts. As can be seen from Fig. 5a, the throughput of the smart contract varies between 3 and 4.5s. However, the highest throughputs are achieved at 100–200 tps, shown in Fig. 5b because the throughputs of reading operations are more reliable than other operations since they only perform reading queries from the databases.

5 Conclusion

The agricultural sector's base problem is the middlemen, which gives a severe impact on farmers due to which the whole agricultural cycle is affected, such as supply-demand chain, better facilities, and prices on crops. Our work targets the construction of trusted and transparent blockchain-based applications that mainly focus on eliminating mediators by providing direct communication via technology to connect the farmers and the government and provide better engagement between producers and consumers or retailers. The application is filled with better features to provide feasible ways for farmers to trade the produce much effectively. Features such as selling and buying crops, results in getting better prices by contacting directly without any mediators, also providing live updated data about the MSPs for each crop, weather conditions in the region, location-based facilities like farmers training centers and warehouses nearby available in that area and also giving updates about the government schemes and better loan providers, to avoid farmers from getting into debt traps. It will help the agricultural sector and the trade market to connect nationwide and will positively impact the economy of the country.

Further future implementations and services can be added, such as upgrading the ripe track feature, which will help the farmers to get better info regarding the crops and improve weather forecasting to provide live details about the surroundings for better caretaking. Enhancing the buying and selling features and integrating the whole agricultural market in the application will gain more consumers as well as customers nationwide in a common trade place.

References

1. Andrian HR, Kurniawan NB, Suhardi (2018) Blockchain technology and implementation : a systematic literature review. In: 2018 International conference on information technology systems and innovation (ICITSI), pp 370–374
2. Dissanayeke U, Perera A, Hewagamage K, Wikramanayake G (2015) Mobile based collaborative learning tool to facilitate instructor-mediated informal learning in agriculture. In: 2015 Fifteenth international conference on advances in ICT for emerging regions (ICTer), pp 99–105. IEEE
3. Jebaraj P (2020) 60% of farmers faced losses: survey. https://www.thehindu.com/news/national/60-of-farmers-faced-losses-survey/article31635197.ece
4. Kamilaris A, Fonts A, Prenafeta-Boldύ FX (2019) The rise of blockchain technology in agriculture and food supply chains. Trends in Food Sci Technol 91:640–652
5. Lin Q, Wang H, Pei X, Wang J (2019) Food safety traceability system based on blockchain and epcis. IEEE Access 7
6. Lin W, Huang X, Fang H, Wang V, Hua Y, Wang J, Yin H, Yi D, Yau L (2020) Blockchain technology in current agricultural systems: from techniques to applications. IEEE Access 8:143920–143937
7. Madhusudhan L (2015) Agriculture role on Indian economy. Bus Econ J
8. Nalinipriya G, Sangeetha R, Saniya K, Navarath SSD (2019) Agro bidding-a smart dynamic system for enhancement of farmer's lifestyle. In: 2019 International conference on smart structures and systems (ICSSS), pp 1–4. IEEE
9. NCRB: 10349 farmers committed suicide in 2018: NCRB (2020). https://economictimes.indiatimes.com/news/politics-and-nation/10349-farmers-committed-suicide-in-2018-ncrb/articleshow/73173375.cms?from=mdr
10. Patel H, Patel D (2016) Survey of android apps for agriculture sector. Int J Inf Sci Tech 6(1–2):61–67
11. Ray S (2017) Expert blames middlemen for farmers' suicides. https://timesofindia.indiatimes.com/city/patna/expert-blames-middlemen-for-farmers-suicides/articleshow/60135918.cms
12. Shahid A, Almogren A, Javaid N, Al-Zahrani FA, Zuair M, Alam M (2020) Blockchain-based agri-food supply chain: a complete solution. IEEE Access 8:69230–69243
13. Shriram P, Mhamane S (2018) Android app to connect farmers to retailers and food processing industry. In: 2018 3rd International conference on inventive computation technologies (ICICT), pp 284–287. IEEE
14. Szabo N (1997) Formalizing and securing relationships on public networks. First Monday 2(9)
15. Tripoli M, Schmidhuber J (2018) Emerging opportunities for the application of blockchain in the agri-food industry. FAO and ICTSD: Rome and Geneva. Licence: CC BY-NC-SA 3

16. Vishnu Padmanabhan PD (2020) The geography of farmer suicides. https://www.livemint.com/news/india/the-geography-of-farmer-suicides-11579108457012.html
17. Xiong H, Dalhaus T, Wang P, Huang J (2020) Blockchain technology for agriculture: applications and rationale. Front Blockchain 3(7)
18. Yadav VS, Singh A (2019) A systematic literature review of blockchain technology in agriculture. In: Proceedings of the international conference on industrial engineering and operations management, pp 973–981

Stochastic Based Part of Speech Tagging in Mizo Language: Unigram and Bigram Hidden Markov Model

Morrel V. L. Nunsanga, Partha Pakray, Mika Lalngaihtuaha, and L. Lolit Kumar Singh

Abstract The process of assigning words in a corpus to the corresponding specified tags based on the context and its definition is part of speech (POS) tagging. It is always been a big challenge yet a very important task in language processing. The task of a part of speech tagging is more challenging in low resource languages, for example, Mizo language. This paper presents the development of a data-driven part of speech tagging system for the Mizo language. This research work includes creating of tagset and annotated Mizo corpus, development of stochastic based taggers such as unigram and bigram Hidden Markov model. The highest accuracy obtained using the proposed unigram and bigram Hidden Markov Model (HMM) based part of speech (POS) taggers are 70.61% and 75.19% respectively.

Keywords Mizo language POS tagger · Hidden Markov model · Part of speech tagging · Unigram · Bigram-HMM · Part of speech tagging in Mizo language

1 Introduction

There are numerous numbers of languages in the world, spoken by human beings. These languages are very complex, diverse and unique in its own. According to the 23rd Edition of the Ethnologue [1], there are around 7117 living human languages in the world today, out of which English has the most speakers, with around 1268 million speakers from different 146 countries. Whereas considering native speakers only, Mandarin Chinese has the highest number in the world, with 917 speakers followed by Spanish, with around 460 million native speakers. Of the world's 142 language families, Trans-New Guinea, Niger-Congo, Indo-European, Austronesian, Sino-Tibetan and Afro-Asian are the main families of languages making up five-sixths of the world's population.

M. V. L. Nunsanga (✉) · M. Lalngaihtuaha · L. Lolit Kumar Singh
Mizoram University, Aizawl, India
e-mail: morrelhmar@mzu.edu.in

P. Pakray
NIT Silchar, Silchar, India

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_53

Mizo language is one of the 453 languages in India, spoken mainly by the people of Mizoram, which is one of the 27 states in India. Other than the mainland of Mizoram, this language is also spoken in Myanmar, Bangladesh and other parts of India, such as Tripura, Meghalaya, Assam, Manipur, and Nagaland. According to 2011 Census, there are around 8.3 lakh people speaking Mizo language in India and 8.45 lakhs users in all over the world. The language is categorized to endangered language by the UNESCO [2] Mizo language need more attention and more efforts to make the language developed with the modern language technology. Research in the field of language processing has shown giant leap along with the advent of new technologies in recent years. Lot of researchers have given attention in the research field of computational linguistics and lexical resources for many languages become stable and give open space for further development of the language in the language processing. In spite of all development in computational linguistics fields for various languages, Mizo language is far behind other major Indian languages. The main aim of this study is to lay a groundwork for development of Mizo language in Natural Language Processing field.

