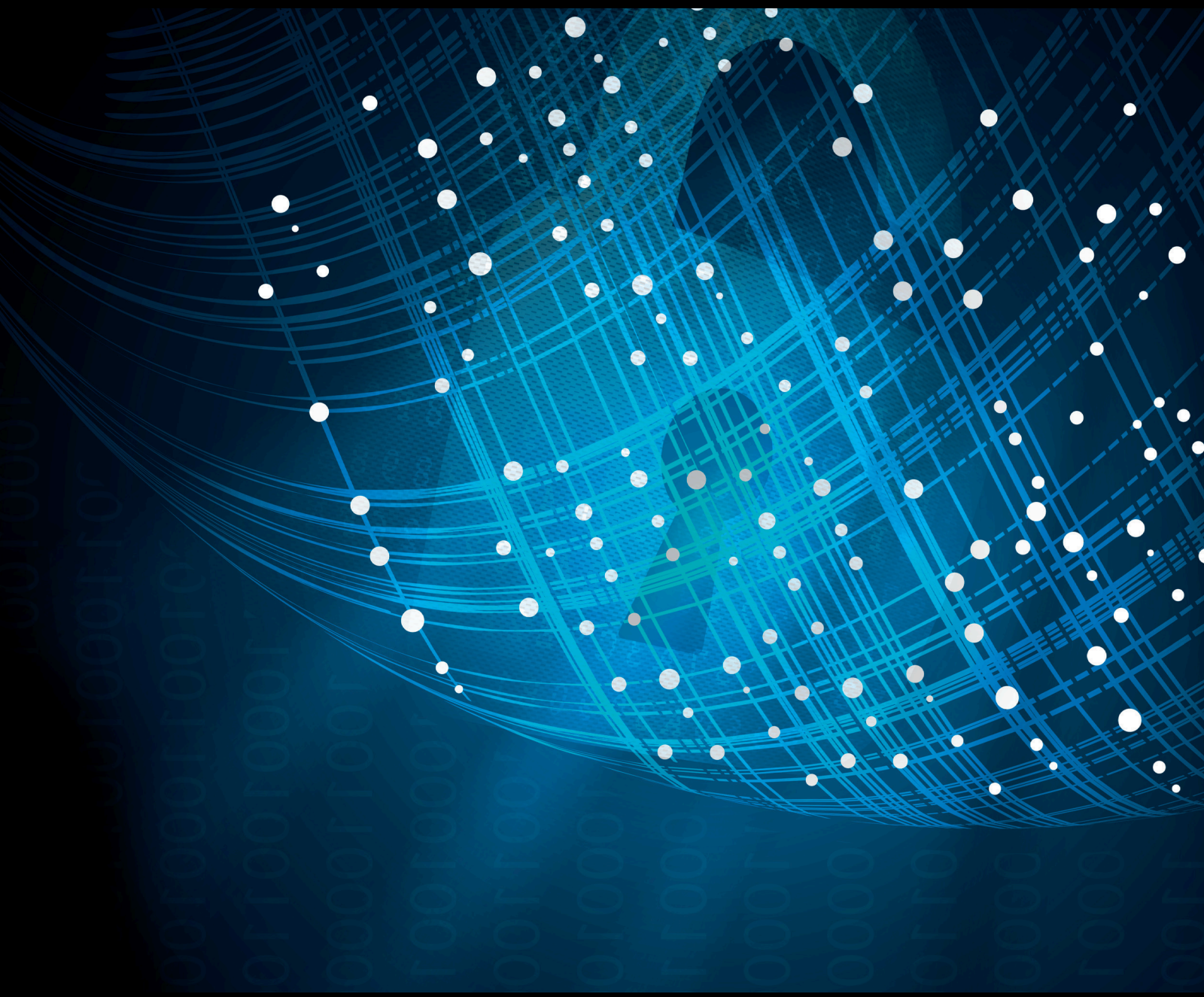
Security and Communication Networks

Machine Learning and Applied Cryptography

Lead Guest Editor: Amir Anees
Guest Editors: Iqtadar Hussain, Umar M. Khokhar, Fawad Ahmed, and Sajjad Shaukat



Machine Learning and Applied Cryptography

Security and Communication Networks

Machine Learning and Applied Cryptography

Lead Guest Editor: Amir Anees
Guest Editors: Iqtadar Hussain, Umar M. Khokhar, Fawad Ahmed, and Sajjad Shaukat

Contents

Contents

Hindawi
Security and Communication Networks
Volume 2022, Article ID 9797604, 3 pages
https://doi.org/10.1155/2022/9797604

WILEY | Hindawi

Editorial

Machine Learning and Applied Cryptography

Amir Anees,[1] **Iqtadar Hussain**,[2] **Umar M. Khokhar**,[3] **Fawad Ahmed**,[4] **and Sajjad Shaukat**[5]

[1]*La Trobe University, Melbourne, Australia*
[2]*Qatar University, Doha, Qatar*
[3]*IT Georgia Gwinnett College, University System of GA, Lawrenceville, GA 30043, USA*
[4]*HITEC University, Taxila, Pakistan*
[5]*King Khalid University, Abha, Saudi Arabia*

Correspondence should be addressed to Amir Anees; amiranees@yahoo.com

Received 30 November 2021; Accepted 30 November 2021; Published 27 January 2022

Machine learning (ML) and cryptography have many things in common, for instance, the amount of data to be handled and large search spaces. The application of ML in cryptography is not new, but with over 3 quintillion bytes of data being generated every day, it is now more relevant to apply ML techniques in cryptography than ever before. ML generally automates analytical model building to continuously learn and adapt to the large amount of data being fed as input. ML techniques can be used to indicate the relationship between the input and output data created by cryptosystems. ML techniques such as boosting and mutual learning can be used to create the private cryptographic key. Methods such as naive Bayesian, support vector machine, and AdaBoost, which come under the category of classification, can be used to classify the encrypted traffic and objects into steganograms used in steganography. Besides the application in cryptography, which is an art of creating secure systems for encrypting/decrypting confidential data, ML techniques can also be applied in cryptanalysis, which is an art of breaking cryptosystems to perform certain side-channel attacks. The aim of this special issue was to create a volume of recent works on advances in different aspects of ML applications in cryptosystems and cryptanalysis. We have selected twenty research articles which deal with different aspects of ML and cryptography.

In the paper entitled "Distributed Outsourced Privacy-Preserving Gradient Descent Methods among Multiple Parties," Z. Tan et al. presented two new outsourced privacy-preserving gradient descent method schemes over horizontally or vertically partitioned data among multiple parties, respectively. Compared to previously proposed solutions, their methods improved in comprehensiveness in a more general scenario.

In the paper entitled "Survey on Reversible Watermarking Techniques of Echocardiography," R. Ghafoor et al. presented a survey on the comparison of state-of-the-art reversible watermarking techniques. The imperceptibility and payload were balanced through a tradeoff. It has been observed in the literature that most of the reversible watermarking methods lack robustness, and very small-scale robustness has been achieved in this domain of watermarking.

In the paper entitled "Machine Learning Technologies for Secure Vehicular Communication in Internet of Vehicles: Recent Advances and Applications," E. Sayed Ali et al. provided theoretical foundations for machine learning and the leading models and algorithms to resolve Internet of Vehicles applications' challenges. This paper has conducted a critical review with analytical modeling for offloading mobile edge-computing decisions based on machine learning and deep reinforcement learning approaches for the Internet of Vehicles.

In the paper entitled "Fusion of Machine Learning and Privacy Preserving for Secure Facial Expression Recognition," A. Ullah et al. presented a novel framework and proposed an effective and robust solution for facial expression recognition under an unconstrained environment; it also helped to classify facial images in the client/server model along with preserving privacy. There are a lot of cryptography techniques available, but they are

computationally expensive; on the contrary, the authors have implemented a lightweight method capable of ensuring secure communication with the help of randomization.

In the paper entitled "Protect Mobile Travelers Information in Sensitive Region Based on Fuzzy Logic in IoT Technology," I. Memon et al. explored the possible flaws associated with security for IoT environment insensitively meant for transfer conditions. They proposed a novel design aimed at detecting a spoofing attack that inspects the probability distributions of received power found for the regions designed for mobile (moving) users.

In the paper entitled "An Improved Method to Evaluate the Synchronization in Neural Key Exchange Protocol," Y. L. Han et al. proposed an improved method for evaluating the synchronization of neural networks timelier and accurately. First, the frequency that the two networks have the same output in previous steps was used for assessing the degree of them roughly. Second, the hash function was utilized to judge whether the two networks have achieved full synchronization precisely when the degree exceeds a given threshold.

In the paper entitled "The Effect of the Primitive Irreducible Polynomial on the Quality of Cryptographic Properties of Block Ciphers," S. S. Jamal et al. introduced 16 affine power affine transformations, and, for fixed parameters, they obtained 16 distinct S-boxes. Here, they thoroughly studied S-boxes with all possible primitive irreducible polynomials and their algebraic properties. All of these boxes were evaluated with the help of nonlinearity test, strict avalanche criterion, bit independent criterion, and linear and differential approximation probability analyses to measure the algebraic and statistical strength of the proposed substitution boxes.

In the paper entitled "Towards an Improved Energy Efficient and End-to-End Secure Protocol for IoT Healthcare Applications," A. Ahmad et al. proposed local coordination Expected Message Authentication Code as an extension of Expected Message Authentication Code. Expected Message Authentication Code is an asynchronous duty cycle medium access control protocol. Expected Message Authentication Code used one important technique of short preamble which is to allow sender nodes to quickly send their actual data when the corresponding receivers wake up.

In the paper entitled "Multicriteria Decision and Machine Learning Algorithms for Component Security Evaluation: Library-Based Overview," J. Zhang et al. developed a new system based on the reusable components as reusability of components is recommended to save time, effort, and resources as such components are already made. Security of components is a significant constituent of the system to maintain the existence of the component as well as the system to function smoothly. Component security can protect a component from illegal access and changing its contents.

In the paper entitled "Evaluating Security of Internet of Medical Things Using the Analytic Network Process Method," X. Huang and Shah Nazir evaluated the security of the Internet of Medical Things by using the analytic network process. The proposed approach was applied using the ISO/IEC 27002 (ISO 27002) standard and some other important features from the literature. The results of the proposed research demonstrated the effective Internet of Medical Things components which can further be used as secure Internet of Medical Things.

In the paper entitled "Secure Framework Enhancing AES Algorithm in Cloud Computing," I. A. Awan et al. presented a framework with key features including enhanced security and owner's data privacy. It modified the 128 AES algorithm to increase the speed of the encryption process, 1000 blocks per second, by the double round key feature.

In the paper entitled "Convolution Neural Network-Based Higher Accurate Intrusion Identification System for the Network Security and Communication," Z. Gu et al. followed a deep learning-based approach for the accurate intrusion detection purposes to ensure the high security of the network. A convolution neural network-based approach was followed for the feature classification and malicious data identification purposes. In the end, comparative results were generated after evaluating the performance of the proposed algorithm to other rival algorithms in the proposed field.

In the paper entitled "Android Malware Detection Based on a Hybrid Deep Learning Model," T. Lu et al. proposed an Android malware detection algorithm based on a hybrid deep learning model which combines the deep belief network and gate recurrent unit. First, they analyzed the Android malware; in addition to extracting static features, dynamic behavioral features with strong antiobfuscation ability were also extracted. Then, they built a hybrid deep learning model for Android malware detection.

In the paper entitled "A Smart Agent Design for Cyber Security Based on Honeypot and Machine Learning," N. E. Kamel et al. presented an introduction of machine learning and honeypot systems, and based on these technologies, they designed a smart agent for cyberattack prevention and prediction.

In the paper entitled "Preprocessing Method for Encrypted Traffic Based on Semisupervised Clustering," R. Zheng et al. analyzed the differences between benign and malicious traffic produced by benign applications and malware, respectively. To fully express these differences, this study proposed a new set of statistical features for training a clustering model. Furthermore, to mine the communication channels generated by benign applications in batches, a semisupervised clustering method was adopted.

In the paper entitled "A Systematic Literature Review on Using Machine Learning Algorithms for Software Requirements Identification on Stack Overflow," A. Ahmad et al. reported a systematic literature review collecting empirical evidence published up to May 2020. This review study found 2,484 published papers related to requirements engineering and Stack Overflow. The data extraction process of the review showed that (1) latent Dirichlet allocation topic modeling is among the widely used machine learning algorithms in the selected studies and (2) precision and recall are amongst the most utilized evaluation methods for measuring the performance of these machine learning algorithms.

In the paper entitled "Deep Learning-Based Cryptanalysis of Lightweight Block Ciphers," J. So proposed a generic cryptanalysis model based on deep learning, where the model tries to find the key of block ciphers from known plaintext-ciphertext pairs. The author showed the feasibility of the deep learning-based cryptanalysis by attacking on lightweight block ciphers such as simplified DES, Simon, and Speck. The results showed that the deep learning-based cryptanalysis can successfully recover the key bits when the key space is restricted to 64 ASCII characters.

In the paper entitled "Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms," L. G. Jun et al. proposed the applications of the machine learning-based spam detection method for accurate detection. In this technique, machine learning classifiers such as logistic regression, *K*-nearest neighbor, and decision tree were used for the classification of ham and spam messages in mobile device communication.

In the paper entitled "Modelling Features-Based Birthmarks for Security of End-to-End Communication System," M. Li et al. proposed a mathematical model, which is based on a differential system, to present feature-based software birthmark. The model presented in this paper provided an exclusive way for the feature-based birthmark of software and then can be used for comparing birthmark and assessing security of end-to-end communication systems. The results of this method showed that the proposed model is efficient in terms of effectiveness and correctness for the feature-based software birthmark comparison and security assessment purposes.

In the paper entitled "Detection and Blocking of Replay, False Command, and False Access Injection Commands in SCADA Systems with Modbus Protocol," L. Rajesh and Penke Satyanarayana worked for False Command Injection attack, False Access Injection attack, and replay attacks on the Modbus protocol. Initially, a real-time SCADA test bed was set up, and they envisaged the impact of these attacks on Modbus protocol data using the test bed. They proposed and developed a method (a) to detect replay attacks by incorporating timestamp and sequence number in Modbus communications and (b) a frame filtering module which will block unauthorized attacks such as False Command Injection and False Access Injection attacks to reach PLC.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this special issue.

Amir Anees
Iqtadar Hussain
Umar M. Khokhar
Fawad Ahmed
Sajjad Shaukat

Hindawi
Security and Communication Networks
Volume 2021, Article ID 8887666, 15 pages
https://doi.org/10.1155/2021/8887666

Research Article

Detection and Blocking of Replay, False Command, and False Access Injection Commands in SCADA Systems with Modbus Protocol

Rajesh L and Penke Satyanarayana

Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, K L Deemed to be University, Vaddeswaram, Guntur 522 502, Andhra Pradesh, India

Correspondence should be addressed to Rajesh L; locharalarajesh@gmail.com

Received 2 September 2020; Revised 21 April 2021; Accepted 9 August 2021; Published 27 September 2021

Academic Editor: Fawad Ahmed

Industrial control systems (ICS) are being used for surveillance and controlling numerous industrial process plants in national critical infrastructures. Supervisory control and data acquisition (SCADA) system is a core component in ICS systems for continuous monitoring and controlling these process plants. Legacy SCADA systems are working in isolated networks and using proprietary communication protocols which made them less exposed to cyber threats. In recent times, these ICS systems have been connected to Internet and corporate networks for data sharing and remote monitoring. They are also using open protocols and operating systems. This leads to vulnerabilities of the system to cyberattacks. Cybersecurity threats are more prevalent than ever in ICS systems. These attacks may be external or internal. Modbus is a widely deployed communication protocol for SCADA communications. There is no security in design of Modbus protocol, and it is vulnerable to numerous cyberattacks. In this paper, we worked for False Command Injection attack, False Access Injection attack, and replay attacks on Modbus protocol. Initially, a real-time SCADA testbed was set up, and we envisaged the impact of these attacks on Modbus protocol data using the testbed. In this work, we used local area network (LAN) environment only for simulating the attacks. We assumed that the attacks penetrated the LAN network. We proposed and developed (a) a method to detect replay attacks by incorporating time stamp and sequence number in Modbus communications and (b) a frame filtering module which will block unauthorized attacks like False Command Injection and False Access Injection attacks to reach programmable logic controller (PLC). Numbers of attacks were simulated and the performance of the method was measured using attack block rate (ABR). It blocked 97% of malicious Modbus transactions or attacks to reach the PLC. It protects SCADA systems from attackers, which is a core component of industrial control systems. The solution enhanced the security of SCADA systems with Modbus protocol.

1. Introduction

Industrial control systems are very crucial for automation of process plants. They are used for surveillance and controlling a plant [1]. These systems are in different forms such as supervisory control and data acquisition (SCADA) systems and distributed control systems (DCS). SCADA system contains data acquisition servers, display-monitoring and controlling clients, historian systems, field instruments, and programmable logic controllers (PLC) [2]. Figure 1 displays a typical SCADA system [3]. It contains different levels of connectivity, systems to differentiate the functionality, and services provided by them. The physical parameters such as pressure, temperature, and pump status will be sensed by suitable sensors and the electrical signal will be passed to PLC through input-output cards. The PLC will collect the data from IO boards, process the data, and send it to human machine interface (HMI) through a suitable communication protocol [4]. In old days, these systems were connected in local area network (LAN). They used mostly proprietary communication protocols. As technology advances, the SCADA data is required for other applications like enterprise resource planning (ERP), energy management system (EMS), and so forth. These systems are connected to Internet

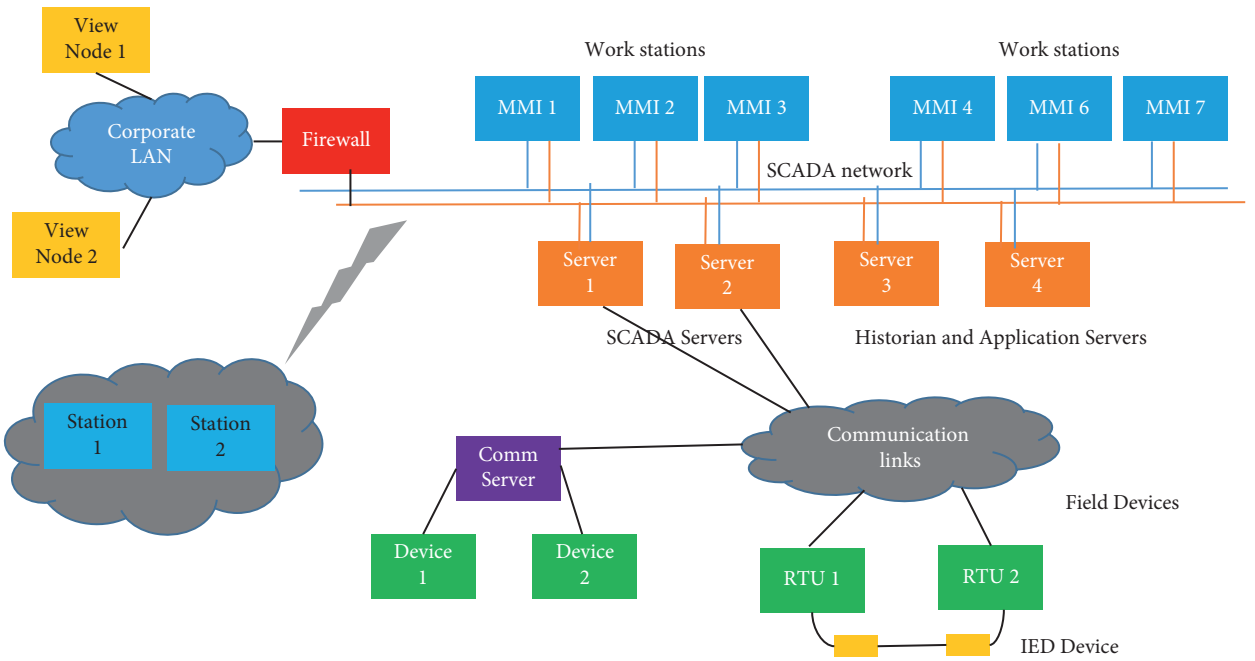


Figure 1: A typical SCADA system architecture [3].

for remote monitoring and data sharing. These systems are also connecting to corporate highly interconnected networks for remote monitoring. The connectivity provides remote surveillance but opens the doors for attackers. Hence, these systems are vulnerable to cyberattacks [5–7]. These attacks may be from internal or external sources.

There were a number of incidents reported all over the world on ICS systems. Some of the examples are Malware Stuxnet attack on SCADA systems in nuclear plants in Iran in 2010, attack on Maroochy Water in 2000, and so forth [8]. The U.S. Department of Homeland Security's (DHS) Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) released a number of attacks reported on ICS system. Every year the attacks are increasing. Even though there are a number of potential areas of vulnerabilities in SCADA components, cyberattacks to communication protocols are a paramount issue [9].

1.1. Modbus Protocol and Vulnerabilities. A SCADA system uses a suitable communication protocol like Modbus, DNP, and so forth for bidirectional data transfer between HMI and PLC [10]. Sensors will sense (measure) the parameters and PLC will collect data from sensors located in the field. This data will be transferred to HMI through a communication protocol [11]. Modbus is most widely deployed communication protocol in control and automation industry. Modbus protocol is an open protocol that was designed in the 1970s [12]. It is a simple request response message protocol at application layer. Modbus is available on two varieties: Modbus Serial and Modbus TCP [13]. The frame format of this protocol is shown in Figure 2. It contains two parts: (i) MBAP Header and (ii) Protocol Data Unit (PDU). The header contains the required details for transmission like

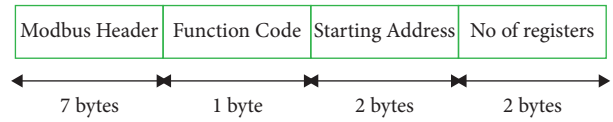


Figure 2: Modbus frame structure [14].

slave ID, protocol ID, and length of the frame to determine the boundaries of the frame. If the physical layer is serial communication, then the nodes participating in the communication are called server and client. They are also called master and slave, if the physical layer is Ethernet (TCP). Modbus protocol contains the following important fields [14]:

(i) Function code

(ii) Starting address of register

(iii) Number of registers

In SCADA communication, data acquisition server or HMI will have Modbus client component (master) and PLC will have its pair component, that is, server component (slave), of Modbus protocol. The master has to send the Modbus request to slave device to poll the data. The slave device will respond to the request with required response and send it to the master. If the request is not correct, then the slave will send exception response to the master. Function code determines the action to be done at Slave. Table 1 gives the details of various function codes and their corresponding actions to be performed. The table contains frequently used function codes in our testing. Starting address and number of registers communicate to the slave how many tags/registers are to be polled by the master [13, 14].

Table 1: Various Modbus function codes and their actions on data.

Function code	Action to be performed
1	Reading of coils
2	Reading of status registers
3	Reading holding registers
4	Reading of input registers
5	Writing to single coil
6	Writing to single holding register
15	Multiple coils: write
16	Multiple holding registers: write

Modbus protocol was designed without considering security aspects because ICS systems were isolated from outside network in earlier days. There was no awareness of cyberattacks when Modbus was designed. Modbus protocol lacks various security features described as follows [15–17]:

(i) Integrity of Modbus frame is not checking by peer devices [18, 19]. The frame can be modified by any attacker and peer devices cannot identify this activity.

(ii) There is no facility for maintaining confidentiality of messages. The Modbus frames will be transferred in plain text and any Man-in-the-Middle attacker can sniff the packet and get the frame information.

(iii) It does not support time stamp for the frames. This is one of the crucial problems because peer devices do not know whether the received response is obtained for the recent query (request) or old query. Any drastic change may happen due to mismatch of real-time field values.

(iv) Modbus is an open protocol and it had simple frame formats. A simple Wireshark tool can be used by attacker to retrieve the information from the network.

As this protocol lacks above security features like integrity, confidentiality, and nonrepudiation, the protocol is more vulnerable to numerous cyberattacks like Man-in-the-Middle attacks in the form of False Command Injection (FCI), False Access Injection (FAI), or False Response Injection; replay attack; and Denial-of-Service (DoS) attacks [20–22]. Out of above attacks, we concentrated on False Command Injection, False Access Injection, and replay attack in this paper. We demonstrated impact of these attacks and proposed and developed methods to block these attacks to reach PLC and HMI. We considered only local area network for simulating the attacks.

1.2. Attacks on Modbus Protocol. Modbus protocol has been suffering from a number of attacks as discussed in Section 1.1. In this research, we worked for False Command Injection, False Access Injection, and replay attacks as follows.

1.2.1. Replay Attacks. The Man-in-the-Middle attacker will store the Modbus messages and will send these messages to target nodes, that is, HMI or PLC, after some time intentionally [23]. As Modbus frame does not have time stamp field, PLC or HMI cannot differentiate whether the response was obtained for recent request frame or old frame. The response in the frame may be old status of field parameters, but HMI will process the frame and erroneous values will be displayed on SCADA mimics. In the same way, the PLC will process the control command and trigger the actuators. This will hamper the operations and may lead to irreversible loss to human life and economy of nation. This attack is very distinctive type that looks like accurate data, but it contains old status [24, 25].

1.2.2. False Command Injection (FCI) Attack. Attacker can send erroneous commands to the PLC like stop the valves while pumping is running, start emergency plant shutdown, and so forth. The attacker can send command to PLC which will disrupt the plant [26]. The aim is execution of arbitrary commands to take control.

1.2.3. False Access Injection (FAI) Attack. The attacker can send false access commands by sending Modbus requests with trial-and-error function codes and starting addresses. It will impose load on the PLC and make it busy. The attacker can send wrong Modbus requests at very high frequency. The PLC will be busy in processing these requests with exception responses. It cannot respond to legitimate HMI commands and requests. This leads to a DoS attack [27].

1.3. Existing Related Works. A number of research scholars worked for security issues of ICS/SCADA systems, communication protocol security, and cyber security of industrial control systems. They also provided some solutions to these problems. The literature survey was conducted in a logical way. Initially, we conducted survey on importance of security of communication protocols in industrial control systems. Next, we described the existing works and shortfalls of those methods to provide security in communication protocols. The next paragraph explains the literature survey on existing works.

Ghosh and Sampalli [28] emphasized the importance of security of SCADA systems. They provided a classification of attacks based on security requirements and network protocol layers. In this paper, they organized security schemes based on current standards, as well as detection and prevention of attacks. They also addressed future challenges in SCADA system security. The authors highlighted that those methods for detection of attacks are more available than methods for prevention of attacks in literature. From this paper, we can understand that it requires to work on prevention of cyberattacks.

Priyanga et al. [29] developed a novel hypergraph-based anomaly detection technique with enhanced principal component analysis and convolution neural network (EPCA-HG-CNN) to detect deviant behaviors of such systems. This method was useful to detect the attacks not for prevention of attacks.

Qian et al. [30] proposed a method based on machine learning techniques where nonparallel hyperplane based fuzzy classifier was used to detect replay and DoS attacks. They informed that hackers even do not need to tamper any data for replay attack; the only thing they need to do is to send the data package eavesdropped from the sensors at another time. The package cannot be detected to be false because there is no time stamp in the protocol used in SCADA systems and the data are in the right form. This method detects only the attacks happening, and it cannot block the attacks. The solution did not provide any remedy for these attacks. Li et al. [31] developed methods that are capable of distinguishing equipment faults from bona fide cyberattacks like replay attack. This method was also used for detection of replay attacks, not for blocking these attacks.

Yusheng et al. [32] explained about various vulnerabilities of Modbus protocol. They proposed real-time deep inspection for Modbus TCP traffic in intrusion detection of industrial control systems based on Modbus protocol. Their method consisted of two modules: rule extraction and deep inspection. The first part separates the Modbus TCP packet into network layer, transport layer, and application layer. The second module generated normal and abnormal rules based on the correlation among those three subparts. The deep inspection module was used for continually correlating the classifications to determine whether the normal or abnormal rule currently applied is a false positive. It is not clear how the rules would be updated over time to detect new attacks or if the rule extraction module could be susceptible to manipulation on the sliding window process due to slow-rate attacks. The paper presented a method to detect the attacks again, not for their prevention.

Fovino et al. [33] developed a secure Modbus protocol based on RSA signature and SHA hash but they did not verify the protocol specific parameters at controller. The frame was transferring in plain text format. They did not address all types of attacks on Modbus protocol. Shahzad et al. [34, 35] developed cryptographic solution using AES, RSA, and SHA for achieving security in Modbus protocol and there was no filtering mechanism. The authors in [36] used AES and hashing for securing IEC 60870-5-104 and Modbus in multicasting polling scenarios. They did not consider integrity of the Modbus frames.

Dudak et al. [37] described enhancement of features of serial Modbus protocol. They developed uBUS protocol by adding some of the protocol specific features. They concentrated on features of Modbus protocol instead of security features and attacks. You and Ge [38] developed Modbus protocol for building security. Erez and Wool [39] developed anomaly detection in Modbus SCADA control registers. The method detects irregular changes in Modbus/TCP SCADA control register values. Hayes and Khatib [40] enhanced security in Modbus by using hash-based message authentication codes and stream transmission control protocol. The authors presented a new method based on stream transmission control protocol (SCTP) and authentication messages based on keyed-hash functions (HMAC) in order to provide a security solution for transactions on Modbus TCP and to establish a mutual authentication mechanism. In this case, all the solutions are software-based and do not include an additional hardware for securing the storage of the secret keys used.

Jung et al. [41] proposed whitelist for SCADA traffic using repetitive communication characteristics of the SCADA system. Kang et al. [42] developed techniques for preparation of whitelist for firewall of SCADA traffic. They provided the concepts of firewall using whitelist generation only. Barbosa et al. [43] proposed approach incorporates a learning phase in which a flow whitelist is learned by capturing network traffic over a period of time and aggregating it into flows. After the learning phase is complete, any nonwhitelisted connection observed generates an alarm. The evaluation of the approach focuses on two important whitelist characteristics: size and stability. The applicability of the approach is demonstrated using real-world traffic traces captured at two water treatment plants and at an electric-gas utility. Serkan et al. [44] studied the FDI attack and simulated the attack in testbed to change the billing information in smart grid application. They explained how this attack can be simulated and proposed methods to detect the attack in the network. They proposed monitoring continuously the ARP and IP changes in the network and alerting the system. This method cannot block or stop to trigger the attack; it will alert the system by critical alarms.

Most of the authors [45–47] worked with testbed using simulation and smart grid domain only. In another section of studies, mathematical modeling [48] and graphical theoretical approach to the network modeling [49] were used for the detection of attacks. In the majority of researches on the security of industrial control systems, no implementation has been done to a real system. Simulation testbed cannot reflect actual real-time system. Zhao and Smidts [50] proposed a two-step hypothesis testing method for detecting and distinguishing replay attacks from other anomalies. The proposed method was demonstrated using a steam generator in a nuclear power plant. This method is suitable for specific domain NPP, not a general model.

From the literature review, it was understood that cyber security of SCADA/ICS systems was a paramount issue in these days. Modbus protocol is vulnerable to cyberattacks; it needs to provide secure data transfer in ICS systems through Modbus protocol. The existing methods were used to detect the attacks not for prevention of these attacks. They did not provide a complete solution to address all the security attacks of Modbus protocol. The FCI, FAI, and replay attacks were less addressed in the literature. The existing methods did not provide end-to-end security in Modbus protocol. In this paper, we analyzed the impact of False Command Injection, False Access Injection, and replay attacks on Modbus protocol using a real-time testbed. After that, we proposed and developed a new method which would provide end-to-end security to detect and block these attacks. We simulated these attacks a number of times and performance was measured using various metrics. In this work, we simulated the attacks in LAN environment. We assumed that the attacks penetrated from outside through Internet connectivity of IT hardware into the LAN (SCADA network). Hence, these attacks were simulated by connecting

the systems in local LAN. Hence, this study is limited to LAN connectivity only. We will work for WAN/Internet in the near future.

The rest of the paper is organized as follows: Section 2 describes the testbed and simulation of attacks on testbed. Section 3 explains the design and development of the method. Section 4 describes the testing and results of the research. The paper is concluded in Section 5.

2. Impact of Cyberattacks on Modbus Protocol

In this section, we described the experimental testbed, simulation of attacks, and impact of these attacks on Modbus communication. We simulated attacks on a real-time testbed for analyzing the impact of these attacks on Modbus.

2.1. Testbed Setup. We set up a real-time testbed in our lab for research. The testbed was shown in Figure 3. It contains one SCADA HMI or DAQ Server with Modbus protocol, one PLC, and one attacker's PC which was used to simulate cyberattacks as *internal* source. These systems are connected in a local LAN switch. We used a Netgear make managed switch for network connectivity. We also connected a web server in the same SCADA LAN for providing web interface of SCADA system. The web server was connected to Internet using a public IP address. A laptop was connected to the Internet and used as *external* attack simulator. The details of the systems are furnished in Table 2. We developed a SCADA application as shown in Figure 4 for testing. The mimic was developed using graphics builder tool of Vijeo Citect SCADA package which was loaded in SCADA HMI. The SCADA package contains Modbus protocol for communicating Schneider make M340 PLC. Initially the PLC was connected directly without gateway PC. The proposed module was running in gateway PC which will protect PLC from the attacks. Detaching the proposed functionality into separate gateway modules will provide an open and standard solution which will be used for existing legacy systems and provides interoperability between multiple manufacturing vendors.

The IP address of PLC can be accessed by (1) accessing host file of SCADA server: IP addresses and domain names are available in host file of SCADA server system. This file lists computer names, IP addresses of SCADA servers, work stations, PLCs, and alias names for routing within the network. (2) Web server was connected to Internet to provide web access and it was connected to SCADA LAN for updating SCADA values. The web server contained public IP address and attacks could access private IP addresses (like PLC, SCADA server) through accessing this public IP address. (3) SCADA systems have been providing remote access through various tools like remote desktop, AnyDesk, and so forth. Attacks can access server, PLC using these tools very easily. Sometimes SCADA systems are distributed geographically over a large space and RTU/PLCs are connecting to SCADA server through GPRS connectivity. Attackers can connect to RTU at site and can destroy the system. Once attacker knows the IP address of PLC by any

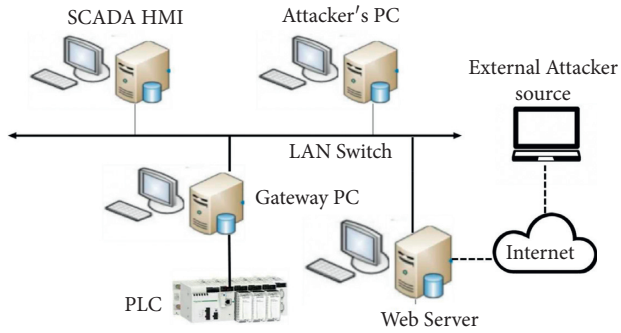


Figure 3: SCADA testbed used for research.

method from the methods mentioned above, the system can be destructed in any manner.

The SCADA application contained two tanks with water. These two tanks were connected using a pump. If the pump starts, then the water will transfer from Tank A to Tank B. We used two Panasonic make level sensors MS-UA11-1 to measure the tank levels. We developed interlock logic in PLC like as follows: Whenever destination tank level reaches 90 cm, the pump will stop automatically by logic. The logic block will continuously monitor destination tank level and it compares it with 90 cm value. If the tank level is more than 90 cm, then it will send pump stop command. The pump will be in state S0 initially. Whenever the destination tank level is more than 90 cm, it will be in state S1. The state diagram was shown in Figure 5. The pump will start automatically once the destination tank reaches zero value. The pump will start whenever the destination tank level reaches 90 cm. But the stop command was sent to pump because of replay attack. The solenoids valves are new ASCO make 24VDC EF8317G35. We used Crompton 0.5HP SP Aquagold 50 Water Pump. These transmitters gave 4 to 20 mA output current and connected to Analog Input and Digital Input modules of PLC. The start and stop commands of motor were controlled by PLC through relay contact which is connected to Digital Output card of PLC.

2.2. Simulation of Attacks. Next, we simulated replay attack, False Command Injection attack, and False Access Injection attack on the testbed from attacker's PC and explained the procedure in subsequent paragraphs.

2.2.1. Replay Attack. The Modbus simulator was loaded in attacker's PC. We used Ettercap tool to scan the target devices and bind the MAC address with target IP address using ARP protocol. Ettercap is a free and open-source network security tool for simulating Man-in-the-Middle attacks in the network. It can be used for computer network protocol analysis and security auditing. Next all messages between PLC and HMI were routed through this attacker's PC. Next, we loaded Wireshark tool to capture the Modbus packet frames. We stored the frame packets and sent them to PLC and HMI after some time.

We observed the impact of this attack on fluid transfer between tanks in the testbed. The graph in Figure 6 shows

Table 2: Details of testbed in lab.

System	Make and model	Specifications	IP address	Other information
SCADA HMI	HP Z420	8 GB RAM with Intel processor	192.168.1.1	Schneider Vijeo Citect SCADA HMI package was loaded in this system
PLC	Schneider make	M340 model	192.168.1.10	PLC logic was written using programming software, Unity Pro
Attacker's PC, web server, and gateway PC	HP Z420	4 GB RAM with Intel processor	192.168.1.3	Modbus simulator was loaded

the normal behavior of pump operations. This graph was a trend plot in SCADA package. Whenever destination tank level reached 90 cm, the HMI sent pump stop command. Figure 7 displays the effect of replay attack. The old values of destination tank level were reported and the pump was stopped intermediately as per interlock logic running in PLC. The destination tank level was reported at HMI as more than 90 cm and the pump start signal was triggered by HMI; but this value was not a real-time current value; it was an old value. But HMI did not differentiate these values and triggered the command. The experiment described the effect of replay attack on Modbus communication.

2.2.2. False Command and Access Injection Attacks. Next, we simulated False Command Injection by sending pump stop command, while fluid was transferring between tanks. Figure 8 shows the graph of the levels of the tanks. When the data values are correct, the level of Tank A will be decreasing and Tank B level will be increasing. The trend of tank level looks like a ramp signal as shown in this figure. Next, we simulated False Command Injection to stop the pump, while fluid was transferring. Then the ramp trend was stopped suddenly as shown in Figure 9 at 11 : 06 : 28. It displayed the impact of FCI attack. In this experiment, the pump was stopped unexpectedly because of simulation of False Command Injection.

From the above testing, it is concluded that Modbus is vulnerable to False Command and Access Injection attacks and replay attacks and it needs to provide a solution for detection of and blocking these attacks. Section 3 explains the proposed and developed solution for this problem.

3. Design of Methods

In this section, we describe our proposed and developed methods for prevention of False Command Injection and False Access Injection attacks and replay attacks. Initially the mathematical modeling of the system was presented. Next, the methodology was proposed and developed.

3.1. Mathematical Modeling

3.1.1. Replay Attacks. In this section, we explain our developed solution to mitigate and detect replay attacks in Modbus protocol. Initially we would like to describe the replay attack in mathematical form. After that, we provide the actual solution.

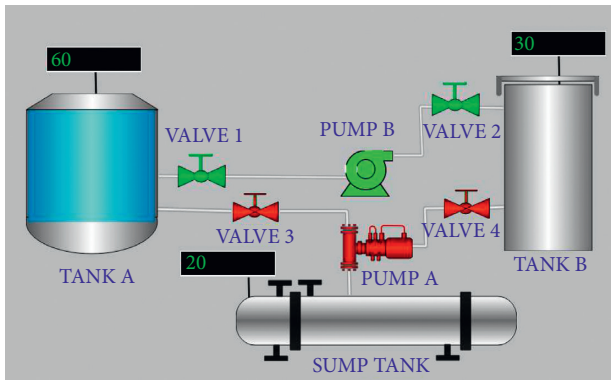


Figure 4: SCADA mimic developed for testing.

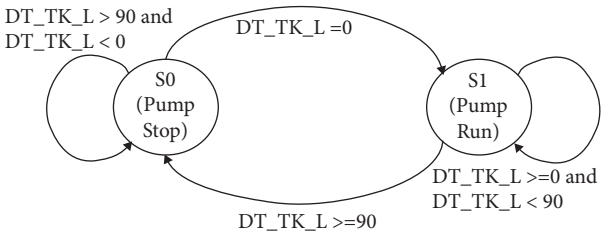


Figure 5: State transition diagram for pump states.

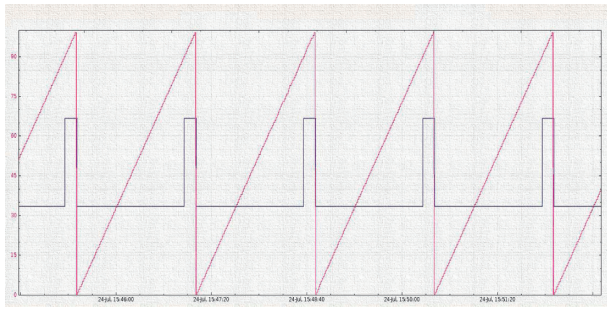


Figure 6: Normal operation of SCADA application.

Let the Modbus request be M_{REQ} containing PLC slave ID, function code, start address, and number of registers to scan.

$$M_{REQ} = \{\text{PLC slave ID, function code, start address, number of reg}\}. \quad (1)$$

This request will be initiated by Modbus client module which will run in SCADA server. SCADA server will poll

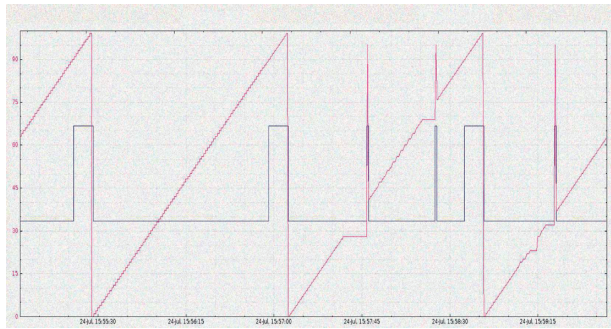


Figure 7: Replay attack on SCADA operation.

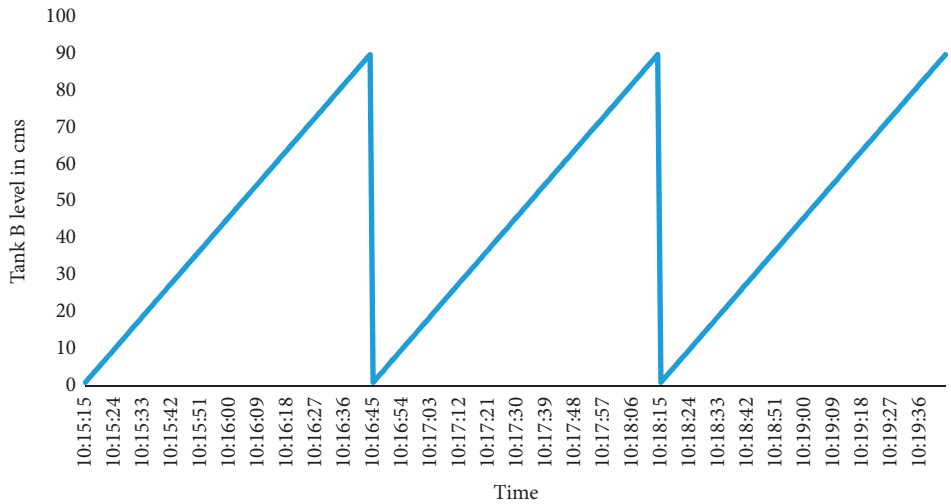


Figure 8: Trend on SCADA of Tank B level during normal fluid transfer.

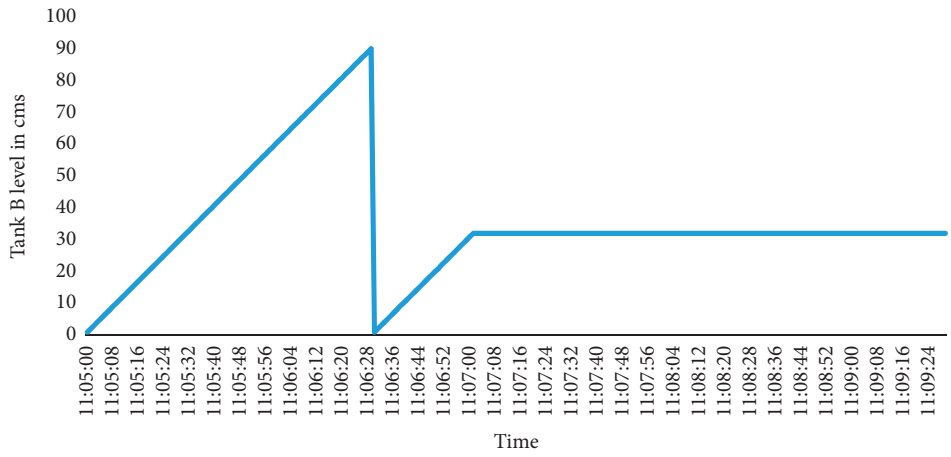


Figure 9: Impact of FCI attack on level of Tank B.

RTU/PLC using Modbus protocol by sending the above-mentioned request to it.

Let the Modbus response to this request be M_{RES} containing the following frame structure as shown in Figure 2:

$$M_{RES} = \{\text{PLC slave ID, function code, frame length, data value}\}. \quad (2)$$

The PLC/RTU will frame the response to the above request and send it to DAQ Server. The DAQ Server will send Modbus requests to PLC like M^1_{REQ}, M^2_{REQ}, and so forth.

$$M_f = \left\{M^i_{\text{Req}},\ i = 0, 1, 2, 3, 4 \ldots\right\}. \quad (3)$$

The PLC will respond to the requests with responses like M^1_{RES}, M^2_{RES}, and so forth.

$$M_r = \left\{M^i_{\text{Res}},\ i = 0, 1, 2, 3, 4 \ldots\right\}. \tag{4}$$

The response will be correct if response for i^{th} query or request is matched with i^{th} response. The response is invalid, if response for i^{th} query is not matched with j^{th} response.

$$M^i_{\text{Req}} \longrightarrow^{\text{yields}} M^i_{\text{Res}}, \quad \text{response is correct,} \tag{5}$$

$$M^i_{\text{Res}} \longrightarrow^{\text{yields}} M^j_{\text{Res}}, \quad \text{response is not correct.} \tag{6}$$

Replay attack detects when (6) occurs at peer device. The sequence number in Modbus request is not the same as sequence number in Modbus response.

To detect the replay attack in Modbus protocol, we included two parameters in the Modbus frame structure:

(a) Sequence number

(b) Time stamp of the frame

We included a sequence number for each and every Modbus frame. The sequence number SEQ_NO_TX of request frame will be copied in response frame by PLC. Modbus client module at HMI will check the sequence number in response frame. If SEQ_NO_RX is matching with one of the SEQ_NO_TX, then it will accept the frame and send it to next step; otherwise, it will reject. Next, it will check the time stamp of response frame. The time difference between request frame and response frame will be calculated and if the difference is more than threshold value it will reject the frame because the response is a delayed frame.

Each and every frame will have a sequence number in the request frame. The PLC will copy this number in response frame. The sequence number is a numerical number (integer) which ranges from 0 to 255 and again it will start from 0. The frame will be changed as follows:

$$\begin{aligned} \text{M}_{\text{REQ}} &= \{\text{Seq no, PLC slave ID, function code, start address, number of reg}\}, \\ \text{M}_{\text{RES}} &= \{\text{Seq no, PLC slave ID, function code, frame length, data value}\}. \end{aligned} \tag{7}$$

The HMI will check the sequence number of response frame. If it matches with the sequence number of request frame, which was sent recently, then it will accept the frame; otherwise, it will reject it.

$$\begin{aligned} &\text{Accept frame if SEQ\_NO\_TX}_{\text{req}} = \text{SEQ\_NO\_RX}_{\text{res}}, \\ &\text{reject frame if SEQ\_NO\_TX}_{\text{req}} \neq \text{SEQ\_NO\_RX}_{\text{res}}. \end{aligned} \tag{8}$$

This method may provide some level of checking but this is not suitable for the following cases:

(a) When multiple requests are sent to PLC without waiting for responses

(b) When the sequence number in response frame matches with request frame but it is after "n" iterations of 255 values

We attended to this problem by implementing time stamp in the frame. Each and every frame will be time-stamped by PLC and HMI. As we discussed above, Modbus did not support time stamp of frame. We add an 8-byte time stamp field to Modbus frame as shown in Figure 10. This time stamp field will be used for checking the freshness of Modbus frame. Generally, Modbus request and responses are periodic in nature. Once the plant was commissioned, the data blocks with tags were fixed. The HMI will poll PLC with fixed configuration for every periodic scan time. HMI will send Modbus requests to PLC with periodic time intervals. If replay attack was launched, then time stamp of the response frame would be very much lag of present time. The module will detect the time gap, reject the frames, and prevent the replay attacks. The threshold for checking the delay of time stamps can be configured by user. The module will automatically adjust this threshold value based on traffic conditions.

In our method, the time delay was calculated between present current time and time stamp of the frame. If the time delay or duration was more than threshold value, the frame was declared as old frame and rejected. The threshold value depends on scanning or polling interval and number of data blocks. If data blocks are high, the PLC will take more time for execution of request. The polling interval may be fixed in some projects, but it may vary and depends on PLC execution time for each query. For example, in our project, HMI polls PLC for every 100 ms for each data block. The data block contains starting address, number of registers, function code, and slave ID. The project contains 4 data blocks for scanning. Then HMI started first data block at 0 ms and PLC took 4 ms for execution of request. HMI received response frame at 64 ms with 60 ms of transmission delay. Next, PLC may wait for 40 ms and initiate next cycle of scanning or may start polling of next data block immediately to achieve fast data acquisition. In the second case, the time duration for one cycle of data blocks is 60 ms in our project. But this time duration is not fixed and it may vary depending on PLC execution time for every request and transmission delay which depends again on data traffic in the network. Hence, it is required to select optimum threshold value for detecting replay attack.

3.1.2. False Command Injection and False Access Injection Attacks. The block diagram for mathematical modeling of the module is shown in Figure 11. Let the input to the frame filtering module be Modbus Frame F_i. It contains Modbus function code (FC), start address (SA), and number of registers (NR) with MBAP Header. It can be expressed as follows:

$$F_i = \{\text{MBAP Header}||FC||SA|NR\}. \tag{9}$$

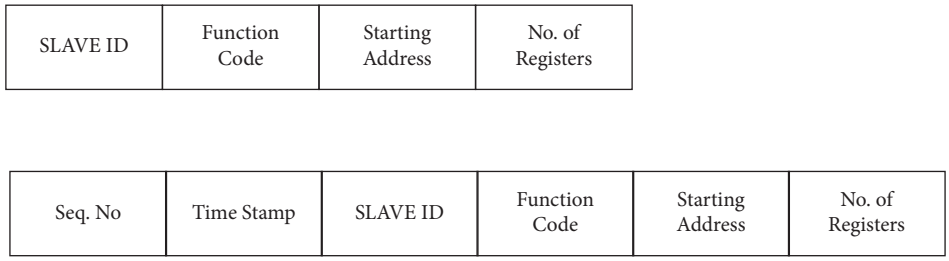


Figure 10: Modbus frame structure, actual and modified.

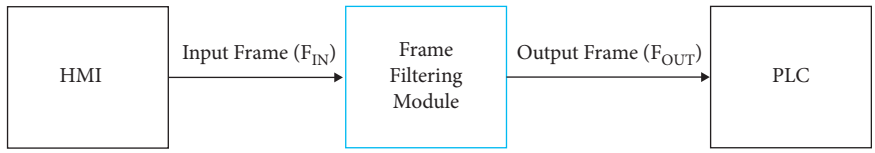


Figure 11: Mathematical modeling of frame filtering module.

The total input to frame filtering module is F_{IN} which is sequence of Modbus frames F_i. The number of frames depends on database in HMI and PLC. Let the number of requests be N. F_{IN} can be expressed as follows:

$$F_{IN} = \{F_1, F_2, F_3, \ldots F_N\}. \tag{10}$$

The module will pass the frame, F_i , if it satisfies the configuration of the module; otherwise, it will block the frame. The output of the frame filtering module is F_o.

$$\begin{aligned} F_o &= F_i \quad \text{if } F_i \text{ matches the configuration,} \\ F_o &= 0 \quad \text{if } F_i \text{ does not match the configuration.} \end{aligned} \tag{11}$$

The total output of frame filtering module is F_{OUT} consisting of F_i or no frame. It can be expressed as follows:

$$F_{OUT} = \{F_1, 0, 0, F_4 \ldots F_N\}. \tag{12}$$

The performance of the frame filtering module can be evaluated by measuring frame rejection rate or attack block rate, which is defined as ratio of number of attacks blocked to number of attacks generated as shown in the following equation:

$$\text{Attack Block Rate (ABR)} = \frac{\text{no.of attacks blocked}}{\text{no.of attacks generated}}. \tag{13}$$

Let the generated attack frames at input of the module be F_g; let the number of blocked attack frames be F_b; then attack block rate can be defined as

$$\%\text{ABR} = \frac{F_b}{F_g} * 100. \tag{14}$$

In this work, we designed a frame filtering module which will send only authorized commands and Modbus requests to PLC. This module will be loaded in a gateway PC. It is basically a computer system. Instead of connecting to network, PLC will be connected to this PC. The communication module of PLC had one RJ45 port for Modbus interface. The port was connected to gateway PC LAN port directly with LAN cable. The destination IP address of Modbus driver at SCADA package was assigned with gateway PC. Hence, instead of connecting to PLC directly, DAQ Server/HMI will connect to this frame filtering module in gateway PC. Generally, every PLC will perform configuration with memory addresses. The memory contains the starting address and number of registers. For example, analog values will use holding registers. The memory range of holding registers contain from 4×00001 to 4×05000. The holding register address shall be between these limits; otherwise, PLC returns exception error.

The flow chart of this module is shown in Figure 12. This module has to be configured using a configuration file. In the configuration of the device, the user has to provide the IP address of the Master device, port number of the connection, allowed function codes, and allowed memory addresses (register addresses). The module will be in listen mode to accept the connection from Master device (connection from DAQ Server/HMI). It will accept the connection if the IP address is an authorized one. It will check the IP address in accept socket call with configured addresses. If the IP address is not matching with any configured IP address, then the connection will be dropped or rejected and the socket will be closed. Next it will check whether the function code is allowed or not. If it is allowed, then it will check whether the starting address and number of registers are within the limits of configured memory or not. If all the parameters are satisfied, then it will pass the request to PLC; otherwise, the request or command will be dropped. In this way, it will protect the PLC from unauthorized or nonconfigured attacks. Modbus has user configurable and future use function codes. Attacker can send malicious commands using these function codes; but the frame filtering module will not accept those commands and block them. Even the Modbus requests with allowed parameters will be accepted and all other requests or commands will be rejected by this module.

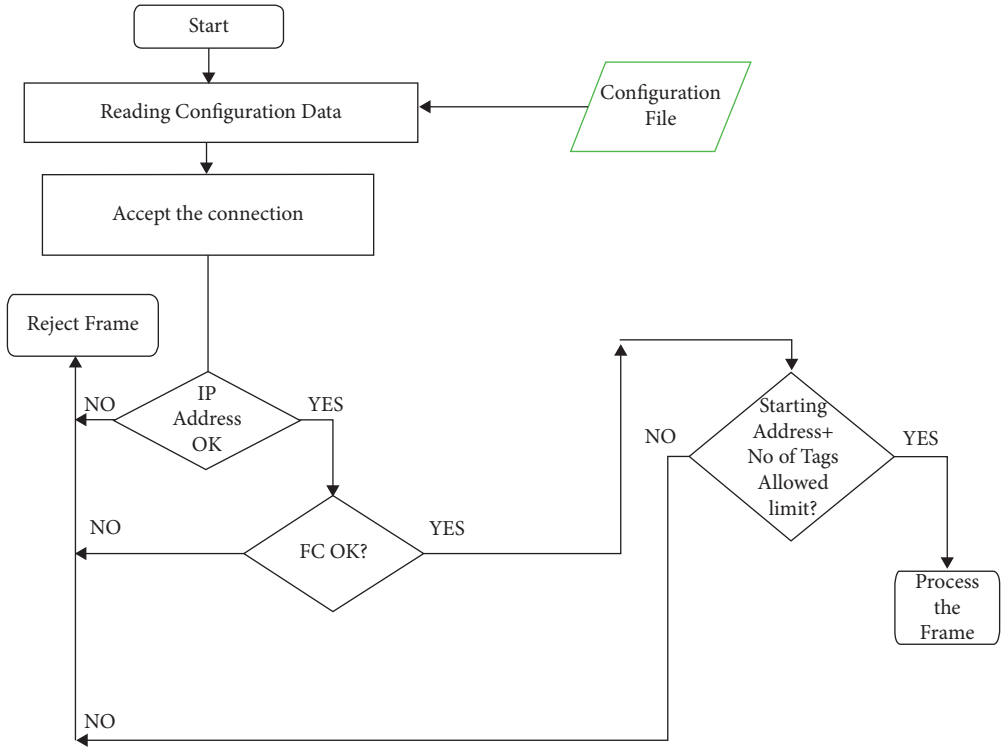


Figure 12: Flow chart of the frame filtering module.

3.2. Open Solution. Even though the above solution provides remedy for the attacks, it changed the frame format of Modbus protocol. Hence, it is not compatible with other manufacturing vendors. It does not provide interoperability. We proposed a solution for this problem. Instead of directly modifying the Modbus frame by DAQ Server or PLC, two gateway modules were placed between DAQ Server and PLC as shown in Figure 13. These gateway modules will modify the frame formats. The gateway at DAQ Server (called gateway1) will add the two new fields to the actual, original Modbus frame (F) initiated by DAQ Server. The modified frame ($F\sim$) will be sent to gateway2 module. This module will remove these fields and send the actual Modbus request (F) to PLC. Then the PLC will frame the response to the request and send it to gateway2 module. Again, gateway2 module will add these new two fields and send them to gateway1. This gateway1 module will remove the fields and send actual response to DAQ Server/HMI. The checking of sequence number and time stamp will be carried out at gateway1 module and gateway2 module instead of checking at DAQ server and HMI. In this way, the gateway modules can be installed in separate PCs and can be easily interfaced with existing legacy systems and any Modbus compliant PLC or device. The gateway2 PC in testbed can be used for this purpose and another PC is required for gateway PC1.

4. Results, Discussion, and Feature Scope

Initially we run the system with modified Modbus without any attack and the system was running normally without any break for 5 hours. Next, we loaded Ettercap tool, Modbus simulator in attacker's PC. We used Ettercap tool for simulating Man-in-the-Middle attack. We sniffed the Modbus packet using Wireshark tool. Using Ettercap tool, we scanned the list of IP addresses in the network. We got IP addresses and MAC addresses of all systems in the network. The attacker's PC used the MAC address from previous scan with ARP spoofing, which resulted in linking of attacker's PC MAC with IP address of HMI/Server. An Ettercap filter can be used to modify Modbus TCP frame originating from HMI and destination to PLC to simulated MITM attack. In our testing, we stored the Modbus requests which are generated by HMI and sent to PLC after various time intervals like 10 ms, 50 ms, 100 ms, 200 ms, 500 ms, 1 sec, 5 sec, 10 sec, 30 sec, 1 min, 30 min, 1 hr, 3 hrs, and 5 hrs as shown in Figure 14. We simulated 100 requests for each time interval. The threshold value was set at 500 ms. The requests received after 500 ms were rejected or dropped by the module. The HMI accepted the Modbus requests with time difference between Modbus frame time stamp and current time being less than 500 ms, and the sequence number matched. From the graph, it was observed that the number of attacks passing through the module is high near to threshold value and the number of attacks was less near to origin because the chance of getting same sequence number at 10 ms is less. The sequence number is a cyclic integer value; hence, it will take $n * t$ ms time to reset it to zero or restart the sequence again. Here "n" is number of Modbus requests and "t" is inter-Modbus request time. The chance of getting same sequence number with less time interval is less; hence, the number of attacks passing through the module is also less.

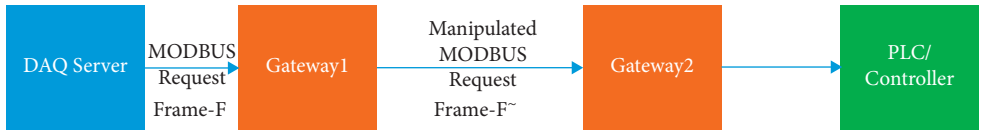


Figure 13: Open solution using gateway modules.

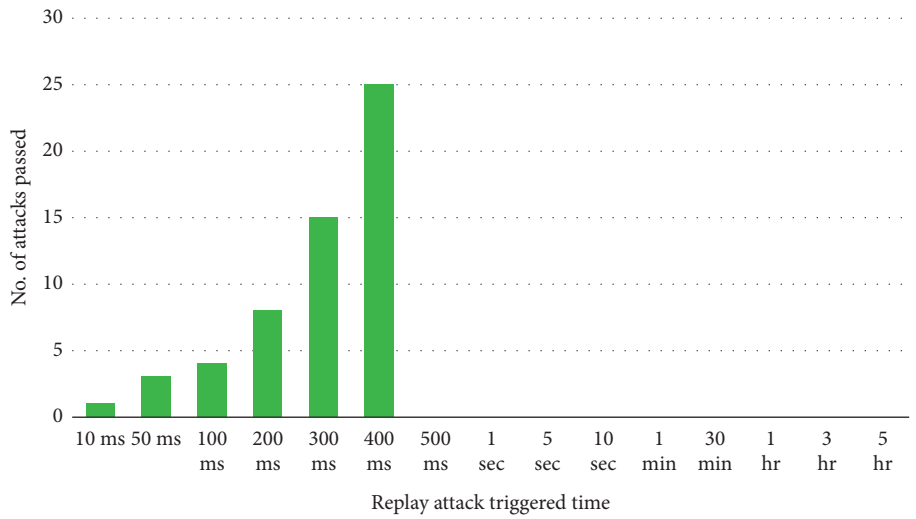


Figure 14: Graph displaying number of replay attacks passing through the module with 500 ms threshold value.

Table 3: Configuration of frame filtering module.

IP address	Function code	Starting address	Number of registers
192.168.1.1	1	0×10000	100
	2	0×10000	250
	4	0×15000	300
192.168.1.2	1	0×10000	200
	4	0×25000	50
192.168.1.3	1	0×02001	4000
	3	0×40000	600
192.168.1.4	2	0×06000	200
	3	0×70000	1250

We configured the frame filtering module with the following configuration as shown in Table 3. It allows Modbus master connections from 4 IP addresses. Each Master will be allowed only some of the function codes and memory addresses. After that, we loaded Modbus simulator in attacker's PC. The simulator can be used for simulating False Command Injection and False Response Injection attacks. We simulated false commands (False Command Injection) using the simulator with function code (FC) = 5, starting address (SA) = 0×0001, and value = 1 (ON). The simulator used IP address of gateway PC as destination IP address. The connection was received at gateway PC. The frame filtering module (FFM) rejected the connection because the IP address (192.168.1.6) was not configured in the module. Next, we simulated the False Command Injection attacks by binding the HMI IP address which is configured in module. We used Ettercap tool to bind the IP address to target device MAC using ARP protocol. The connection was accepted by frame filtering module because IP address of the connection was impersonated by HMI IP address (192.168.1.3). But the frame was rejected as shown in Table 4 because it was not matching the memory configuration of frame filtering module.

Hence, the frame filtering module successfully blocked 97% of attacks. The results can be described as True Negative Rate of 3% and True Positive Rate of 97%. The results are shown in Figure 14 for three iterations. The frame filtering module successfully blocked False Command Injection and False Access Injection attacks to reach the PLC. The HMI and PLC were protected from cyberattacks; hence, the industrial control system was protected from attackers. In this paper, we worked for three attacks on SCADA systems. In the future, we will work for other cyberattacks.

Next, we simulated number of attacks with numerous combinations of function codes and memory addresses. The wrong commands were rejected because of mismatch of configuration. Table 4 displays example of the commands simulated and output of frame filtering module. The table

Table 4: Output of frame filtering module (A: accept, R: reject).

IP address (A/R)	Function code (A/R)	Starting address of registers (A/R)	Number of registers (A/R)	A/R	Reason for rejecting the frame
192.168.1.1 (A)	2 (A)	0×10001 (A)	10 (A)	A	Frame is accepted
192.168.1.2 (A)	5 (A)	0×10005 (A)	1 (A)	A	Frame is accepted
192.168.1.3 (A)	3 (A)	0×40200 (A)	500 (R)	R	Starting address + number of registers crosses the memory access limit
192.168.1.4 (A)	4 (R)			R	Function code 4 is not allowed for this master
192.168.1.6 (R)				R	IP address of master device not matching the list of configured addresses
192.168.1.1 (A)	4 (A)	0×30526(R)		R	The starting address is out of memory access
192.168.1.3 (A)	5 (A)	0×00001	1	R	The memory address is not configured
192.168.1.1 (A)	2(A)	0×10010(A)	25(A)	A	Frame is accepted

displays the reason for rejecting the frame. The Modbus request from IP address 192.168.1.2 was accepted, and the function code and number of registers are within the limit as per configuration; hence, the request was accepted. But the Modbus request from system with IP address 192.168.1.4 was rejected because of configuration mismatch. For some Modbus requests, the IP address and FC were allowed, but the memory access area was out of configured memory; hence, the frame was rejected. Next, we simulated 100, 150, and 200 False Command Injection and False Access Injection attacks in three iterations of each command or request for 10 ms and the frame filtering module blocked 98, 148, and 197 commands, respectively, as shown in Figure 15. It only allowed 2, 2, and 3 commands. The same experiment was conducted 100 times. The mean value of allowed commands in total experiment was 3 commands. We calculated Attack Block Rate (ABR) as given in equation (14). We acquired ABR value of 97%.

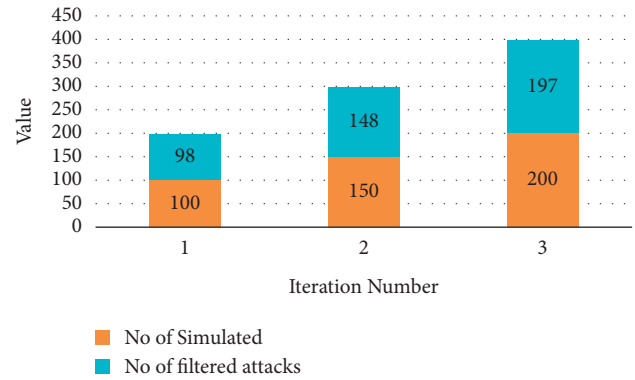


Figure 15: Performance of frame filtering module.

4.1. Overhead of the Solution. This solution provides security in Modbus protocol for secure data transmission between DAQ Server/HMI and PLC, but it adds overhead on Modbus frame length and total time duration. The overhead on Modbus frame length and total time duration are explained in the following sections.

4.1.1. Frame length. We included two fields to the original Modbus frame. The size of sequence number is 2 bytes; time stamp is 8 bytes. Total extra size of frame is 10 bytes. This extra byte size is constant for any Modbus frame. Table 5 indicates how much overhead will be added in the Modbus frame for each type of request.

4.1.2. Time Delay in Total Round Trip Time Duration. As this solution adds processing time for inclusion/deletion of two extra fields and checking of sequence number and time frame, there was a delay in round trip time duration. The total round trip time duration was calculated by time difference between Modbus command triggered at DAQ Server and Modbus response processed at DAQ Server. We conducted the experiment 100 times with different poll intervals from DAQ Server. We got 16 ms average time delay for 1000 ms poll interval. It is 1.6% overhead in total time duration.

Table 5: Overhead of solution on Modbus frame size.

Sl. no.	Type of Modbus request	% overhead on Modbus frame
1	Reading coils/registers	4
2	Writing coils/registers	77

4.2. Comparison of Solution with Existing Related Works. The solution was compared with existing related works. We took references from literature survey and Table 6 describes the differences. Our solution overheads only 1.6% time delay compared to 66% time delay or latency as per [51]. It adds 4% overhead on frame size, compared to 291% as per [33].

In this work, we used testbed with local area network (LAN) connectivity and simulated various attacks by assuming that these attacks penetrated through WAN or Internet into LAN. This study is applicable to internal attacks only. In the future, we will work for external attacks.

Table 6: Comparison of solution with existing related works.

Parameter	Existing works	Our solution
Overhead on frame size/packet length	291% overhead on frame length [[33]]	4% on request, 77% on command frames
Total round trip time duration	66% latency [[51]]	1.6% latency

5. Conclusion

Industrial control systems are using open access networks to leverage efficiency and are more vulnerable to cyberattacks. Modbus is most widely used communication protocol in ICS/SCADA systems, which is one of the core components in Industrial Control Systems. SCADA systems are suffering from security issues and are more vulnerable to cyberattacks. In this paper, we developed an integrated framework solution to prevent the main attacks on SCADA systems. In this framework, we designed a frame filtering module to protect PLC from unauthorized access attacks like False Command Injection and False Access Injection attacks. The module successfully blocked 97% of these attacks. Hence, it protected the PLC from FCI and FAI attacks. We also developed another module in this framework to detect replay attacks in Modbus protocol. The total integrated framework solution was successfully detected and blocked the attacks to reach PLC and HMI. The solution enhances the security for Industrial Control Systems. In this work, we used Local Area Network only for simulating the attacks. We assumed that attacks penetrated through Internet into SCADA network. In the future, we will work for other cyberattacks and WAN/Internet environment.

Abbreviations

SCADA: Supervisory control and data acquisition
ICS: Industrial control systems
PCS: Process control systems
PLC: Programmable logic controllers
RTU: Remote telemetry unit
IED: Intelligent electronic device
LDS: Leak detection system
ERP: Enterprise resource planning
EMS: Energy management system
ICS-CERT: Industrial Control Systems Cyber Emergency Response Team
MAC: Media access control
DAQ: Data acquisition
ARP: Address resolution protocol
PC: Personal computer
FCI: False Command Injection
FAI: False Access Injection.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors would like to acknowledge the management of Koneru Lakshmaiah Education Foundation (K L Deemed to be University) to allow the research in the university.

References

[1] M. Cheminod, L. Durante, and A. Valenzano, "Review of security issues in industrial networks," *IEEE Trans. Ind. Informatics*, vol. 9, pp. 277–293, 2013.

[2] B. Karabacak, S. O. Yildirim, and N. Baykal, "A vulnerability-driven cyber security maturity model for measuring national critical infrastructure protection preparedness," *International Journal of Critical Infrastructure Protection*, vol. 15, pp. 47–59, 2016.

[3] *Pacific Northwest National Laboratory, U.S. Department of Energy, The Role of Authenticated Communications for Electric Power Distribution, Pacific Northwest National Laboratory*, U.S. Department of Energy, Washington, DC, USA, 2006.

[4] K.-C. Kao, W.-H. Chieng, and S.-L. Jeng, "Design and development of an IoT-based web application for an intelligent remote SCADA system," in *Proceedings of the IOP Conference Series: Materials Science and Engineering*, vol. 323, no. 1, pp. 12–25, IOP Publishing Ltd., Komputer, Indonesia, December 2018.

[5] D. Dzung, M. Naedele, T. P. V. Hoff, and M. Crevatin, "Security for Industrial control systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1152–1177, 2015.

[6] B. Babu, T. Ijyas, P. Muneer, and J. Varghese, "Security issues in SCADA based industrial control systems," in *Proceedings of the 2nd International Conference on Anti-cyber Crimes (ICACC)*, pp. 47–51, IEEE, Abha, Saudi Arabia, March 2017.

[7] L. Fillatre, I. Nikiforov, and P. Willett, "Security of SCADA systems against cyber–physical attacks," *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 5, pp. 28–45, 2017.

[8] L. J. Trautman and P. C. Ormerod, "Industrial cyber vulnerabilities: lessons from Stuxnet and the internet of things," *University of Miami Law Review*, vol. 72, p. 761, 2017.

[9] ICS-CERT, "Year in review 2015," 2020, https://ics-cert.us-cert.gov.

[10] W. Su, A. Antoniou, and C. Eagle, "Cyber security of industrial communication protocols," in *Proceedings of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–4, IEEE, Limassol, Cyprus, September 2017.

[11] K. Imtiaz and M. J. Arshad, "Security challenges of industrial communication protocols: threats vulnerabilities and solutions," *International Journal of Computer Science and Telecommunications*, vol. 10, no. 4, 2019.

[12] *Modbus over Serial Line Specification and Implementation Guide V1.02*, 2006, http://www.Modbus-IDA.org.

[13] *Modbus Messaging on TCP/IP Implementation Guide V1.0b*, 2006, http://www.Modbus-IDA.org.

[14] *Modbus Application Protocol Specification V1.1b3*, 2012, http://www.Modbus-IDA.org.

[15] R. Nardone, R. J. Rodríguez, and S. Marrone, "Formal security assessment of Modbus protocol," in *Proceedings of the 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 142–147, IEEE, Barcelona, Spain, December 2016.

[16] A. Volkova, M. Niedermeier, R. Basmadjian, and H. D. Meer, "Security challenges in control network protocols: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 619–639, 2019.

[17] L. Rosa, M. Freitas, S. Mazo, E. Monteiro, T. Cruz, and P. Simões, "A comprehensive security analysis of a SCADA protocol: from OSINT to mitigation," *IEEE Access*, vol. 7, Article ID 42156, 2019.

[18] A. M. Abdul and S. Umar, "Data integrity and security [DIS] based protocol for cognitive radio ad hoc networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, no. 1, pp. 187–195, 2017.

[19] K. Rambabu and N. Venkatram, "Contemporary affirmation of security and intrusion handling strategies of internet of things in recent literature," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 9, pp. 2729–2744, 2018.

[20] S. Bhatia, N. Kush, C. Djamaludin, A. Akande, and E. Foo, "Practical modbus flooding attack and detection," in *Proceedings of the Twelfth Australasian Information Security Conference (AISC 2014)[Conferences in Research and Practice in Information Technology, Volume 149]*, pp. 57–65, Australian Computer Society, Inc., Auckland, New Zealand, January 2014.

[21] A. M. Abdul and S. Umar, "Attacks of denial-of-service on networks layer of OSI model and maintaining of security," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, no. 1, pp. 181–186, 2017.

[22] L. Rajesh and P. Satyanarayana, "Detecting flooding attacks in communication protocol of industrial control systems," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, 2020.

[23] Y. Hu, A Yang, L. Hong, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, Article ID 1550147718794615, 2018.

[24] Y. Yang, K. McLaughlin, T. Littler et al., "Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in smart grid SCADA systems," in *Proceedings of the International Conference on Sustainable Power Generation and Supply (SUPERGEN 2012)*, p. 138, Hangzhou, China, September 2012.

[25] V. Somu, D. B. K. Kamesh, J. K. R. Sastry, and S. N. M. Sitara, "Snort rule detection for countering in network attacks," *Advances in Intelligent Systems and Computing*, vol. 515, pp. 573–583, 2017.

[26] B. Chen, N. Pattanaik, A. Goulart, L. Karen, B. Purry, and D. Kundur, "Implementing Attacks for Modbus/TCP Protocol in a Real-Time Cyber Physical System Test bed," in *Proceedings of the 2015 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, pp. 1–6, IEEE, Charleston, SC, USA, May 2015.

[27] P. Maynard, K. McLaughlin, and B. Haberler, "Towards understanding man-in-the-middle attacks on IEC 60870-5-104 SCADA networks," in *Proceedings of the 2nd International Symposium for ICS & SCADA Cyber Security Research 2014 (ICS-CSR 2014)*, pp. 30–42, St. Poelten, Austria., April 2014.

[28] S. Ghosh and S. Sampalli, "A survey of security in SCADA networks: current issues and future challenges," *IEEE Access*, vol. 7, Article ID 135812, 2019.

[29] P. Priyanga, K. Krithivasan, S. Pravinraj, and V. S. Sriram, "Detection of cyberattacks in industrial control systems using enhanced principal component analysis and hypergraph-based convolution neural network (EPCA-HG-CNN)," *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 4394–4404, 2020.

[30] J. Qian, X. Du, B. Chen, B. Qu, K. Zeng, and J. Liu, "Cyber-physical integrated intrusion detection scheme in SCADA system of process manufacturing industry," *IEEE Access*, vol. 8, Article ID 147471, 2020.

[31] D. Li, N. Gebraeel, and K. Paynabar, "Detection and differentiation of replay attack and equipment faults in SCADA systems," *IEEE Transactions on Automation Science and Engineering*, 2020.

[32] W. Yusheng, K. Fan, Y. Lai et al., "Intrusion detection of industrial control system based on Modbus TCP protocol," in *Proceedings of the IEEE 13th International Symposium on Autonomous Decentralized System (ISADS)*, pp. 156–162, IEEE, Bangkok, Thailand, March 2017.

[33] I. N. Fovino, A. Carcano, M. Masera, and A. Trombetta, "Design and implementation of a secure modbus protocol," *IFIP Advances in Information and Communication Technology Critical Infrastructure Protection*, vol. III, pp. 83–96, 2009.

[34] A. Shahzad, M. Lee, Y.K. Lee et al., "Real time MODBUS transmissions and cryptography security designs and enhancements of protocol sensitive information," *Symmetry*, vol. 7, no. 3, pp. 1176–1210, 2015.

[35] A. Shahzad, S. Musa, and M. Irfan, "Security solution for SCADA protocols communication during multicasting and polling scenario," *Trends in Applied Sciences Research*, vol. 9, no. 7, pp. 396–405, 2014.

[36] X. Luo and Y. Li, "Security enhancement mechanism of modbus TCP protocol," in *Proceedings of the DEStech Transactions on Computer Science and Engineering ICITI*, Seattle, WA, USA, 2018.

[37] J. Dudak, G. Gaspar, S. Sedivy, P. Fabo, L. Pepucha, and P. Tanuska, "Serial communication protocol with enhanced properties–securing communication layer for smart sensors applications," *IEEE Sensors Journal*, vol. 19, no. 1, pp. 378–390, 2019.

[38] W. You and H. Ge, "Design and implementation of modbus protocol for intelligent building security," in *Proceedings of the IEEE 19th International Conference on Communication Technology (ICCT)*, pp. 420–423, IEEE, Xi'an, China, October 2019.

[39] N. Erez and A. Wool, "Control variable classification, modeling and anomaly detection in Modbus/TCP SCADA systems," *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 59–70, 2015.

[40] G. Hayes and K. E. Khatib, "Securing modbus transactions using hash-based message authentication codes and stream transmission control protocol," in *Proceedings of the 2013 Third International Conference on Communications and Information Technology (ICCIT)*, pp. 179–184, IEEE, Beirut, Lebanon, June 2013.

[41] W. Jung, J. Yun, S. Kim, K. Shim, and M. Kim, "Structured whitelist generation in scada network using prefixspan algorithm," in *Proceedings of the 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, p. 326, September 2017.

[42] D. H. Kang, B. K. Kim, N. C. Jung, and K. S. Jhang, "Whitelist generation technique for industrial firewall in SCADA networks," in *Frontier and Innovation in Future Computing and Communications*, pp. 525–534, Springer, Dordrecht, Netherland, 2014.