Natural Language Processing (NLP) is a sub discipline of Artificial Intelligence (AI) that discuss with how words and their surrounding meaning can be interpreted by a computer programme. If focus on enabling computers to understand human languages. Natural language processing turns unstructured text data to structured data to be analysed further. The majority of data in textual form is usually highly in unstructured form. In order to produce significant information inside from the data, it is important to get acquainted with the techniques of text analysis and proper presentation.

In any language processing, part of speech tagging plays a significant role. It is the way toward labelling each token in a corpus with a designated word class to indicate its grammatical information about the token. The defined set of labels that is used to indicate the part of speech of a language and its grammatical information about the token is called tagset. These tagsets are differed for different languages and are usually designed carefully to provide the abstract representation of the morphological features of the text in the corpus. This information helps in further analysis of the corpus and the text processing. POS tagging is one of the initial processing modules in any language processing pipelines that can be considered as the prerequisite task which simplifies many complex problems in the computational linguistic fields.

Different techniques and methods have been employed for tagging system in the language processing and they are called POS taggers. These taggers performed differently on different dataset depends on the type of the dataset in terms of efficiency and accuracy. They can roughly be divided into three categories [3] such as Rule based approaches, stochastic or statistics approaches and transformation based approaches. Rule based tagging systems implicate with detailed rules written manually for disambiguation in the sentences and words are tagged based on these written rules. These rules could be diverged for different languages as the syntactic structures of different languages vary. Stochastics or Statistics based tagging system are also called probabilistic based tagging system. They are depending on the training dataset to compute the probability of the certain tag sequence occurring. The Transformation based tagging system is also named as Brill tagging system. It takes the advantage of both the previous architectures—rule based and stochastic based.

The language of Mizo is a grammatically rich and complex language with less computational research tools. It is possible to drive forward research in this language by creating more lexical tools. The main contribution of this research work is development of basic works of stochastics based POS tagger using bigram Hidden Markov Model (HMM) and unigram tagger as a comparative study. This exploration work includes building of a reliable annotated corpus and development of tagset, which is designed to represent morphological features of the Mizo language.

2 Related Works

In this section, we present the research works related with part of speech tagging system for different languages. Since the inception of coding of part of speech tagging system in the year 1960s, lots of improvements have been made with different techniques and methods. System based on statistical method is one of the most popular tagging systems for different languages. HMM based part of speech tagging method was introduced [4] in the mid-1980s.

HMM based Part of speech tagger for Arabic is discussed in [5]. The paper presents characteristics of Arabic languages and 55 tagsets have been proposed. They have developed a 9.15 MB corpus of native Arabic articles. Words of 23,554 verbs, 27,594 nouns, 5384 proper nouns, and 5722 adjectives were chosen to train the tagger. 944 words were used as tested corpus and achieved accuracy of 97%.

Statistical POS tagging system in Persian text is presented in [6]. They have created tagged corpus and evaluated statistical based TnT tagger on Persian language. The experiments were repeated several times on 80% and 15% of the corpus as training data and test data respectively and the data were selected randomly. The obtained overall average accuracy is 96.59%.

A Part Of Speech tagging system for Urdu Language based on statistical model is discussed in [7]. In this paper, a supervised learning, n-gram Markov model tagging method has been used. The experiments have been performed based on the unigram, bigram and back off methods on small and large tagset. They achieved higher accuracy with smaller tagset and with the back off method, and they could achieved 95% accuracy.

Arabic Part of Speech Tagging based on Parallel HMM is discussed in [8]. In this paper, they proposed a new approached tagging system relies on two HMM working together in parallel in the system. The first one is the main model, and the second model is used as reference for low probabilities tags. Both the models are trained using dual corpus. To overcome the time complexity, the system was implemented using multithreading approached. The average accuracy of 75.38% was obtained on small dataset. Though the concept is novel, the system is tested on very small dataset which consist of 40 numbers of sentences (845 words). The performance of the system is yet to be trialed on large dataset to see the actual performance of this method.

Cahyani and Vindiyanto [9] discuss presented POS tagger for Indonesian language based on the HMM N-gram (bigram and trigram) approach and Viterbi algorithm. They have compared HMM bigram and HMM trigram on the Indonesian language corpus and found that HMM bigram scored better with the accuracy of 77.56% whereas 61.67% accuracy was obtained with HMM trigram.

Joshi [10] presented a POS tagging system for Hindi language based on Hidden Markov Model. 15,200 sentences were utilized to train the system and the IL POS tag set were used in the system. They obtained 92% accuracy on the test data.

HMM based POS Tagging system for Kayah Language is discussed in [11]. They have developed 16 tagsets to disambiguate words in the Kayah language and they achieved the average accuracy of 87%.

Singh et al. [12] presented part of speech tagging on Marathi using statistical method. They have implemented and compared unigram, bigram, trigram and basic HMM. They used tagset developed by IIIT Hyderabad, and a test corpus of 25,744 words (1000 sentences) was developed to see the performance of the system. They achieve accuracy 77.38%, 90.30%, 91.46% and 93.82% for unigram, bigram, trigram and HMM respectively.

Mohammed [13] discussed stochastic based Part of speech tagging system for Somali language, which is low-resourced language. He presented the first Part of speech tagging system for Somali using different approaches such as HMM, Conditional Random Fields and Neural Networks. 14,369 words were used to train the system and obtained the average accuracy of 87.51%. There are many more papers related with part of speech tagging based Hidden Markov Model (HMM) for different languages [4, 14–17].

3 Proposed System Description

Data collection, preprocessing, tokenization, creating tagset, building corpus, training and development of stochastic based tagging system are the main methods of this system, and the same is described in detail as follows:

3.1 Data Collection

Unformatted raw texts were collected from different sources and from various topics, majority from daily newspaper ‘Vanglaini’ online. This collection of texts includes different topics such as daily news, health, culture, politics, and sports. In this research work, a corpus consisting of 23,319 words (688 sentences) has been build.

3.2 Preprocessing

Raw texts in Mizo language need proper cleaning and normalizing for further computational processing. Due to lack of stability of writing styles in Mizo language, many differences are found in the raw text. It is due to the fact that Mizo language does not have proper grammatical guidelines for writing words and sentences. In this phase, removal of unnecessary punctuation in the sentences, correction of spelling mistakes, standardization of writing styles have been made.

3.3 Tokenization

Tokenization is the foremost process while dealing and modelling with text data. In general, it is the process of splitting chunk of text into smaller unit. It could be splitting of paragraphs into sentences or breaking up of sentences into words or words into characters. Each of these smaller units are termed as tokens. In this work, collected sentences are tokenized into words, which are separated blank spaces. Symbols and punctuations are also separated out from words thus forming separate tokens.

3.4 Development of Tagset

Tagset is a list of collection of tags or labels, designed to indicate the morphological classes of each word in the sentences. Tagset are usually designed for specific language since the morphological structures of languages are different for different languages. It is essential to design a proper tagset to signpost the grammatical information about each token in the corpus. In this paper, a list of tagset, consisting of 45 tags, has been developed to meet the morpho-syntatic requirements of the Mizo language. The tagset used for this works is shown in Table 1.

3.5 Building the Mizo Corpus

Corpus, in computational linguistics context, is a collection of structured text data. They are usually designed for specific purpose with a specific format. It is essential task to build a large annotated corpus to perform part of speech tagging using training based techniques. They are the main language resources and knowledge beds for language processing to perform statistical analysis.