[43] R. R. R. Barbosa, R. Sadre, and A. Pras, "Flow whitelisting in SCADA networks," *International journal of critical infrastructure protection*, vol. 6, no. 3-4, pp. 150–158, 2013.

[44] H. H. G. Serkan, S. Useyin, Y. N. Ercan, U. Furkan, and K. Gokce, "False Data Injection Attacks and the Insider Threat in ¨ Smart Systems," *Computers & Security*, vol. 97, 2020.

[45] D. Myers, S. Suriadi, K. Radke, and E. Foo, "Anomaly detection for industrial control systems using process mining," *Computers & Security*, vol. 78, pp. 103–125, 2018.

[46] Y. Li and Y. Wang, "False data injection attacks with incomplete network topology information in smart grid," *IEEE Access*, vol. 7, pp. 3656–3664, 2019.

[47] X. Liu, P. Zhu, Y. Zhang, and K. Chen, "A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2435–2443, 2015.

[48] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.

[49] A. Anwar, A. N. Mahmood, and Z. Tari, "Identification of vulnerable node clusters against false data injection attack in an AMI based smart grid," *Information Systems*, vol. 53, pp. 201–212, 2015.

[50] Y. Zhao and C. Smidts, "A control-theoretic approach to detecting and distinguishing replay attacks from other anomalies in nuclear power plants," *Progress in Nuclear Energy*, vol. 123, Article ID 103315, 2020.

[51] M. K. Ferst, H. F. M. D. Figueiredo, G. Denardin, and J. Lopes, "Implementation of secure communication with modbus and transport layer security protocols," in *Proceedings of the 2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, pp. 155–162, IEEE, São Paulo, Brazil, November 2018.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 8876893, 16 pages
https://doi.org/10.1155/2021/8876893

Research Article

Distributed Outsourced Privacy-Preserving Gradient Descent Methods among Multiple Parties

Zuowen Tan,[1] Haohan Zhang ,[2] Peiyi Hu,[1] and Rui Gao[2]

[1]*Department of Computing Science and Technology, Jiangxi University of Finance & Economics, Nanchang 330032, China*
[2]*Research Center for Information Technology Security, CEPREI, Guangzhou 510610, China*

Correspondence should be addressed to Haohan Zhang; zhang.haohan@aliyun.com

Received 13 April 2020; Revised 31 January 2021; Accepted 25 March 2021; Published 22 April 2021

Academic Editor: Amir Anees

The Internet of Things (IoT) is one of the latest internet evolutions. Cloud computing is an important technique which realizes the computational demand of largely distributed IoT devices/sensors by employing various machine learning models. Gradient descent methods are widely employed to find the optimal coefficients of a machine learning model in the cloud computing. Commonly, the data are distributed among multiple data owners, whereas the target function is held by the model owner. The model owner can train its model over data owner's data and provide predictions. However, the dataset or the target function's confidentiality may not be kept in secret during computations. Thus, security threats and privacy risks arise. To address the data and model's privacy mentioned above, we present two new outsourced privacy-preserving gradient descent (OPPGD) method schemes over horizontally or vertically partitioned data among multiple parties, respectively. Compared to previously proposed solutions, our methods improve in comprehensiveness in a more general scene. The data privacy and the model privacy are preserved during the whole learning and prediction procedures. In addition, the execution performance evaluation demonstrates that our schemes can help the model owner to optimize its target function and provide exact prediction with high efficiency and accuracy.

1. Introduction

The Internet of Things (IoT) is the latest internet evolution which provides multifarious novel digital, smart services and products by integrating abundant devices into networks [1]. It enables the communication between the physical world and the cyberspace [2]. IoT system contains radio-frequency identifications, wireless sensor networks, and the cloud computing [3]. Cloud computing realizes the computational demand of large-scale distributed IoT devices or sensors through various machine learning methods. Since IoT devices have tiny memory, the collected data are required to be stored and managed by the cloud servers [3–5]. Data can be downloaded from the cloud for different purposes such as machine learning. However, since there may exist sensitive data such as physiological data, location data, and some other data which are closely related to our personal information [6], this exposes the data to security breaches. Therefore, IoT not only provides convenience but also brings about security and privacy issues [7]. How to deal with security, privacy, and trust has been one of the main barriers in developing IoT in the real world [8, 9]. Most of the existing work on the protection of sensitive data is based on the secure communication channels and authorization [10]. In our paper, we focus on the protection of sensitive data in machine learning or deep learning. The data can be protected during the transmission phase, the computation phase, and the prediction phase. Furthermore, the computation and prediction results' privacy can also be preserved.

In machine learning or deep learning, the prediction function is usually called the decision model. The model coefficients' quality determines the accuracy of the model. In order to minimize the error of the model, the optimal coefficients are indispensable. This process is called model learning. Gradient descent methods are effective methods to

find the optimal coefficients of the decision model, such as linear regression, hyperplane decision classification, and neural networks. Gradient descent methods conclude four types: classical gradient descent method (GD), stochastic gradient descent method (SGD), minibatch stochastic gradient descent method (minibatch SGD), and momentum. Through these methods, the optimal prediction function can be obtained after several iterations.

In the cloud computing, the cloud server offers huge storage and computing capacity. The model owner initializes the prediction function, and the training data are distributed among different data owners who hope to get desired results with these data by cloud servers without exposing their privacy. These data form an enormous training dataset which is divided into different disjoint subsets held by different data owners. The dataset partition can be horizontal or vertical. The number of data owners can be two, even more than two. As is known to all, the channel transmission is not secure in our real life. In addition, data owners, the model owner, and the cloud server do not trust each other. When they train a decision model together, they worry about that any other participant may get information from their own data. So, they encrypt their training data or the decision model with their own public keys or blind their data to preserve confidentiality before delivering them to the cloud server. The training data and the decision model can be kept confidential during the whole cloud computing. After finishing training the decision model, the model owner learns the model securely based on the training dataset with the help of the cloud server. At this time, the clients can get the prediction about their request data from the cloud server according to this decision model.

At present, although a lot of researchers focus on the data privacy protection or the model privacy protection when gradient descent methods are utilized to optimize machine learning models, few schemes can provide both data privacy and model privacy at the same time. Beyond that, some privacy-preserving gradient descent schemes can protect data owners' privacy, but they are not applied to an outsourcing computation. In addition, the dataset's partition is usually horizontal or vertical in the distributed system. In many previous literature studies, few schemes can be applied to two different partitioned datasets at the same time. Besides, both training data and the decision model are held only by data owners rather than the model owner. In fact, it is more practical that the models are held by the model owner rather than the data owner. Motivated by the above, we construct two novel outsourcing gradient descent methods to solve these problems.

Generally speaking, it is necessary to preserve the privacy of the training data, the decision model, and the request data during the model training. Assume that there exists a training dataset X, and the corresponding label vector is $\mathbf{y}$. Each row of the dataset represents one sample $\mathbf{x}$ with a set of attributes. By $f(\mathbf{x})$, we denote the prediction function which maps the sample $\mathbf{x}_i$ into its corresponding category label y_i. According to the partition of the dataset, each data owner has part of data samples or part of the attributes. The model owner holds the coefficients of the prediction function $f(\mathbf{x})$. The target of data owners and the model owner is to minimize the error of the prediction function and obtain the optimal coefficients ultimately through the gradient descent methods. Thus, the model owner holds the optimal decision model. Then, it can provide the client accurate prediction. In this paper, we focus on outsourced gradient descent methods over distributed data among multiple parties which conclude data owners, the model owner, the cloud server, and the client. Both horizontal and vertical partition of the dataset are discussed. For the horizontally partitioned dataset, two or multiple data owners hold different samples with the same attributes, whereas two or more data owners hold all same samples but with different sets of attributes when the dataset is vertically partitioned.

1.1. Contributions. To address the privacy when performing gradient descent methods by multiple parties via the cloud computing, we propose two OPPGD schemes over horizontally or vertically distributed data. Our main contributions of this paper are summarized as follows:

(1) We design an outsourced privacy-preserving scalar product (OPPSP) algorithm. The cloud server computes the inner product of two vectors encrypted under different keys securely. For example, one data owner and the model owner hold one vector, respectively. Both parties first encrypt their own vector with their own key and send the encrypted vector to the cloud server. Then, the cloud server computes the scalar product of these two encrypted vectors.

(2) We propose two secure and comprehensive schemes to perform OPPGD over horizontally or vertically distributed dataset, respectively. The number of data owners can be two or more than two. The prediction functions are linear regression or neural networks. The OPPGD schemes are applied to classical GD, SGD, minibatch SGD, and momentum. It is worth noting that our schemes are with higher applicability and practicability contrasted with other schemes.

(3) We demonstrate that our OPPGD schemes are privacy-preserving. The computational cost and communication complexities are discussed. The analyses show that our OPPGD schemes are with high efficiency and accuracy.

1.2. Organization. The remainder of this paper is as follows. In Section 2, we discuss the related works on privacy-preserving gradient descent methods. In Section 3, we briefly introduce some preliminaries, Elgamal homomorphic cryptosystem [11], and gradient descent methods. In Section 4, we describe the system model, problem statements, the threat model, and the system requirements. We present two OPPGD schemes and prove their correctness, security, and complexity in Section 5. The performance evaluation of the schemes is analyzed in Section 6. Section 7 makes a conclusion on our OPPGD schemes.

2. Related Works

In this section, we review works on privacy-preserving gradient descent methods among parties. According to the existence or absence of cloud servers, the existing works can be classified into two categories.

2.1. The Absence of Cloud Servers. Wan et al. [12] presented the first privacy-preserving scheme for gradient descent methods. They proposed a generic formulation of gradient descent methods by defining the prediction function $f(x)$ as a composition $g^{\circ}h(x)$. The formulation is used to perform the specific iteration-based algorithm in linear regression or neural networks. In our paper, we also use this formulation. However, the partition of the dataset discussed in their scheme [12] is only vertical. Han et al. [13] extended the scheme [12] to the horizontally distributed dataset and proposed the least square approach to perform gradient descent methods. Both schemes [12, 13] utilize a secure scalar product to gain their privacy preservation, but they cannot be applied to the outsourced model. Gabor Danner and Jelasity [14] designed a novel fully distributed privacy-preserving minibatch SGD that can avoid collecting any personal data centrally. Their scheme does not require the precise sum of gradients. A tree topology and homomorphic encryption are employed to produce a "quick and dirty" partial sum. The protocol can resist collusion attacks. Hegedus and Jelasity [15] adopted differential privacy technology to solve privacy-preserving stochastic distributed gradient descent methods. Mehnaz et al. [16] designed two secure gradient descent schemes over horizontally partitioned data and vertically partitioned data via a secure sum protocol. Later, they designed a secure gradient descent method scheme [17] without Yao's circuits over the arbitrarily partitioned dataset. Based on output perturbation, Wu et al. [18] devised a novel "bolt-on" differentially private algorithm for stochastic gradient descent.

2.2. The Existence of Cloud Servers. Liu et al. [19] designed an encrypted gradient descent method. Both data owners and the cloud server perform operations collaboratively to learn the target function without leaking any data privacy. They extended their scheme to the outsourced model by utilizing the BGN cryptosystem. However, their protocol is only suitable for a two-party scenario. Shokri and Shmatikov [20] learnt an accurate neural network model without sharing input datasets by using the stochastic gradient descent method. After the parameter server initializes the parameter vector, it updates the parameters with the help of the cloud server without leaking any privacy. Kim et al. [21] provided a practical frame for mainstream learning models such as logistic regression. They calculated the gradient descent method securely by using homomorphic encryption, but this is inefficient. Since the required bit length of ciphertext modulus per iteration is too long, it also takes up too much space. Francisco-Javier et al. [22] realized training supervised machine learning over ciphertext. Through the gradient descent method, the server optimizes the predicted training model without exposing the data or the training model. Mohassel and Zhang [23] used the stochastic gradient descent method to construct new and efficient privacy-preserving machine learning protocols for linear regression, logistic regression, and neural network. Their protocol is involved with a two-server model. Data providers distribute their private data among two noncolluding servers, while the servers train models on the joint data through secure two-party computation techniques. Li et al. [24] also presented a multikey privacy-preserving deep learning scheme in the cloud computing environment. Their protocols realize outsourced multilayer backpropagation network learning via the gradient descent methods. Ma et al. [25] took advantage of two noncolluding servers' framework to build a new outsourced model of the privacy-preserving neural network. However, the model owner can only make prediction rather than learning the model itself.

2.3. The Other Works on Privacy Preservation for Machine Learning. Aside from the above privacy-preserving gradient descent methods, there are also plenty of other works on privacy-preserving computation over distributed data among multiple parties under the cloud environment. Liu et al. [26] constructed an efficient privacy-preserving method to compute outsourced data. They [27] also proposed a privacy-preserving outsourced calculation toolkit, which allows data owners to securely outsource their data to the cloud for storage and calculation. Rady et al. [28] designed a new architecture that achieves the confidentiality and integrity of query results of the outsourced database. Yu et al. [29] devised a verifiable outsourced computation scheme over encrypted data by employing fully homomorphic encryption and polynomial factorization algorithm. Chamikara et al. [30] presented an efficient and scalable nonreversible perturbation algorithm of data mining without leaking privacy of big data via optimal geometric transformations. Li et al. [31] proposed a novel outsourced privacy-preserving classification scheme based on homomorphic encryption. In their scheme, multiple parties outsource securely their sensitive data to an untrusted evaluator for storing and processing. Li et al. [32] devised a novel scheme for a classifier owner to provide users with the privacy-preserving classification service by delegating a cloud server. However, they focus on two concrete secure classification protocols: naive Bayes classifier and hyperplane decision classifier. Park et al. [33] described a privacy-preserving naive Bayes protocol. No intermediate interactions are required between the server and the clients. Hence, their protocols can alleviate the heavy computational cost of fully homomorphic encryption. Li et al. [34] proposed an outsourced privacy-preserving C4.5 decision tree algorithm over both horizontally and vertically partitioned datasets. They used the BCP cryptosystem to present an outsourced privacy-preserving weighted average protocol. Rong et al. [35] presented a series of privacy-preserving building blocks for verifiable and privacy-preserving association rule mining under the hybrid cloud environment. Li et al. [36] used an efficient homomorphic encryption with multiple keys to design an outsourced privacy-preserving ID3 data mining

solution. Xue et al. [37] built a differential privacy-based privacy-preserving classification system for secure edge computing. Yang et al. [38] realized privacy-preserving medical record sharing in the cloud computing environment. Kaur et al. [39] devised an efficient privacy-preserving collaborative filtering for the healthcare recommender system over arbitrary distributed data. In our work, we aim at designing outsourced privacy-preserving gradient descent methods among multiple parties. To the best of our knowledge, there has not been any work which addresses the issue comprehensively.

3. Preliminaries

In this section, we introduce some preliminaries for our outsourced privacy-preserving gradient descent schemes.

3.1. The Elgamal Homomorphic Cryptosystem. The Elgamal cryptosystem [11] comprises the following algorithms: preparation, key generation, encryption, and decryption:

Preparation (λ): given a security parameter λ. The system generates the public parameter PP as follows. The system first chooses a large prime number N and a random number g less than N. And it publishes the multiplicative cyclic group G of prime order N with the generator g. The public parameter PP = (Ng)

KeyGen (PP): taking PP as the input, each party P_i randomly selects a number sk_i less than N as its private key and computes $\mathrm{pk}_i = g^{\mathrm{sk}_i} \bmod N$ as its public key.

$\mathrm{Enc}_{\mathrm{pk}_i}(M_i)$: P_i selects a random integer r_i which is coprime to $(N-1)$ and encrypts its plaintext M_i with the public key pk_i to generate the ciphertext $C_i = (C_{i1}C_{i2})$

$$\begin{aligned} C_{i1} &= g^{r_i} \bmod N \\ C_{i2} &= \mathrm{pk}_i^{r_i} m \bmod N \end{aligned} \tag{1}$$

$\mathrm{Dec}_{\mathrm{pk}_i}(C_i)$: each party P_i decrypts C_i with its secret key sk_i and obtains the plaintext M_i. The decryption process is

$$M_i = C_{i2}C_{i1}^{-\mathrm{sk}_i} \bmod N \tag{2}$$

Its correctness is early confirmed.

$$\begin{aligned} C_{i2}C_{i1}^{-\mathrm{sk}_i} \bmod N &= \left(\mathrm{pk}_i^{r_i} M_i\right)\left(g^{r_i}\right)^{-\mathrm{sk}_i} \bmod N \\ &= \left(g^{\mathrm{sk}_i r_i} M_i\right)\left(g^{r_i}\right)^{-\mathrm{sk}_i} \bmod N \\ &= M_i \end{aligned} \tag{3}$$

The semantic security of the Elgamal cryptosystem is based on the hardness assumption of discrete logarithm problems over finite fields.

3.2. The Key Conversion System. As for the secure outsourced computation over the dataset among multiple parties, the essential difficulty is how to deal with different ciphertexts encrypted under different keys which are sent from multiple parties. Based on Gentry's fully homomorphic encryption [40], we transform the ciphertext under different keys into the ciphertext under the same key. Take two parties, Alice and Bob, as an example. Assume that their respective key pairs are $(\mathrm{pk}_a\mathrm{sk}_a)$ and $(\mathrm{pk}_b\mathrm{sk}_b)$. For a plaintext m, its ciphertext encrypted under key pk_a is $[m]_{\mathrm{pk}_a}$. The goal is to switch encrypted $[m]_{\mathrm{pk}_a}$ into a new ciphertext $[m]_{\mathrm{pk}_b}$ which is encrypted under the public key pk_b. The conversion can be divided into the following steps:

Rekey generation $(\mathrm{pk}_b\mathrm{sk}_a)$: taking pk_b and sk_a as the input, it outputs the rekey $\widetilde{\mathrm{sk}_a} = \{\widetilde{\mathrm{sk}_{ai}}\}_{i=1}^{I}$, where $\widetilde{\mathrm{sk}_{ai}} \longleftarrow \mathrm{Encrypt}(\mathrm{pk}_b\mathrm{sk}_{ai})$ is the i-th binary representation of sk_a

Reencryption $(\mathrm{pk}_b, [m]_{\mathrm{pk}_a})$: taking public key pk_b and ciphertext $[m]_{\mathrm{pk}_a}$ as inputs, it outputs $\widetilde{m} = \{\widetilde{m}_i\}_{i=1}^{I}$, where $\widetilde{p}_i \longleftarrow \mathrm{Encrypt}(\mathrm{pk}_b[p_i]_{\mathrm{pk}_a})$

Evaluation algorithm $(\mathrm{pk}_b D_\Pi, \widetilde{p}_i\widetilde{\mathrm{sk}_{ai}})$: taking the public key pk_b, rekey $\widetilde{\mathrm{sk}_a}$, ciphertext $\widetilde{m}$, and a decryption circuit D_Π, it outputs $[m]_{\mathrm{pk}_b} \longleftarrow \mathrm{Evaluate}(\mathrm{pk}_b D_\Pi \widetilde{m}\widetilde{\mathrm{sk}_a})$

3.3. Gradient Descent Methods. Assume that D is the dataset of data samples, $\{(\mathbf{x}_i y_i) | i = 1, 2 \ldots N\}$, where the vector $\mathbf{x}_i = [x_{i1}x_{i2} \ldots x_{im}]$ presents the i-th sample's m attributes and y_i denotes the target attribute. The goal is to determine a prediction function $f(x)$ such that $f(\mathbf{x}_i)$ is as close to y_i as possible. Thus, when one makes prediction about the test data, the basic strategy is to make the prediction function to produce the smaller error. Gradient descent methods are always applied to search $f(x)$'s optimal coefficients. The technique can minimize the prediction error. The whole process can be described as follows. At the beginning, one determines the loss function $L(x)$, randomly initializes a coefficient vector of $f(x)$, and calculates the current error about the learning dataset. If the current error is not ideal, one can take the derivative of $L(x)$ with respect to the vector, modify the coefficient vector, and update $f(x)$ based on the derivative. Then, one recalculates the loss and repeats optimizing its model until the minimum error appears. To this end, one can generate the optimal value through several iterations.

There are four main gradient descent methods, such as classical GD, SGD, minibatch SGD, and momentum. In classical GD, the loss function is determined by all samples in each iteration which leads to high computational complexity. For SGD, the loss function is determined by a random sample every iteration which reduces computing overhead. However, this method has one weakness that, sometimes, the final coefficient vector may be the local optimal value rather than the global optimal value. When the loss function is determined by a batch of random samples every iteration, the gradient descent method is called minibatch SGD. The minibatch SGD has classical GD's and SGD's advantages and overcomes their weaknesses. So far, SGD is the most widely applied in machine learning. Momentum is the latest gradient descent method which greatly improves the accuracy and speed of the prediction. Beside

the learning rate η, the coefficient vector $\omega = \gamma\omega - \eta\nabla\omega$ in momentum contains a new parameter γ, the attenuation rate. However, our schemes can be applied to the above four main gradient descent methods.

The error function of every sample $\mathbf{x}_i$ is $E(f(\mathbf{x}_i)y_i) = (y_i - f(\mathbf{x}_i))^2$. Given l arbitrary samples, the loss function is (12)

$$L = \frac{1}{l}\sum_{i=1}^{l} E(f(\mathbf{x}_i)y_i) = \frac{1}{2l}\sum_{i=1}^{l}(y_i - f(\mathbf{x}_i))^2 \quad (4)$$

The prediction function $f(x)$ is a composition function of two functions $g(z)$ and $z = h(x)$, where $g(z)$ is any differentiable function and $h(x)$ is a linearly separable function: $h(\mathbf{x}_i) = \sum_{j=1}^{m}\omega_j x_{ij}$, where $\omega = (\omega_1\omega_2 \ldots \omega_m)$ is the coefficient vector of the prediction function. When $l = 1$, the method is SGD, when $1 < l < N$, the method is minibatch SGD, whereas when $l = N$, the method is GD. Subsequently, we update the coefficient vector $\omega = \omega - \eta\nabla\omega$, where $\nabla\omega = \partial L\partial\omega$ and η is a constant parameter called learning rate. When the coefficient vector is $\omega = \gamma\omega - \eta\nabla\omega$, where γ is a constant parameter called attenuation rate, this method is momentum.

For each sample $\mathbf{x}_i$, there is a derivative $\partial E(f(\mathbf{x}_i)y_i)\partial\omega$. Thus, we calculate $\nabla\omega = (\partial L\partial\omega) = (1l)\sum_{i=1}^{l}\partial E(f(\mathbf{x}_i)y_i)\partial\omega$

$$\begin{aligned}\frac{\partial E(f(\mathbf{x}_i)y_i)}{\partial\boldsymbol{\omega}} &= \frac{\partial E(f(\mathbf{x}_i)y_i)}{\partial f(\mathbf{x}_i)}\frac{\partial f(\mathbf{x}_i)}{\partial\boldsymbol{\omega}} \\ &= \frac{\partial E(f(\mathbf{x}_i)y_i)}{\partial f(\mathbf{x}_i)}\frac{\partial g(h(\mathbf{x}_i))}{\partial h(\mathbf{x}_i)}\frac{\partial h(\mathbf{x}_i)}{\partial\boldsymbol{\omega}}\end{aligned} \quad (5)$$

As the function $f(x)$ changes, $\nabla\omega$ is also different. Here, we discuss two specific functions used in linear regression and neural network.

In linear regression, the prediction function for an arbitrary sample $\mathbf{x}_i$ is $f(\mathbf{x}_i) = \sum_{j=1}^{m}\omega_j x_{ij}$. Then,

$$\begin{aligned}\frac{\partial E(f(\mathbf{x}_i)y_i)}{\partial\boldsymbol{\omega}} &= \frac{\partial}{\partial\boldsymbol{\omega}}\left(\frac{1}{2}(y_i - f(\mathbf{x}_i))^2\right) \\ &= \frac{1}{2}\mathbf{x}_i(y_i - f(\mathbf{x}_i)) \\ &= \frac{1}{2}(\mathbf{x}_i f(\mathbf{x}_i) - \mathbf{x}_i y_i)\end{aligned} \quad (6)$$

In neural networks, $f(x)$ is also called as activation function that is a sigmoid function, $(z) = 1(1 + e^{-\alpha z})$, or tanh function, $f(z) = (e^{\alpha z} - e^{-\alpha z})(e^{\alpha z} + e^{-\alpha z})$. If the function is a sigmoid function, the prediction function for an arbitrary sample $\mathbf{x}_i$ is $f(\mathbf{x}_i) = 1(1 + e^{-\alpha\sum_{j=1}^{m}\omega_j x_{ij}})$. Then,

$$\begin{aligned}\frac{\partial E(f(\mathbf{x}_i)y_i)}{\partial\boldsymbol{\omega}} &= \frac{\partial}{\partial\boldsymbol{\omega}}\left(\frac{1}{2}(y_i - f(\mathbf{x}_i))^2\right) \\ &= -\frac{1}{2}\alpha(y_i - f(\mathbf{x}_i))f(\mathbf{x}_i)(1 - f(\mathbf{x}_i)) \\ &= -\frac{1}{2}\alpha(y_i - f(\mathbf{x}_i))f(\mathbf{x}_i)(1 - f(\mathbf{x}_i)) \\ &= \frac{1}{2}\alpha\left(\mathbf{x}_i y_i f^2(\mathbf{x}_i) + \mathbf{x}_i f^2(\mathbf{x}_i) - \mathbf{x}_i f^3(\mathbf{x}_i) - \mathbf{x}_i y_i f(\mathbf{x}_i)\right)\end{aligned} \quad (7)$$

Through the Taylor expansion formula, the function $f(x)$ can be expanded into a polynomial $T(a)$. Then, we have

$$\begin{aligned}\frac{\partial E(f(\mathbf{x}_i)y_i)}{\partial\boldsymbol{\omega}} \approx \frac{1}{2}\alpha\Bigg(&\mathbf{x}_i y_i T^2\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) + \mathbf{x}_i T^2\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) \\ &- \mathbf{x}_i T^3\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) - \mathbf{x}_i y_i T\left(\sum_{j=1}^{m}\omega_j x_{ij}\right)\Bigg)\end{aligned} \quad (8)$$

4. Models and Requirements

4.1. System Model. As shown in Figure 1, the system comprises five entities: data owners, a model owner, a cloud server, a key conversion server, and a trusty decryption server. Each entity is described as follows:

Data owner (DO): after receiving the public parameter PP, each DO generates their own key pair and encrypts their data. Then, DOs send their respective ciphertext to the cloud server, depicted as Step 1 in Figure 1. After MO has finished training the model, one DO can request the CS and MO to make prediction.

Cloud server (CS): assume that CS can provide DOs and MO with unlimited computation and storage service. After receiving vectors encrypted by every DO and MO, the CS executes the OPPSP algorithm and finally sends the encrypted results back to DOs as Step 2.

Model owner (MO): MO holds the target function which contains the coefficient vector, learning rate, or the attenuation rate. MO encrypts the target function's coefficients with its own key and then sends the ciphertext to the CS and executes the OPPSP algorithm. After receiving $\nabla\omega$, MO updates its model until it gets the optimal coefficients. Moreover, it can provide DO with prediction services.

Key conversion server (KCS): KCS runs the conversion algorithm and switches different ciphertexts encrypted under DOs' respective keys into a new intermediate ciphertext under the same key, which is depicted as Step 3.

Trusty decryption server (TDS): assume that TDS is trusty. It only provides decryption service. TDS will not conspire with other parties. After receiving new encrypted results from the KCS as Step 4, the TDS decrypts these results and performs few computations to acquire the final results. In the end, TDS sends the intermediate results back to the MO, as depicted in Step 5.

In our system model, each entity is semihonest except TDS. All the entities have some background knowledge of the attribute names, class names, and the number of their attributes. Each data owner has a part of the complete dataset, which can be partitioned horizontally or vertically.

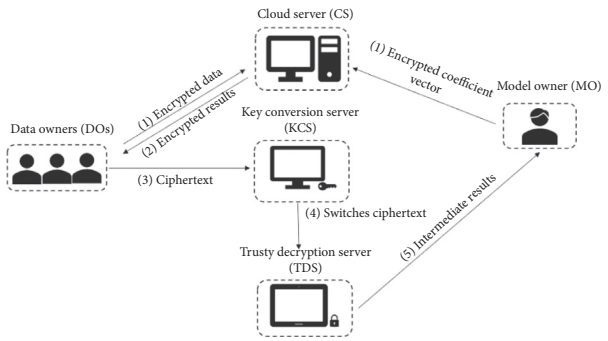


Figure 1: The system model.

When the dataset is distributed vertically, all data owners have the class value vector. The complete attribute dataset X is of size $n \times m$, and the target vector $\mathbf{y}$ is represented as follows:

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}$$
$$y = [y_1 y_2 \cdots y_n]^T \tag{9}$$

where y_i is $\mathbf{x}_i$'s corresponding class value.

For the horizontally partitioned dataset, each data owner has n samples with all the attributes and the corresponding class value, as described in Figure 2. For the vertically partitioned dataset, each data owner has m_i attributes with all the samples and the corresponding class value. The data owner P_i's data can be depicted as in Figure 3

The scheme consists of the preparation phase, the training phase, and the prediction phase. An overview of the scheme can be described as follows:

Preparation phase: according to the public parameter PP, DOs and MO generate their respective key pairs. They also share a secret value k in advance. Then, DOs encrypt their dataset with their respective keys, while MO encrypts the coefficient vector of its model with his public key. Then, DOs and MO send their ciphertext to CS, respectively.

Training phase: CS performs the OPPSP algorithm and sends the results back to DOs. Next, DOs perform decryption and send the results to the KCS. KCS switches these encrypted results and sends the final results to the TDS. TDS decrypts the results and sends the results to the MO. MO can update the coefficients to optimize the model.

Prediction phase: with the help of the CS, the MO makes prediction for the DO's query.

4.2. Problem Statement. Let D_i be the dataset of data owner DO_i. All datasets are disjoint and composed of the complete dataset. Each dataset $D_i = \{\mathbf{x}_j^i\} \subset X$ is of size p_i, where the integers $j \in [1 p_i]$ and $i \in [1t]$. If the dataset is partitioned horizontally, $\mathbf{x}_j^i \in R^m$. If the dataset is partitioned vertically, $\mathbf{x}_j^i \in R^n$. MO holds the coefficient vector $\omega \in R^m$ of the target function $f(x)$ and the target vector $\mathbf{y} \in R^m$

$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$\cdots$	$x_{1,m}$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$\cdots$	$x_{2,m}$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$\cdots$	$x_{3,m}$
$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$\cdots$	$x_{4,m}$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$x_{n-1,1}$	$x_{n-1,2}$	$x_{n-1,3}$	$\cdots$	$x_{n-1,m}$
$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$\cdots$	$x_{n,m}$

Figure 2: Horizontally partitioned dataset.

$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$\cdots$	$x_{1,m}$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$\cdots$	$x_{2,m}$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$\cdots$	$x_{3,m}$
$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$\cdots$	$x_{4,m}$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$x_{n-1,1}$	$x_{n-1,2}$	$x_{n-1,3}$	$\cdots$	$x_{n-1,m}$
$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$\cdots$	$x_{n,m}$

Figure 3: Vertically partitioned dataset.

Our goal is to train the MO's target function with DOs' datasets. MO needs to get $\nabla\omega$ to optimize the coefficients of the target function $f(x)$ after renewing coefficients over MO's coefficients and DOs' datasets. We discuss two kinds of machine learning methods: linear regression and neural network. For linear regression, each MO's task is to obtain encrypted $\mathbf{x}_i(y_i - f(\mathbf{x}_i))$ of every sample $\mathbf{x}_i$ with the help of

the CS. For the neural network, each MO's task is to obtain $\mathbf{x}_i f(\mathbf{x}_i)(y_i - f(\mathbf{x}_i))(1 - f(\mathbf{x}_i))$ for every sample $\mathbf{x}_i$. After getting the results $\nabla\omega$, MO chooses one gradient descent method to refresh its coefficients. In the end, MO can provide accurate prediction services about the query through its optimal target function.

Since each P_i encrypts D_i with its public key pk_i and MO encrypts its coefficients with its public key pk, CS performs computations only over encrypted data. TDS performs the decryption algorithm of the final results, while DO and MO share a secret value k. This can prevent the TDS from getting the information about the coefficients.

4.3. Threat Model. Assume that all the entities except TDS are semihonest, honest-but-curious. In other words, these entities follow the protocol, but they may try to obtain as much as secret information from the message which they receive.

Consider two kinds of adversaries in this model: an external adversary and an internal adversary. An external adversary may obtain some information, i.e., encrypted data or encrypted results, during every iteration via public channels. An internal adversary could refer to a malicious data owner DO, the model owner MO, the cloud server CS, or the key conversion server KCS. The goal of a malicious DO is to extract the coefficients of target function $f(x)$. An internal adversary KCS tries to extract the intermediate results and the MO's coefficient vector, while the goal of an adversary MO is to reveal the information of each DO's partitioned dataset. In addition, if the CS is an internal adversary, it tries to acquire MO's coefficients or DO's datasets.

4.4. Privacy Requirements. In the outsourced gradient descent schemes, privacy preservation is essential. In our model, we assume that the cloud server is semihonest. In order to measure the extent of privacy preservation, now, we define two privacy preservation levels.

Definition 1. Explicit privacy leakage means that privacy may be exposed during the computation of the cloud server or among the message transmission over public channels. If an outsourced computation scheme can prevent the explicit privacy leakage, we call it achieving the level-1 privacy.

Definition 2. Implicit privacy leakage means that one's privacy may be leaked by deducing from results of the cloud server. If an outsourced computing scheme can prevent the implicit privacy leakage, we call it achieving the level-2 privacy.

In our OPPGD scheme, DOs' data and MO's coefficient vector are uploaded to the cloud server in the ciphertext. Explicit privacy leakage means that DOs' data or MO's coefficient vector and final desired results are leaked during the scheme. Implicit privacy leakage means that it is impossible to deduce DOs' data or MO's coefficient vector from intermediate results. Our OPPGD schemes can realize level-1 privacy or level-2 privacy.

5. Two OPPGD Schemes

In this section, we present two outsourced privacy-preserving gradient descent schemes over horizontally partitioned data or vertically partitioned data. For simplicity, we make the following assumptions. When data are horizontally partitioned, each DO has only one record with all the attributes and the class value. When data are vertically partitioned, each DO has one attribute of all the samples and the corresponding class vector. An outsourced privacy-preserving gradient descent scheme is composed of the preparation phase, the training phase, and the prediction phase. Now, we first describe the OPPGD scheme over horizontally partitioned data.

5.1. OPPGD Scheme over Horizontally Partitioned Data

5.1.1. Preparation Phase. The phase is involved with several essential algorithms, parameter generation, key pair generation, and encryption.

Step 1: the system runs Algorithm 1 to generate PP = (g, N) and SP = k

Step 2: after receiving the PP, DOs, MO, and TDS operate Algorithm 2 to obtain their own key pair $(\mathrm{pk}_i \mathrm{sk}_i)$, $(\mathrm{pk}_M\, \mathrm{sk}_M)$, and $(\mathrm{pk}_*\, \mathrm{sk}_*)$

Step 3: DO encrypts its $\mathbf{x}_i$ and y_i to be $[\mathbf{x}_i]_{\mathrm{pk}_i} = \mathrm{pk}_i^{r_i}\mathbf{x}_i \bmod N$ and $[y_i]_{\mathrm{pk}_i} = \mathrm{pk}_i^{r_i} y_i \bmod N$. Then, MO encrypts its coefficient vector ω to be $[[\omega_1]_{\mathrm{pk}_M}[\omega_2]_{\mathrm{pk}_M} \ldots [\omega_m]_{\mathrm{pk}_M}]$ by Algorithm 3, where $\mathbf{x}_i = (x_{i1}x_{i2} \ldots x_{im})$ and $[\omega_i]_{\mathrm{pk}_M} = \mathrm{pk}_M^{r_M}\omega_i \bmod N$

5.1.2. Training Phase

Step 4: each DO sends their encrypted $[\mathbf{x}_i]_{\mathrm{pk}_i}$ and $[y_i]_{\mathrm{pk}_i}$ to the CS, and MO sends $[\omega]_{\mathrm{pk}_M}$ to the CS.

Step 5: CS operates Algorithm 4 and obtains the encrypted scalar product vector $\mathbf{S}$ after receiving $[\mathbf{x}_i]_{\mathrm{pk}_i}$, $[y_i]_{\mathrm{pk}_i}$, and $[\omega]_{\mathrm{pk}_M}$ from DOs and MO, where $s_i = [\mathbf{x}_i]_{\mathrm{pk}_i} \cdot [\omega]_{\mathrm{pk}_M}$. In addition, CS also makes some other computations over some components of $\nabla\omega$. To be specific, CS computes $\mathbf{I}_{i1}$ and $\mathbf{I}_{i2}$ in the linear regression model or computes $\mathbf{I}_{i3}$, I_{i4}, $\mathbf{I}_{i5}$, and $\mathbf{I}_{i6}$ in the neural network model,

Input: the security parameter λ
Output: the public parameter PP = $\{g, N\}$, a secret value k
(1) generate a prime N, choose a primitive element g in Z_N^*
(2) generate a secret value k
(3) **end**

Algorithm 1: Parameter generation.

Input: the public parameter PP$\{gN\}$ and a secret value k
Output: the key pair (pk sk)
(1) choose sk < N
(2) compute pk = $g^{\text{sk}} \bmod N$
(3) **end**
(4) return (pk sk)

Algorithm 2: Key pair generation.

Input: the key pair (pk sk), a message m, and a random integer r_i which is a coprime to $N - 1$
Output: the encrypted message $[m]_{\text{pk}}$
(1) choose a random integer r which is a coprime to $N - 1$
(2) compute $[m]_{\text{pk}} = \text{pk}^r m \bmod N$, g^r **mod N**
(3) **end**
(4) **return** $[\mathbf{m}]_{\mathbf{pk}}$ and $g^r \bmod N$

Algorithm 3: Encryption.

Input: two encrypted vectors $[a]_A$ and $[b]_B$, where $a = (a_1 a_2 \ldots a_m)$ and $b = (b_1 b_2 \ldots b_m)$
Output: encrypted scalar product s
(1) CS computes $s = \sum_{j=1}^{m} [a_i]_A \cdot [b_i]_B$
(2) **end**
(3) return **s**

Algorithm 4: Outsourced privacy-preserving scalar product.

Input: an encrypted message $[m]_{\text{pk}}$, its corresponding key pair (pk sk), and ciphertext $g^r \bmod N$
Output: m
(1) compute: $m = [m]_{\text{pk}} \cdot g^{-r\text{sk}} \bmod N$
(2) **end**
(3) **return m**

Algorithm 5: Decryption.

$$\begin{aligned}
\mathbf{I}_1' &= g^{2\mathrm{sk}_i r_i}\mathbf{x}_i y_i \bmod N\\
\mathbf{I}_2' &= g^{\mathrm{sk}_M r_M}\mathbf{x}_i \sum_{j=1}^{m}\omega_j x_{ij} \bmod N\\
\mathbf{I}_3' &= g^{2\mathrm{sk}_M r_M}\mathbf{x}_i y_i T^2\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) \bmod N\\
\mathbf{I}_4' &= g^{2\mathrm{sk}_M r_M}\mathbf{x}_i T^2\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) \bmod N\\
\mathbf{I}_5' &= [\mathbf{x}_i]_{\mathrm{pk}_i} s_i^3 = g^{3\mathrm{sk}_M r_M}\mathbf{x}_i T^3\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) \bmod N\\
\mathbf{I}_6' &= [\mathbf{x}_i]_{\mathrm{pk}_i}[y_i]_{\mathrm{pk}_i} s_i = g^{\mathrm{sk}_M r}\mathbf{x}_i y_i T\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) \bmod N
\end{aligned} \tag{10}$$

Step 6: CS sends the above encrypted results to the DO.

After receiving encrypted scalar product **S**, DO performs decryption operation. The TDS and the MO perform decryption as shown in Algorithm 5

Step 7: once DOs receive encrypted results $\mathbf{I}_{i1}\mathbf{I}_{i2}$ or $\mathbf{I}_{i3}$, $\mathbf{I}_{i4}$, $\mathbf{I}_{i5}$, $\mathbf{I}_{i6}$ from CS, DO runs Algorithm 5 to get the new ciphertext:

$$\begin{aligned}
k\mathbf{I}_1^{'} &= [k\mathbf{x}_i y_i]_{\mathrm{pk}^*}\\
k\mathbf{I}_2^{'} &= \left[k\mathbf{x}_i \cdot \sum_{j=1}^{m}\omega_j x_{ij}\right]_{\mathrm{pk}^*}\\
k\mathbf{I}_3^{'} &= \left[k\mathbf{x}_i y_i T^2\left(\sum_{j=1}^{m}\omega_j x_{ij}\right)\right]_{\mathrm{pk}^*}\\
k\mathbf{I}_4^{'} &= \left[k\mathbf{x}_i T^2\left(\sum_{j=1}^{m}\omega_j x_{ij}\right)\right]_{\mathrm{pk}^*}\\
k\mathbf{I}_5^{'} &= \left[k\mathbf{x}_i T^3\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) \bmod N\right]_{\mathrm{pk}^*}\\
k\mathbf{I}_6^{'} &= \left[k\mathbf{x}_i y_i T\left(\sum_{j=1}^{m}\omega_j x_{ij}\right)\right]_{\mathrm{pk}^*}
\end{aligned} \tag{11}$$

Step 8: DO blinds above ciphered data with the security parameter k to be $k\mathbf{I}_1' k\mathbf{I}_2'$ in the linear regression model or $k\mathbf{I}_3'$, $k\mathbf{I}_3'$, $k\mathbf{I}_5'$, $k\mathbf{I}_6'$ in the neural network model.

Step 9: DO sends these blinded encrypted results to the KCS.

Step 10: KCS operates Algorithm 6 to convert the blinded encrypted results $k\mathbf{I}_1'$, $k\mathbf{I}_2'$ or $k\mathbf{I}_3'$, $k\mathbf{I}_3'$, $k\mathbf{I}_5'$, $k\mathbf{I}_6'$ to be new results $k\mathbf{I}_1^{'}$ and $k\mathbf{I}_2^{'}$ in the linear regression model or $k\mathbf{I}_3^{'}$, $k\mathbf{I}_4^{'}$ $k\mathbf{I}_5^{'}$, and $k\mathbf{I}_6^{'}$ in the neural network model,

$$\begin{aligned}
k\mathbf{I}_1^* &= k\mathbf{x}_i y_i\\
k\mathbf{I}_2^* &= k\mathbf{x}_i \sum_{j=1}^{m}\omega_j x_{ij}\\
k\mathbf{I}_3^* &= k\mathbf{x}_i y_i T^2\left(\sum_{j=1}^{m}\omega_j x_{ij}\right)\\
k\mathbf{I}_4^* &= k\mathbf{x}_i T^2\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) \quad \omega_j x_{ij}\\
k\mathbf{I}_5^* &= k\mathbf{x}_i T^3\left(\sum_{j=1}^{m}\omega_j x_{ij}\right)\\
k\mathbf{I}_6^* &= k\mathbf{x}_i y_i T\sum_{j=1}^{m}
\end{aligned} \tag{12}$$

which are all encrypted under the TDS's key pk^*

Step 11: subsequently, the KCS sends the above intermediate results $k\mathbf{I}_1^{'}$, $k\mathbf{I}_2^{'}$ or $k\mathbf{I}_3^{'}$, $k\mathbf{I}_4^{'}$, $k\mathbf{I}_5^{'}$, $k\mathbf{I}_6^{'}$ to the TDS.

Step 12: TDS runs Algorithm 5 and gets where

$$\begin{aligned}
\mathbf{I}_{i1} &= [\mathbf{x}_i]_{\mathrm{pk}_i}[y_i]_{\mathrm{pk}_i} = g^{2\mathrm{sk}_i r_i}\mathbf{x}_i y_i \bmod N\\
\mathbf{I}_{i2} &= [\mathbf{x}_i]_{\mathbf{pk}_i} s_i = g^{2\mathrm{sk}_i r_i + \mathrm{sk}_M r_M}\mathbf{x}_i \sum_{j=1}^{d}\omega_j x_{ij} \bmod N\\
\mathbf{I}_{i3} &= [\mathbf{x}_i]_{\mathrm{pk}_i}[y_i]_{\mathrm{pk}_i} s_i^2\\
&= g^{4\mathrm{sk}_i r_i + 2\mathrm{sk}_M r}\mathbf{x}_i y_i\left(\sum_{j=1}^{m}\omega_j x_{ij}\right)^2 \bmod N\\
I_{i4} &= [x_i]_{\mathrm{pk}_i} s_i^2 = g^{3\mathrm{sk}_i r_i + 2\mathrm{sk}_M r_M} x_i T^2\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) \bmod N\\
\mathbf{I}_{i5} &= [\mathbf{x}_i]_{\mathrm{pk}_i} s_i^3\\
&= g^{4\mathrm{sk}_i r_i + 3\mathrm{sk}_M r_M}\mathbf{x}_i T^3\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) \bmod N\\
\mathbf{I}_{i6} &= [\mathbf{x}_i]_{\mathrm{pk}_i}[y_i]_{\mathrm{pk}_i} s_i\\
&= g^{3\mathrm{sk}_i r_i + \mathrm{sk}_M r}\mathbf{x}_i y_i T\left(\sum_{j=1}^{m}\omega_j x_{ij}\right) \bmod N
\end{aligned} \tag{13}$$

Input: the ciphertext $[\mathrm{ms}]_{\mathrm{pk}_o}$ of the message ms with the original public key pk_o, the decryption circuit D_{Π_o} of the original encryption, and the target key pk_*
Output: the ciphertext $[\mathrm{ms}]_{\mathrm{pk}_*}$ of the message ms.
(1) compute: $[\mathrm{ms}]_{\mathrm{pk}_*} \leftarrow \mathrm{Evaluation}\ (\mathrm{pk}_b D_{\Pi_o}\ [\mathrm{ms}]_{\mathrm{pk}_o} [\mathrm{sk}_o]_{\mathrm{pk}_*})$
(2) **end**
(3) **return** $[\mathbf{ms}]_{\mathrm{pk}_*}$

Algorithm 6: Ciphertext conversion.

Input: the update information $\nabla\omega$, the coefficient vector ω, the learning rate η, and the attenuation rate γ
Output: the renew coefficient vector ω'
(1) compute $\omega' = \omega - \eta\nabla\omega$ or $\omega' = \gamma\omega - \eta\nabla\omega$
(2) **end**
(3) return ω'

Algorithm 7: Renewing the coefficient vector.

Initialization: DO's encrypted query feature vector $[q_i]_{\mathrm{pk}_i}$, the corresponding key pair $(\mathrm{pk}_i, \mathrm{sk}_i)$, MO's encrypted coefficient vector $[\omega]_{\mathrm{pk}_M}$, and the corresponding key pair $(\mathrm{pk}_M, \mathrm{sk}_M)$, where $q_i = (q_{i1}q_{i2} \ldots q_{im})$ and $\omega = (\omega_1\omega_2 \ldots \omega_m)$

Target: prediction result pr'
Step 1: DO and MO send $[q_i]_{\mathrm{pk}_i}$ and $[\omega]_{\mathrm{pk}_M}$ to the CS, respectively.
Step 2: CS computes pr, whereas $\mathrm{pr} = \sum_{j=1}^{m} [q_{ij}]_{\mathrm{pk}_i} [\omega_{ij}]_{\mathrm{pk}_M} = \sum_{j=1}^{m} g^{\mathrm{sk}_i r_i + \mathrm{sk}_M r_M} q_{ij} \cdot \omega_{ij} \bmod N$
Step 3: CS sends pr to the MO.
Step 4: MO runs Algorithm 5 and decrypts pr with its key pair $(\mathrm{pk}_M, \mathrm{sk}_M)$ and obtains $\overline{\mathrm{pr}} = \sum_{j=1}^{m} g^{\mathrm{sk}_i r_i} q_{ij}\omega_{ij} \bmod N$
Step 5: MO sends $\overline{\mathrm{pr}}$ to each DO.
Step 6: MO runs Algorithm 5 to decrypt $\overline{\mathrm{pr}}$ with its key pair $(\mathrm{pk}_M, \mathrm{sk}_M)$ and gets access to the desired prediction result: $\mathrm{pr}' = \sum_{j=1}^{m} q_{ij} \cdot \omega_{ij} \bmod N$

Algorithm 8: Subprotocol prediction.

and then TDS makes some simple computations: $k\mathbf{l}_1 = k\mathbf{I}_2^* - k\mathbf{I}_1^* = k(\mathbf{x}_i \sum_{j=1}^{m} \omega_j x_{ij} - \mathbf{x}_i y_i)$ in the linear regression model or $k\mathbf{l}_2 = k\mathbf{I}_3^* + k\mathbf{I}_4^* - k\mathbf{I}_5^* - k\mathbf{I}_6^* = k(\mathbf{x}_i \sum_{j=1}^{m} \omega_j x_{ij} - \mathbf{x}_i y_i)$ in the neural network model to get the final results $k\mathbf{l}_1$ or $k\mathbf{l}_2$

Step 13: TDS sends $k\mathbf{l}_1$ or $k\mathbf{l}_2$ to the MO.

5.1.3. Prediction Phase

In this phase, DO requests prediction with the help of the CS and MO.

Step 14: MO receives $k\mathbf{l}_1$ or $k\mathbf{l}_2$ and removes the security parameter k to obtain different $\nabla\omega$ of each sample $\mathbf{x}_i$

Step 15: then, the MO chooses one gradient descent method and then optimizes its coefficient vector through Algorithm 7

Step 16: each of the DO encrypts a query feature vector $[\mathbf{q}_i]_{\mathrm{pk}_i}$, and the MO encrypts its optimal coefficient vector $[\omega]_{\mathrm{pk}_M}$

Step 17: each of the DO and MO sends $[\mathbf{q}_i]_{\mathrm{pk}_i}$ and $[\omega]_{\mathrm{pk}_M}$ to the CS, respectively.

Step 18: finally, MO, CS, and DO operate together to help the DO to extract the prediction results by operating subprotocol prediction (Algorithm 8).

5.2. OPPGD Scheme over Vertically Partitioned Data. The OPPGD scheme over vertically partitioned data is a little different from the OPPGD scheme over horizontally partitioned data. After receiving $[\mathbf{x}_i]_{\mathrm{pk}_i}$, $[y_i]_{\mathrm{pk}_i}$, and $[\omega]_{\mathrm{pk}}$, CS executes Algorithm 4 n times in the first scheme, whereas CS operates Algorithm 4 nm times in the second scheme. This is because one record's m attributes are sent to the CS by its DO,

respectively. In addition, when the KCS receives the blinded encrypted results, it needs to add blinded encrypted results together m times to get the inner product of a record and the coefficient vector. For simplicity, we omit the same steps of the OPPGD scheme over vertically partitioned data as the steps of the OPPGD scheme over horizontally partitioned data.

5.3. Scheme Correctness. Now, we prove the correctness of our proposed OPPGD scheme over horizontally partitioned data. The correctness of the other scheme can be verified in a similar manner.

Theorem 1. *MO can correctly obtain $\nabla\omega$ to update its coefficient vector.*

Proof. After receiving $[\mathbf{x}_i]_{\text{pk}_i}$, $[y_i]_{\text{pk}_i}$, and $[\omega]_{\text{pk}}$, CS computes an encrypted scalar product $\mathbf{S}$, where $s_i = \sum_{j=1}^{m} g^{\text{sk}_i r_i + \text{sk}_M r} \omega_j x_{ij} \bmod N$. For linear regression, CS calculates $\mathbf{I}_1$ and $\mathbf{I}_2$, whereas for the neural network, CS calculates $\mathbf{I}_3$, $\mathbf{I}_4$, $\mathbf{I}_5$, and $\mathbf{I}_6$. After receiving the encrypted results from the CS, each DO decrypts the message sent from the CS and obtains $\mathbf{I}_1'$ and $\mathbf{I}_2'$ or $\mathbf{I}_3'$, $\mathbf{I}_4'$, $\mathbf{I}_5'$, and $\mathbf{I}_6'$ in linear regression or the neural network, respectively. Then, it blinds these encrypted results with k to be $k\mathbf{I}_1'$ and $k\mathbf{I}_2'$ or $k\mathbf{I}_3'$, $k\mathbf{I}_4'$, $k\mathbf{I}_5'$, and $k\mathbf{I}_6'$ and sends them to the KCS. Consequently, KCS converts the ciphertext into $k\mathbf{I}_1'$ and $k\mathbf{I}_2'$ or $k\mathbf{I}_3'$, $k\mathbf{I}_4'$, $k\mathbf{I}_5'$, and $k\mathbf{I}_6'$ under the key pk^* of the TDS. TDS decrypts the above intermediate results through Algorithm 5 to produce $k\mathbf{I}_1^*$ and $k\mathbf{I}_2^*$ or $k\mathbf{I}_3^*$, $k\mathbf{I}_4^*$, $k\mathbf{I}_5^*$, and $k\mathbf{I}_6^*$. Then, it computes $k\mathbf{I}_2^* - k\mathbf{I}_1^*$ or $k\mathbf{l}_2 = k\mathbf{I}_3^* + k\mathbf{I}_4^* - k\mathbf{I}_5^* - k\mathbf{I}_6^*$ and generates the final results $k\mathbf{l}_1$ or $k\mathbf{l}_2$ for linear regression or the neural network. Ultimately, after the MO receives them, he removes the security parameter k and obtains $\nabla\omega = \mathbf{x}_i \sum_{j=1}^{m} \omega_j x_{ij} - \mathbf{x}_i y_i$ in linear regression or $\nabla\omega = (12)\alpha(\mathbf{x}_i\ y_i T^2(\sum_{j=1}^{m} \omega_j x_{ij}) + \mathbf{x}_i T^2(\sum_{j=1}^{m} \omega_j x_{ij}) - \mathbf{x}_i T^3(\sum_{j=1}^{m} \omega_j x_{ij}) - \mathbf{x}_i y_i T(\sum_{j=1}^{m} \omega_j x_{ij}))$ in the neural network which are equal to equation (3) or equation (5), respectively. Then, MO can achieve accurate $\nabla\omega$ □

6. Privacy and Complexity Analysis

We will analyze the privacy, computational cost, and communication overhead of the OPPGD scheme over horizontally partitioned data. We can perform analysis of the OPPGD scheme over vertically partitioned data in terms of the privacy, computational cost, and communication overhead in almost the same way. For simplicity, we omit the latter.

6.1. Privacy Analysis. According to the definitions of two different privacy levels in Section 4.4, we conduct the privacy analysis of our proposed OPPGD scheme over horizontally partitioned data.

Proof. Upon the hardness assumption of the Diffie–Hellman problem, our proposed OPPGD schemes achieve level-1 privacy against any probabilistic polynomial-time adversary. □

Proof. Now, we show that our scheme can preserve MO's model privacy and DO's data privacy.

In Step 3 of Algorithm 3, MO and DO hide their input via Elgamal encryption. After receiving $[\mathbf{x}_i]_{\text{pk}_i}$, $[y_i]_{\text{pk}_i}$, and $[\omega]_{\text{pk}}$, the CS runs Algorithm 4 and obtains the encrypted scalar product $\mathbf{S}$. Especially, MO's and every DO's encrypted input are $g^{\text{sk}_M r_M}\omega \bmod N$ and $\{g^{\text{sk}_i r_i}\mathbf{x}_i \bmod N\, g^{\text{sk}_i r_i} y_i \bmod N\}_{i=1}^{n}$. Upon the hardness assumption of the Diffie–Hellman problem, although CS knows MO and DO's public keys $g^{\text{sk}_i} \bmod N$ and $g^{\text{sk}_r} \bmod N$, it is still impossible for them to acquire their secret keys sk_i and sk. Since the randomness r_i and r are chosen by DO and MO, respectively, any adversary who attempts to solve $\{g^{\text{sk}_i r_i} y_i \bmod N,\ g^{\text{sk}_M r_M}\omega \bmod N\}$ from the public keys $\{g^{r_i} \bmod N,\ g^{r_M} \bmod N\}$ will have to be faced with two instances of Diffie–Hellman problems. Thus, DO's $\mathbf{x}_i$ and y_i and MO's ω will not be exposed to other parties. When the KCS performs Algorithm 6 to convert the encrypted results $\{\mathbf{I}_1'\mathbf{I}_2', \mathbf{I}_3'\mathbf{I}_4', \mathbf{I}_5', \mathbf{I}_6'\}$, it receives MO and DO's secret keys encrypted under the TDS's public key. However, TDS is a trusty decryption server, so KCS cannot obtain TDS's secret key, which means KCS knows nothing about MO and DO's secret keys and their private value. So, the encrypted results $\{k\mathbf{I}_1'k\mathbf{I}_2'k\mathbf{I}_3'k\mathbf{I}_4'k\mathbf{I}_5'k\mathbf{I}_6'\}$ cannot leak any secret information. Next, TDS runs Algorithm 5 and obtains encrypted $\nabla\omega$. However, without the secret value k, TDS cannot obtain $\nabla\omega$. Hence, MO's model parameters will not be exposed.

Since MO's coefficient vector, gradient $\nabla\omega$, and DO's data will not face the privacy problem, our OPPGD schemes can provide level-1 privacy. □

Theorem 3. *Upon the hardness assumption of knapsack problems, our OPPGD schemes can provide level-2 privacy against any probabilistic polynomial-time adversary.*

Proof. After receiving the encrypted results from the CS, DOs run Algorithm 5 to generate new encrypted results under MO's key. For linear regression, DO knows $\{\mathbf{I}_2', g^{\text{sk}_r} \bmod N, \mathbf{x}_i y_i\}$. For neural networks, DO knows $\{\mathbf{I}_3'\mathbf{I}_4', \mathbf{I}_5', \mathbf{I}_6', g^{\text{sk}_r} \bmod N, \mathbf{x}_i y_i\}$. However, with the knowledge of the information, it is still impossible to acquire ω. This is because that the knapsack problem is assumed to be difficult: given a scalar product z and a vector $\mathbf{a}$, it is hard to find vector $\mathbf{b}$ that satisfies $z = \mathbf{ab}$

Consequently, MO's coefficient vector and gradient results $\nabla\omega$ cannot be deduced from the intermediate results all over the scheme.

Therefore, we conclude that our schemes can achieve level-2 privacy. □

6.2. Theoretical Efficiency Analysis. Now, we carry out the theoretical efficiency analysis of the schemes. We consider the situation for linear regression. Assume that the MO chooses the SGD method to update its coefficient vector within one epoch. In essence, MO optimizes its coefficients within several epochs. In the following, we analyze the feasibility of our proposed schemes in detail in terms of

Table 1: Complexity cost of the proposed OPPGD scheme.

			Phase			
			Preparation	Training	Prediction	Total
Computation cost	DO	Multiplications	$\mathcal{O}(\mathrm{mn}+n)$	$\mathcal{O}(3\mathrm{mn})$	$\mathcal{O}(2m)$	$\mathcal{O}(4\mathrm{mn}+n+2m)$
	MO	Multiplications	$\mathcal{O}(m)$	$\mathcal{O}(\mathrm{mn})$	$\mathcal{O}(m+1)$	$\mathcal{O}(2m+\mathrm{mn}+1)$
		Additions		$\mathcal{O}(m)$		$\mathcal{O}(m)$
	CS	Multiplications		$\mathcal{O}(3\mathrm{mn})$	$\mathcal{O}(m+1)$	$\mathcal{O}(3\mathrm{mn}+m+1))$
		Additions		$\mathcal{O}(\mathrm{mn}-n)$	$\mathcal{O}(m-1)$	$\mathcal{O}(\mathrm{mn}+m+n-1)$
	KCS	Multiplications		$\mathcal{O}(2\mathrm{mn})$		$\mathcal{O}(2\mathrm{mn})$
	TDS	Multiplications		$\mathcal{O}(2\mathrm{mn})$		$\mathcal{O}(2\mathrm{mn})$
		Additions		$\mathcal{O}(\mathrm{mn})$		$\mathcal{O}(\mathrm{mn})$
Communication overhead	Total round		$\mathcal{O}(n+1)$	$\mathcal{O}(2n+2)$	$\mathcal{O}(4)$	$\mathcal{O}(3n+7)$
	Total bits to transmit		$\mathcal{O}((\mathrm{mn}+2m)\lvert N\rvert)$	$\mathcal{O}(6\mathrm{nm}\lvert N\rvert)$	$\mathcal{O}(4\mathrm{nm}\lvert N\rvert)$	$\mathcal{O}((11\mathrm{nm}+2m)\lvert N\rvert)$

Table 2: Running time of the OPPGD scheme with the dataset of n tuples.

Tuples	KeyGen (ms)	Encryption (ms)	Training (ms)	Prediction (ms)
1000	279.47	293.07	31146.57	0.06
2000	559.03	585.97	31338.61	0.06
3000	838.42	878.63	31507.91	0.06
4000	1117.97	1172.11	31677.20	0.06
5000	1397.61	1464.20	31846.51	0.06
6000	1677.09	1756.98	32015.80	0.06

computational cost and communication overhead. Both computational cost and communication overhead are shown in Table 1

6.2.1. Computational Cost. Assume that the dataset contains n records, each of which has m attributes, and one class value in the OPPGD scheme over horizontally partitioned data. In Step 3, DOs and MO operate Algorithm 3 $\mathcal{O}(\mathrm{mn}+n)$ and $\mathcal{O}(m)$ times, respectively. Thus, $\mathcal{O}(\mathrm{mn}+n+m)$ multiplications are required. In Step 5, CS performs OPPSP to calculate encrypted scalar product $\mathbf{S}$. It requires 3mn multiplications and $\mathcal{O}(n(m-1))$ additions. In Step 7, DO performs nm decryptions to generate the encrypted results which are under the MO's key. In Step 8, DO needs $\mathcal{O}(2\mathrm{mn})$ multiplications to blind encrypted results with the security parameter K. In Step 10, KCS performs $\mathcal{O}(2\mathrm{mn})$ multiplications to convert the encrypted results into new results. In Step 12, TDS performs $\mathcal{O}(2\mathrm{mn})$ decryptions and makes mn subtractions to obtain the final results. In Step 14, MO performs $\mathcal{O}(\mathrm{mn})$ multiplications and obtains $\nabla\omega$. MO operates SGD to update its coefficient vector by executing $\mathcal{O}(m)$ times of multiplications and additions. In Step 16, both DO and MO perform $\mathcal{O}(m)$ encryption operations to encrypt their query and the optimal coefficient vector. In Step 18, in order to generate prediction results, CS performs $\mathcal{O}(\mathrm{mn})$ multiplication and $\mathcal{O}(m-1)$ additions, while both DO and MO perform one encryption operation, respectively.

6.2.2. Communication Overhead. Next, we analyze the communication complexity of each entity in our proposed schemes. In Step 4, DO and MO communicate with CS n rounds and one round, respectively. It takes

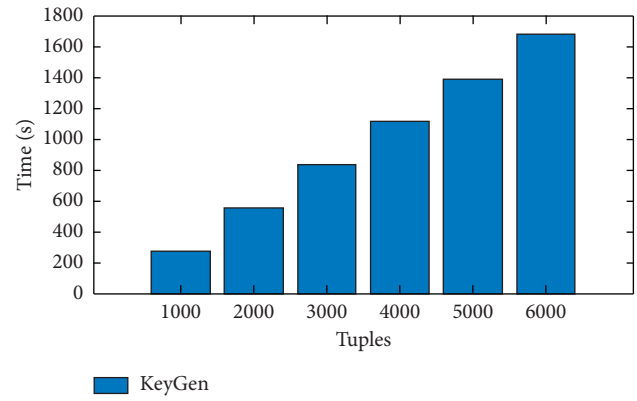


Figure 4: The running time in the KeyGen algorithm.

$\mathcal{O}((2\mathrm{nm}+2m)\lvert N\rvert)$ bits to transmit. In Step 6, since CS sends the encrypted results to DOs, its communication overhead required is $\mathcal{O}(n)$. So, it requires $\mathcal{O}(2\mathrm{nm}\lvert N\rvert)$ bits. Moreover, in Step 9, when each DO sends blinded encryption results to the KCS, the communication overhead is $\mathcal{O}(n)$. Thus, $\mathcal{O}(2\mathrm{nm}\lvert N\rvert)$ bits are required to be transmitted. In Step 11, KCS sends new intermediate results to TDS via $\mathcal{O}(1)$ round with $\mathcal{O}(\mathrm{nm}\lvert N\rvert)$ bits. In Step 13, the communication overhead between TDS and MO is $\mathcal{O}(1)$. It costs $\mathcal{O}(\mathrm{nm}\lvert N\rvert)$ bits to transmit. In Step 17, DO and MO send the encrypted feature vectors to CS, respectively, with the communication overhead of $\mathcal{O}(2)$ which costs $\mathcal{O}(2\mathrm{nm}\lvert N\rvert)$ to transmit. In Step 18, $\mathcal{O}(2)$ communication cost is required for the DO to obtain its desired prediction results, while $\mathcal{O}(2\mathrm{nm}\lvert N\rvert)$ bits are transmitted. Hence, the communication cost of the scheme is $\mathcal{O}(3n+7)$ in total.

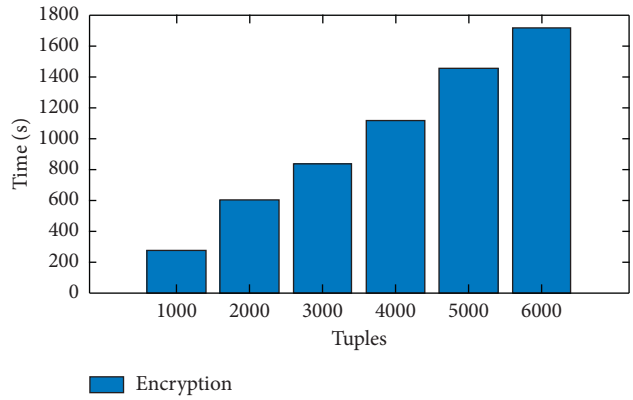


Figure 5: The running time in the encryption algorithm.

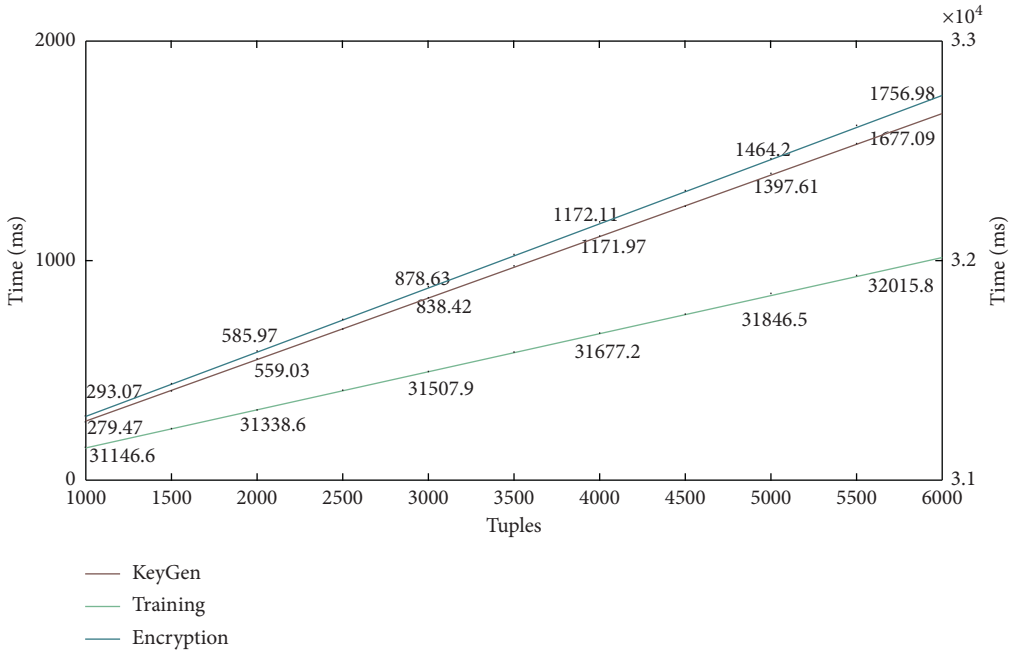


Figure 6: The running time in the KeyGen algorithm, the encryption algorithm, and the training phase.

Table 3: The running time of the key generation algorithm and the evaluation algorithm based on the key dimension.

Key bit-length	KeyGen (s)	Evaluation (s)	Training (s)
2048 bits	40	31	31.8
8192 bits	480	180	180.8

7. Performance Evaluation

In this section, we evaluate the efficiency of the OPPGD scheme over horizontally partitioned data by using a custom simulator built in JAVA. The running time of the OPPGD scheme over vertically partitioned data can be evaluated in a similar way. The scenario we focus on in our paper is the data are partitioned among multiple data owners, and the target function is owned by the model owner. The model owner can not only train its model over data owner's data but also provide users with predictions. To the best of our knowledge, no other prior work in the literature discusses this scenario. So, we present detailed performance evaluation of our schemes rather than comparing them to previous works.

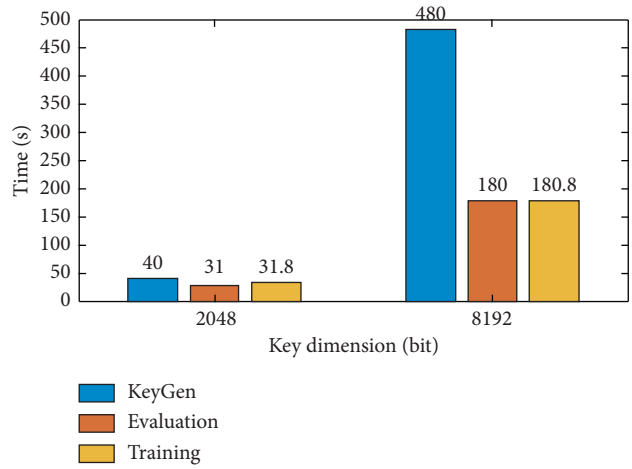


Figure 7: The running time of the key generation algorithm and the evaluation algorithm based on the key dimension.

Table 4: The total time cost of each entity in the OPPGD scheme.

Tuples	DO (ms)	MO (ms)	CS (ms)	KCS (ms)	TDS (ms)
1000	52.48	13.09	41.95	26.09	28.43
2000	104.93	26.13	83.89	52.19	57.67
3000	157.38	39.18	125.66	78.28	86.5
4000	209.83	52.23	167.76	104.38	115.34
5000	262.28	65.27	209.43	130.47	144.17
6000	314.73	78.33	251.31	156.57	173.01

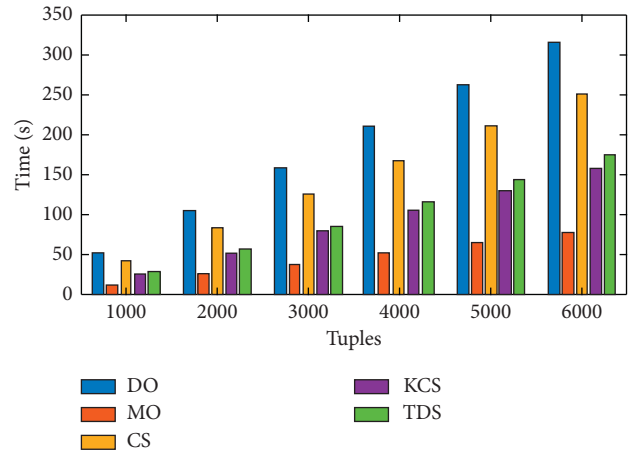


Figure 8: The running time of each entity in the OPPGD scheme with different key bit-lengths.

There are five entities in the scheme: the model owner MO, the data owner DO, the cloud server CS, the key conversion server KCS, and the trusty decryption server TDS.

We run the data owners DOs and the model owner MO on a laptop with Intel Xeon(R) E5-1620 3.50 GHZ CPU processor and 16 GB RAM memory. The cloud server CS, the key conversion server KCS, and the trusty decryption server TDS sides are operated on a computer with Intel(R) Core (TM) i7-4770 3.40 GHz CPU processor and16 GB RAM memory.

In our experiments, DO's data X are represented as one $n * m$ matrix, where n ranges from 1000 to 6000 and $m = 50$. We evaluate the computational efficiency of our OPPGD schemes without considering communication latency. We simulate four stages: the KeyGen algorithm, the encryption algorithm, the training phase, and the prediction phase. As the data size n changes, the corresponding time cost is also different. When the key bit-length is 2048 bits, the running time of each stage of the schemes with the number of data tuples can be seen from Table 2. The calculation of the OPPGD scheme is mainly in the training stage, while the calculation cost of the rest stages is very low. We use the histogram to explicitly present the running time in the KeyGen algorithm and the encryption algorithm in Figures 4

and 5. The running time in the KeyGen algorithm, the encryption algorithm, and the training phase is shown in Figure 6. In addition, when the data dimension is 6000, the running time mainly verified in the KeyGen algorithm, the evaluation algorithm, and the training phase based on various key bit-lengths is different. So, we simulate these stages and the running time, as shown in Table 3 and Figure 7. When the key bit-length is 2048 bits, the total running time of each entity in our OPPGD scheme is shown in our Table 4. According to the variation of the tuples or key bit-lengths, the running time of each party is shown in Figure 8

8. Conclusion

Massive work on the protection of sensitive data of IoT devices is based on the secure communication channels and authorization. In our paper, we focus on the protection of data which are collected by the IoT devices, stored, and calculated on the cloud end and the privacy of the machine learning model which is held by the MO. Gradient descent methods are employed comprehensively to train a machine learning model in the cloud computing environment. In order to preserve data privacy and model privacy during the cloud computing, we propose two secure schemes to perform outsourced privacy-preserving gradient descent methods over a horizontally or vertically distributed dataset. The proposed schemes enable the model owner (MO) to train its learning model and obtain the optimal coefficient vector based on the dataset owned by the DO with the help of CS, TDS, and KCS. After the MO improves its model, it can offer prediction service to the DO. Both the privacy of the MO's model and DO's dataset can be protected. Complexity and performance evaluation are also given in detail. In the future work, we will try to optimize our system to reduce the number of entities.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Ministry of Science and Technology of the People's Republic of China (Grant no. 2018YFB0803505), the National Natural Science Foundation of China (Grant nos. 61862028 and 61702238), the Natural Science Foundation of Jiangxi Province (Grant no. 20181BAB202016), and the Science and Technology Project of Provincial Education Department of Jiangxi (GJJ160430 and GJJ180288).

References

[1] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?" *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41–49, 2018.

[2] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for internet of things," *Journal of Network and Computer Applications*, vol. 42, pp. 120–134, 2014.

[3] I. Andrea, C. Chrysostomou, and G. C. Hadjichristofi, "Internet of things: security vulnerabilities and challenges," 2015.

[4] N. Kshetri, "Can blockchain strengthen the internet of things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.

[5] M. T. Thai, "A disruptive integration," *Computer*, vol. 51, pp. 48–53, 2018.

[6] L. Chen, S. Thombre, K. Järvinen et al., "Robustness, Security and privacy in location-based services for future IoT: a Survey," *IEEE Access*, vol. 5, pp. 8956–8977, 2017.

[7] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. M. A. Kazmi, and C. S. Hong, "Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges," *Future Generation Computer Systems*, vol. 92, pp. 265–275, 2019.

[8] J. Daubert, A. Wiesmaier, and P. Kikiras, "A view on privacy & trust in IoT," 2015.

[9] Z. Zhang, M. C. Y. Cho, C. Wang, C. Hsu, C. Chen, and S. Shieh, "IoT security: ongoing challenges and research opportunities," 2014.

[10] P. P. Jayaraman, X. Yang, A. Yavari, D. Georgakopoulos, and X. Yi, "Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation," *Future Generation Computer Systems*, vol. 76, pp. 540–549, 2017.

[11] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[12] L. Wan, W. K. Ng, S. Han, and V. C. S. Lee, "Privacy-preservation for gradient descent methods," 2007.

[13] S. Han, W. K. Ng, L. Wan, and V. C. S. Lee, "Privacy-preserving gradient-descent methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 6, pp. 884–899, 2010.

[14] G. Danner and M. Jelasity, "Fully Distributed Privacy preserving mini-batch gradient descent learning," *IFIP International Conference on Distributed Applications and Interoperable Systems*, vol. 9038, pp. 30–44, 2015.

[15] I. Hegedűs and M. Jelasity, "Distributed differentially private stochastic gradient descent: an empirical study," 2016.

[16] S. Mehnaz, G. Bellala, and E. Bertino, "A secure sum protocol and its application to privacy-preserving multi-party analytics," 2017.

[17] S. Mehnaz and E. Bertino, "Privacy-preserving multi-party analytics over arbitrarily partitioned Data," 2017.

[18] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton, "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics," 2017.

[19] F. Liu, W. K. Ng, and W. Zhang, "Encrypted gradient descent protocol for outsourced data mining," 2015.

[20] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," 2015.

[21] M. Kim, Y. Song, S. Wang, Y. Xia, and X. Jiang, "Secure logistic regression based on homomorphic encryption: Design and Evaluation," *JMIR Medical Informatics*, vol. 6, no. 2, p. 19, 2018.

[22] F. Gonzalez-Serrano, A. Amor-Martin, and J. Casamayón-Antón, "Supervised machine learning using encrypted training data," *International Journal of Information Security*, vol. 7, no. 2, pp. 365–377, 2017.

[23] P. Mohassel and Y. Zhang, "SecureML: a system for scalable privacy-preserving machine learning," 2017.

[24] P. Li, J. Li, Z. Huang et al., "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.

[25] X. Ma, X. Chen, and X. Zhang, "Non-interactive privacy-preserving neural network prediction," *Information Sciences*, vol. 481, pp. 507–519, 2019.

[26] X. Liu, B. Qin, R. H. Deng, and Y. Li, "An efficient privacy-preserving outsourced computation over public data," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 756–770, 2017.

[27] X. Liu, R. Deng, K. R. Choo, Y. Yang, and H. Pang, "Privacy-Preserving outsourced calculation toolkit in the cloud," *IEEE Transactions on Dependable and Secure Computing*, vol. 2018, p. 1, 2018.

[28] M. Rady, T. Abdelkader, and R. Ismail, "Integrity and confidentiality in cloud outsourced data," *Ain Shams Engineering Journal*, vol. 10, no. 2, pp. 275–285, 2019.

[29] X. Yu, R. Zhang, and Z. Rui, "Verifiable outsourced computation over encrypted data," *Information Sciences*, vol. 479, pp. 372–385, 2019.

[30] M. A. P. Chamikara, P. Bertok, D. Liu, S. Camtepe, and I. Khalil, "Efficient privacy preservation of big data for accurate data mining," *Information Sciences*, vol. 2019, 2019.

[31] P. Li, J. Li, Z. Huang, C. Gao, W. Chen, and K. Chen, "Privacy-preserving outsourced classification in cloud computing," *Cluster Computing*, vol. 21, no. 1, pp. 277–286, 2018.