To the best of our knowledge, there is no proper tagged corpus in Mizo language till today. So, raw digital texts are collected from different sources and from different topics, majority from Vanglaini daily news and articles (online version). This manual

Table 1 Proposed tagset for Mizo language

S. No.	Tag	Description	Examples
1	CMN	Common Noun	*Sava, Thing, Khua*
2	PPN	Proper Noun	*Johny, Aizawl*
3	ABN	Abstract Noun	*Lungaihna, Remna*
4	PSP	Personal Pronoun	*Ka (I), Kan (We), In (You), An (They)*
5	POP	Possessive Pronoun	*Ka (My), Kan (Our), In (Your), An (Their)*
6	RLP	Relative Pronoun	*Kha, Khing*
7	MP	Demonstrative Pronoun	*Hei hi*
8	IP	Interrogative Pronoun	*Tunge, khawi zawk, hei*
9	VB	Verb base form	*Hmu, Hnek, kal*
10	NVB	Nounal Verb	*Aizawl, e.g. Ka va Aizawl ang e*
11	DVB	Double Verb	*Kal kal suh*
12	RB	Adverb base form	*Lutuk*
13	DRB	Double Adverb	*Char char, den den*
14	MRB	Demonstrative Adverb	*Rawn, han, zu*
15	IRB	Interrogative adverb	*Engtinnge*
16	JJ	Adjective base form	*Lian, mawi, sang*
17	MJJ	Demonstrative Adjective	*Heng hi*
18	DJJ	Double Adjective	*Mawi mawi, em em*
19	CJJ	Comparative Adjective	*Sang zawk*
20	SJJ	Superlative Adjective	*Sang ber*
21	IJJ	Interrogative Adjective	*Eng thil nge i siam?*
22	PPT	Postposition	*Atan*
23	CC	Coordinating Conjunction	*Leh, &*
24	UH	Interjection	*Ekhai!, Karei!*
25	PT	Particles	*I kal tur a ni*
26	SYM	Symbol	*[,], @, #, %, **
27	,	Comma	*,*
28	.	Fullstop	*.*
29	:	Colon	*:*
30	;	Semi colon	*;*
31	?	Question mark	*?*
32	QM	Quotation Mark	*“, ”, ’, ’*
33	CD	Cardinal number	*1, 2, 3, pali, panga*
34	NG	Negation	*Lo*
35	ET	Date	*19th March, 2020, 10.11.2020*
36	RBP	Adverb of place	*Aizawl-ah, school-ah*

(continued)

Table 1 (continued)

37	RBT	Adverb of time	*Zana, zinga*
38	SF	SUFFIX	*Ah, te, in*
39	AT	ARTICLE	*Chu, chuan*
40	RBM	Adverb of manner	*Na tuar*
41	FW	Foreign Word	*Conference, field*
42	CRB	Comparative Adverb	*Tisual zawk*
43	SRB	Superlative Adverb	*Hriat tam Ber*
44	SPRB	Specifying Adverb	*Hmanga, kaltlanga*
45	VBN	Verbal Noun	*Thutna, kalna*

Table 2 Summary of the developed corpus

Particulars	Count
Total number of words (including symbols)	23,319
Total no. of sentences	688
No. of unique tags	45
No. of unique vocabulary	4442

task of annotation was done carefully with the 45 tagset to handle different ambiguities exist in Mizo language and a lot of help from linguistics experts was obtained to establish a reliable corpus for Mizo POS tagging. Summary of the developed Mizo corpus during this research work is given in Table 2.

3.6 Development of Stochastic Based Tagging System

Any tagging model that contains frequency or probability in some way can be properly classified as stochastic, i.e., it uses frequency, probability or statistics to assign a tag to the term. The main object of this phase is to train the system and provide the most probable tag to the given word. Two language models such as unigram and bigram Hidden Markov Model based taggers have been developed.

Unigram Tagger: Unigram tagger, also known as 1-gram tagger, is one of the simplest stochastic based tagger. It finds the most likely tag from the corpus. It uses a training corpus to evaluate which tag is most likely for each token. As the name infers, it is a tagger that lone uses a single word as its setting for deciding the POS (Part-of-Speech) tag. A unigram tagger, in simple words, is a tagger with context-based whose context is a single token. In the unigram tagging system, only the likelihood of a word for a given tag is considered, the encompassing meaning of that word is not taken into account. Probability of unigram is given as

$$P(W_i) = C(W_i)/N,$$

where $C(W_i)$ is count of occurrence of W_i in the training dataset and N is the total words count in the training data.

Bigram Hidden Markov Model. Hidden Markov Model (HMM) is one of the popular statistics or probabilistic based model, which does not need much expert knowledge about the morphological structure of a language. Use of HMM for tagging system is a special case for Bayesian inference [3], and a paradigm that is trying to choose the best tag sequence that corresponds to the sequence of words in a corpus. It is a task of finding the sequence of POS tags t_1^n that is most probable tag sequence from a given word sequence w_1^n. So, we have

$$T = \underset{t_1^n}{\operatorname{argmax}}\, P\left(t_1^n \middle| w_1^n\right). \tag{1}$$

Using Bayes' rule, we have an equation for conditional probability

$$P(a|b) = P(b|a)P(a)/P(b). \tag{2}$$

So, by using above Eq. (2),

$$T = \underset{t_1^n}{\operatorname{argmax}}\, P\left(w_1^n \middle| t_1^n\right) * P\left(t_1^n\right) \middle| P\left(w_1^n\right).$$

For each tag sequence, $P(w_1^n)$ remains the same, so it can be neglected.
Therefore,

$$T = \underset{t_1^n}{\operatorname{argmax}}\, P\left(w_1^n \middle| t_1^n\right) * P\left(t_1^n\right). \tag{3}$$

where t_1^n is a tag sequence and w_1^n is a word sequence. The term $(w_1^n|t_1^n)$ is the likelihood of the word string, and $P(t_1^n)$ is the probability of the tag sequence.

HMM is based on the Markov assumption, which says that the likelihood of a tag depends on two assumptions. The first premise is that the probability of a word occurring depends on its part of speech tag,

$$P\left(w_1^n \middle| t_1^n\right) \approx \coprod_{i=1}^{n} P(w_i|t_i).$$

The second premise is that the probability of tag sequence can be calculated as the product of the probabilities of its consequent n-gram. In bigram assumption, it depends only on the previous tag.

$$P\left(t_1^n\right) \approx \coprod_{i=1}^{n} P(t_i|t_{i-1}). \tag{4}$$

Substituting in Eq. 3 by 4, we have

$$T = \underset{t_1^n}{\operatorname{argmax}}\, P\left(w_1^n \middle| t_1^n\right) * P\left(t_1^n\right) = \underset{t_1^n}{\operatorname{argmax}} \approx \coprod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}).$$

$P(w_i|t_i)$ represents the likelihood, and $P(t_i|t_{i-1})$ represents the transition probabilities. For finding the highest probable tag sequence, Viterbi algorithm is used.

4 Experimental Results and Analysis

This section discusses briefly about the experimental works performed based on the manually annotated corpus consisting of 23,319 words and presented the results and its analysis.

4.1 Tagset Distribution in the Corpus

Out of 45 tagsets used for labelling words in the corpus, Fig. 1 presents the tag frequency distribution in the whole corpus with higher than 2% in our complete corpus. It is observed that Verb (VB) has the highest occurrence, followed by adverb (RB), common noun (CMN) and personal pronoun (PSP).

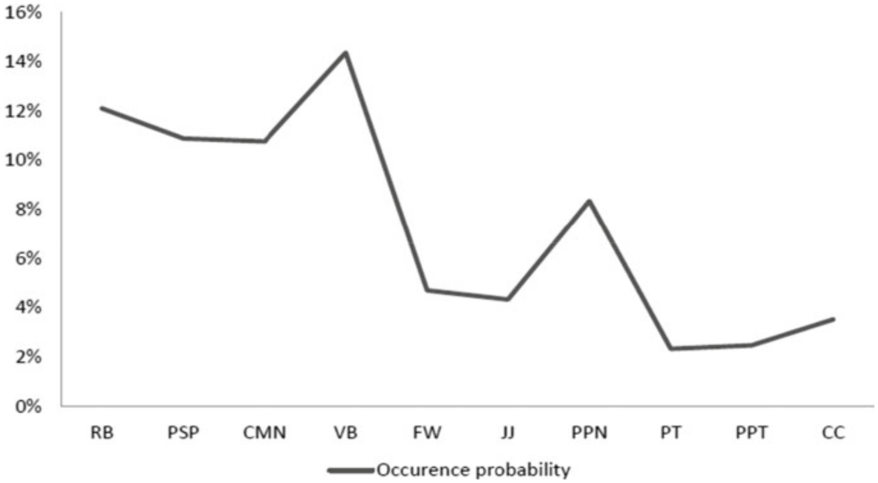
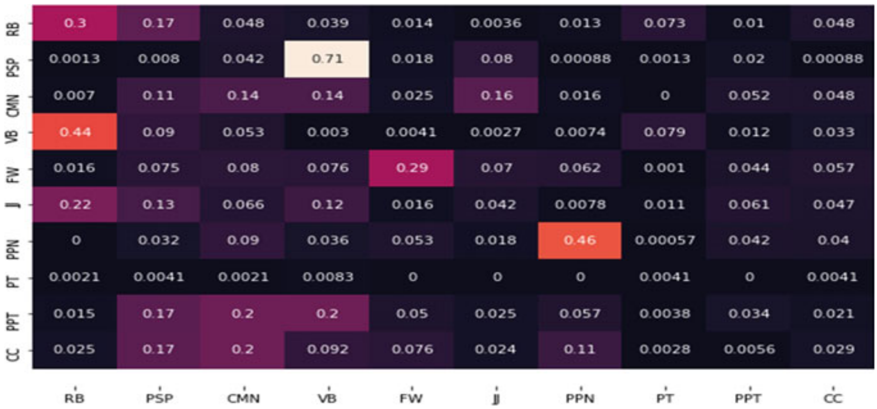


Fig. 1 Tag set occurrence in the corpus



	RB	PSP	CMN	VB	FW	JJ	PPN	PT	PPT	CC
RB	0.3	0.17	0.048	0.039	0.014	0.0036	0.013	0.073	0.01	0.048
PSP	0.0013	0.008	0.042	0.71	0.018	0.08	0.00088	0.0013	0.02	0.00088
CMN	0.007	0.11	0.14	0.14	0.025	0.16	0.016	0	0.052	0.048
VB	0.44	0.09	0.053	0.003	0.0041	0.0027	0.0074	0.079	0.012	0.033
FW	0.016	0.075	0.08	0.076	0.29	0.07	0.062	0.001	0.044	0.057
JJ	0.22	0.13	0.066	0.12	0.016	0.042	0.0078	0.011	0.061	0.047
PPN	0	0.032	0.09	0.036	0.053	0.018	0.46	0.00057	0.042	0.04
PT	0.0021	0.0041	0.0021	0.0083	0	0	0	0.0041	0	0.0041
PPT	0.015	0.17	0.2	0.2	0.05	0.025	0.057	0.0038	0.034	0.021
CC	0.025	0.17	0.2	0.092	0.076	0.024	0.11	0.0028	0.0056	0.029

Fig. 2 Transition probabilities

4.2 Transition Probabilities

The transition probability is one of the very important calculations in Bigram HMM tagger. It is the likelihood of the particular tag sequence, considering two tags, i.e., tag-tag pairs. As per the Bigram HMM tagger assumption, the probability of a tag depends on the preceding tag. Figure 2 depicts the top higher transition probability generated when the splitting ratio is 0.9:0.1 (only some portion is shown here due to space limitation). It can be seen that Personal pronoun (PSP) follows verb has the highest transition probability.

4.3 Accuracy of the Taggers

In order to evaluate the performance of the two taggers, the manually tagged corpus is used and is split into two sets: one is for training dataset and another is for test set. The accuracy of the taggers is evaluated on different size of the training set and test set by splitting the corpus into percentage ratios of 70:30, 75:25, 80:20, 85:15, 90:10 as training set and test set respectively. The accuracy of the taggers is calculated using the following formula:

$$\text{Accuracy} = (\text{No of correct tags/No of words}) \times 100$$

The accuracy of the two taggers on the different sized of the training and data set is show in Table 3.

The performance of the unigram shown in the table above is relatively good but sometimes this approach may give tag sequences that conflicts the grammar rules of a language. The unigram tagger points out the most commonly used tag for a

Table 3 Table captions should be placed above the tables

Train set: Test set	Unigram (%)	Bigram (%)
70:30	68.61	72.39
75:25	69.09	72.52
80:20	69.81	73.98
85:15	70.53	74.30
90:10	70.61	75.19

particular word in the annotated training data and it utilizes this information to label the word in the unannotated text. For example, the training set contains a Mizo word *in* (drink) 20 times tagged as 'verb', and *in* (house) 10 times tagged as 'Noun', then all the words '*in*' in the test set will be tagged as 'verb'.

Unlike the Unigram tagger, the Bigram HMM taggers depends not only on the frequency of a single word, instead, it considers the probability of a tag sequence with the previous tag. The result obtained is presumed to be more reliable than unigram tagger. The accuracy increases as the size of training data increases and in our experiment the maximum accuracy of 75.19% is obtained when the training data is highest. So, it is expected to improve the performance of the taggers with the increase in the size of the corpus.

5 Conclusion and Future Works

It is challenging and exciting to work on the development of language processing tools for under-resource language. In this paper, we have designed a model for stochastics based part of speech tagging system for under-resourced language, in the case of Mizo language. Preprocessing and cleaning of raw texts collected from different domains have been carried out carefully. A reliable Mizo corpus consisting of 21,300 words was created and annotated the corpus manually with the proposed 45 tags. The performances of the developed taggers were evaluated on this corpus. The experiment was repeated and evaluated with different splitting ratios of the corpus. The experiment results show that the average accuracy of the unigram tagger is 70.61% and the accuracy of the bigram HMM tagger is 75.19%.

This research work is the fundamental establishment works of part of speech labelling framework, which is one of the significant modules of each language improvement. There are heaps of spaces for development in computational linguistics field in Mizo language. The major limitation of performance of the taggers is the lack of availability of the resource. Therefore, by increasing the amount of data in the corpus, the accuracy of the tagging system is expected to improve and it will be a good to attempt to standardize the writing styles of Mizo language. It is also possible to use certain rule-based components to detect and correct current model defects. Further, comparative study of the performance of different approaches such as trigram, Support Vector machines (SVM), and Conditional Random Fields (CRF) could be explored in the further studies.