[32] T. Li, Z. Huang, P. Li, Z. Liu, and C. Jia, "Outsourced privacy-preserving classification service over encrypted data," *Journal of Network and Computer Applications*, vol. 106, pp. 100–110, 2018.

[33] H. Park, P. Kim, H. Kim, K.-W. Park, and Y. Lee, "Efficient machine learning over encrypted data with non-interactive communication," *Computer Standards & Interfaces*, vol. 58, pp. 87–108, 2018.

[34] Y. Li, Z. L. Jiang, L. Yao, X. Wang, S. M. Yiu, and Z. Huang, "Outsourced privacy-preserving C4.5 decision tree algorithm over horizontally and vertically partitioned dataset among multiple parties," *Cluster Computing*, vol. 22, pp. 1581–1593, 2019.

[35] H. Rong, H. Wang, J. Liu, F. Tang, and M. Xian, "Verifiable and privacy-preserving association rule mining in hybrid cloud environment," 2019.

[36] Y. Li, Z. L. Jiang, X. Wang, and S. M. Yiu, "Privacy-preserving ID3 data mining over encrypted data in outsourced environments with multiple keys," *IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, vol. 1, pp. 548–555, 2017.

[37] W. Xue, Y. Shen, C. Luo, W. Hu, and A. Seneviratne, "A privacy-preserving system for edge-based classification," 2018.

[38] J.-J. Yang, J.-Q. Li, and Y. Niu, "A hybrid solution for privacy preserving medical data sharing in the cloud environment," *Future Generation Computer Systems*, vol. 43, pp. 74–86, 2015.

[39] H. Kaur, N. Kumar, and S. Batra, "An efficient multi-party scheme for privacy preserving collaborative filtering for healthcare recommender systemficient Multi-party Scheme for Privacy Preserving Collaborative Filtering for Healthcare Recommender System," *Future Generation Computer Systems*, vol. 86, pp. 297–307, 2018.

[40] R. Gennaro, C. Gentry, B. Parno et al., "Non-interactive verifiable computing: outsourcing computation to untrusted workers," in *Proceedings of the 30th Annual Cryptology Conference Advances in Cryptology-CRYPTO 2010*, Lecture Notes in Computer Science, vol. 6223, pp. 465–482, Springer, Berlin, Germany, 2010, https://www.iacr.org/archive/crypto2010/62230459/62230459.pdf.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 8820082, 19 pages
https://doi.org/10.1155/2021/8820082

Research Article

Survey on Reversible Watermarking Techniques of Echocardiography

Rabiya Ghafoor,[1] Danish Saleem,[1] Sajjad Shaukat Jamal,[2] M. Ishtiaq,[1] Sadaf Ejaz,[1] Arif Jamal Malik,[1] and M. Fahad Khan[1]

[1]*Department of Software Engineering, Foundation University Islamabad, Islamabad 46000, Pakistan*
[2]*Department of Mathematics, College of Science, King Khalid University, Abha, Saudi Arabia*

Correspondence should be addressed to M. Ishtiaq; imishtiaq@gmail.com

Received 3 September 2020; Revised 6 November 2020; Accepted 30 December 2020; Published 26 March 2021

Academic Editor: Emanuele Maiorana

In critical domains such as medical and military, reversible watermarking (RW) has been used. In the medical domain, different modalities are used to store patient information. The current study focuses on the application of RW in echocardiography data. Mostly, RW is applied to protect patient data without affecting the quality of the decoded image. The RW methods are benchmarked as per imperceptibility, robustness, and payload. The survey presents a comparison of state-of-the-art RW techniques. The imperceptibility and payload are balanced through a tradeoff. It has been observed in the literature that most of the RW methods lack robustness, and very small-scale robustness has been achieved in this domain of watermarking. Different types of RW, i.e., fragile, semifragile, and robust methods, are being compared and reviewed. Mostly, fragile methods are developed on the error-expansion techniques built on histogram shifting-based approach. In this study, several RW methods are compared and the results are presented.

1. Introduction

Medical imaging (such as echocardiography and ultrasonography) holds vital details about the patient. Echocardiography with its different flavors is eminent for rendering information regarding the size and shape of the heart, identification of cardiological risks, pumping capacity, and location and severity of tissue damage [1–3]. The information comprised by these modalities is outstandingly critical that makes it very challenging for video processing and image processing. It ultimately results in a strong urge for lossless and robust data hiding for such data processing [4, 5].

Continuous advancements in technology have done a marvelous job transforming every facet of human life, and the quest for further improvement and technological advancement is perpetual. Computer technology, with all its power, has shrunk into palms with the advent of smartphones and other handheld devices. This has completely revolutionized the communication and data processing domains with an ever-increasing data transfer volume. In the information technology jargon, this transfer could be broken into Store, Retrieve, and Transmit.

One of the biggest beneficiaries of this transformation is the medical domain which has benefited tremendously over the years with the innovations in information technology. Health information technology (HIT) has gained popularity, and many health institutes are exploiting information technology for storing patient's information. This electronic medical record (EMR) holds complete information about a patient's current and past ailments, laboratory test data, medical imaging results, diagnosis, and drug information, etc. All these data can be transformed into a single database that can be stored, accessed, and transmitted to other medical institutions as per requirement. This encourages a paperless environment and enables different stakeholders to access complete information about the patient with a single mouse click.

This digitization of medical records potentially poses a threat to the data integrity from both intentional or unintentional changes and misuse which has increased the tendency of attacks on digital data [6]. Manipulation, unauthorized access, deletion, and distribution of medical data are increasing [7, 8]. Patient data theft crimes are on the rise, and this can have a severely adverse effect on the efficacy of telemedicine and E-health applications [9, 10].

Although techniques like DICOM (Digital Imaging and Communications in Medicine) mitigate many of these risks that arise during transmission of digitized medical data, the overhead of extra information makes it a lesser candidate if the same is to be transmitted over public networks or low bandwidth connections [10].

All the data protection techniques such as cryptography, steganography, and watermarking, etc., have their own advantages, but for authentication of medical data [11], watermarking has been the most popular and robust technique adopted [12]. A digital identification mark or message is embedded into the cover medium, i.e., the digital medical data like images, videos, text, etc. This not only secures the data authenticity but also nullifies any ownership disputes [8, 12, 13]. When used in conjunction with cryptographic techniques, digital watermarking provides a highly resolute solution against data tampering [10]. Another perk is that it is a well-established research area and the application spans over various prominent areas such as broadcast monitoring. Therefore, a lot of research onus has been towards the medical image and video watermarking to secure the communication of highly confidential patient data. Modern data hiding techniques and their application classification are shown in Figure 1. A general framework of the digital watermarking technique for medical images is presented in Figures 2(a) and 2(b). The data embedding process requires a cover image, digital watermark, and a noncompulsory secure key. The output of this embedding function will be a watermarked image that can be transmitted or stored as per requirement. When the watermark is extracted, the encoding process is generally performed in a reverse order to extract the embedded data. These data are subsequently used to authenticate the original content and for tamper detection [15, 16].

An efficient watermarking system must ensure certain attributes that are pivotal in ensuring data security. These parameters or attributes are generally conflicting and require a certain degree of compromise made between them by the watermarking algorithm [9]. The four major characteristics are capacity, security, perceptibility, and robustness. In this paper, we aim to review reversible data hiding techniques which are being utilized for hiding information in echocardiography (ultrasound) for the security of healthcare systems.

2. Literature Review

In 2019, Kaw et al. [17] submitted an RDH scheme using optimal pixel repetition for securely hiding information. It results in a histogram invariant stego image. High robustness is achieved from statistical attacks due to histogram invariance. PSNR acquired after experiments is 42 dB for 1.25 bpp payload by applying the scheme for X-ray.

In 2018, Geetha and Geetha [18] proposed a scheme for hiding data utilizing expansion on prediction error on image points obtained by integer-integer wavelet transform (I-IWT). Subbands are used in this technique for employing I-IWT, and embedding of secret bits is done on subbands. 52.6 PSNR is obtained for 75285 bpp payload for an X-ray image.

In 2020, spatial domain-based method was proposed by Dutta et al. [19] for watermarking. By exploiting spatial redundancy of vastly correlated pixels, this technique achieved robustness and visual quality. Accurate reversibility is demonstrated even in error-prone transmission channels. Parah et al. [20] purposed a reversible and high capacity system that utilized the pixel repetition technique. The proposed system has the capability of detecting tampering by using Block Checksum Embedding. Experiments were carried out on CT scans and witnessed an average of 46 dB PSNR as an output for a 1,96,608-bit payload.

Bhardwaj and Aggarwal [21] proposed an encrypted RDH algorithm based on hierarchal absolute moment block truncation coding. Hierarchical AMBTC is used to obtain a table having a high mean, low mean table, and bitmap sequence table. Encryption is done at a first place through a homomorphic-based cryptosystem, and then, every grey pixel having a high mean table is utilized for data embedding. In addition to that, each pixel having tables with low mean and bitmap sequence (except zero (0) value) is utilized in the embedding process without any underflow or overflow issue. Investigational outcomes showed that the proposed method was performed possessing a peak embedding rate of 1.80 bits per pixel and 49.89 dB PSNR value. Geetha and Geetha presented the RDH technique with high embedding capacity without degrading the quality of the cover image [22].

Interpolated points of the cover image are predicted by employing the Rhombus Mean Interpolation technique. For the discovery of tampering and authenticity of the content, the patient's data as secret information along with checksum for the nonoverlapping of 2×2 pixel block are inserted. The results for the CT scan depicted that 42.398 dB PSNR achieved 1.5 bpp payload.

In 2014, Acharjee et al. [4] proffered the technique of watermarking for echocardiograph video. After computing the motion vector of two adjacent frames, the image is watermarked. Results showed SSIM between the original and restored frames (2–6) was between 0.9902 and 0.9943 whereas PSNR of the original and recovered frames for the same number of frames was between 163.95 dB and 165.23 dB.

Dey et al. proposed a DWT-DCT-SVD-based technique for watermarking intravascular ultrasound video [23]. Watermark embedding is achieved by splitting IVUS video into frames and with the employment of discrete wavelet transformation (DWT) along with discrete cosine transformation (DCT) followed by singular value decomposition

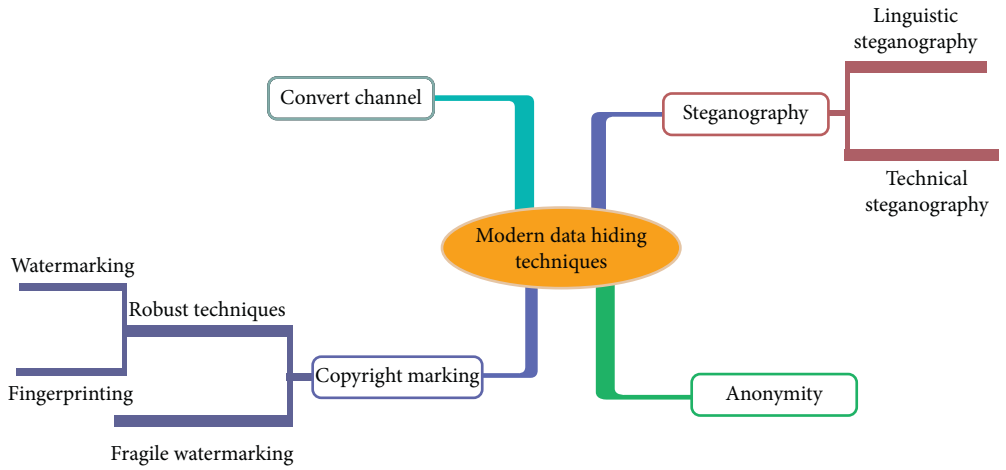


Figure 1: Classification of modern data hiding techniques [14].

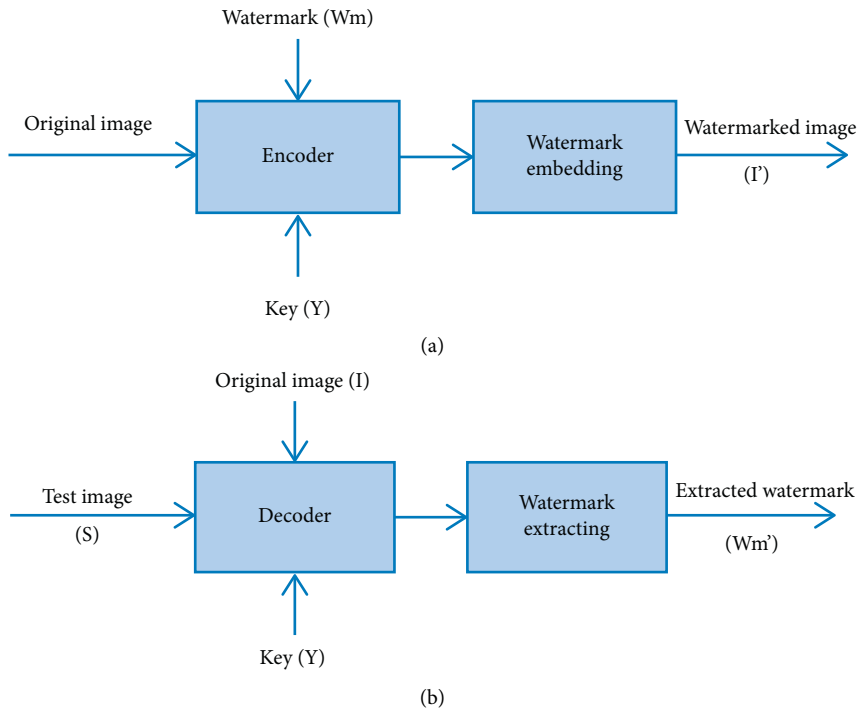


Figure 2: Watermarking. (a) Embedding and (b) extraction process [15].

(SVD). The extraction process involved the application of inverse of both DWT and DCT along with SVD. The assessment of the proffered technique is done through PSNR along with the correlation coefficient. 50.8122 PSNR achieved between the original IVUS video and watermarked video, and the correlation coefficient was 0.9990 between the original and watermarked video. The reversible watermarking technique was presented by Fadoua and Hamid [24] for the security of medical video. Patient identity is embedded into the host video, for that matter polynomial transform methods were utilized for the identification of optimal frames in each video shot. After experimentation average, PSNR and NC obtained are 61.4394 and 1.0000, respectively, for the frame size 120×120.

3. Methodologies

In this section, three types of watermarking techniques are discussed, i.e., spatial domain, frequency domain, and hybrid domain methods. These methods are mostly classified into fragile, semifragile, and robust techniques. In Table 1, the methods are being summarized.

3.1. Spatial Domain

3.1.1. Watermarking with Tamper Detection and Retrieval Using Lossless Watermarking with LSB Modification. In 2012, Agung et al. proposed a reversible technique based on

Table 1: Different types of watermarking.

Watermarking type	Characteristics
Fragile	The fragile methods are mostly used in application where content authentication is desired. Detection of forgeries/tempers, unauthorized processing, and transmission is carried out using fragile methods of watermarking.
Semifragile	Application which has privacy constraints along with content authentication uses semifragile methods.
Robust	The method of watermarking is used in situation where protection of copyright and digital rights management is desired. These kinds of watermarks survive malicious/nonmalicious attacks on the marked media.

LSB modification for watermarking the medical images [25]. Liew's design-based technique [26] used in this paper in terms of original LSBs is embedded in the region of interest by utilizing run-length encoding. Jasni's scheme [27] for temper detection and watermark recovery is also the base of this technique. Synopsis of the technique is presented as follows.

(a) Image preparation: for experimentation, images of ultrasound (8 bit) with a resolution of 640×480 pixels are used. The image is segregated into RONI and ROI. For temper detection and recovery, ROI is utilized for watermarking. RONI is utilized for embedding the original LSBs for the reversibility of the watermark. For generalizing, the static sizes of samples of ROI along with RONI are used as demonstrated.

The red rectangular region depicts the ROI in Figure 3. For achieving the best precision of temper localization and better quality of the recovered image, smaller block size of 6×6 is used instead of 8×8. A mapping ($A \longrightarrow B \longrightarrow C \longrightarrow D \longrightarrow \ldots \longrightarrow A$) sequence prepared for subjecting ROI to embed watermark is one to one block mapping, and a specific block is denoted by each symbol. Information for the reconstruction of every previous block is injected in the adjacent next block [27].

$$\overrightarrow{B} = [(kxB) \mathrm{mod} N_b] + 1, \tag{1}$$

where k denotes the prime number and N_b is the representation of total blocks that exist in ROI. A distinct integer $\{1, 2, 3, \ldots, N_b\}$ as B is allocated to each ROI block. Raster scan {left-right top-bottom} is utilized for the allocation of a number to every block. After that, RONI is split into 6×1 pixel blocks. The compression of actual LSBs is done by applying RLE, and a block in the region RONI is then utilized for embedding the resultant package of RLE.

(b) Watermarking procedure in ROI: the watermarking technique is based on Jasni's scheme [27] with a block size of 6×6 pixels in ROI. Ahead of inserting the watermark, the native LSBs are removed and each one is set to zero. Each 6×6 block is segregated further into 3×3 pixel subblock. For every single subblock, a watermark of 3×3 bits consists of an authentication watermark of a 2-bit and recovery watermark having 7 bits (r).

Create a watermark v for every single subblock after calculating the average intensity of block (Avg_B) and its subblock (Avg_Bs) as

$$v = \begin{cases} 0, & \text{if AvgBs} > \text{Avg}_B, \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

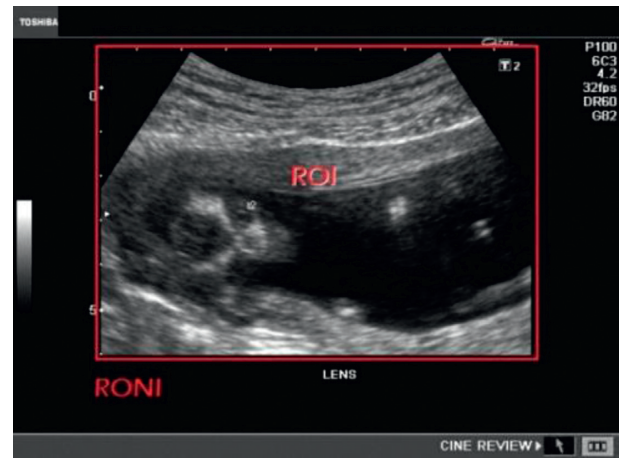


Figure 3: Locations of ROI and RONI [26].

The watermark p is generated as

$$v = \begin{cases} 1, & \text{if Avg}_{\text{Bs}} \text{ is odd,} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The watermark (v, p, r) is then embedded for every single subblock in its LSBs as illustrated in Figure 4.

(c) Embedding in the RONI: the actual LSBs are stored in RONI after compression. These stored values are then used for the image restoring process. Removed LSBs are depicted as a 640×480x1-bit binary LSB matrix. For achieving higher embedding capacity, an RLE-based compression is used. A matrix having LSBs is split into 3×3 bit LSB blocks first, and then, all the bits of the block are converted into decimal according to the demonstration mentioned in Figure 5.

The decimal value range is 0–511(9 bits). An acquired collection of converted values of every single LSB block is then subjected to a raster scan, and obtained data run is then utilized as an input of RLE. The result of the RLE process will be a package containing the run value (RV) and run count (RC).

Each 12-bit RLE package of injected into 6×1 pixel block in the RONI. RV range is 511, so 9-bit storage is required whereas RC is limited to 7, for that matter only 3 bits are required. A complete scheme for the RLE package having 12 bits in RONI is depicted in Figure 6.

Established from the experiments, second LSB (LSB 2) needs to be used to avoid storage issues and embedding all the RLE packages in RONI.

(d) Identification of tamper and recovery: the presumed image is decomposed into RONI and ROI and ROI is split into 6×6 pixel block. Each block of ROI is then further

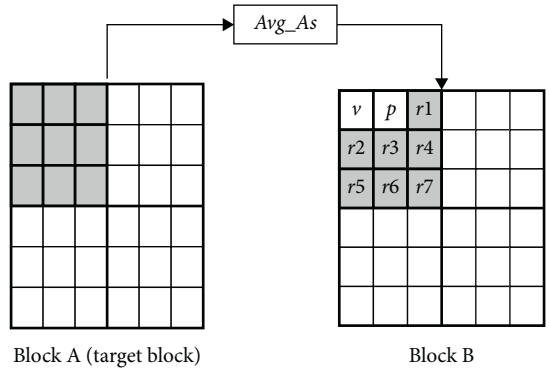


Figure 4: Watermarking in ROI [25].

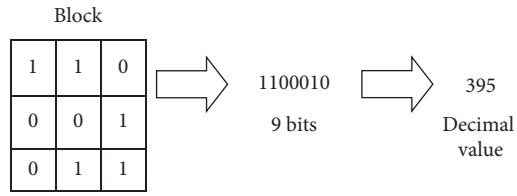


Figure 5: Decimal conversion of 9-bit LSB block [25].

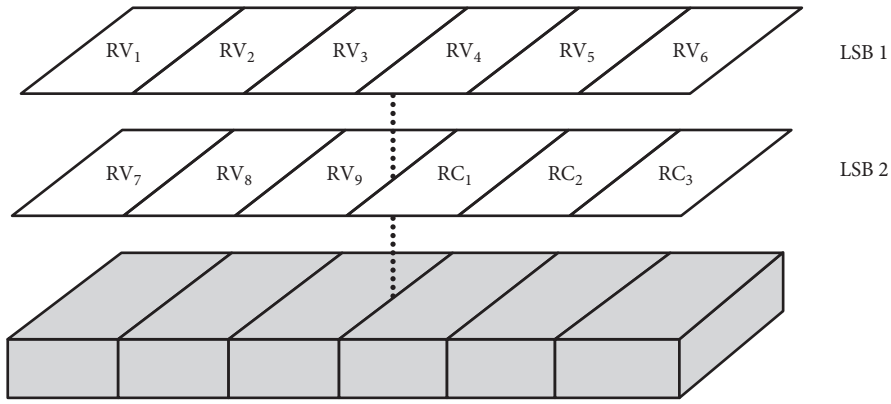
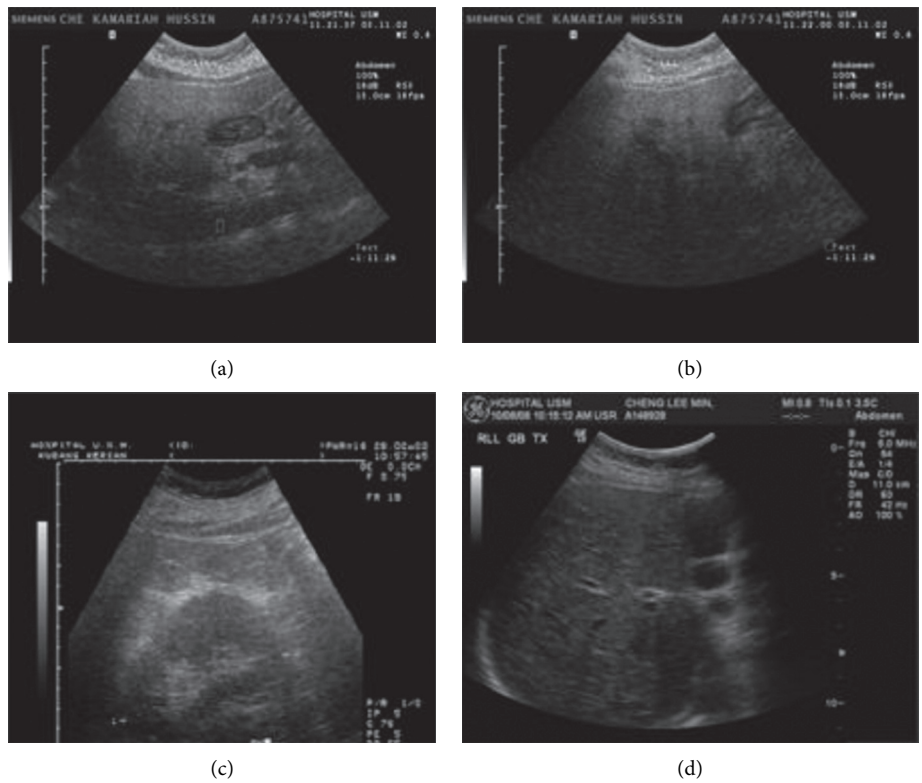


Figure 6: Embedding of an RLE package into a RONI block [25].

divided into subblock of 3×3 pixels. As proposed in [27], for every subblock, parity bit p and authentication bit v will be extracted. After adjusting the block's LSBs to zero, calculate the average intensity of block (Avg_B) and its subblock (Avg_Bs). Populate the parity bit p' and authentication bit as v' and compare these bits with the extracted bits p and v for the authentication of the block whether it is tempered or not. Tempered blocks will be revived by spotting recovery blocks utilizing the continuous pattern of the mapping which is utilized in the preparation of the image. As illustrated in Figure 7, let A be a block that is tempered. For the recovery of the block, information embedded in the block B will be used. For every single subblock in A block, substitute 7-bit MSB containing recovery data acquired from its corresponding subblock in the block B.

(e) Restoring the actual LSBs: after splitting RONI into 6×1 pixel block, the block of RONI which is utilized for injecting each RLE package will be extracted and decoded as shown in Figure 6. A collection of decimal values in the range of 0–511 will be obtained after decoding RLE. Each decimal value will be transformed into a binary number of $3 \times 3 \times 1$ bits. This whole procedure is opposite to the one illustrated in Figure 6. A matrix containing $3 \times 3 \times 1$ bits will be the 3×3 pixel block's LSBs. After dividing the image into 3×3 pixels block by using a raster scan, each $3 \times 3 \times 1$ bit LSB is then restored to each block of the image.

The second LSB used to embed RLE packages cannot be restored because that was not stored. Due to one of the qualities of ultrasound that nearly every pixel value of RONI is depicting black having value zero, the second LSB is set to zero for every pixel with a value less than 3 to boosting the PSNR of the recovered image. For the case of ROI, it can be claimed that every pixel is reversible. MATLAB is used for the implementation purpose to evaluate the watermark reversibility as well as to spot the tamper and repossession of the image.



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

Figure 7: Images utilized for the experiment [28]. (a) Ultrasound 1 (888 × 666 pixels), (b) ultrasound 2 (888 × 666 pixels), (c) ultrasound 3 (704 × 576 pixels), and (d) ultrasound 4 (640 × 480 pixels).

Results illustrated the PSNR of the watermarked image is above 46 dB which means that the technique can originate watermarked images with lower deterioration and identical to the actual.

From the reversibility point of view, the results depicted in Figure 8 show that the technique is reversible for that region which comprises the ROI, but the RONI-acquired image is close to the original due to the inclusion of the second LSB. Temper detection results showed that the scheme performed 100% for temper localization whereas for sharpening attack it performed 99%, and for brightness and contrast adjustment attacks, the performance was between 60% and 98%. Recovery performance was 100% in terms of recovery and PSNR for the area <20%. For the manipulations containing sharpening adjustments, brightness adjustments, and contrast adjustments, the recovery rate was low (6%–48%). Further improvements are desired to make this scheme completely lossless for a wider range of attacks.

3.1.2. ROI-Based Tamper Detection and Recovery for Medical Images Using Reversible Watermarking Technique. Al-Qershi and Khoo [28] coined a technique comprised of difference expansion for reversibly embedding watermark in ultrasound images. Embedding is done into changeable and expandable groups, and difference groups are distinguished by the location map. This technique can be utilized for hiding information related to the patient, authentication of ROI, locating the area tempered inside the ROI, and recovery of the tempered regions. After splitting the original image into ROI, RONI, and border regions, RONI is used for embedding payload through 2D-DE. The location map is generated which is then used at the extraction phase. The payload is the combination of the following:

(a) Patient's information

(b) ROI's hash message for authentication

(c) ROI pixels for locating tempering

(d) Recovery along with the LSBs of pixels located border of the image

(a) Embedding phase:

(i) After decomposition of the image into ROI, RONI, and border

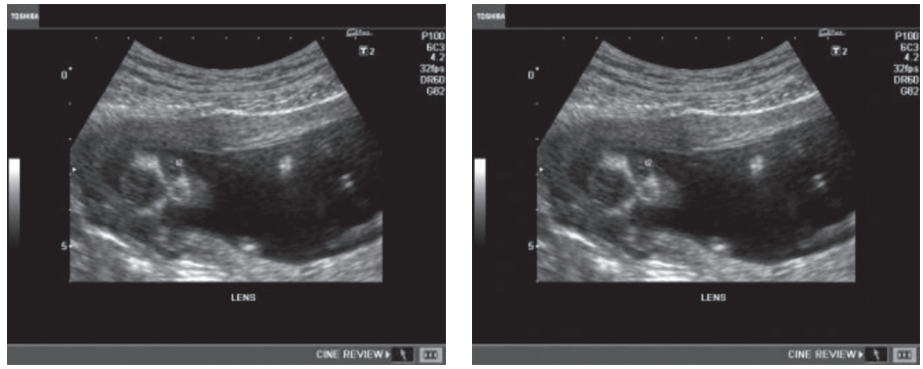
(ii) By utilizing the MD5 algorithm a hash message "H" is computed for ROI after dividing it into 16 × 16 pixels

(iii) Bits of ROI pixels "P" and LSBs of border pixels "L" are collected

(iv) The patient's information "D" is appended in these collected bits "P" and "H" along with the hash message "H"

(v) After the concatenation of bits in the previous step, the payload is formed by compressing string through Huffman coding

(vi) RONI is utilized to embed payload in it using 2D-DE [29]



(a) (b)

Figure 8: (a) Actual image and (b) watermarked image [27].

(vii) The location map generated is then appended with the coordinates of ROI and after compression embedded in border pixels LSBs

(b) Extraction phase

(i) For extraction, the location map is populated along with the extraction of ROI coordinates by decompressing the bitstream collected from border pixels.
(ii) RONI is identified by utilizing ROI coordinates.
(iii) Extraction of the payload is done from RONI which is disintegrated into H, P, L, and D parts after decompressing.
(iv) For the authenticity of the image, the hash is calculated for the ROI and matched with the extracted one. If values are different, that shows tempering.
(v) The tempered part is recovered by dividing the ROI into 16×16 pixel block, computing the average of each block, and replacing with the average value of corresponding pixels of extracted ROI "P". Bits of L are utilized to recover LSBs of border pixels.

The proposed technique is experimented using four DICOM images shown in Figure 7 having different sizes of the ROI and distinct data sizes of patients.

Reversibility can be assessed with the help of pixel-by-pixel comparison of the original and recovered image whereas tempering can be located by the comparison of hash values collected at the embedding phase and extraction phase.

Temper detection and recovery were demonstrated by replacing some pixels from RONI with pixels in ROI. During extraction, the scheme successfully located the tempered region and recovered area which is shown in Figure 9.

The embedding results with variant sizes of ROI and patient's data sizes are illustrated in Table 2.

Results illustrated that technique performed well with regard to PSNR, increased capacity of embedding, and reversibility in case of no major tempering. The abovementioned technique performed very well in terms of higher hiding capacity as well as visual quality by utilizing hiding techniques that provide higher embedding roam to compensate for the issues of ROI size and image size. Further improvements in terms of multiple-ROI concept can also be incorporated for facilitating medical informatics.

3.1.3. Harris Corner Detection and Fuzzy C-Means. In 2013, Debalina Biswas proposed a method that uses a clustering method for the recognition of the pattern known as fuzzy c-means (FCM) [30]. In this technique of clustering, one set of data can belong to two or even more than two clusters. It also permits the data when they belong to more than one class; then, its membership function degree will vary between 0 and 1.

There is a rare advantage of the FCM that linguistic variables can be grading based on pro-rata data and after the accurate analysis fit easily in the domain of the discrete. FCM uses an iterative approach to reduce the function of the dissimilar and computes the centers of the cluster. After updating, the centroid within the pixels set will be shifted to the right location.

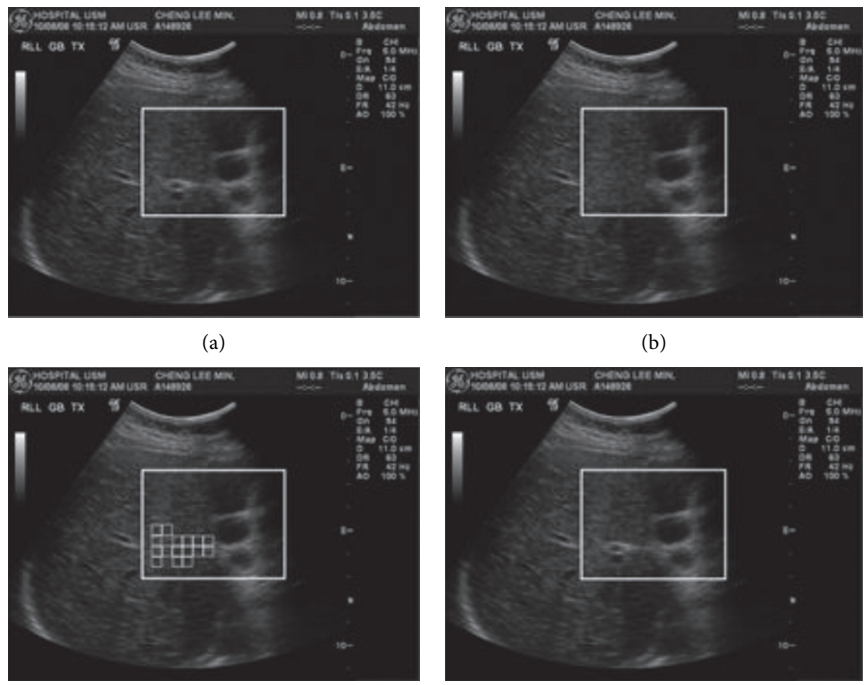
The membership matrix, u_{ij} is used for the membership degree for the points of data of x_i cluster which starts from

$$\sum_{i=1}^{c} u_{ij} = 1, \quad \forall j = 1, \ldots, n, \tag{4}$$

performance index (PI) which is used for the center of cluster and matrix of membership is provided as given below:

$$\begin{aligned} J(U, c1, c2, \ldots, Cc) &= \sum (i = 1, \ldots, c) Ji \\ &= \sum (i = 1, \ldots, c) \sum (j = 1, \ldots, n) uij^{\wedge m} d_{ij^2}. \end{aligned} \tag{5}$$

Euclidian distance (d_{ij}) is between the data point and the center of the cluster. The weighting exponent is m which belongs to [1, infinity]. There are two conditions which can be used to reduce the function of dissimilarity given below:



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

Figure 9: Temper detection and recovery [28]. (a) Watermarked image, (b) image after tempering, (c) temper detection, and (d) recovered image.

Table 2: Embedding results with variant sizes of ROI.

	2K			6K			10K		
ROI size	Payload (bits)	Capacity of hiding (bpp)	PSNR	Payload (bits)	Capacity of hiding (bpp)	PSNR	Payload (bits)	Capacity of hiding (bpp)	PSNR
Ultrasound 1									
4%	169,424	0.287	37.01	209,520	0.354	36.42	246,768	0.417	36.07
8%	260,704	0.441	35.99	299,760	0.507	35.54	335,312	0.567	35.28
12%	395,680	0.669	35.63	436,528	0.738	35.60	n/a	n/a	n/a
Ultrasound 2									
4%	155,808	0.264	37.46	194,816	0.329	36.81	230,112	0.389	36.46
8%	245,376	0.415	36.45	283,872	0.480	36.09	319,024	0.539	35.86
12%	376,368	0.636	36.13	417,056	0.705	36.09	n/a	n/a	n/a
Ultrasound 3									
4%	122,848	0.303	41.88	163,136	0.402	40.13	200,176	0.494	39.15
8%	207,776	0.512	39.36	245,824	0.606	38.34	281,200	0.694	37.75
12%	263,984	0.651	38.24	n/a	n/a	n/a	n/a	n/a	n/a
Ultrasound 4									
4%	82,832	0.270	41.25	119,088	0.388	39.81	153,968	0.501	38.57
8%	151,552	0.493	38.84	191,024	0.622	37.94	n/a	n/a	n/a
12%	190,912	0.622	38.10	n/a	n/a	n/a	n/a	n/a	n/a

$$ci = \sum \frac{(j = 1, \ldots, n) uij^m xj}{uij^m}. \quad (6)$$

(a) FCM algorithm: following are the steps:

(i) The matrix which is used for the membership is initialized randomly.

(ii) Using equation (3), centers of clusters are calculated.

(iii) Using equation (2), dissimilarity function is computed. Halt if the perfections in the results are below the taught pixel of the line of the threshold.

(iv) Using equation (4), a new matrix of membership is calculated and then the steps are repeated.

(b) Alpha blending: this blending technique is used to blend the first image source with the pixel equivalent in the

second image source. Following is the execution of the alpha blending:

$$\text{pixel} = \text{alpha} * (\text{source pixel of first image}) + (1 - \text{alpha}) * (\text{source pixel of the second image}). \quad (7)$$

For the blending of the images, the blending factor is taken from the first image that is why it is known as alpha blending [31, 32]. The range which is used for the alpha in the algebraic is between 0.0 and 1.0. Using this technique, watermark is generated which is given as follows:

$$(\text{WII}) = \text{alpha} * (\text{CI}) + (1.0 - \text{alpha}) * (\text{WI}), \quad (8)$$

where WII is the watermarked image, CI is the selected image, and WI is the selected equivalent image. The output of the process is shown in Figure 10.

(c) Embedding process of watermarking: in the embedding process of watermarking, the grey medical image segmentation is applied using fuzzy c-means. Then, the binarization is employed on the image. From the image, which is binarized, an open binary area is used to clean the smaller objects. For the detection, Sobel edge is employed. Then, the algorithm of Harris corner detection [33, 34] is practical. The maximum diameter is taken, and based on the points of Harris, the center is calculated. The boundary is marked between the region of the noninterest (RONI) and region of interest (ROI) based on the circle which is drawn on the diameter and the center point. Watermarked image size is used to select an area from the RONI. The alpha blending technique is used to embed the watermarking on the area which is selected.

(d) Discussion and results: for the study of the effect of watermark embedding on the medical images, results were compiled and shown in Figure 11.

Section of RONI is exploited for the insertion of the watermark using the alpha blending technique shown here.

(e) PSNR applied on the signal of watermark: PSNR between the watermarked MRI image and the original image is 28.8255, and PSNR between the watermarked IVUS image and the original image is 23.504.

3.1.4. Protection of Ultrasound Image Sequence: Employing Motion Vector Reversible Watermarking. With the induction of new techniques for medical imaging, the diversity of imaging modalities has increased twofold. Every modality has its own requirements for a reversible watermarking strategy to be employed for data safety for HIS. In 2019, Habib and Al-Fayez [36] presented an approach for digital security of ultrasound imaging videos or image sequences using motion vectors. Motion vector prediction is a popular technique in video compression and utilizes a comparison between two images or frames for providing data compression at the encoder side. This is a very promising and computationally inexpensive (when large blocks are used) method for videos and image sequences that have a low relative motion rate between frames such as ultrasound imaging.

The proposed methodology, after a histogram preprocessing stage (to minimize overflow/underflow), subjected the image sequence to a full search block-matching algorithm (BMA) with a 16×16 block size. The magnitude of the calculated motion vector designated as $\text{MV}_{i,j}$, where I is the i^{th} frame and j is the j^{th} motion vector of the frame, is used to define an embedding threshold value. These horizontal and vertical magnitude values MV_H and MV_V of the motion vectors, selected on the basis of the threshold value, are used to calculate the phase angle using $\theta = \tan^{-1}(\text{MV}_V/\text{MV}_H)$ which is subsequently used for data embedding. The side information about the frames that were calculated during histogram modification and motion estimation is also embedded alongside the watermark. This ensures accurate data extraction at the decoder/extraction side and increases the robustness.