References

1. Ethnologue, languages of the world. https://www.ethnologue.com/ethnoblog/gary-simons/welcome-23rd-edition. Last accessed 2020/11/01
2. Wikipedia. The free encyclopedia. https://en.wikipedia.org/wiki/List_of_endangered_languages_in_India. Last accessed 2020/11/01
3. Jurafsky D (2000) Speech & language processing, 2nd edn. Pearson Education, India
4. Kupiec J (1992) Robust part-of-speech tagging using a hidden Markov model. Comput Speech Lang 6(3):225–242
5. Al Shamsi F, Guessoum A (2006) A hidden Markov model-based POS tagger for Arabic. In: Proceeding of the 8th international conference on the statistical analysis of textual data, France, pp 31–42
6. Tasharofi S, Raja F, Oroumchian F, Rahgozar M (2007) Evaluation of statistical part of speech tagging of Persian text. In: 2007 9th international symposium on signal processing and its applications. IEEE, pp 1–4
7. Anwar W, Wang X, Li L, Wang XL (2007) A statistical based part of speech tagger for Urdu language. In: 2007 international conference on machine learning and cybernetics, vol 6. IEEE, pp 3418–3424
8. Kadim A, Lazrek A (2018) Parallel HMM-based approach for Arabic part of speech tagging. Int Arab J Inf Technol 15(2):341–351
9. Cahyani DE, Vindiyanto MJ (2019) Indonesian part of speech tagging using hidden Markov model–Ngram & Viterbi. In: 2019 4th international conference on information technology, information systems and electrical engineering (ICITISEE). IEEE, pp 353–358
10. Joshi N, Darbari H, Mathur I (2013) HMM based POS tagger for Hindi. In: Proceeding of 2013 international conference on artificial intelligence, soft computing (AISC-2013), pp 341–349
11. Linn ZZ, Patil PB (2019) Part of speech tagging for Kayah language using hidden Markov model. In: 2019 4th international conference on electrical, electronics, communication, computer technologies and optimization techniques (ICEECCOT). IEEE, pp 228–233
12. Singh J, Joshi N, Mathur I (2013) Development of Marathi part of speech tagger using statistical approach. In: Advances in computing, communications and informatics (ICACCI)
13. Mohammed S (2020) Using machine learning to build POS tagger for under-resourced language: the case of Somali. Int J Inf Technol 12:717–729
14. Wikipedia. The free encyclopedia. https://en.wikipedia.org/wiki/Part-of-speech_tagging. Last accessed 2020/11/01
15. Yimin L, Degen H (2005) Chinese part-of-speech tagging based on full second-order Hidden Markov model. Comput Eng 10
16. Daimary SK, Goyal V, Barbora M, Singh U (2018) Development of part of speech tagger for Assamese using HMM. Int J Synth Emotions (IJSE) 9(1):23–32
17. Bandyopadhyay S, Ekbal A (2007) HMM based POS tagger and rule-based Chunker for Bengali. In: Advances in pattern recognition, pp 384–390

An Approach to Analyze Rumor Spreading in Social Networks

Ravi Kishore Devarapalli and Anupam Biswas

Abstract Rumor source identification is an emerging research area in social network analysis, which depends on the process of rumor spreading. To identify rumor source, it is crucial to understand how rumor spreads over the network. In rumor spreading, the position of spreader in the network plays an important role. The position of spreader is generally associated with its connectivity with other users in the network. In this paper, rumor spreading has been analyzed from the perspective of rumors source node position in the network. Experimental results on different networks indicate the dependency of rumor spreading on source node positions in social networks.

Keywords Centrality measures · Information diffusion · Rumors · Rumor source detection · Social networks

1 Introduction

Social networking platforms such as Facebook, Whatsapp, Twitter and YouTube play an important role in disseminating information. Nowadays, these social networking platforms are very popular for sharing information as it reaches people very fast. Due to the large-scale connectivity, cyber-criminals prefer these platforms to circulate misinformation or rumors for creating false narrative among the people and execution of their criminal activities. A rumor is defined as a statement that has an unverified or false value of truth [16], which contains either disinformation or misinformation. The studies show rumors destroy the self-confidence of the individual and often lead to anxiety, depression, suicidal thoughts and other issues as well [17]. Recently, many people spread the COVID-19 information to their friends, family members, and

R. K. Devarapalli (✉) · A. Biswas
Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam, India
e-mail: ravi5129@cse.nits.ac.in

A. Biswas
e-mail: anupam@cse.nits.ac.in

R. Patgiri et al. (eds.), *Edge Analytics*, Lecture Notes in Electrical Engineering 869,
https://doi.org/10.1007/978-981-19-0019-8_54

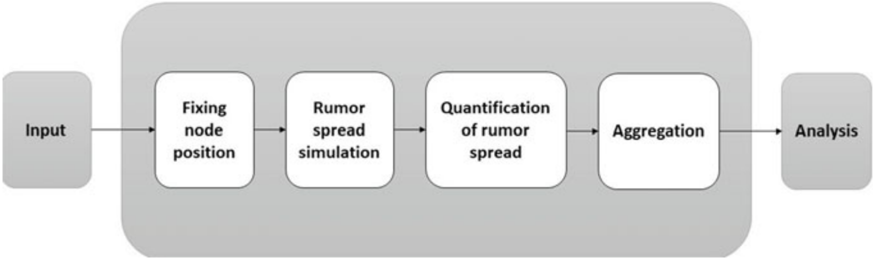


Fig. 1 Analyzing methodology

colleagues, etc., including facts as well as rumors. Therefore, to control the circulation of rumors, there is a need for detection of rumors and their source early [19, 21].

Numerous studies have been carried out for detection of rumors [4, 5, 7, 9, 15] as well as for identification of rumor source [8, 19, 20]. These studies indicate that both detection of rumors and identification of rumor source depend on how the rumors spread in the network. Thus, it is important to analyze the factors affecting rumor spreading in order to detect rumors and their sources. In this paper, a methodology is proposed to analyze rumor spreading in the social network. The methodology is designed to analyze how the position of rumor source node affects the spreading of rumors in the network. To distinguish the position or locality of a node in the network, different centrality measures are considered. Information diffusion model is considered for realistic simulation of rumor spreading in the network. Two metrics are designed to quantify the spreading of rumors in the network. The analysis shows significant dependency of rumor spreading on source node position in the network.

Rest of the paper is organized as follows. Section 2 elaborates the analysis methodology which includes the importance of the source node position, how to change the position of source using the centrality measures and evaluation metrics. Section 3 details the experimental analysis such as experimental setup, dataset details and result analysis of experiments on different networks. Section 4 discusses the observations and insights of the experimental results. Section 5 concludes with key remarks and future perspective of the analysis.

2 Analyzing Methodology

The methodology has several steps as shown in Fig. 1, and they are as follows: first input, second, fixing node position to fix the position of a source node in the network. Third, rumor spread simulation to spread the rumor in the network by any diffusion model. Fourth, quantification of rumor spread to quantify the rumor spread using some metrics. Next, aggregation and analysis explains the procedure and analysis based on the quantification. Each of the steps is explained in the following sections.

2.1 Fixing Node Position

In order to analyze the affects on rumor spread by the position of the source node, first, it is needed to focus on how to fix the position of the source node. For this reason, consideration must be given to centrality measures to fix the position of the source node. Suppose, centrality measure of each node in the network is available to us, that makes it easy to change the position of the source node by changing the centrality measures. Therefore, just update the centrality measure each time to change the position of the source nodes. As a result, the position of the node that has the same centrality measure has been updated automatically. There is a major contrast between the infected graphs obtained from two source nodes from various environments, one is from the highest centrality and the other is from the lowest centrality. Because, in any network, the connectivity of the nodes of each centrality are different. Therefore, we used centrality measures to change or fix the position of the rumor source so as to evaluate the affects on rumor spread by the position of a seed node in the network.

Centrality measures: The centrality measures are to be considered as one of the important factors for fixing the position of the source node. These measures are calculated to assign a score in the range [0, 1] to each node [11]. We have considered three centrality measures as follows:
Betweenness centrality (B_c): B_c is defined as a node, i.e., a bridge between any two other nodes, that has the shortest path between them [3]. This measure assigns scores to each node, such as how much specific node serves as a bridge to all other network nodes. It is noticed that a node with higher betweenness centrality does not have the highest degree which is essential for the spread of information [14]. B_c is calculated using Eq. (1).

$$B_c(v) = \sum_{s \neq v \neq t} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \tag{1}$$

where sigma$_{s,t}(v)$ is the number of the shortest paths from s to t via v. *Closeness centrality* (C_c): C_c is defined in the graph as the closest distance between any node to all other nodes [11]. This measure assigns the score to each node from range [0, 1] as it calculates the shortest path from a certain node to all other nodes in the network. Closeness of a vertex is calculated by the following equation (2).

$$C_c(v) = \frac{1}{\sum_{w \epsilon G} d(v, w)} \tag{2}$$

where $d(v, w)$ is distance between the nodes v and w.

Degree centrality (D_c): This is defined as the total number of edges connected to a particular node in the network [2]. This measure assigns a score (degree) to each node in the network depending on the number of edges attached to it. In the real world, popular personalities like politicians, actors and sports stars have a high degree of centrality for those on the network. Social networks like Facebook and Twitter are confirming this, since more popular celebrities have more friends and followers in their profiles. The following equation (3) is used to determining the degree centrality.

$$D_c(G) = \sum_{v \epsilon G} \frac{|\deg(v_*) - \deg(v)|}{|H|} \tag{3}$$

where $H = (|v| - 1)(|v| - 2)$, and

v_* = vertex with highest degree.

Once each centrality measure assigns its score to each network node, each score must be chosen so that corresponding node with the same score as the source node is used to spread the information across the network.

2.2 Rumor Spread Simulation

In order to analyze the rumor spread, first, we have to simulate the real scenario of the rumor spreading. In our assumption, once node is infected, it will be in the same state and will not be recovered. Several models are available to simulate the rumor spread in networks [12]. However, susceptible infected (SI) model is the best suited model for our assumptions mentioned above, because the state of infected node will not change in this model. For this reason, we considered SI model for simulating the rumor spreading.

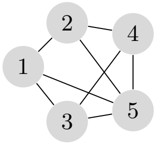
SI model: The SI model was introduced in 1927 by Kermack. In this model, during the diffusion, a node is allowed to change its status only from susceptible (S) to infected (I). This model assumes that if, during the diffusion process, a susceptible node comes into contact with an infected one, it becomes infected with probability β: once a node becomes infected, it stays infected (the only transition allowed is S→ I) [12]. Further, during the simulation, each node can experience the two statuses such as "0" for susceptible node and "1" for infected node [18].

2.3 Quantification of Rumor Spread

After simulation, the next step is quantification of the rumor spreading. In order to quantify the spreading, the consideration must be given to some metrics. In this paper, we considered two metrics to quantify the spreading; they are neighbor overage and whole graph coverage. The quantification of these two metrics are useful to explore how the rumors are spreading after the simulation by any diffusion model, and these two metrics have been explained as follows.

Neighbor Coverage (NC): NC is one of the metrics to be addressed for assessing the effects of change in the rumor source position. NC is a division between sum

Fig. 2 A network example



of the fraction of infected neighbors and the total number of infected nodes after a particular iteration. We compute NC in each iteration by using the equation (4).

$$\mathrm{NC}_i = \sum_{i=1,n}^{k} \frac{(\frac{I_{i,n}}{N_{j,n}})}{T_{i,s}} \tag{4}$$

where,

NC_i = neighbor coverage at ith iteration,

$I_{i,n}$ = infected neighbors of node n at ith iteration,

$N_{j,n}$ = total neighbors of node n at jth iteration, and

$T_{i,s}$ = total number of infected nodes after ith iteration.

Initially, $\mathrm{NC}_i = 0$ because we assume that only the root node is infected, but not its neighbors at the first iteration, i.e., $i = 0$. Whenever i increases, the root node infects its neighbors on the basis of the infection rate of the diffusion model. Infected neighbors often infect their neighbors simultaneously by taking into account the increase in i and the infection rate.

For example, consider a graph $G(V, E)$ with five vertices and eight edges as shown in Fig. 2, to analyze the effects on rumor spreading by position of source node using the metric NC, where V is the set of vertices and E is the set of edges. Suppose rumor spread starts from node 1, then only that node is infected for $i = 0$ as shown in Fig. 3a, so $\mathrm{NC}_0 = 0$, because no neighbors are infected when $i = 0$ even though it has three neighbors such as 2, 3 and 5. Further, source node starts infecting its neighbors based in increment in i. In this example, as shown in Fig. 3b, source node infects two of its neighbors 3 and 5 when $i = 1$. NC at each iteration is calculated as following,

$$\mathrm{NC}_1 = \frac{\frac{I_{1,1}}{N_{1,1}} + \frac{I_{1,3}}{N_{2,2}} + \frac{I_{1,5}}{N_{3,3}}}{T_2, 1} = \frac{\frac{2}{3} + \frac{2}{3} + \frac{2}{4}}{3} = \frac{1.83}{3} = 0.61 \tag{5}$$

This process will continue and source node infects entire network when $i = 3$ as shown in Fig. 3d, and the obtained NC values are, for $i = 2$, $\mathrm{NC}_2 = 0.77$ and for $i = 3$, $\mathrm{NC}_3 = 1$. Rumors spread in a similar way in real-world networks such as Facebook and Twitter; when the number of iterations increased, the coverage of the network also increases as shown in Fig. 3.

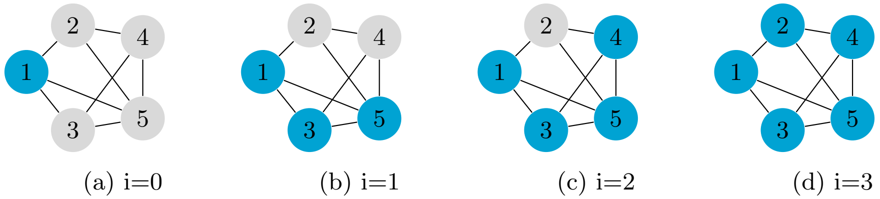


Fig. 3 Iteration wise infection

Whole graph coverage: This section quantifies the effects on rumor spreading by the position of source node. Analysis is based on the metric whole graph coverage. This metric measured in terms of number of iterations required to cover the whole graph from a particular source node. To find the effects by position of source node, it is needed to change the source node position every time and find the required iterations from each position. As explained in Sect. 2.1, just change the centrality measure to change the position of the source node, because if centrality measure changes, then associated nodes which have the same centrality measure selected as source node, which ultimately changes the position of source node.

In order to analyze the effects on rumor spreading using the metric whole graph coverage, it is first important to find the centrality measures for each node in the network. If the centrality measures have been found, sort all the values to find the minimum, min-median, median, max-median and maximum centrality's. Now, assign the related nodes to each of these five centrality's. Many nodes are reported to have the same centrality. Remember, for example, nodes 1, 2, 3 and 4 with the same degree (degree centrality), i.e., 3 as shown in Fig. 1. The associated nodes are greater than or equal to one for each centrality measure, i.e., (C_m : len(associated nodes) $>=$ 1). Now, from each centrality measure, we must consider each associated node as a source node, and spread the rumor until it spreads the whole network. After the dissemination of rumor has been performed by each source node, count the amount of iterations needed to propagate the whole network by each node of the each centrality measure. The number of iterations taken by each node is noted to be different, since each node is selected from different network positions. Furthermore, it is found that the source node selected from the group of nodes with the highest centrality measure takes less iterations for spreading the rumors across the entire network, whereas it requires more iterations from other settings. Thus, the rumor spreading is affected by changes in the position of source node from one environment to another.