The authors have been able to achieve infinite PSNR between the sending and receiving side images, meaning both were identical after data extraction. PSNR value of the original versus the watermarked image was around 41 dB with an SSIM of about 0.92. The maximum embedding capacity achieved was 0.29 bits per block, and the ratio of side information to the watermark data was about 1.65% which is highly promising. The downside, as mentioned by the authors, is the applicability of the algorithm being limited to large block sizes only as reducing the block size not only increased the computational overhead but also involved adding large amounts of side information data mandatory accurate retrieval.

3.2. Frequency Domain

3.2.1. A New Approach to Fully Reversible Watermarking in Medical Imaging with Breakthrough Visibility Parameters. Alešat et al. formulated a reversible scheme for watermarking of medical images based on the alloying advantages of different three conventional approaches—zero, reversible, and RONI watermarking [37]. The basic purpose of combining different approaches is to eradicate the disadvantages and exploit the strengths of the approaches. The input media is split into different two categories RONI and ROI; RONI is the zone of the image where minute variations do not affect the medical information as a whole. Area, other than RONI, falls under the umbrella of ROI.

The identification of RONI is based on the pair's comparison of neighbor vectors where vectors consist of the pixel's values of rows along with columns. A comparison between neighbor vectors is made in each direction. A threshold is used to differentiate between the boundary of ROI and RONI. For experimentation for this technique, the defined threshold has 10% similarity based on the different tests [38]. The sample utilized for experimentation purposes in this study contained RONI of size 11%.

(a) Watermark concealment.

(i) After the detection of ROI, it is subjected to the DT-CWT transform.

(ii) Select DT-CWT numbers LL.

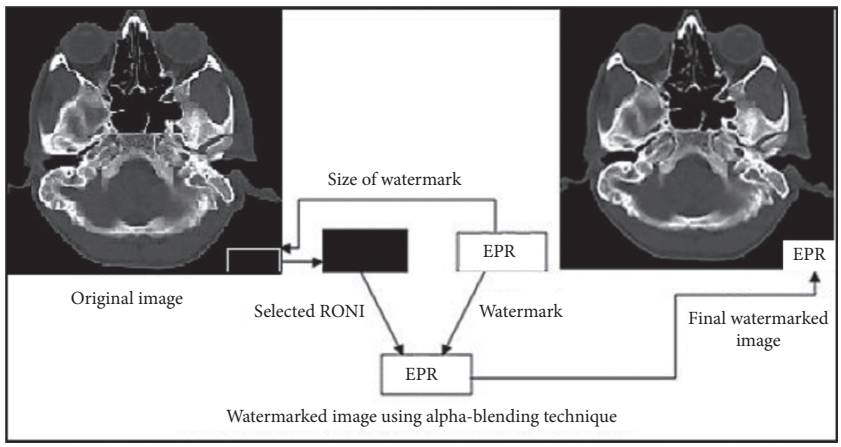


Figure 10: Alpha blending technique [30].

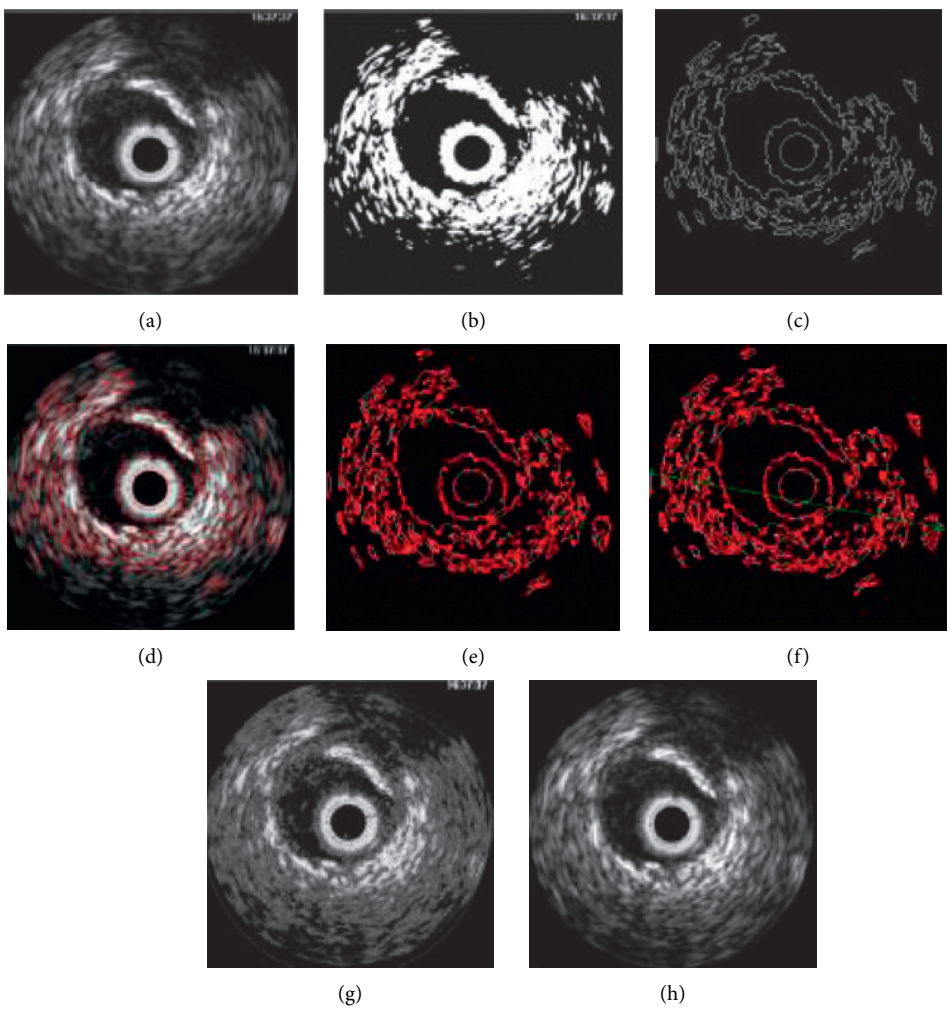


Figure 11: Watermarking embedding process in IVUS [35].

(iii) Determine the arithmetic average of coefficients ($\varnothing_{LL}$).

(iv) The random location of pixels comprising the same number of pixels as a watermark is identified in the LL coefficient region. Locations must not be too close to the edge of the image.

(v) Let $R_i(x, y)$ be the i^{th} random location, then neighboring region comprising 7×7 pixels is elected by placing $R_i(x, y)$ at the center.

(vi) A feature image F equivalent to watermark size. The average of neighboring 7×7 block selected for the i^{th} random location must be equal to the individual averages of feature image F. The following formula is utilized to create matrix B from feature image F:

$$B(x, y) = \begin{cases} 1, & \text{if } F(x, y) \geq \varnothing_{LL}, \\ 0, & \text{if } F(x, y) < \varnothing_{LL}, \end{cases} \quad (9)$$

where $\varnothing_{LL}$ is the arithmetic average of LL band coefficients.

(vii) A secret share is generated which is used as a watermark to RONI by combining matrix B and a watermark as shown in Figure 12.

(b) Watermark extraction.

(i) Region of noninterest and region of interest of the image which is watermarked are isolated.

(ii) LL subband of coefficients is selected after the DT-CWT transformation with RONI.

(iii) The same seeding process would be repeated with the help of secret key S for the generation of the same random pixels within the LL region as in the embedding process.

(iv) For each $R_i(x, y)$ location, by keeping $R_i(x, y)$ at the center and selecting the neighbor 7×7 pixels, the average is obtained.

(v) This value is utilized at the time of feature image which must be of the size of the inserted watermark.

(vi) The feature image is then used to create binary matrix B from the same formula used in the concealment process. The public share will be generated by the combination of the binary matrix B and Figure 13.

(vii) For the extraction of watermark, logical OR operation is performed created in the present process with the undisclosed share and free share.

(viii) The utilization of the process depicted in Figure 12. Reduction of watermark than its actual size.

Reversible contrast mapping is utilized for the reversible watermarking. RCM utilization required segmentation of the image into a group of two pixels called pairs. For the transformation, pairs are utilized:

$$x' = 2x - y, \quad y' = 2y - x. \quad (10)$$

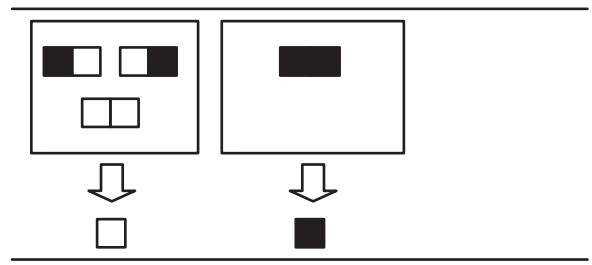


Figure 12: Reduction process [38].

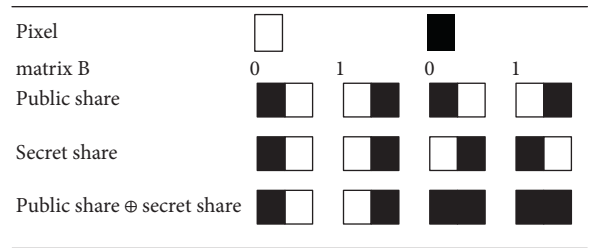


Figure 13: Cookbook for public share and secret share [37].

However, (x, y) are the original and (x', y') are the values that are transformed. To avoid the underflow overflow, put a limit on the transformation subdomain.

(c) Watermark embedding.

(i) Pixels (x', y') are isolated in the region of noninterest.

(ii) For every value of (x, y) if the values of the x and y belong to the domain and if its conformation is based on the odd value of pixels, then the pair is exposed to transformation. The new value of x LSB is set as "1" and LSB of the new value of y is possibly utilized for data insertion.

(iii) If the values of x and y belong to the domain and if its conformation is not based on the odd value of pixel, then the new value of x LSB is set as "0" and LSB of the new value of y is possibly utilized for insertion of data.

(iv) If the values of x and y do not belong to the domain, zero is assigned for LSB and the actual value is persisted. The locations identified for bits above are utilized to embed a watermark in RONI.

(d) Watermark extraction.

(i) To the new values of x and y, the region of noninterest is isolated.

(ii) For every new value of x and y, apply steps (iii)–(v):

(iii) If the new value of x has LSB of 1, then LSB of y' is appended after retrieval in the watermark sequence. LSBs of both transformed pixels are set to zero, and the original values are filled by taking the inverse form of transform.

(iv) If the LSB of the new value of x is zero and the pairs have their LSBs set to one and are in the domain, then the LSB of the new value of y is appended to the

sequence of watermark and the original pair is returned by fixing the LSB of new pair to one.

(v) If the new value of x has its LSB zero and the new pair having LSB one is not in the domain, original values are filled by replacing the LSB of the new value of x with the concerned value from the already having sequence of the watermark.

For examination purposes, experiments are performed using 6000 selected medical images from the database which contains 60,000 medical images [38]. Test images consist of 9 different modalities including ultrasound. The average PSNR obtained is 81 when the RONI size was 10% of the original image. The average SSIM obtained is 0.999974. For extensively testing the proposed technique, it can be tested on a larger medical database.

3.2.2. Detection of Accurate Tamper in Region of Interest. In 2014, Eswaraiah proposed a watermarking technique for medical images [5, 39, 40]. In the medical images in the form of different shapes, there is an area of the region that is of no interest to the physician. In this work, those medical images are considered which contain a single ROI. The border of the image is the pixels outermost three lines. Pixels are divided into three respective sets: pixels of ROI, pixels of border, and pixels of RONI. For authentication purposes, SHA-1 technique is used to calculate the ROI hash code. Any kind of change in the code of hash will be quickly identified. Areas of medical images ROI are divided into 4×4 blocks and those of RONI are divided into 8×8 blocks, so they do not overlap. Now the ROI blocks are plotted to the RONI blocks. This is done to embed the information to be recovered. The following equation is used:

$$B_{\mathrm{RONI}} = \left[\left(k \times B_{\mathrm{ROI}}\right) \mathrm{mod} N_b\right] + 1. \tag{11}$$

Lastly, the ROI information and value of hash are implanted into the pixels of border LSBs.

(a) Algorithm for the embedding: medical images are divided into sections of 3 which contain pixels of region of interest, pixels of border, and pixels of region of noninterest. SHA-1 technique is used to calculate the ROI hash value. ROI area is divided into the blocks of 4×4. RONI is divided into the blocks of 8×8. Map each individual block of ROI on the area of RONI. In this way, the data to be recovered are embedded. Secret key ROI information and the value of the hash are used for encrypted. Pixels of the border are used for the bits of encryption to be embedded. Now the medical image can be sent to remote locations.

(b) Algorithm for the extraction: pixels of the border are used for the encryption bit extraction. Decryption is applied to get the ROI information and value of the hash. SHA-1 technique is used to calculate the value of the hash of ROI. Both values of the hash will be compared, and if the values get equaled, the process of extraction will be stopped. ROI is divided into the blocks of 4×4 and RONI into the blocks of 8×8. For each block of ROI, steps are repeated so that the block of ROI which is tampered will be identified. The value of variance and average for each block will be calculated. If both variance and the average value of the blocks are not equaled, it means it is tampered and replaced with the pixel bits.

3.2.3. A Novel Texture Quantization-Based Reversible Watermarking Scheme for Information Health Systems. In 2017, Turuk and Dhande [41] presented a novel approach for making transactions of information via health information systems (HIS) more secure. An argument in the favor of digital watermarking is a promising solution aptly suited for securing HIS data from sniffing and snooping by unauthorized users. The proposed novel approach used a hybrid method that utilized a combination of image quantization and its texture treated as a feature to hide patient data segmented into multiple watermarks. The preliminary feature extraction was done using a discrete wavelet transform (DWT) with a "HAAR" mother wavelet, and further quantization approach for transform coefficients was presented by the authors which were the most noteworthy contribution.

The proposed method first used a level 2 DWT to find the downsampled coefficients from the cover image, as depicted in Figure 14. As DWT is very effective in segregating the textured areas and edges, it provides a good basis for improving the imperceptibility of the watermark addition. Figure 15(a) shows an example of the level 2 DWT being applied to a CT modality image. To increase the robustness, the energy of each decomposed subband was calculated using the following equation:

$$e_k = \frac{1}{N_k M_k} \sum i \sum j \left| J_k(i, j) \right|, \tag{12}$$

where k points to the approximation and subbands at a certain decomposition level while J_k is the subband coefficient of dimension $N_k M_k$. A texture map was generated by obtaining a relationship between the spatial change and the energy variance by combining coefficients of the lowest matrix of the subband (LL) with the energy as given below:

$$\mathrm{txtFeatures}_r(i, j) = \left[W_r(i, j)\right] 2 + \mathrm{var}\left[\mathrm{LL}_d\left[\left(\frac{i}{2}\right] + 1, \left[\frac{j}{2}\right] + 1\right), \mathrm{LL}_d\left(\left[\frac{i}{2}\right] + 2, \left[\frac{j}{2}\right] + 2\right)\right]. \tag{13}$$

This texture map is used to calculate the threshold value to preserve the most prominent edge features and texture details. Based on this threshold value, all subbands apart from LL are used to embed the watermark data. As wavelet coefficients obtained are real numbers, their mapping to binary watermark data involved a

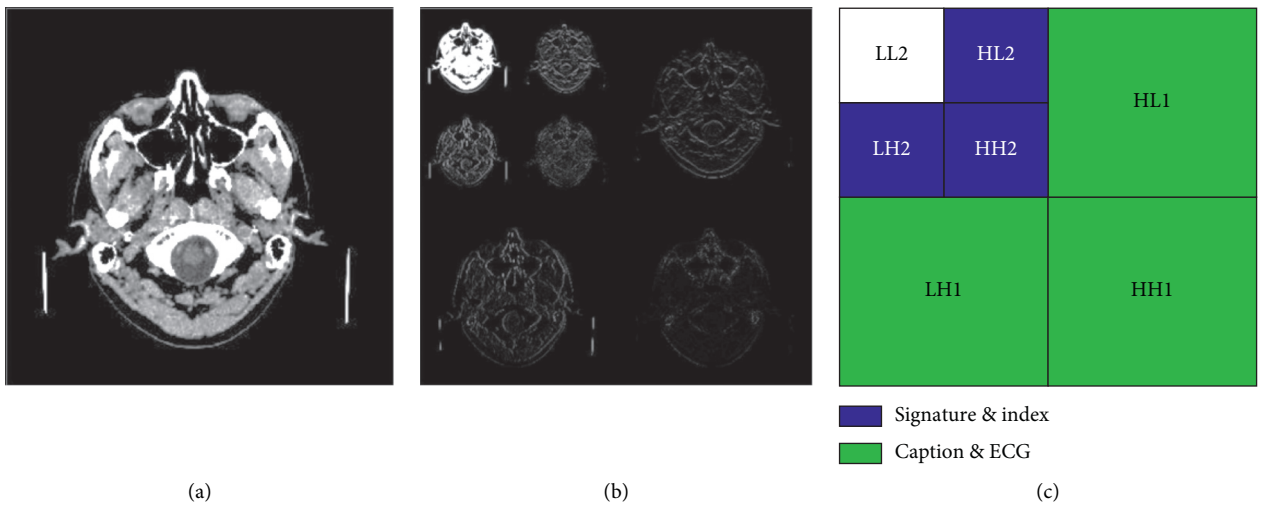


Figure 14: DWT of CT modality image and texture areas and texture map of the subbands [41].

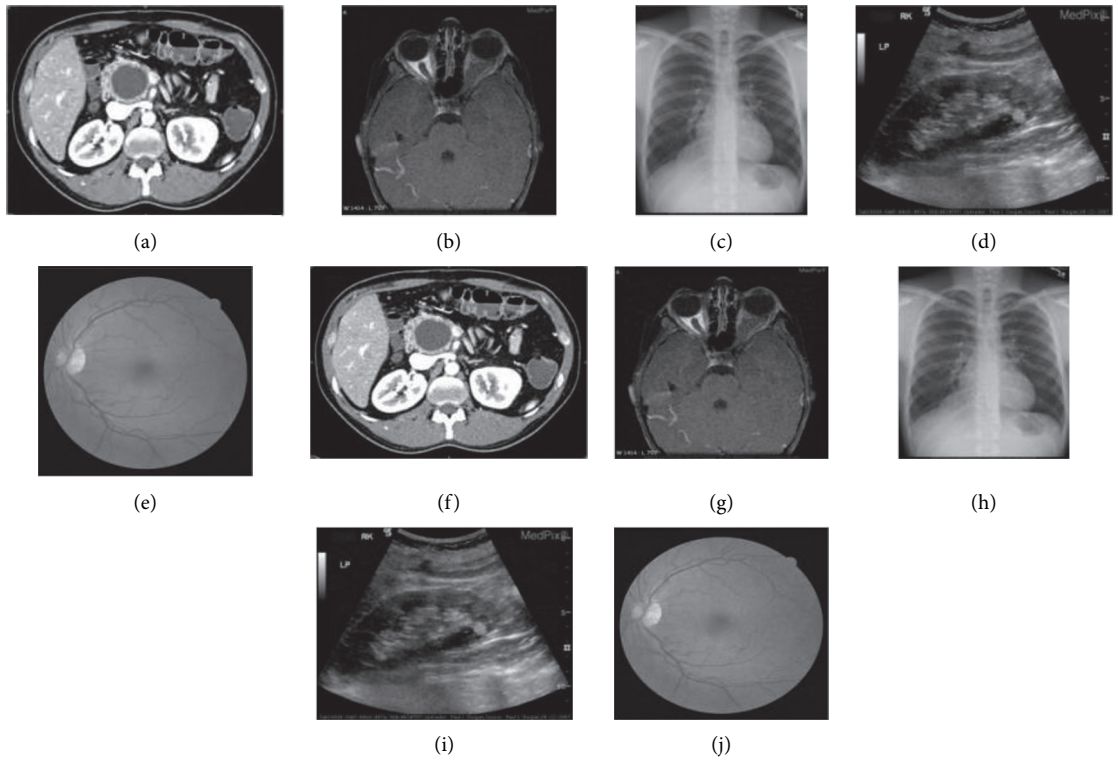


Figure 15: Evaluating the watermarking imperceptibility. (a–e) Original images. (f–j) Watermarked images [42].

quantization function $Q(f)$ and its variation was defined as follows:

$$Q(f) = \begin{cases} 0, & \text{if floor}\left(\frac{f}{\Delta}\right) \text{is even}, \\ 1, & \text{if floor}\left(\frac{f}{\Delta}\right) \text{is odd}, \end{cases} \quad (14)$$

where $\Delta = 2L$ in which L is the decomposition level. As the same constant value is being added and subtracted, to ensure data integrity, a tracking key was used to signify the coefficient sign change and duplicate coefficients. Watermark data were created using the BCH error-correcting code which ensures data integrity. In the experiment, the watermarked data were performed which comprised of a caption, an ECG signal, the physician's electronic signature, and ICD-10 diagnostic data. The researchers were able to embed a maximum of 22,600 bits

with a PSNR of 53.60 along with encouraging numbers for both MSSIM and UIQI. This was a major improvement considering the contemporary methods and made this research a major contribution to reversible medical image watermarking.

3.2.4. A Hybrid Cryptography-Based Watermarking Technique for the Secure Transmission of Medical Data for Telehealth Applications. Thakur et al. [43] presented an approach employing a hybrid cryptographic-watermarking methodology for the applications of telehealth. Logistic maps, which were initially employed for modeling animal populations, have been extensively studied as an effective cryptographic tool as they provide pseudorandomness and chaotic behavior. The researchers used two-dimensional logistic maps to encrypt the watermarked host image enabling secure transmission/communication.

In a bid to improve imperceptibility, the authors used a combination of three transform domain techniques, i.e., DCT, DWT, and SVD, on the host image. The watermark image was subjected to DCT and SVD only to get a singular matrix $S1$. In the first step, a 2nd level "Haar" DWT is performed on the host image resulting in the approximate image (cA1) and the coefficient matrices. cA1 is further transformed using DCT and the resulting matrix is singularized using SVD (matrix S). SVD matrices are used for watermark embedding (based on gain factor α), and inverse DCT and DWT are performed to get the final watermarked image. The inverse resultant matrix of DCT was then obtained.

(a) The host image which is of size 512×512 read as $\mathrm{Cw}_{\mathrm{image}}$ and the image which is watermarked of size (256×256) read as $\mathrm{WM}_{\mathrm{image}}$

(b) Utilize the second level of DWT on the DWT cover

(c) Utilize DCT which will provide B

(d) Utilize the SVD on B which provides S

(e) For the image of watermark, utilize DCT on $\mathrm{WM}_{\mathrm{image\ to}}$ provide D

(f) Utilize SVD on the D which will provide $S1$

(g) Using the gain factor, embed the watermark image in the cover; now utilize the inverse of DWT, the inverse of DCT, and the inverse of SVD to get watermark with the cover image

(h) Using private key, encryption is applied to the $\mathrm{Wd}_{\mathrm{image}}$

The time of encryption and decryption of the proposed method and survival against different attacks were tested in the experiments, and all parameters and the results were highly promising for the proposed method. NC, SSIM, and PSNR were greater than 74.61, 0.998, and 0.900 dB, respectively, which proved that the proposed method is a highly secure and robust methodology for telemedicine applications. In the start, the MRI image of the brain having size 512×512 and the size of the part of the body between the neck and abdomen 256×256 is taken as cover and watermark image, respectively.

The proposed algorithm is implemented using MATLAB R2013a on a 64-bit machine having a Core-i5 processor and 4 GB RAM.

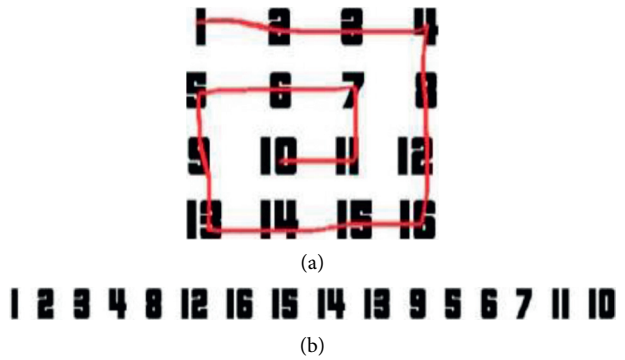
3.2.5. A Novel Robust Reversible Watermarking Scheme for Protecting Authenticity and Integrity of Medical Images. In 2019, Liu et al. [42] presented a technique for protecting medical imaging content from tampering. The methodology was based on an improved ROI- and RONI-based reversible watermarking technique that ensured the integrity of both regions and avoided the spatial segmentation of the image. The watermarking scheme consisted of four major proposals for improvement. The first was slantlet wavelet transform- (SWT-) and singular value decomposition- (SVD-) based reversible watermarking method which provided watermark robustness which ensured the lossless recovery of both ROI and RONI. The second generation of the watermark is based on a combination of integrity and authenticity data for increased image security.

This included the hash value of the hospital logo as the validity of data and the information of tamper detection for both the image and the ROI. Thirdly, a grouping of integer wavelet transform (IWT) and block truncation coding (BTC) was used for encoding tamper recovery data of ROI which offered a great compromise between the recovered visual quality and embedding capacity. Finally, there was no image segmentation in the domain of spatial and watermarks embedded in the whole image which mitigated any additional security risks.

Watermark generation consisted of a generation of authenticity data (SHA-1 hash of hospital logo), tamper detection (SHA-1 hash of image), tamper localization (CRC-16 based error-checking of ROI), and tamper recovery information. The watermark was embedded using SWT which offers better spatial and frequency localization as compared to DWT and hence better imperceptibility versus robustness results. An added layer of SVD, which was utilized for selecting the significant (from the generated singular matrix) values for watermarking, reinforced the watermark robustness.

The authors performed experiments on various image modalities that included 200 medical images and 40 each of X-ray, CT, ultrasound, and fundus images. All the experiments demonstrated high levels of imperceptibility with an average PSNR of 41.2995 and SSIM of 0.9607. The robustness of the scheme was also commendable as the average of the mean BERs of all experiments was 0.0476, and NCCs were 0.9624. The scheme also performed better than the spatial-domain methodologies when subjected to various attacks.

3.2.6. Spiral Order Technique Reversible Watermarking. In 2018, S. Lakshmi and Ayyappan [44] proposed a method of data hiding which is reversible in nature in the medical images using spiral order. The order of traversal is depicted in Figure 16. It is suitable to use for images that are confidential in nature as the patient data are very important to hide the data in the medical images using this technique. In the grayscale image, there are no data present in the region of



(a)

(b)

Figure 16: Spiral order technique [8].

the background, and to hide the data, this region can be utilized. For the embedding of data modification, a histogram is utilized, and to hide the data, spiral order is applied.

(a) Using spiral order to hide data: the cover image is read first for the hiding of data and then spiral order is used to convert it into the form of a vector. To hide the information, positions are identified first. There are four stages involved: take the first row from the rest of the rows, the last column from the rest of the column, the last row from the rest of the rows, and the first column from the rest of the columns. The results of the technique are presented in Figure 17.

(b) Modification through histogram: the difference between the adjacent pixels is calculated. It will be close to zero as these are closely packed. The high point will be the most occurred value of difference. In three ways, pixels are altered depending on the difference value how much it is above or below the high point.

(c) Recover the information: the bit of text can be extracted using the following equation:

$$B = \begin{cases} 0, & if\ |Yi - Ai - 1| = PP, \\ 1, & if\ |Yi - Ai - 1| = PP + 1. \end{cases} \quad (15)$$

The values of the pixels of the image were originally reconstructed using the following equation:

$$A_i = \begin{cases} Yi + 1, & \text{if} \left| Yi - Ai - 1 \right| > \text{PP and } Yi < Ai - 1, \\ Yi - 1, & \text{if} \left| Yi - Ai - 1 \right| > \text{PP and } Yi > Ai - 1, \\ Yi, & \text{otherwise.} \end{cases} \quad (16)$$

(d) Experiments and results: PSNR and MSE are the performance metrics to be applied to the medical images. To check the similarity among the images, SSIM is used. Improvement in the results with the proposed method is shown as follows.

3.2.7. Hybrid Estimate Reversible Watermarking. In 2018, Fatima proposed estimate-based hybrid reversible watermarking [44] in the domain of medical images. It is a four-step process that contains cover image, processing of the

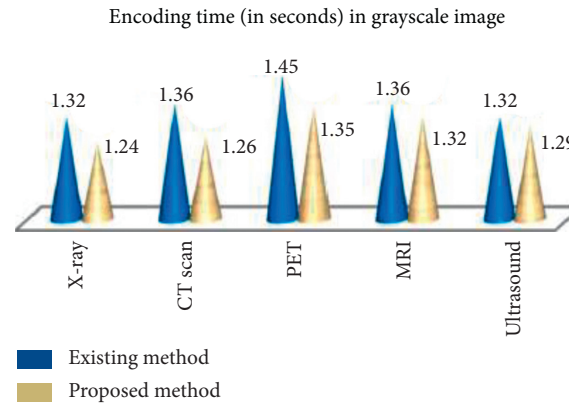


Figure 17: Comparison of embedding time [8].

images, encoding using watermarking, and lastly decoding using watermarking.

(a) Encoding using watermarking: break the original image which consists of four phases in encoding through watermarking. The image is taken as input and then it is further divided into subblocks of 8×8, and for each individual block, representation of four phases is used. A predictor of three different types is utilized for the expected value of the pixel. A window of 3×3 is used in both perpendicular and parallel directions so that the initial position of the pixel varies. Pixels in the neighboring are used for the variance calculation, and as a result, predicted values are found out. To improve the quality of the results, projected error is also calculated.

(b) Utilizing sliding window four-stage representation: taking two as the size of the step, the window glides through the image cover. The center value of the pixel is to be projected. The context of projection will include eight values of adjacent pixels. The projection will be accomplished by using eight values of adjacent pixels to calculate the variance. Projection values in the range are found out and replaced with the value of a pixel in the center.

(c) Division of the image using four stages: blocks are divided into 8×8 of the image which is taken as input, and on each individual phase, hybrid prediction is applied to find out the pixel projection value. 8 adjacent values of pixels are used to compute the variance which is used to find out the

value of projection. The procedure is repeated for the rest of the blocks. Encoding and decoding are done on the phases.

(d) Hybrid projection: the reason to use hybrid projection is to lessen the distortion in the image which occurs owing to reversible watermarking. Here, the position is found out to embed the watermark. According to the different adjacent ranges of the variance, three types of projection are used. The projection comes in the range of 0–10 considered as low range of variance, 10–40 as a medium range of variance, and greater than 40 as a high range of variance. Mean projection is utilized for a low range of variance, the trimmed mean projection is utilized for a medium range of variance, and MED projection is utilized for a high range of variance. For the computation of the variance, mean projection is used. For the calculation of the variance, trimmed mean projection uses the 8 adjacent pixel values. For finding the projection value, the mean of the values sorted is utilized. The window is slid over the image to calculate the variance. Change in the context of the projection means the order of sorting is changed in the encoding and decoding. Variance is computed using the following equation:

$$\sigma x = \frac{1}{8}\sum(i = 1, \ldots, 8)(xi - \mu), \tag{17}$$

where xi represent the adjacent pixel values and μ represents the mean. To define the projection pixel value of the center, the following equation is utilized:

$$C' = \begin{cases} \sum_{i=1}^{8} \frac{X_i}{8}, & \text{if } \sigma_x \geq 0 \text{ and } \sigma_x < 10, \\ \sum_{i=3}^{6} \frac{K_i}{4}, & \text{if } \sigma_x \geq 10 \text{ and } \sigma_x < 40, \\ \sum_{i=2}^{3} \frac{KM_i}{2}, & \text{if } \sigma_x > 40. \end{cases} \tag{18}$$

All four corners of the window are utilized for the MED projection. KM is the total count of the values when arranged in order of increasing.

(e) Error of prediction: the input image is used for finding the projection value of the image. Projection error is computed between the image original version and the projection image, and when the error is expanding, data will be embedded into the image. The value of the edge is set over the error of prediction and depends on how the edge of encoding and decoding is done.

(f) Process of embedding: edge values lie in the range of the error of prediction. If the text fits within the range, then the edge values are utilized; otherwise, the values of error projection will be moved left.

(g) Message digest 5 algorithm: for the secure transfer of the data, MD5 has been used which generates a 128-bit value of hash. This value of the hash is represented in the hexadecimal number. Further, this value will be converted into the binary form using LSB embedding. For the secure data transmission, both hash values should be matched the one which is extracted and text bit.

(h) Extraction: a decoder is used to recover the original image and data which were embedded. For the full retrieval of the data, it is important to match the hash values of the extracted data and text bit. The original image is recovered by adding the error of the original projection and the value of projection.

(i) Experimental results: PSNR and the capacity of embedding performance metrics are used for reversible watermarking. Both are inversely proportional to each other. A higher value of PSNR means the quality of the image is good. A medical image of size 256×256 is used. Testing is important for the complete recovery of the image. The results are shown in Table 3.

3.2.8. Reversible Watermarking for Medical Video. Faduao and Hamid proposed the reversible watermarking technique for medical videos by exploiting the polynomial transformation methods [24]. Polynomial transformation methods are used for the identification of suitable watermark regions. The cover video is initially fragmented into keyframes, and subsequently, these frames are shattered into substantial components for opting suitable regions for watermarking.

After selecting the keyframes of individual video shots, their location is persisted in key information K.

Polynomial transform is applied to the selected keyframes to obtain a textured region. After dividing the texture component into 8∗8 blocks, the watermark w_i bit is inserted in k blocks $\mathrm{FB}_i(\mathrm{FB}_i, \ldots, \mathrm{FB}_k)$.

Watermark insertion is done by the following equation:

$$\mathrm{FB}_i^* = \mathrm{FB}_i(1 + w_i * S), \tag{19}$$

where S is the strength of watermarking. Watermark blocks are used to construct the watermarked texture. Original structure parts and watermarked texture are combined to obtain a watermarked image I_w.

Similar action with inverse logic is applied for the extraction of the watermark. The watermarked video is divided into shots. Key information K (as provided before) is utilized to determine the keyframe from the individual shot with its location. Polynomial transform was applied to disintegrate the watermarked frame F_w into the best regions: watermarked texture. The watermarked texture component is segmented into 8∗8 blocks to get the altered frame's blocks $\mathrm{FB}^*(\mathrm{FB}^*, \mathrm{FB}^*_{\mathrm{k}})$.

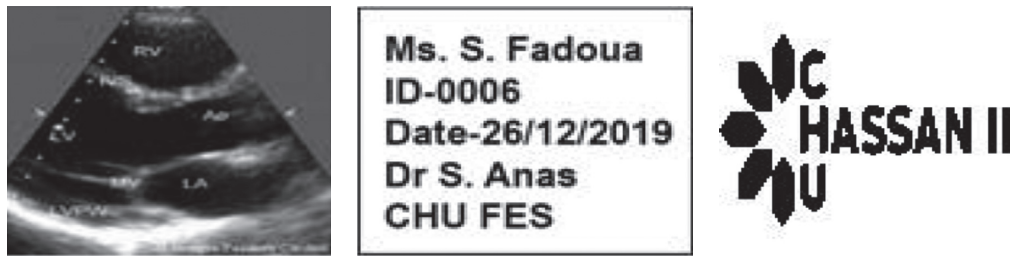
$w_i^* = \mathrm{FB}_i^*/\mathrm{FB}_i$ is utilized to compute the new coefficients (w_{11}^*, w_{88}^*). If $w_i^* < 0$, then $\mathrm{watermark}_i = 0$ bit; else, $\mathrm{watermark}_i = 1$ bit, and this way, watermark is constructed.

Two optimized frames as keyframes from individual video shots of echocardiography are used. 512×512 is frame size medical video whereas the watermark size is 120×120. The vital ingredient for these tests is opted experimentally and fastened on $S = 0.1$ for this particular technique. Figure 18 demonstrates the sample frame of the original medical video and two watermarks, which are utilized for visual illustration.

PSNR and NC are used for performance assessment of the abovementioned scheme. For shot 1 and shot 2 obtained, PSNR results are 60.1278 dB and 60.4540 dB, respectively,

Table 3: Medical image capacity and PSNR.

Positive threshold	Heart	Brain	Kidney	Liver	Ultrasound
Embedding capacity					
2	0.0007	0.0058	0.0048	0.0050	0.0010
5	0.0047	0.0065	0.0079	0.0060	0.0014
7	0.0048	0.0070	0.0230	0.0068	0.0015
10	0.0052	0.0074	0.0250	0.0076	0.0120
15	0.0396	0.0098	0.0312	0.0085	0.0170
PSNR					
2	74.31	73.67	75.25	72.88	63.13
5	64.51	72.17	68.07	63.21	60.66
7	63.94	66.18	64.80	59.11	55.40
10	61.95	58.41	59.43	54.35	55.35
15	53.86	50.21	52.45	48.75	43.89



(a) (b) (c)

Figure 18: (a) Ultrasound, (b) watermark 1, and (c) watermark 2 [24].

Table 4: Comparison of reviewed watermarking techniques.

Articles	Parameters							
	Reversible	Robust	PSNR (dB)	NC	SSIM	Temper detection	Recovery	Media type
Turuk and Dhande [41]	✓	×	53.64	×	~1	×	×	Images
Thakur et al. [43]	✓	✓	74.60	0.910	0.999	×	×	Images
Habib and Al-Fayez [36]	✓	Semifragile	Inf	×	≈0.92	×	×	Video
Faduao and Hamid [24]	✓	✓	60.2909	1.0000	×	×	×	Video
Agung et al. [25]	✓	×	58.5	×	×	✓	✓	Images
Al-Qershi and Khoo [28]	✓	×	37	×	×	✓	✓	Images
Rŏcek et al. [37]	✓	×	81	×	0.999974	×	×	Images
Liew and Zain [26]	✓	✓	23.5042	×	×	×	✓	Images

whereas for NC result obtained for the same both shots are 1.0. Hence, the performance of this technique was without exhibiting any apparent variance between actual and watermarked videos. Besides, it addressed the watermark capacity problem as well.

For the evaluation of the robustness of the technique, watermarked frames were subjected to 3 different attacks: JPEG compression, median filtering, and histogram equalization. The techniques illustrated above are compared in Table 4.

4. Conclusion

In today's era, the protection of the data is crucial to overcome the problems of tempering, integrity, and authentication of data and the rights of ownership. A comprehensive survey of reversible watermarking techniques of echocardiography was presented. The increasing spectrum of a threat to digital security demands consistent efforts for safeguarding data integrity. The intricacy of conflicting performance parameters in data hiding and reversible watermarking makes it difficult to put forth a comprehensive technique. This provides heaps of room for improvements in this field in the near and far future.

Data Availability

The detailed results which cannot be included in this article due to the length of the article will be available on the authors' website after the acceptance/publication of the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The author Dr. Sajjad Shaukat Jamal extends his gratitude to the Deanship of Scientific Research at King Khalid University for funding this work through research groups program under Grant number R.G.P. 1/234/41.

References

[1] J. A. Panza, "Real-time three-dimensional echocardiography: an overview," *The International Journal of Cardiac Imaging*, vol. 17, no. 3, pp. 227–235, 2001.

[2] J. S. Gottdiener, "Overview of stress echocardiography: uses, advantages, and limitations," *Current Problems in Cardiology*, vol. 28, no. 8, pp. 485–516, 2003.

[3] T. Araki, N. Ikeda, F. Molinari et al., "Effect of geometric-based coronary calcium volume as a feature along with its shape-based attributes for cardiological risk prediction from low contrast intravascular ultrasound," *Journal of Medical Imaging and Health Informatics*, vol. 4, no. 2, pp. 255–261, 2014.