2.4 Aggregation and Analysis

In this methodology, first, we have to fix the position of source node, for this find the centrality scores like minimum, min-med, median, max-med and maximum etc. Now, consider each score from these five and select associated nodes belong to those centrality measures, which give the difference between the position of selected nodes. After fixing the position of source node, start the simulation using SI model to spread the rumor over the network. Next, to quantify the rumor spread, we have taken two metrics, namely neighbor coverage and whole graph coverage. Based on the quantification of those two metrics, we analyze the effects on rumor spreading by the position of source node.

3 Experimental Analysis

The following setup is needed in order to examine the effects on rumor spreading via its source node position.

3.1 Experimental Setup

We considered various networks such as real-world networks like football network (small-scale), Facebook network (medium-scale), and synthetic networks such as Erdos-Renyi and Barabasi-Albert etc. for the metrics NC and whole network coverage. A set of Python language packages are used for designing, modifying, researching, to use the network functions and to generate the synthetic networks. Those packages are Networkx, NDlib, Json, pandas, matplotlib and numpy. The Networkx is used to find centrality of each node in the network [6], and NDlib is for using the diffusion models to spread the rumor [12, 18].

After computing scores for each centrality measure, it is important to consider some of those scores from each centrality which represent the whole network, because if all the scores are considered for evaluation, then it is difficult to distinguish the results from those. There are two options available for this, which make it easy to take into account any amount of required scores from each centrality. They are median, which gives the middle element of the set of elements given, and mean, which gives the mean value of the set of elements given. Because multiple scores covering the entire network need to be chosen, we allow the median to pick the various scores. Next, find a minimum, maximum and then median of those two centrality levels. The min-med, which is the median between minimal and median scores, and the max-med, which is the median between median and maximum scores, etc., must also be identified. The entire network is described by these five, such as minimal, min-med, median, max-med and maximum scores for each centrality.

3.2 Dataset Details

The datasets used to evaluate the impact on the propagation of rumor are a small network, a synthetic network and a real-world network as listed in Table 1.

Table 1 Dataset considered for the experiments

Network	Nodes	Edges	Density
Football	115	613	0.0934
Facebook	1024	26,749	0.0511
Barabasi-Albert	20,000	119,965	0.0006
Erdos-Renyi	20,000	400,431	0.0020

- **Small-scale networks**: Consider the famous and real-world "American football" network, which involves "American football games between Division IA colleges during the regular season of Fall 2000." The number of nodes and edges in the football dataset is seen in Table 1.
- **Medium-scale networks**: Many real-world datasets, such as Facebook and Twitter, are publicly accessible on the Stanford Large Network collection [13]. Find the Facebook network because this data collection consists of "circles" from Facebook. In addition , the data collection comprises node functions, circles and ego networks. This has been changed by replacing the internal Facebook ID for each account with a new value [13]. For number of nodes and edges in each network, see Table 1.
- **Synthetic networks**: These networks are scale-free networks and both synthetic networks such as Erdos-Renyi (ER) model [10] and Barabasi-Albert (BA) model [1] are considered. The ER model [6, 10] is used to generate scale-free networks of subjective degree distributions. This model spontaneously creates a network of linked nodes. Each edge is integrated with a probability p graph that is independent of any edge in the network. The BA model [1, 6] is used to build scale-free networks by adopting the power law distribution. This network begins with the underlying linked network nodes, where additional nodes are connected to the network one at a time.

3.3 Result Analysis

It has been found that per each iteration neighbor coverage is maximum whenever the source node belongs to the environment where the centrality measure of the corresponding nodes is maximum. Figure 4, illustrates the similar one, the neighbor coverage of source node that belongs to the environment maximum centrality has maximum coverage in each iteration compared to nodes from other centrality measures. Related findings were found for all centrality measures, such as betweenness, closeness and degree centrality. Another important finding is the rumor spreading from the environment with minimum centrality, which brings negative growth in neighbor coverage after several iterations. As shown in Fig. 4, there is a negative growth in neighbor coverage by source node picked from minimum centrality of

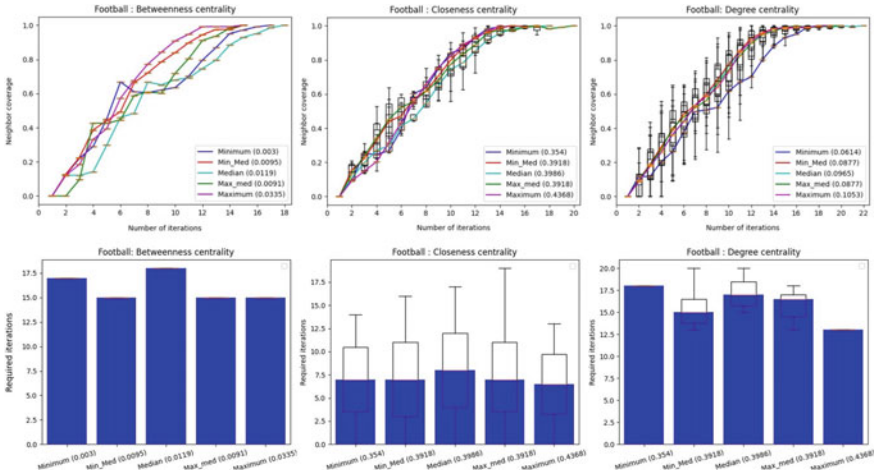


Fig. 4 Effect of source node position on football network

betweenness and closeness. This is because in most of the scenarios, nodes with lowest centrality have less number of neighbors.

The next metric is the whole graph coverage, though you can find the number of iterations used to cover the whole graph. In this case, the source node takes fewer iterations from the corresponding nodes with maximal centrality to infect the whole graph. Otherwise, it takes more iterations to cover the whole graph. With the exception of degree centrality, in Fig. 4, it takes fewer iterations to cover the entire graph while the source node is from the highest centrality.

Experiments for metrics NC and whole graph coverage are implemented on different networks like Facebook, Barabasi-Albert and Erdos-Renyi etc. The metrics NC and whole graph coverage give the similar results as shown in Figs. 5, 6 and 7. If the source node is from the highest centrality, the neighbor coverage is greater, and the number of iterations needed for the whole graph coverage are less.

4 Discussion

In our methodology, we used two metrics, such as NC and whole graph coverage, to analyze the affects on rumor propagation by its source node location. Using Eq. (4) to determine NC to find the coverage of contaminated neighbor nodes at each iteration. It has been found that there is a discrepancy in NC's per iteration that is obtained by source nodes from different locations. That means changes in the source node position often alter the NC per iteration. For this analysis, we used different datasets, suppose source node is from the position where it has the maximum centrality measure; therefore, NC is maximum per iteration, whereas source node from minimum centrality measure NC is less per iteration.