[4] S. Acharjee, R. Ray, S. Chakraborty, S. Nath, and N. Dey, "Watermarking in motion vector for security enhancement of medical videos," in *Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kanyakumari, India, July 2014.

[5] B. Lei, E. L. Tan, S. Chen, D. Ni, T. Wang, and H. Lei, "Reversible watermarking scheme for medical image based on differential evolution," *Expert Systems with Applications*, vol. 41, 2014.

[6] S. M. Mousavi, A. Naghsh, and S. A. R. Abu-Bakar, "Watermarking techniques used in medical images: a survey," *Journal of Digital Imaging*, vol. 27, no. 6, pp. 714–729, 2014.

[7] W. Mazurczyk and S. Wendzel, "Information hiding:Challenges for forensic experts," *Communications of the ACM*, vol. 61, pp. 86–94, 2018.

[8] M. T. Ahvanooey, Q. Li, H. J. Shim, and Y. Huang, "A comparative analysis of information hiding techniques for copyright protection of text documents," *Security and Communication Networks*, vol. 2018, Article ID 5325040, 22 pages, 2018.

[9] A. Singh, B. Kumar, G. Singh, and A. Mohan, *Medical Image Watermarking: Techniques and Applications*, Springer, Berlin, Germany, 2017.

[10] C. K. Tan, J. C. Ng, X. Xu, C. L. Poh, Y. L. Guan, and K. Sheah, "Security protection of DICOM medical images using dual-layer reversible watermarking with tamper detection capability," *Journal of Digital Imaging*, vol. 24, no. 3, pp. 528–540, 2011.

[11] L. Chia-Pin, J. Honye, C.-J. Chang, and C.-T. Kuo, "Clinical application of intravascular ultrasound in coronary artery disease: an update," *Acta Cardiologica Sinica*, vol. 27, 2011.

[12] L. Singh, A. K. Singh, and P. K. Singh, "Secure data hiding techniques: a survey," *Multimedia Tools and Applications*, vol. 79, no. 23-24, pp. 15901–15921, 2020.

[13] A. Umamageswari, M. Ferni Ukrit, and G. R. Suresh, "A survey on security in medical image communication," *International Journal of Computer Applications*, vol. 30, no. 3, pp. 41–45, 2011.

[14] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding-a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[15] D. D. Brabin and J. J. Tamilselvi, "Reversible data hiding: a survey," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 3, 2013.

[16] A. Sonal and C. Lakshmi, "A review on reversible data hiding techniques," *International Journal of Applied Engineering Research*, vol. 13, no. 5, pp. 2857–2864, 2018.

[17] J. A. Kaw, N. A. Loan, S. A. Parah, K. Muhammad, J. A. Sheikh, and G. Bhat, "A reversible and secure patient information hiding system for IoT driven ehealth," *International Journal of Information Management*, vol. 45, 2019.

[18] R. Geetha and S. Geetha, "Improved reversible data embedding," in *Proceedings of the International Conference on Next Generation Computing Technologies*, Dehradun, India, November 2018.

[19] T. Dutta, R. Bagi, and H. Gupta, "Robust reversible watermarking for grayscale medical images," in *Advances in Data and Information Sciences*Springer, Berlin, Germany, 2020.

[20] S. A. Parah, F. Ahad, J. A. Sheikh, and G. M. Bhat, "Pixel repetition technique: a high capacity and reversible data hiding method for E-healthcare applications," in *Intelligent Techniques in Signal Processing for Multimedia Security*-Springer, Berlin, Germany, 2017.

[21] R. Bhardwaj and A. Aggarwal, "Hiding clinical information in medical images: an enhanced encrypted reversible data hiding algorithm grounded on hierarchical absolute moment block truncation coding," *Multidimensional Systems and Signal Processing*, vol. 31, no. 20, pp. 1051–1074, 2020.

[22] R. Geetha and S. Geetha, "Embedding electronic patient information in clinical images: an improved and efficient reversible data hiding technique," *Multimedia Tools and Applications*, vol. 79, no. 20, pp. 12869–12890, 2020.

[23] N. Dey, P. Das, A. B. Roy, A. Das, and S. S. Chaudhuri, "DWT-DCT-SVD based intravascular ultrasound video watermarking," in *Proceedings of the 2012 World Congress on Information and Communication Technologies*, Trivandrum, India, November 2012.

[24] S. Fadoua and T. Hamid, "Medical video watermarking scheme for telemedicine applications," in *Proceedings of the 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Meknes, Morocco, March 2020.

[25] T. Agung, B. W. Adiwijaya, and F. P. Perman, "Medical image watermarking with tamper detection and recovery using reversible watermarking with LSB modification and run length encoding (RLE) compression," in *Proceedings of the 2012 IEEE International Conference on Communication, Networks and Satellite (ComNetSat)*, vol. 3, Bali, Indonesia, July 2012.

[26] S.-C. Liew and J. M. Zain, "Reversible medical image watermarking for tamper detection and recovery with run length encoding compression," *World Academy of Science, Engineering and Technology*, pp. 674–678, 2010.

[27] J. Zain and A. Fauzi, "Medical image watermarking with tamper detection and recovery," in *Proceedings of the 2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, New York, NY, USA, September 2006.

[28] O. M. Al-Qershi and B. E. Khoo, "ROI-based tamper detection and recovery for medical images using reversible watermarking technique," in *Proceedings of the 2010 IEEE*

International Conference on Information Theory and Information Security, Beijing, China, December 2010.

[29] Khoo and O. M. Al-Qershi, "Reversible watermarking scheme based on two-dimensional difference expansion (2D-DE)," in *Proceedings of the The 2010 International Conference on Computer Research and Development*, Kuala Lumpur, Malaysia, May 2010.

[30] M. G. Jessica Fridrich, "Invertible authentication," in *Proceedings International Conference on Information Technology: Coding and Computing*, Las Vegas, NV, USA, April 2001.

[31] N. Dey, P. Das, A. Das, and S. S. Chaudhuri, "Feature analysis for the blind–watermarked electroencephalogram signal in wireless telemonitoring using alattar's method," in *Proceedings of the Fifth International Conference on Security of Information and Networks-SIN'12*, New York, NY, USA, October 2012.

[32] N. Dey, S. Mukhopadhyay, A. Das, and S. S. Chaudhuri, "Analysis of P-QRS-T components modified by blind watermarking technique within the electrocardiogram signal for authentication in wireless telecardiology using DWT," *International Journal of Image, Graphics and Signal Processing*, vol. 4, 2012.

[33] N. Dey, A. B. Roy, A. Das, and S. S. Chaudhuri, "Stationary wavelet transformation based self-recovery of blind-watermark from electrocardiogram signal in the wireless telecardiology," *Communications in Computer and Information Science,Recent Trends in Computer Networks and Distributed Systems Security*, Springer, Berlin, Germany, 2012.

[34] N. Dey, P. Maji, P. Das, A. Das, and S. S. Chaudhuri, "An edge based watermarking technique of medical images without devalorizing diagnostic parameters," in *Proceedings of the International Conference on Advances in Technology and Engineering (ICATE)*, Mumbai, India, January 2013.

[35] G. Coatrieux, H. Sankur, and B. Rolland, "Relevance of watermarking in medical imaging," in *Proceedings 2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine*, Arlington, VA, USA, November 2000.

[36] R. Habib and F. Al-Fayez, "Protection of ultrasound image sequence: employing motion vector reversible watermarking," *International Journal of Advanced Computer Science and Applications*, vol. 10, 2019.

[37] A. Rŏcek, K. Slavíček, O. Dostál, and M. Javorník, "A new approach to fully-reversible watermarking in medical imagingwith breakthrough visibility parameters," *Biomedical Signal Processing and Control*, vol. 29, pp. 44–52, 2016.

[38] A. Rŏcek, K. Slavíce, and M. Javorník, "RONI size and another attributes ofrepresentative sample of medical images in common hospital operation,related to securing by watermarking methods," in *Proceedings of the International Conference on Image Processing, Production and Computer Science*, London, UK, October 2016.

[39] J. J. Eggers, R. Bauml, R. Tzschoppe, and B. Girod, "Scalar costa scheme for information embedding," *IEEE Transactions on Signal Processing*, vol. 51, 2003.

[40] S. Rani and R. Euphrasia, "Dynamic hiding of message in rgb domain based on random channel indicator," *International Journal of Applied Engineering Research*, vol. 10, 2015.

[41] M. Turuk and A. Dhande, "A novel texture-quantization-based reversible multiple watermarking scheme Applied to health information system," *Journal of Digital Imaging*, vol. 31, no. 2, pp. 167–177, 2018.

[42] X. Liu, J. Lou, H. Fang et al., "A novel robust reversible watermarking scheme for protecting authenticity and integrity of medical images," *IEEE Access*, vol. 7, pp. 76580–76598, 2019.

[43] S. Thakur, A. K. Singh, S. P. Ghrera, and M. Elhoseny, "Multi-layer security of medical data through watermarking and chaotic encryption for tele-health applications," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3457–3470, 2019.

[44] F. A. Memon and N. Ahmed, "Reversible watermarking for the security of medical image databases," in *Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC)*, Riyadh, Saudi Arabia, April 2018.

[45] Lakshmi and S. Ayyappan, "A review on reversible data hiding techniques," *International Journal of Applied Engineering Research*, vol. 48, pp. 2857–2864, 2018.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 8868355, 23 pages
https://doi.org/10.1155/2021/8868355

Review Article

Machine Learning Technologies for Secure Vehicular Communication in Internet of Vehicles: Recent Advances and Applications

Elmustafa Sayed Ali,[1,2] **Mohammad Kamrul Hasan**,[3] **Rosilah Hassan**,[3] **Rashid A. Saeed**,[4] **Mona Bakri Hassan**,[1] **Shayla Islam**,[5] **Nazmus Shaker Nafi**,[6] **and Savitri Bevinakoppa**[6]

[1]*Department of Electronics Engineering, Sudan University of Science and Technology, Khartoum, Sudan*
[2]*Department of Electrical and Electronics Engineering, Red Sea University, Port Sudan, Sudan*
[3]*Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia*
[4]*Department of Computer Engineering, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*
[5]*Department of Computer Science, Institute of Computer Science and Digital Innovation, UCSI University Malaysia, 56000 Kuala Lumpur, Malaysia*
[6]*Schools of IT and Telecommunication Engineering, Melbourne Institute of Technology, Melbourne, Australia*

Correspondence should be addressed to Mohammad Kamrul Hasan; hasankamrul@ieee.org

Received 21 July 2020; Revised 30 January 2021; Accepted 24 February 2021; Published 13 March 2021

Academic Editor: Fawad Ahmed

Recently, interest in Internet of Vehicles' (IoV) technologies has significantly emerged due to the substantial development in the smart automobile industries. Internet of Vehicles' technology enables vehicles to communicate with public networks and interact with the surrounding environment. It also allows vehicles to exchange and collect information about other vehicles and roads. IoV is introduced to enhance road users' experience by reducing road congestion, improving traffic management, and ensuring the road safety. The promised applications of smart vehicles and IoV systems face many challenges, such as big data collection in IoV and distribution to attractive vehicles and humans. Another challenge is achieving fast and efficient communication between many different vehicles and smart devices called Vehicle-to-Everything (V2X). One of the vital questions that the researchers need to address is how to effectively handle the privacy of large groups of data and vehicles in IoV systems. Artificial Intelligence technology offers many smart solutions that may help IoV networks address all these questions and issues. Machine learning (ML) is one of the highest efficient AI tools that have been extensively used to resolve all mentioned problematic issues. For example, ML can be used to avoid road accidents by analyzing the driving behavior and environment by sensing data of the surrounding environment. Machine learning mechanisms are characterized by the time change and are critical to channel modeling in-vehicle network scenarios. This paper aims to provide theoretical foundations for machine learning and the leading models and algorithms to resolve IoV applications' challenges. This paper has conducted a critical review with analytical modeling for offloading mobile edge-computing decisions based on machine learning and Deep Reinforcement Learning (DRL) approaches for the Internet of Vehicles (IoV). The paper has assumed a Secure IoV edge-computing offloading model with various data processing and traffic flow. The proposed analytical model considers the Markov decision process (MDP) and ML in offloading the decision process of different task flows of the IoV network control cycle. In the paper, we focused on buffer and energy aware in ML-enabled Quality of Experience (QoE) optimization, where many recent related research and methods were analyzed, compared, and discussed. The IoV edge computing and fog-based identity authentication and security mechanism were presented as well. Finally, future directions and potential solutions for secure ML IoV and V2X were highlighted.

1. Introduction

Intelligent Transportation Systems (ITS) and computational systems' rapid development opened new scientific research in smart traffic safety with comfort and efficient solutions. Artificial Intelligence (AI) has been widely used to optimize traditional data-driven approaches in different research areas [1]. AI-based on the Vehicle-to-Everything (V2X) system obtains information from various sources, i.e., car, train, bus, etc., and enables to increase the realization of drivers and forecast to avoid accidents. This progression has directed to the opportunity to understand smart driving, which was built on the idea of copying real driving comportment, while avoiding human mistakes and bringing comfortable safety to drivers. Many services have been invented from crowd and light road traffic to adapting traffic, a legacy from self-based vehicle systems to the IoV [2]. IoV is addressed to change the interaction between the vehicles, roadside stations, on-board stations, and environments to communicate data and multimedia between various networks. The motivation of IoV is to be adopted and build the human-vehicle-roadside onboard IoT Connected services within the various vehicle and different networks.

Machine Learning (ML) is responsible for a wide range of AI applications. The ML techniques are unsupervised, supervised, and reinforcement learning. In the unsupervised ML scheme, training depends on untagged data. It tries to find an adequate representation of untagged data. While, in supervised learning, it learns from a group of labeled data. In supervised learning, regression and classification schemes train the discrete and continuous data for prediction and decision-making. Reinforcement learning (RL) studies from the learning agent's activities from the consistent reward to capitalize on the notion of cumulative rewards. The Markov Decision Process (MDP) is a sample of RL [2]. This scheme is a perfect technique for taking many issues' research problems in vehicular networks, such as in collaborative optimization of oil consumption for a specific area and optimum path forecasting of electric vehicles and minimizing traffic congestions.

Given the importance of the use of Artificial Intelligence (AI) in IoV, as it provides smart models in most of its applications, this paper contributes a brief concept on one of the AI methods known as machine learning and the possibility of its use in several specific aspects related to the IoV network. In IoV networks, edge computing and caching problems are the most considered challenges requiring an intelligent optimization method. Edge computing and caching challenges are related to many factors, i.e., channel condition, dynamic communication topology, and resource allocation management. In the IoV network architecture, artificial intelligence is in a separated layer responsible for virtual cloud infrastructure. The AI layer act as an information management brain. Deep neural networks are ML algorithms developed to make decisions according to learned IoV resource actions [3].

This paper has conducted a critical review with analytical simulation for offloading mobile edge computing decisions based on learning and Deep Reinforcement Learning (DRL) technologies for vehicular communication in (IoV). We have considered a typical IoV network architecture with one IoV Edge-Computing (IoVEC) and one mobile user. The tasks of the device arrive as a flow in time. Our analytical model performs the offloading decision process of the task flow as a Markov Decision Process (MDP). The optimization object minimizes the weighted sum of offloading latency and power consumption, which is decomposed into the reward of each time slot.

The rest of the paper is organized as follows; the study background and motivation are presented in Section 2, where systematic technical knowledge and motivation of secure ML in the IoV field were discussed. A brief concept of AI in IoV is reviewed in Section 3 by considering using AI in multimedia and IoV edge-based and Vehicle-to-Everything's Internet communications. Section 4 provides a clear concept about the contribution of AI to enabling QoS and QoE optimization, where QoS manages and controls resources of the IoV network by setting various priorities for each data type, while QoE discusses the measurement of the overall system homogeneity and stability of service. Section 5 provides a detailed description of using machine learning algorithms with IoV in different aspects. The most common use cases of ML in IoV applications are presented in Section 6. Section 7 gives a brief review of the possible future research directions and potential ML solutions in IoV. Finally, the conclusion is presented in Section 8.

2. Background and Motivation

Due to the significant research and technology development in wireless communication, the traditional ITS has to care about the vehicular communication field. Recently, the numbers of vehicles have increased due to transporting huge numbers of people from region to region. This increment in the number of vehicles would create issues such as crowding and accidents on the roads. This issue could be considered as one of the main problems in daily life. Most of the general form of vehicular networking is known as the vehicular ad hoc network (VANET) [4]. VANET consists of Vehicle-to-Vehicle (V2V) and Vehicle-to-Roadside (V2R) communications to transfer the vehicles' information. The VANETs' communication depends on the Roadside Unit (RSU) to support Wireless Access in Vehicular Environments (WAVE).

The Roadside Unit (RSU) along the roadwork acts as wireless access points' support communication to the vehicles inside its coverage area [5]. The hybrid vehicular network architecture, interacted with the cellular communication architectures, will operate the cellular communication services, i.e., voice, in collaborations. Due to the current trend to connect vehicular networks to information centers and the need to exchange data, IoV allows enabling Internet access among on-road vehicles. One of the essential IoV applications is to improve the features of VANETs to reduce various issues in urban traffic and accident environments [6]. IoV enables the vehicular road networks to interconnect with different wireless network technologies i.e., Wi-Fi and 4G/LTE for V2I, IEEE WAVE for V2V and

V2R, MOST/Wi-Fi for V2S, and CarPlay NCF for V2P. It is useful to provide a comprehensive presentation to ML's concepts in IoV and explain the areas that could contribute to these networks' development [7].

In recent years, the arising need to introduce artificial intelligence technologies in IoV applications has been facing some challenges. These challenges are related to making particular decisions and forecasting different aspects of IoV, such as traffic monitoring and management, big data processing, energy and resource management, and intelligent interaction with users to provide high-quality services [6, 7]. Several studies have been conducted on using artificial intelligence techniques such as machine learning to develop solutions to most of these challenges [8]. Due to the current developments in the field of AI, especially in using machine learning techniques to make intelligent decisions in several IoV applications, it is useful to provide a comprehensive presentation to study some concepts of using ML in IoV and explain the areas that could contribute to the development of these networks.

3. Artificial Intelligence Methods in the IoV Network

AI technology is more related to the layer responsible for presentation and functionalities in the IoV-layered architecture. A term of virtual cloud infrastructure can describe this layer and be responsible for storing, processing, analyzing the information received from the IoV network, and decision-making based on the analyzed information. In IoV, the computation and analysis are provided by Big Data Analysis (BDA) and Vehicular Cloud Computing (VCC) systems which are used as an information management center [9]. According to the IoV applications, many services can be provided by the IoT cloud environment, requiring intelligent service management. The smart cloud-computing servers provide many smart services, i.e., safety, traffic administration, entertaining, and subscription, which are the foundation of elegance in IoV. The cloud servers based on AI enable the procedure and develop AI in Real-Time (RT) massive data traffic to provide a smart decision for intelligent customer services. The Vehicular Cyber-Physical System (VCPS) is considered a vehicular network model that concerns disseminating information using next-generation Internet [10]. VCPS depends on AI technology to provide smart processing in huge data traffic utilizing fog and cloud computing for civilian and safety applications.

In IoV networks, edge computing and caching problems are the most considered challenges requiring an intelligent optimization method. Edge computing and caching challenges are related to many factors, i.e., channel condition, dynamic communication topology, and resource allocation management. AI in IoV provides an intelligent approach to solve most of these challenges. The use of ML offers a means of interaction to the IoV environment and enables the creation of an agent that learns challenging factors to optimize the overall IoV network utilization [11]. Q-learning and deep neural networks are ML algorithms developed to make decisions according to learned IoV resource actions. In the IoV network architecture, the presentation of artificial intelligence in a separated layer is responsible for virtual cloud infrastructure. The AI layer acts as an information management brain [10, 11]. The AI layer in IoT architecture consists of big data analysis, cloud computing, and expert systems. It plays an essential task in storing, processing, and analyzing the information received from the coordination layer and takes decisions according to the network status.

3.1. Artificial Intelligence Methods for IoV Multimedia Communication. The deployment of IoV in multimedia communications requires a device that allows data exchange and communication with other surrounded devices. This can be achieved by any technology such as Personal Area Networks (PAN), the Internet of Things (IoT), and Wireless Sensor Network (WSN). Data exchange's scalability and flexibility are quite important for IoV by integrating sensors, vehicles, humans, actuators, machines, etc. The sensor in intelligent IoV enhances vehicle and traffic systems' safety, while harmonized traffic data transfer in the IoV system network enhances vehicular system efficiency. However, the amount of energy consumption, required capacity, green buffer-awareness, and message exchange through IoVs may compromise severe data transfer risk [12]. AI based on self-driven vehicles encourages several types of applications with many benefits of intelligence. Especially for the increase in the amount of data and complexity, which the algorithms will be processing, it is precise and effective for future directions. As growing, the high traffic information in IoVs required a smart utility, followed to efficiently monitor and manage the demand for intelligent IT technologies [13]. With the rapid evolution in digital technologies, the development of multimedia depends on the IoV system, and it needs a portable device to collect a voluminous amount of information for aiding and guiding the specific trend for analyzing the transportation industry by IoT-based platforms.

Figure 1 shows the structure of multimedia communication through sensor nodes in the IoV system. Its structure consists of three main parts for IoV data and information network techniques and models. The data and information network techniques and models are redeveloped with the central server. The inter and intravehicle network connections among various sections are executed by transferring urgent and sensitive data throughout the vehicle via adaptive and smart wireless communication. The vehicle's client enables QoS monitoring [13, 14]. In this structure, the IoV traffic can be arranged based on the category containing sensitive/standard, prestored, real-time, or high-definition resolution, respectively. To accomplish the real-time and jitter-tolerant data and information exchange with low buffer storage and scarce power supply, it should be fortified to tolerate the raw unprocessed data and information into

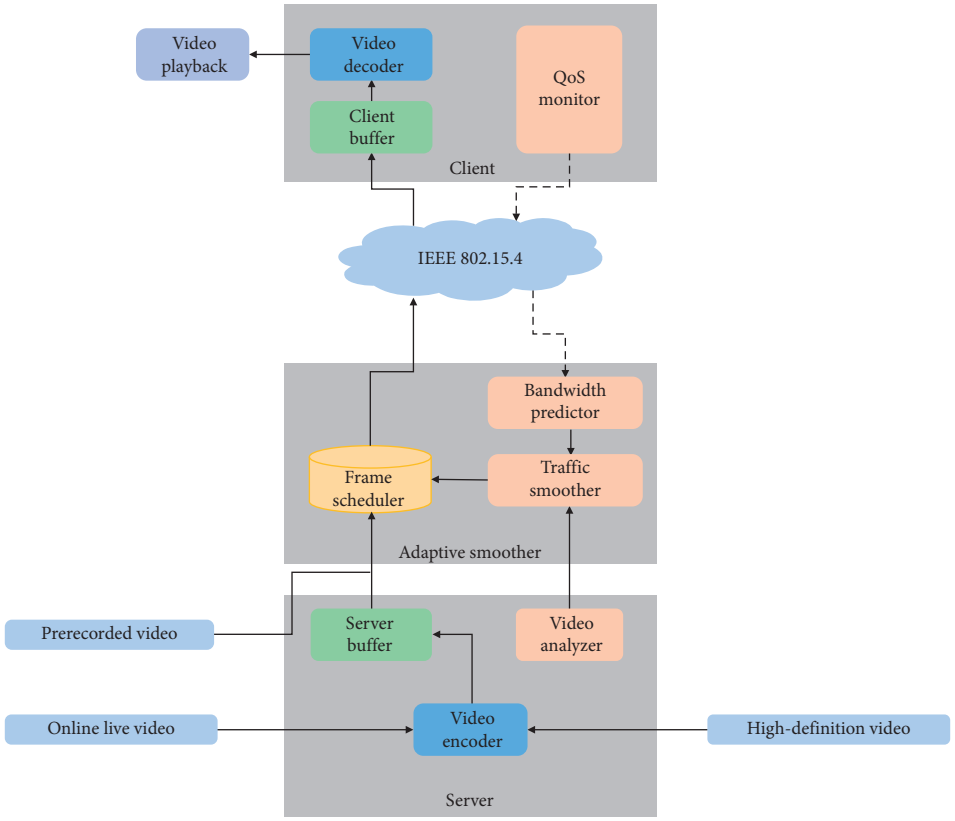


Figure 1: IoV data and information network techniques and models [14].

the regular and synchronized format with good and clean visibility.

3.2. Intelligent IoV Edge-Based Algorithm. IoV Edge Computing (IoVEC) is a new technology that enables vehicles to communicate with cloud computing to directly deliver cloud services from the network edge and support delay-critical IoV applications. It could be achieved by placing computer servers at radio access points or base stations. In edge caching and computing platforms, AI trains and deploys powerful ML models at the edge servers and mobile devices. Edge AI techniques changed the structure of the semiconductor industry [15]. In IoV, the Edge Information System (EIS) plays a vital and unique role. It is able to help the key functionalities of intelligent vehicles, from data acquisition and data processing to actuation. Data processing in the network edge can satisfy the low-latency requirement for mission-critical tasks and save an amount of communication bandwidth. The AI edge-based IoV typically has high spatial locality for road conditions, map information, and temporal locality for traffic conditions. On the contrary, with big sensing data, intelligent vehicles are facing tremendous computation burdens [12, 15].

Offloading computation and load balancing are the most critical factors that determine the maximum system utility in IoV. Cooperative edge caching and edge computing can serve to improve the performance of these factors. But indeed, the edge computing and caching policies are limited in dynamic systems' applications such in IoV networks. AI cognitive capability helps develop edge cognitive computing architecture to provide dynamic computing service [16]. AI cognitive capability will improve energy efficiency and user experience since it is able to interact with other IoV components to perform efficient resource management, as shown in Figure 2. IoV architecture-based AI algorithms enable the perception of vehicular environment information's real-time behavior by interacting with the environment according to the current state related to offloading, cooperative caching, and edge computing [15, 16].

The IoV edge-based AI architecture can efficiently drive the edge computing resources depending on cooperative caching to manage edge computing policies. Such edge-based AI architectures can use deep ML algorithms for efficient IoV resource management. Other considerations related to system utility are IoV network mobility and vehicles'/RSUs' handover mechanisms. These considerations are significant factors that significantly affect temporary storage resources [15]. Therefore, it is necessary to the trade-off between the accuracy of the prediction, the temporary storage of content on the move, and the handoff implementation. AI enables the prediction of handover and intelligent sharing of allocated bandwidth and edge caching.

3.3. Artificial Intelligence Methods for Vehicle-to-Everything. In vehicular applications, AI enables executing tasks intelligently, such as enhancing the Plug-in Electric Vehicle

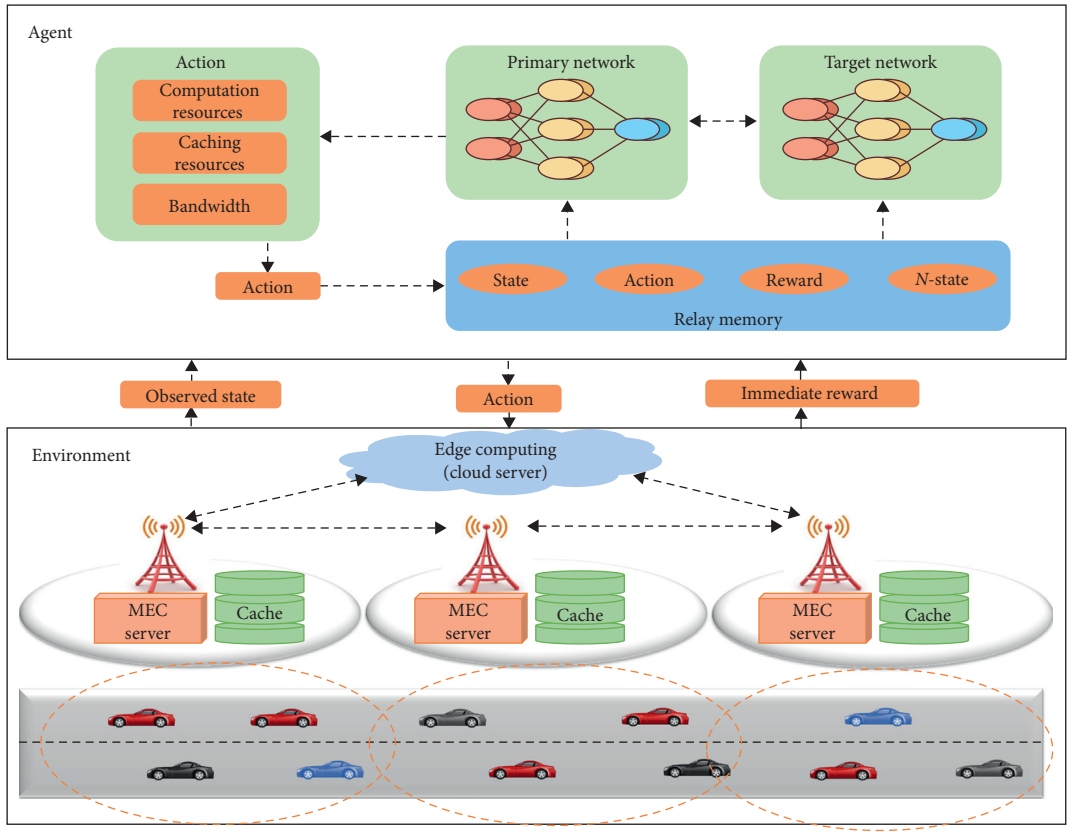


Figure 2: IoV edge-based AI architecture.

(PEV) charge, minimizing fuel consumption, enhancing location-based services, and traffic congestion rectification. The traffic flow information can be obtained from multiple sources such as induction loops, crowd sourcing-based information services and vehicles, and Closed-Circuit Television (CCTV) cameras [17]. Modeling precise and accurate traffic exchange prediction procedures utilizing legacy traffic flow prediction mechanisms is a vital problematic issue. AI techniques have been extensively used for modeling estimation mechanisms in research areas such as robotics, data science, computer vision, natural language processing, and medicine. AI used the data-driven method that facilitates it more efficiently to tackle little and multimedia data. The aims of V2X technology to transportation systems are to enhance safety and efficiency by sharing data among vehicles, infrastructures, and walkers. V2X schemes received a tremendous amount of use in academia, industry, and governments.

There are three fundamental aspects of a V2X communication system: road safety, energy efficiency, and traffic efficiency [18]. The V2X scheme is based on sharing information among Vehicle-to-Infrastructure (V2I), Vehicle-to-Vehicle (V2V), Vehicle-to-Self (V2S), Vehicle-to-Pedestrian (V2P), and Vehicle-to-Road side units (V2R). V2X is an evaluation technology for vehicular networks. AI with V2X can enable new approaches to applications such as traffic flow prediction and management for real-time data, location-based applications, vehicular platoons, data storage in vehicles, autonomous transport facilities, and congestion control. The most widely utilized AI techniques are Heuristic Techniques, Robotics, Game-Theoretic Learning, Expert Systems, Evolutionary Algorithms, Turning Test, Logical AI, Planning, Schedule and Optimization, Natural Language Processing, Swarm Intelligence, Inference, Fuzzy Logic, and Machine Learning [19]. One of the ML-based V2X is autonomous driving where AI is used to enable essential features of human driving. ML in V2X can play a critical role in enhancing safety and efficiency of in-vehicle networks [18]. Modern machines have widely applied it for applications such as competing at the highest level in strategic games, autonomous vehicles, understanding human speech, and intelligent network routing in content delivery networks.

Other considerations are related to security in V2X applications. AI provides many security mechanisms for routing protection against threats and attacks. In addition, the AI swarm intelligent algorithms protect against malicious vehicle attacks. A DL-based technique for anomaly detection in V2X vehicles provides a means of security against different kinds of attacks, i.e., Denial of Service (DOS), rushing attacks, gray hole, and Sybil attacks. The research work presented by Abdallah Moubayed (2020) reviews the concept of using machine learning in fifth generation (5G) IoV for security issues, in addition to discussing various challenges faced by V2X communications [20]. The study presents the considerations related to V2X security and privacy and illustrates different kinds of attacks

related to authentication, confidentiality, data integrity, and accountability. Zeinab El-Rewini (2020) provided a three-layer automotive security framework considering the issues of control, communication, and sensing [21]. The framework enables eavesdropping, jamming, and spoofing attacks.

Moreover, it has the ability to detect different communication layer attacks in the V2X network such as spoofing, man-in-the-middle, and Sybil attack; the research is also providing a survey on using the machine and deep learning for cybersecurity solutions and V2X network security. Haji M. Furqan (2019) had introduced an intelligent security framework for V2X communication based on the AI radio brain model to enable learning information from higher network layers and radio environment [22]. The security framework detects the vehicle condition and considers the channel information to decide the best-suited security level. In this scheme, decision-making depends on vehicle conditions related to location, environment, utility, time, and application. In automotive V2X, traffic safety is an important issue that needs to make data classification and enable secure permitting of the autonomous vehicle. The defensible decision to pass another vehicle is a critical issue according to its dynamic behavior. Jean-Philippe Monteuuis provided a misbehavior classifier related to data classification for multiple road users using machine learning algorithms [23].

Another V2X traffic issue is location spoofing, which can cause traffic congestion. So (2019) reviewed practical spoofing attacks that can pay to pass the security checks in the V2X application layer [24]. The study proposed three physical layer checks to ensure the detection rate and decrease false positives, in addition to enabling comprehensive evaluation of the performance of the security for several types of attacks. The misbehavior detection is depending on machine learning to security checks at the application layer using the VeReMi datasets which are enhanced datasets for several types of attacks. Kang (2016) used the Intrusion Detection System (IDS) based on Deep Neural Network (DNN) for V2V and V2I networks [27]. The proposed study uses unsupervised Deep Belief Network (DBN) pretraining schemes to train the parameters of the DNN to optimize the learning efficiency. The IDS-based DNN enables to train high-dimensional CAN packet data afterwards, in order to extract the statistical properties of normal and attack packets to identify the attack. This method provides security features against hacking packets and identifies any malicious attack to V2X networks. Table 1 presents a brief summary of different research works considering the use of AI technology in Vehicle-to-Everything's security issues.

4. Artificial Intelligence-Enabled Quality of Experience Optimization (QoE)

In IoV, QoE provides measurement deals with network performance and perception, in addition to IoV application experience. The QoE considers the IoV experience to ensure high quality of data transmission by continuously measuring the QoE of the network and updating. For IoV end users and due to the rapid change of IoV communication topology, the user's quality of experience is considered as one of the main challenges in IoV networks [28]. The flexible and scalable connection between integrated components of the IoV system i.e., vehicles, sensors, actuators, humans, and machines, is vital for IoV, which must fit with the requirement of user perception enhancement to decrease the power consumption. Moreover, to improve transportation systems' safety and traffic data exchanging in vehicular networks, power and buffer-aware QoE/QoS via IoV came with a high risk of quality compromise during sensitive IoV applications i.e., in the medical field. For such reason, cost-effective power and buffer-aware QoE optimization solutions for designing and deploying the IoV are required. Quality of Service (QoS) in IoV is related to the routing path quality, impact of velocity, position of vehicles, and network topology. These aspects mainly affect IoVs' energy efficiency [29]. The QoS optimization with energy efficiency regarding the IoV network efficiency is about developing a solution to the multiattribute decision-making and being able to optimize many IoV network operations.

AI techniques have changed the landscape of the IoV through multimedia communication. It improves the overall IoV network by efficiently optimizing the route selection to obtain stable transmitting multimedia content in the IoV system [14, 29]. AI can also help develop energy and buffer-aware optimization mechanisms to optimize the QoE and QoS during multimedia communication in the IoV system. Machine Learning (ML) techniques can provide a framework to analyze the QoE services with the high level of optimization. ML will help in assessing and examining the faults and quality degrading factors prospected from important collected information by IoV systems to enhance the IoV user's satisfaction, in addition to evaluating the QoS by considering several impacts to the IoV network related to communication, energy, and resource management operations [14].

4.1. Buffer-Aware QoE/QoS Optimization. Due to the high demand for video traffic in IoV networks, the development of intelligent solutions must fulfill the expectations and ensure maximum Quality of Experience (QoE). The optimization of QoE during multimedia communication in the IoV system can be obtained by deploying a novel algorithm based on the buffer allocation mechanism, which enables controlling the high peak variable rate of multimedia by allocating the proper buffer size in IoV. The buffer aware QoE optimization must consider the requirements related to energy and video rate adaptation. In IoV applications based on video transmission, the dynamic adapting coding rate of the requested videos can ensure that optimizing the QoE by the encoding rate depends on the video content itself. Machine learning algorithms provide automatic video processing with the additional complexity given by the data's temporal dimension [30]. ML can achieve different video processing schemes in pixel level or higher-level

Table 1: Summary of artificial intelligence methods in secure Vehicle-to-Everything networks.

Year	Source	Security approaches	Features	Advantages	Challenges	Citations
2020	ArXiv	NSL-KDD data mining; Cloud Security Alliance (CSA)	Machine learning in fifth generation (5G) IoV	Security issues related to softwarization, software-defined perimeter, and virtualization	QoS performance and scalability and cost in secure V2X dynamic networks	Abdallah [20]
2020	Elsevier	Controller Area Network (CAN); IDS; Security-Aware FlexRay Scheduling Engine (SAFE); Hardware Security Module (HSM)	AI-based V2X automotive security framework	Detects sensing and communication layers' attacks	Cybersecurity in fully autonomous V2X	El-Rewini [21]
2019	arXiv	Intelligent V2X security (IV2XS); physical layer security (PLS)	Cognitive security based on context-aware proactive security	Security decision-making according to vehicles' channel conditions	Identify the best-suited level of security.	Furqan [22]
2019	WiSec'19	Basic safety messages (BSMs).	Misbehavior detection based on ML for secure V2X traffic	Detects spoofing attacks in the V2X application layer	Identify and detect the V2X location spoofing	So [24]
2018	IEEE	MinMax, MLP, Adaboost, and Random Forest misbehaving classifiers	V2X traffic safety-based ML algorithms	A misbehavior classifier for vehicle data classification	Secure decision for V2X traffic safety	Monteuuis [26]
2016	PLoS ONE	Controller Area Network (CAN) and IDS	Intrusion detection system (IDS) based on deep neural network (DNN)	Extract the statistical properties of normal and attack CAN data packets	Identify malicious attack to V2X networks	Kang [27]

representations obtained after additional preprocessing of raw images. The ML schemes enable the optimization of the process of buffer allocation and dynamic video-rate adaption.

In IoV networks, it is challenging to achieve QoS and efficiency for multimedia streaming, especially in high-mobility features. The buffer-aware streaming approach will allow users to play multimedia streaming over the IoV network. AI-based buffer-aware QoS adopted for vehicle streaming services to evaluate multimedia content preloaded by IoV servers according to the user's mobility information. A buffer-aware QoS streaming approach over the IoV network can provide various priority levels of streaming service [31]. ML will evaluate vehicle mobility's direction and speed, the strength of IoV signals, and the size of media content stored in the buffer to optimize the quality of streaming service on the IoV network.