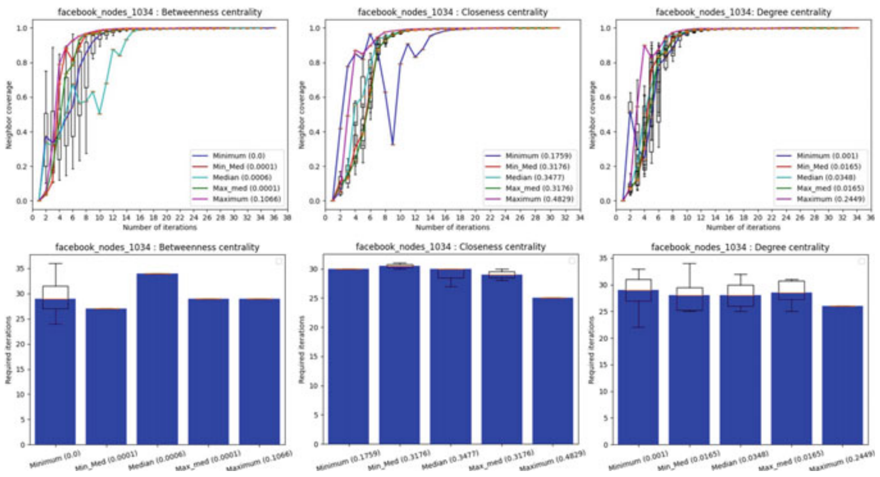


Fig. 5 Effect of source node position on Facebook network

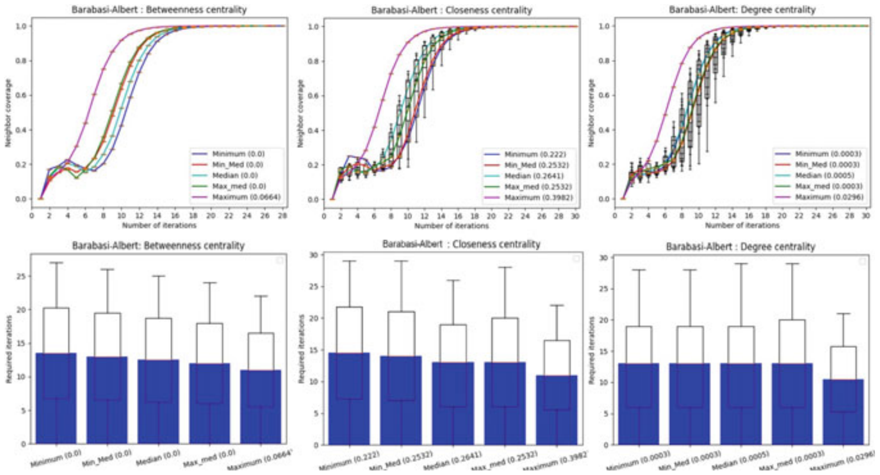


Fig. 6 Effect of source node position on Barabasi-Albert network

The second metric is whole graph coverage; it is used to determine the number of iterations needed to disperse the rumor in the whole graph. We have taken the number of iterations needed to cover the whole graph for this. These iterations are different from different location for each source node chosen. Means, the location of the source node shifts therefore changes the number of iterations needed to cover the whole graph. Suppose the source node position is from the highest centrality, it takes fewer number of iterations to spread the whole network relative to the lowest centrality position of the source node.

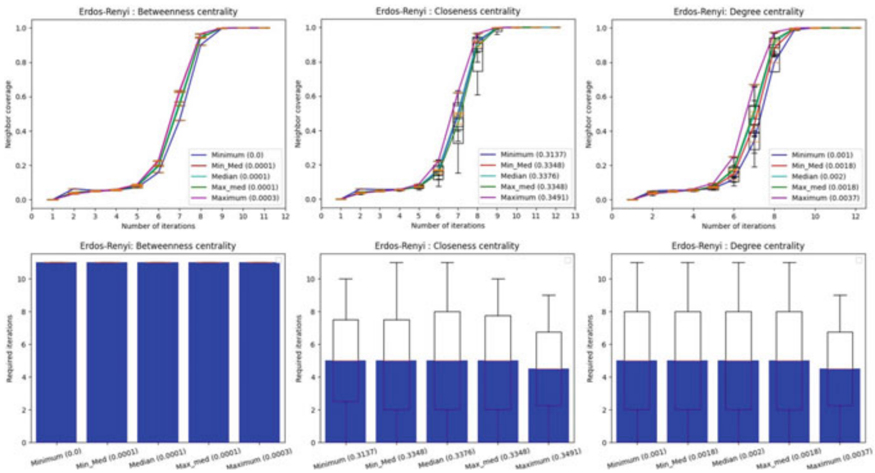


Fig. 7 Effect of source node position on Erdos-Renyi network

5 Conclusion and Future Direction

This paper proposes a methodology to analyze the effects of rumor dissemination by its source location. By using both metrics, NC and whole graph coverage, we measured these impacts. Significantly, the location of the source node influenced the dissemination of rumor. Comparatively, both metrics NC and whole graph coverage showed comparable findings that in any case, source node from the highest centrality environment affects the network more rapidly. Thus, the rumor source location is an important element in the dissemination of rumor. This research plays a crucial role in understanding the affects of the source node position on the network from which environment. Once the source node environment is identified, it is easy to identify the rumor source in the network by applying source node identification techniques.

References

1. Barabási AL, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512
2. Boudin F (2013) A comparison of centrality measures for graph-based keyphrase extraction. In: Proceedings of the sixth international joint conference on natural language processing, pp 834–838
3. Brandes U (2001) A faster algorithm for betweenness centrality. J Math Sociol 25(2):163–177
4. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder-decoder approaches. arXiv preprint arXiv:1409.1259
5. Church K (1996) Conference on empirical methods in natural language processing
6. Developers N (2018) Software for complex networks
7. DiFonzo N, Bordia P (2007) Rumor, gossip and urban legends. Diogenes 54(1):19–35

8. Dong W, Zhang W, Tan CW (2013) Rooting out the rumor culprit from suspects. In: 2013 IEEE international symposium on information theory. IEEE, pp 2671–2675
9. Donovan P (2007) How idle is idle talk? One hundred years of rumor research. Diogenes 54(1):59–82
10. Erdős P, Rényi A (1960) On the evolution of random graphs. Publ Math Inst Hung Acad Sci 5(1):17–60
11. Freeman LC (1978) Centrality in social networks conceptual clarification. Soc Netw 1(3):215–239
12. Kermack WO, McKendrick AG (1927) A contribution to the mathematical theory of epidemics. Proc R Soc Lond Ser A 115(772):700–721
13. Leskovec J, Mcauley JJ (2012) Learning to discover social circles in ego networks. In: Advances in neural information processing systems, pp 539–547
14. Louni A, Subbalakshmi K (2014) A two-stage algorithm to estimate the source of information diffusion in social media networks. In: 2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS). IEEE, pp 329–333
15. Nicholas D, Bordia P, Rosnow R (1994) Reining in rumors. Organ Dyn 23:47–62
16. Qazvinian V, Rosengren E, Radev D, Mei Q (2011) Rumor has it: identifying misinformation in microblogs. In: Proceedings of the 2011 conference on empirical methods in natural language processing, pp 1589–1599
17. Resnick P, Carton S, Park S, Shen Y, Zeffer N (2014) RumorLens: a system for analyzing the impact of rumors and corrections in social media. In: Proceedings of the computational journalism conference, vol 5, p 7
18. Rossetti G, Milli L, Rinzivillo S, Sîrbu A, Pedreschi D, Giannotti F (2018) NDlib: a python library to model and analyze diffusion processes over complex networks. Int J Data Sci Anal 5(1):61–79
19. Shah D, Zaman T (2011) Rumors in a network: who's the culprit? IEEE Trans Inf Theory 57(8):5163–5181
20. Shelke S, Attar V (2019) Source detection of rumor in social network—a review. Online Soc Netw Media 9:30–42
21. Takahashi T, Igata N (2012) Rumor detection on twitter. In: The 6th international conference on soft computing and intelligent systems, and the 13th international symposium on advanced intelligence systems. IEEE, pp 452–457