4.2. Energy-Aware QoE/QoS Optimization. Energy management in IoV systems is considered one of the main challenges faced in IoV applications. It is very important to effectively manage the power resources during communication in the IoV system. In most IoV applications, the Electric Vehicles' (EVs) charging and discharging time negatively impact the QoE. Power-aware QoE Optimization in Vehicle-to-Grid (V2G) networks expresses the degree of satisfaction with the State of Charge (SOC) and charging the cost of using an EV [32]. In the charging schedule, the service of enough CSs is an important QoE metric, especially in the peak charging hours. AI-based charging scheduling schemes must consider the QoE optimization. The QoE of vehicles in the IoV network with a higher vehicle's mobility and limited coverage area of RSU can be degraded and can significantly affect communication quality by decreasing flow satisfaction. In addition, any growth in energy consumption in RSU leads to inefficient IoV energy network management [33]. Moreover, due to the limited IEEE 802.11p-based vehicular communication bandwidth, providing a fair share of network resources among vehicles will face a crucial problem related to flow management [34]. AI-based energy management schemes provide an intelligent decisions' controller to overcome energy operation's complexity by providing efficient solutions. Table 2 presents the key points of most related works in AI technologies that use the IOV QoS/QoE optimizations.

5. Machine Learning Algorithms in the IoV Network

Machine learning has different models, classifications, and training methods widely used for prediction problems and intelligent managing. In IoV applications, Reinforcement Learning (RL) will provide guidance behavior to promote resilience and scalability. It can give path selection or route optimization in IoV networks. The use of ML with the Software Defined Network (SDN) in IoV can ensure delay of minimization and throughput maximization as the operation and maintenance strategy. Together, ML and SDN will improve the IoV network performance with stable and superior routing services [35]. They can ensure optimal routing policy adaptation according to sensing and learning from the IoV environment to achieve better utilization. Figure 3 shows the functions of ML that can be deployed in the IoV networks. In the domain of IoV network security,

Table 2: Artificial intelligence methods in IoV QoS/QoE optimization.

Year	Source	Approaches	Features	Advantages	Challenges	Citations
2020	Sensors	Reinforcement learning; centralized Q-learning	Energy optimization with 5G vehicular social networks	Maximize the energy efficiency and optimization	Ensure communication quality and reduce delays	Park and Lim [33]
2019	IJEAT	SDN-based ML (BAT algorithm)	Prioritize the data packets in IoT cloud storage	Enhance traffic QoS	Traffic delay reduction in IoT multimedia applications	Hasan et al. [28]
2019	Elsevier	Fuzzy-enabled algorithms for buffer and power-aware QoE optimization.	AI-based multimedia communication mechanism and IoV-based QoE optimization framework	Improve multimedia streaming for end users	QOE optimization for multimedia communication in IoV	Sodhro [14]
2018	IEEE	QoE-based ML for video admission control and resource management	Extracting the quality-rate characteristics of unknown video sequences	Improve the service and quality level delivered to end user	Guarantee a minimum service quality level	Islam et al. [30]
2017	EAI	Many-to-one matching game; Stable Matching Algorithm (SMA); Pareto Optimal Matching Algorithm (POMA)	Traveling plan-aware scheduling scheme for EV charging in driving pattern	Improve the QoE in vehicle power grid networks	QoE enhancement in EV industry	Bozkaya and Canberk [34]
2016	Science PG	Fuzzy QoS	Enhance energy efficiency in IoV	Optimize energy QoS	Trade-off between QoS and energy efficiency	Hu [29]

ML with the SDN brings some unique advantages to the deployments of security solutions. For security issues, the centralized control on the software layer with API access will be convenient to develop ML software interaction with the SDN data plane to provide statistical reports to the application layer upon vehicles' requests [36].

In Cognitive Internet of Vehicles' (CIoV) applications, i.e., automatic driving, automation and connectivity are very important in self-driving aspects which should be sufficient of intelligence to reduce road accidents. ML can take control of vehicles to enable error-free driving. CIoV allows cloud-based ML into a transportation system for security risks and privacy issues [37]. In CIoV cognition and control layer, ML provides strategic services for different function levels, i.e., driving behaviors, health monitoring, and pattern and emotion analysis, in addition to network resource allocation and optimization. To improve driving safety and efficiency in the IoV transportation system, deep learning (DL) schemes provide intelligent decision-making to evaluate the critical, influential collision probability factors and risk of possible accidents in the IoV [38]. Different DL techniques can be used for collision prediction and accident forecasting, i.e., Genetic Algorithms (GA), Neural Networks (NN), Fuzzy logic, and Support Vector Machine (SVM).

5.1. ML-Based Edge Caching Mechanisms for IoV. The operational excellence and cost efficiency in IoV depend on the caching and computing design. To efficiently improve the QoS for applications, edge caching placements and computing offloading at the vehicles and the RSUs can ensure efficient QoS. Machine learning provides schemes to tackle problems encountered in caching, computing, and communications for IoV. Many ML schemes can be used for edge caching in IoV [39]. It provides relatively right caching decisions, IoV traffic levels' classification prediction, and content demand in supervised learning. Unsupervised learning can be applied to the edge caching design by clustering numbers' vehicles into different groups according to their behavioral and data request history information [40]. The ML-based clustering scheme can predict the data demand depending on the entire vehicle group's interests or social relations.

The reinforcement learning scheme such as the Q-learning technique will enable distribution cache replacement strategy according to the content popularity. Moreover, it can estimate the unknown popularity of caching contents. Integrated Mobile Edge Computing (MEC) servers in the IoV network will help reduce the workload at roadside stations and make the vehicle requesting content perform data and computation offloading during its movements such as mobility-aware caching and computational scenario, as shown in Figure 4. The use of deep Q-learning will optimize the parameters of caching and computing for resource allocation. Deep Q-learning will determine the optimal actions from the collected status of MEC and RSU servers in addition to each vehicle's mobility, channel information, caching contents, and computing [39, 40]. These actions are forwarded to vehicles. Deep Q-learning will select the best set of caching activity for RSU, MEC, and vehicles to serve the requesting and compute the offloading tasks for the IoV.

Integration of ML with edge caching has challenges related to data processing and analysis. The diffusion and high density of data are challenges for the learning and training process. In addition, insufficient computing resources can manipulate the high-dimensional information

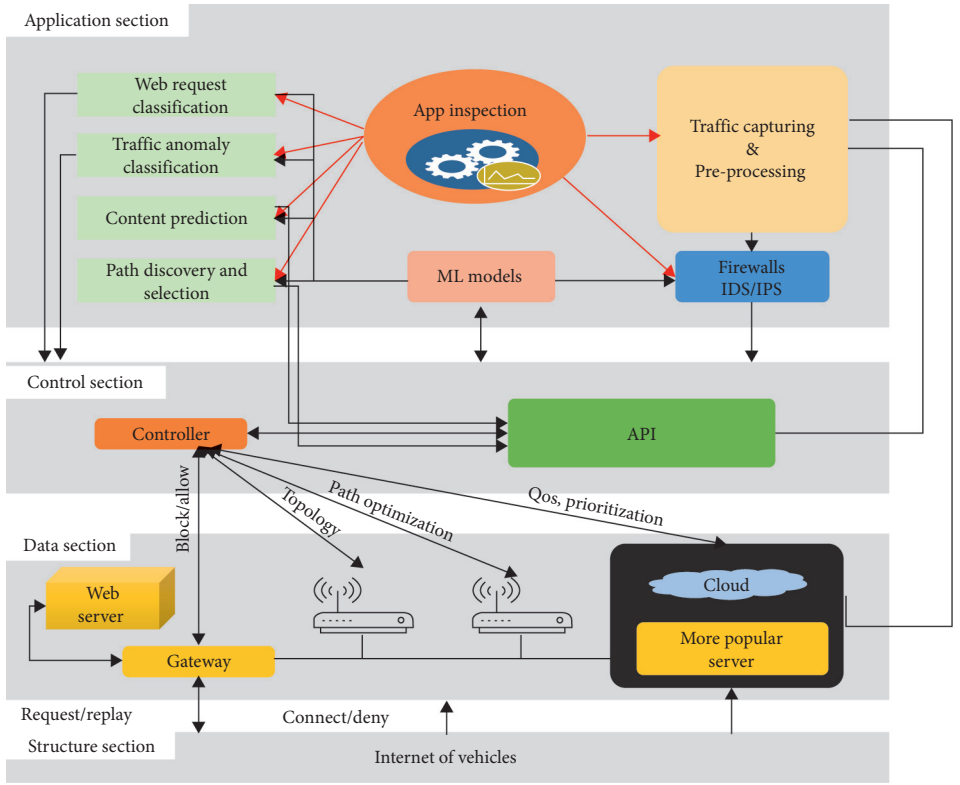


Figure 3: ML functions in the IoV network [36].

that cannot provide precise buffering decisions. To strongly cooperate ML at the IoV network edge to enhance the edge's smart duties, it requires an effective learning approach for massive high-dimensional information that is established to offer a precise estimation of the buffered information at the IoV network edge. Moreover, ML schemes' deployment in IoV applications will extract much sensitive and critical information, and if there is any leakage of information, it can cause serious confidentiality, security, and privacy concerns [40]. For these concerns, an edge-caching system must be secured by security and privacy-preserving schemes and should be developed in different system levels, i.e., transmission/collection, data processing, data access, and storage levels for both edge networks and vehicles.

5.2. Deep Reinforcement Learning-Based Offloading Algorithm. The execution of computing-intensive applications on resource-constrained vehicles still faces a challenge related to offloading the IoV system. Deep Reinforcement Learning (RL) will provide an intelligent offloading system for vehicular edge computing. Integration of Deep RL with vehicular edge computing helps to schedule offloading requests and allocate IoV network resources. Deep RL optimizes the scheduling and resource allocation in IoV to maximize the QoE. In IoV, vehicles calculate utility values related to their available RSUs and pass the offloading requests to the roadside stations. The stations perform task scheduling and resource allocation and inform the RSUs. RSUs are able to receive all vehicle offloading tasks to perform computation offloading. The deep RL algorithms help to optimize the offloading decision by intelligent task scheduling. Figure 5 shows the deep reinforcement learning-based task offloading framework. Offloading requests are scheduled in the task according to the action-value function Q. RSU is selected by vehicles from the available accessing list with probability ε and largest Q-value of the current action-value function [41]. The use of deep RL in IoV offloading optimization will be guaranteeing their venue of network operators by ensuring cooperative offloading in the IoV network, which will maximize the QoE of vehicles.

In IoV offloading computing, optimization parameters are related to the offloading ratio Γi for each task. The vehicle's utility value constraints are related to the limitations of vehicle's CPU and memory resources. Offloading optimization depends on how to minimize the function task latency and energy costs [42]. The cost function can be calculated as follows:

$$\text{Function} = \sum_{i=1}^{N} \left(L_i + w_i E_i\right), \tag{1}$$

where N represents the number of tasks, L_i is denoted for latency cost, and E_i is denoted for energy cost, and w_i is denoted for the weight ratio between latency and energy cost. Each offloading decision is taken depending on the resource unit time slot. This means the flow of scheduling tasks is a sequence in time. For the sequence of N, tasks

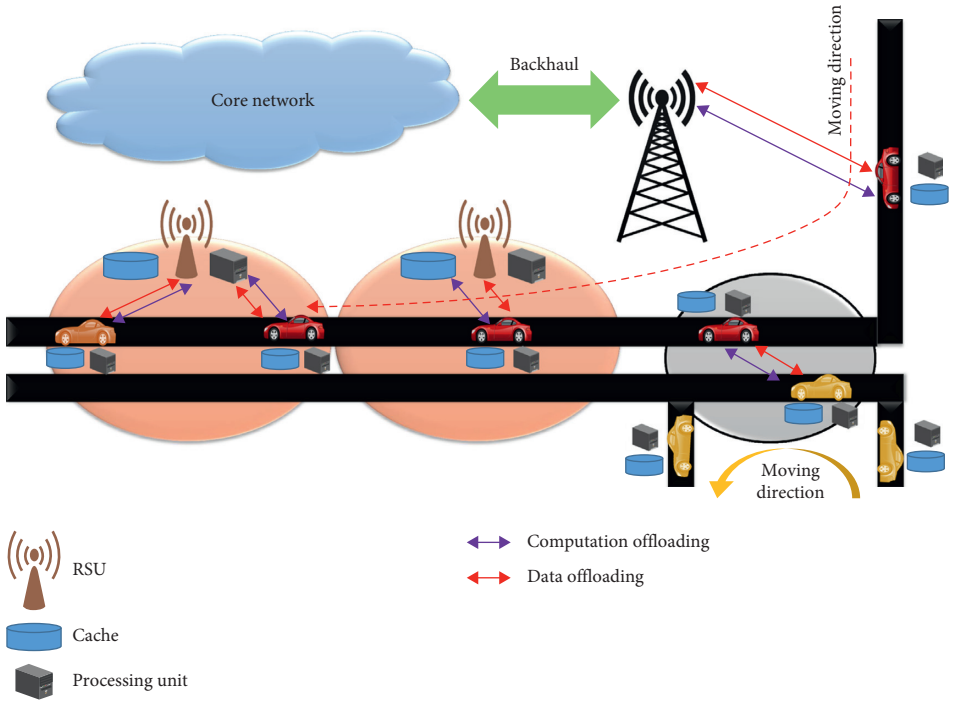


Figure 4: IoV mobility-aware caching and computational scenario [39].

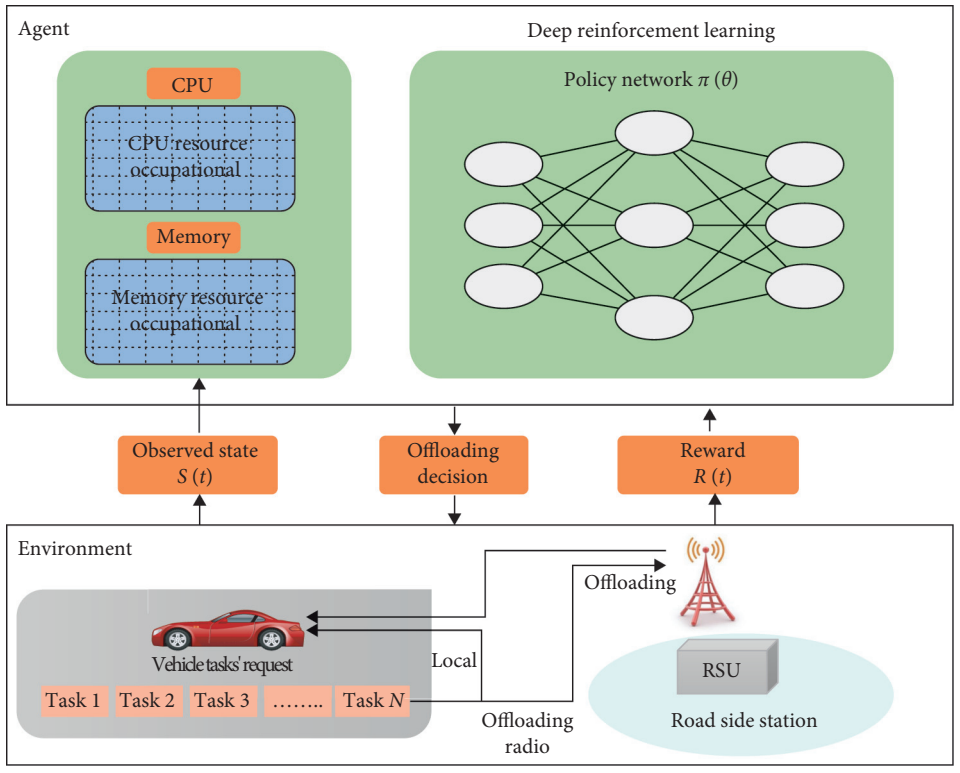


Figure 5: Offloading decision optimization-based deep reinforcement learning.

arrive during a limited observing time L_{obs}, and the cost function F can be calculated depending on the reward function $R(t)$ by

$$\text{Function} = \sum_{t=1}^{L_{obs}} \left(\gamma^{t-1} R(t)\right), \tag{2}$$

where $R(t)$ is the reward of time slot t and γ denoted for the reward discount ratio to describe affection of rewards of the future time slot on the overall cost function. This total cost function can be used to make policy network training. Deep neural networks will provide a policy for mapping from perceived states of the IoV environment to the probabilities of actions to be taken. The policy network-based deep RL training will achieve optimized computing via each time slot [42]. This will minimize the weighted sum of offloading latency and power consumption cost and ensure offloading decision optimization. Figure 6 presents the average cost for power consumption versus probability of task arrival, and Figure 7 shows the average cost for task latency (in Seconds) for different ML for IoV architecture vs. task arrival probability.

The Markov Decision Process (MDP) method is compared with three various techniques i.e., Local, IoVEC, and Random. Local means the zero-offloading technique where all tasks are performed in the vehicle on-board device. IoVEC is known as an integrated IoV edge computing server where a full-offloading procedure has to be maintained and performed. Finally, a random technique selects offloading randomly.

In IoV-based edge computing, vehicles act like clients connecting over the edge-computing node on the roadside without accessing a remote cloud. In this scenario, the offloading decision for heterogeneous resources is considered a complex operation. This is because the environment of vehicular edge computing is changing each time and requires that offloading decisions be re-computed, which will delay providing services. In addition, for vehicular service, the task execution progress cannot guarantee fairness offloading queuing. Deep RL provides a unique decision algorithm to achieve intelligent vehicular-controlled services based on an edge computing model [43]. It helps to learn the service offloading knowledge and the observation functions related to environmental data of vehicular mobility and the edge computing nodes. The offloading decision model is trained at the powerful edge computing nodes and distributes the decision information to the vehicles for services offloading. During decision model training, vehicles transmit the parameters to the roadside station edge-computing node for updating the necessary offloading decision periodically.

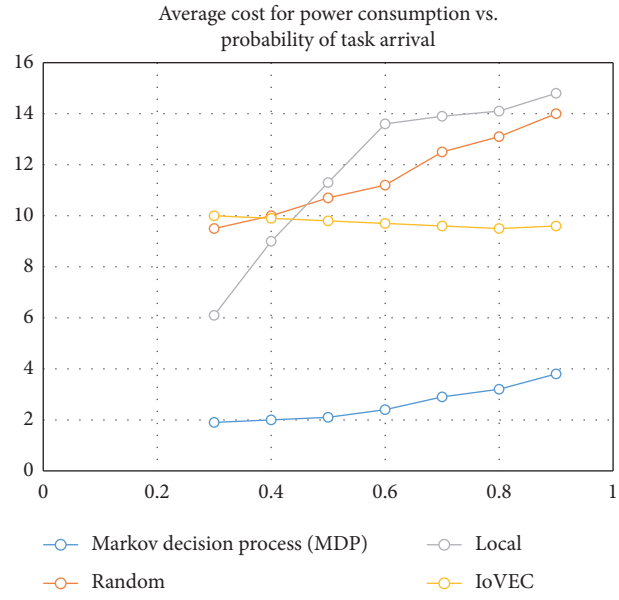


Figure 6: Average cost power consumption (KJ) for different ML for IoV architecture vs. task arrival probability.

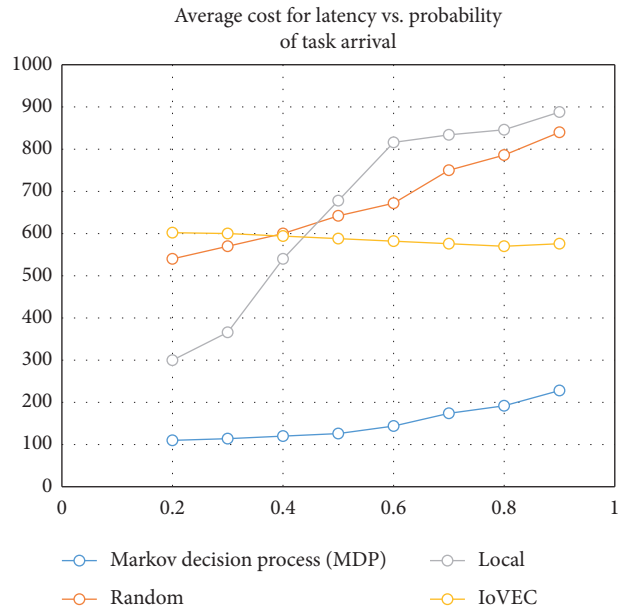


Figure 7: Average cost for task latency (Sec) for different ML for IoV architecture vs. task arrival probability.

5.3. ML for Dynamic and High-Mobility IoV. IoV networks may have dynamic features in many aspects, i.e., topology, traffic, and wireless propagation channels, due to their mobility. An efficient learning and dynamic prediction must provide a degree of optimization in routing, traffic load, and assisting the channel estimation module-tracking channel variations [44]. Machine learning (ML) methods lead to better results for modeling the dynamic changes of vehicular channels and optimizing vehicle routing and traffic flow. ML systems integrated into RSUs help to estimate traffic patterns by collecting information about vehicles. ML can provide intelligent IoV routing protocol with critical information for a highly dynamic environment. It was able to predict the network capability of paths to optimize vehicle route selection based on vehicle mobility and transmission capacity. In dynamic IoV, RSUs-based ML can predict the vehicle's moves and direction [45]. The prediction is depending on the information provided by the vehicle when it moves from RSU to another which will help RSUs to enable estimation of the traffic flows.

5.4. ML-Based Decision-Making in IoV. In recent years, Autonomous Vehicle (AV) growth generates a novel tendency to implement several intelligent approaches and methods to enhance adaptive decision-making efficiency

and quality. The combination of AI, ML, RI, and IoV offers high-efficient control systems that can be exploited in various applications to accommodate more adaptive, automatic, and robust embedded systems [46]. Decision-making in IoV networks requires intelligent algorithms to handle the processes related to driving environment perception, path planning, strategy network control, and resource management. For an intelligent driving vehicle system, a module that integrates the path, behavior, and motion planning is needed to operate in a highly optimized decision-making algorithm. In addition, the decision-making algorithm must take into account the operations of vehicle control. It must be able to predict and learn the information related to vehicle platform faults, trajectory, and energy [47]. These considerations deal with vehicles' platform, as shown in Figure 8. For cognitive driving, localization, semantic understanding, and sensor fusion contribute to the decision-making process.

Furthermore, the intelligent vehicles and IoV systems' applications face the decision-making challenges associated with collecting and distributing IoV big data to vehicles and interested users to enhance road intelligence experience, in addition, in making decisions related to traffic managing, road congestion, and safety. Huge volumes of big data require a more robust and intelligent mechanism in decision-making procedures to reduce road congestion and improve traffic operations in addition to challenges related to useful communication links between different types of vehicles and smart devices and security and privacy problems [48]. Many machine learning methods can be used to contribute to solving the above challenges where these methods enable to model channels in different IoV network scenarios. In addition, it provides intelligent solutions to avoid road accidents by smart learning and analysis of the driving environment using the data collected from the sensors since IoV networks are interested in exchanging messages everywhere and sharing content between smart vehicles [49]. ML-based smart resource management for IoV networks has become crucial to decide the policy of the connection method of power control, selection, and resource allocation and assignment.

5.4.1. Network Control. Higher IoV network performance demands efficient solutions for network operation and optimization. ML in the network domain will leverage ML abilities for new network management for IoV applications. The capabilities of ML will provide an efficient way for intrusion detection and performance prediction. Besides, ML enables the IoV network to make intelligent decisions for network scheduling, and adaptation depends on network characteristics and environment [50]. ML algorithms will facilitate the IoV network to classify and predict traffic patterns and network states. In general, the use of ML in communication networks promises to achieve many solutions for different networking aspects, i.e., data collection and analysis, clustering decision-making and prediction, and model construction validation, in addition to network deployment and interference, as shown in Figure 9 [51]. Because of the IoV characteristics, which depend on the Internet, data and traffic prediction, analysis, and classification are the most critical aspects related to IoV network control.

(1) Traffic Prediction. Data collection and analysis are related to collecting a large amount of representative network data and the ability to characterize the network factors. Based on the IoV application, data collection can be gathered from different network layers. According to the IoV network state, offline data collection with high quality is required for data analysis, while online data collection will enable learning network performance and adaptation [52]. For IoV critical applications, data analysis needs to find a proper network feature, i.e., to predict the best network traffic performance by analyzing the historical data. Data collection and analysis need to prepare network data by normalization, discretization, and missing value completion. ML is an excellent choice to help extract the network feature. For IoV networks, ML plays an essential role in traffic prediction and network management [53]. Accuracy in traffic volume estimation in IoV networks is considered as one of the main factors that impact the performance analysis of network operations, i.e., resource allocation, network routing, congestion, and data streaming control.

(2) Traffic Classification. Traffic classification represents the need for IoV network applications to be matched with the Internet traffic flow. In IoV, the Internet traffic classification is an essential aspect of efficient network quality of service and quality of experience. Moreover, in the network edge, accurate Internet traffic classification is a critical challenge and an essential component of the network security domain. In this case, network traffic classification's importance is to recognize the vehicle network applications and control the traffic flow as needed to balance value or prioritize each other. In the security issue, traffic classifications provide a means of intrusions and malicious attack detection [53]. ML-based statistical features will give a classification scenario to the more realistic situation for IoV network traffic for network control and security. Moreover, it achieves efficiency, adaptability, and performance enhancement.

(3) Traffic Management. Other considerations related to network control are network traffic monitoring and management. In the IoV network, to ensure efficient network optimization, ML enables to adapt the dynamic Internet traffic in IoV and maximize the QoS/QoE without compromising end user experiences. ML provides an adaptation of real-time network conditions and maximizes the user experience [53]. ML can help to overcome the shortcoming of classical TCP congestion control algorithms by classifying a packet loss due to congestion or link errors. ML approaches will be easy to customize best-suited congestion control schemes that can adapt to unique network requirements. ML can systematically prospect important information from data held by vehicles and automatically identify very complex links, allowing vehicles to monitor

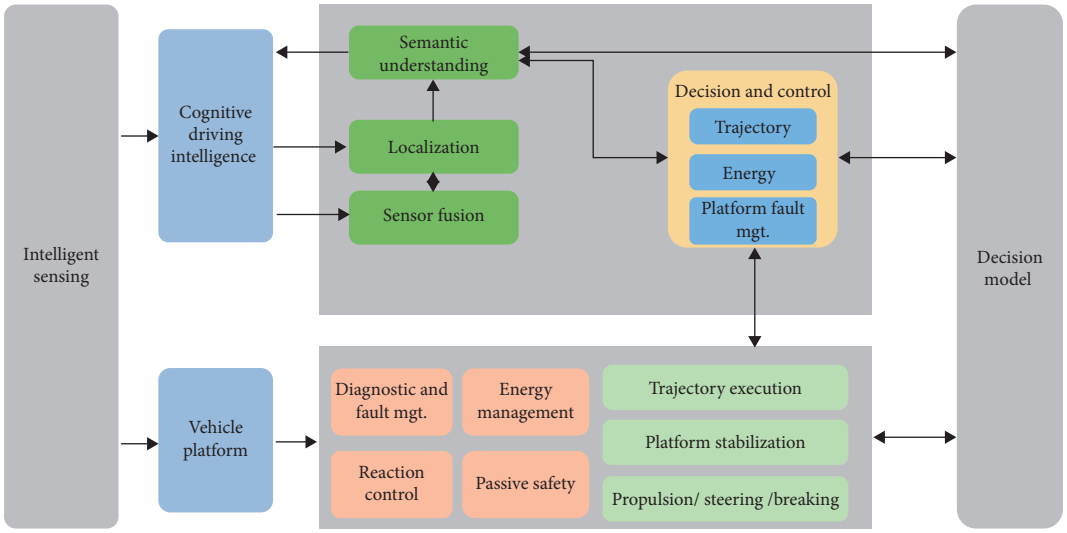


Figure 8: Intelligent driving vehicles' decision-making framework.

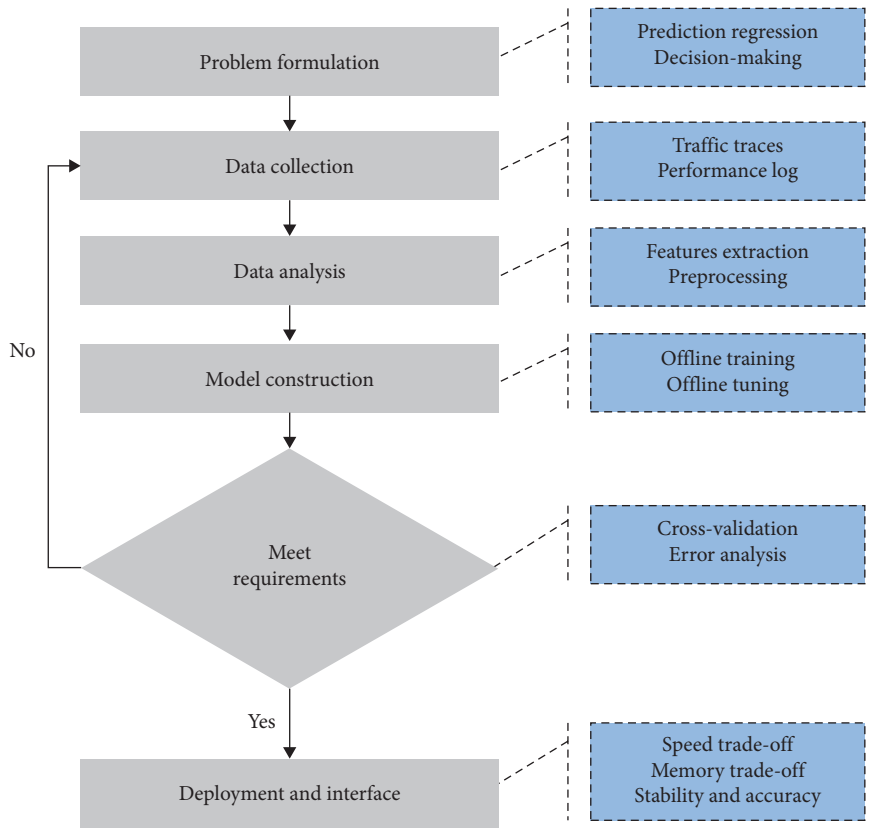


Figure 9: ML for the IoV network control cycle.

their environment intelligently and use data for training purposes [53, 54]. ML enables predicting and adapting to the evolution of environmental features, including wireless channel dynamics and traffic and mobility patterns, and configuring the network, which gives the high possibility to control and manage the network traffic.

Other ML solutions relate to developing accurate channel models in different environments and reducing path loss. These solutions lie in predicting IoV topology and treating severe interference from other IoVs using navigation data and vehicle connectivity. In IoV applications, Internet traffic may be impacted by the weakness of wireless communications [54]. ML technologies can assess wireless conditions without the need for a large number of datasets. Using ANNs' methods, an RSS prediction can be performed in an IoV environment.

5.4.2. Location Prediction. Automation is considered one of the essential advantages of the IoV network. The vehicles contain a perception system to be able to object detection and prediction. In most applications, vehicles' behavior depends on sensory data and the ability to classify the objects in the surrounding environment. These factors help develop autonomous vehicle applications using efficient vehicle behavior prediction and decision-making [55]. The intelligent prediction will help to optimize the decision-making of vehicle trajectories to avoid any risks. Self-driving and autonomous IoV depend on location prediction. The prediction requires information about the position of the vehicle itself and the behaviors of the surrounding vehicles, in addition to the road geometry and traffic rules. Different vehicle behavior prediction models are developed i.e., intention trajectory, maneuver-based, and interaction-aware models [56]. These kinds of models are categorized as input representation and output types' criterion, as shown in Figure 10. In recent years, researchers have tried to use the ML prediction methods to optimize location prediction precision.

ML uses recorded vehicle historical mobility patterns to predict the next location prediction according to mining trajectory patterns. This strategy is depending on the availability of enough historical trajectory data. To gain accurate prediction, ML provides an efficient method to get rid of the problem of suffering from the data sparsity and little historical trajectory and the impact of unknown dynamic contexts, traffic flows, and weather. ML enables the incorporation of this contextual information into the vehicle movement prediction. ML helps to model the contextual information characteristics between the trajectories and builds a learning model by integrating, for example, the neural network with the Long Short-Term Memory (LSTM) to predict the next location, as shown in Figure 11 [57]. The LSTM can easily incorporate heterogeneous features by integrating the trajectory variables to predict the following location effectively.

5.4.3. Intelligent Resource Management. Since IoV applications depend on the IoT, it is found that resource management in this technology is facing many challenges, especially in large-scale IoT networks. These challenges are related to massive channel access, power allocation, interference management, energy management, and coexistence between V2V or V2I and IoT traffic. Massive channel accessing causes overloads to networks and congestion [58]. For resource management, there is a need to develop proper load balancing and access management techniques. The crowded vehicles traveling over the roads make interference problems which requires efficient power allocation and interference management techniques. In the IoV, the IoT's nature is characterized by continuous data traffic, which leads to high energy consumption. Moreover, the harmonious coexistence between the V2V or V2I-existing networks and IoT traffic requires intelligent resource management [58, 59]. ML algorithms play an essential role in addressing the mentioned challenges related to resource management.

ML can make classification, regression, and density estimation for intelligent resource management to exploit data traffic and develop automated solutions for IoV services. ML provides the intelligent prediction for unknown IoV system parameters and system behavior, i.e., RL can control system actions from anonymous monitored system behavior during network activities. Moreover, ML provides suitable solutions for helping careful channel and power allocation and extracts the network parameters to make decisions for CSI, traffic characteristics, and demands of the vehicle's users [59]. Deep learning promises smart solutions to characterize the inherent relationships between the IoV system input and output to develop a traffic control system to optimize the network management and scheduling adaption [50]. This will help to optimize the IoV network QoE.

Another consideration related to the ML use in resource management is maximizing the overall network capacity and guaranteeing the best QoS. Q-learning can attain a substantial regulation and strategy by utilizing the network learning policy to accomplish smart resource control, assignment, and management with the continuous valued activities. It can be employed to obtain an optimal resource allocation approach in V2V communications to maximize the long-term expected accumulated discounted rewards, where the Q function is approximated by a deep neural network [50]. The following equation can find the optimal policy with Q values:

$$Q_{\text{new}}(s_t, a_t) = Q_{\text{old}}(s_t, a_t) \cdot + \alpha\left[r_{t+1} + \gamma \max_{s\varepsilon S} Q_{\text{old}}(s_t, a_t) - Q_{\text{old}}(s_t, a_t)\right]. \tag{3}$$

The observed state is represented by $s\varepsilon S$, where S represents the state space, t denoted for time, s_t is an agent state, and a_t represents action. The Q-learning can be deployed by what is known as Actor-Critic (AC) learning algorithm which is discussed in [50] by (Wang in 2017). The frame of AC learning consists of actor and critic parts which are responsible for control strategy adoption with action selections based on the tested network status and the entered policy of the environment parameter reward function, respectively, as shown in Figure 12. This mechanism enables the IoV vehicles to make decisions based on their learned policy strategy [60]. Each of IoV communication link will observe the current network state, i.e., resource block allocation, channel quality, and QoS requirements to enable selection of actions related to resource block assignment and power level according to the policy strategy to provide a new IoV network state.

6. Machine Learning Applications in IoV

ML contributes to many IoV applications related to emergent message transmission for road safety and dangerous activities. In addition, ML provides new smart solutions for IoV services and entertainment. To minimize the overall energy consumption of the computational facilities and vehicles, while satisfying the delay constraint for

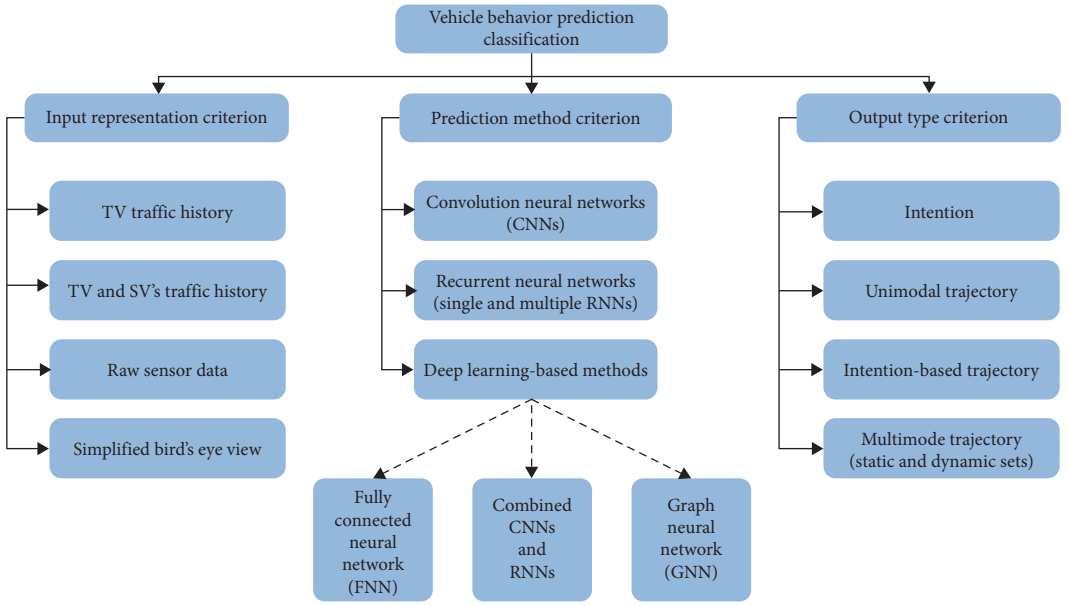


Figure 10: IoV behavior prediction models.

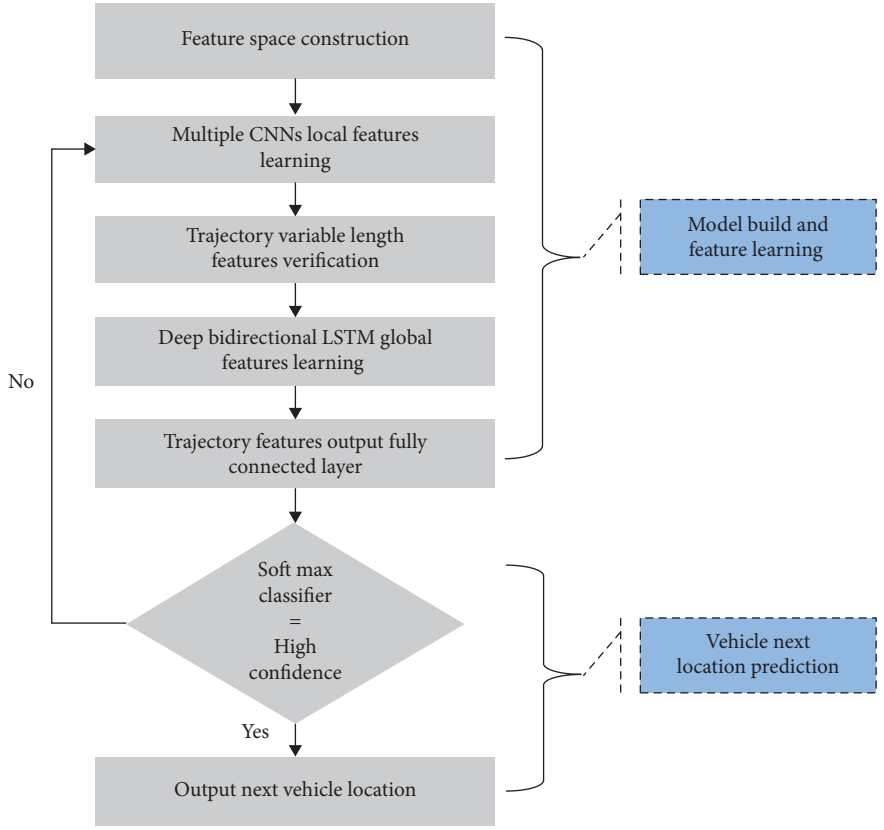


Figure 11: ML algorithm flowchart for IoV location prediction [56].

traffic offloading, ML technology in data mining, pattern recognition, processing, and cognitive computing is an alternative for decision making, which will open new opportunities for intelligent IoV networks, i.e., in driver safety, smart transportation, and autonomous driving applications.

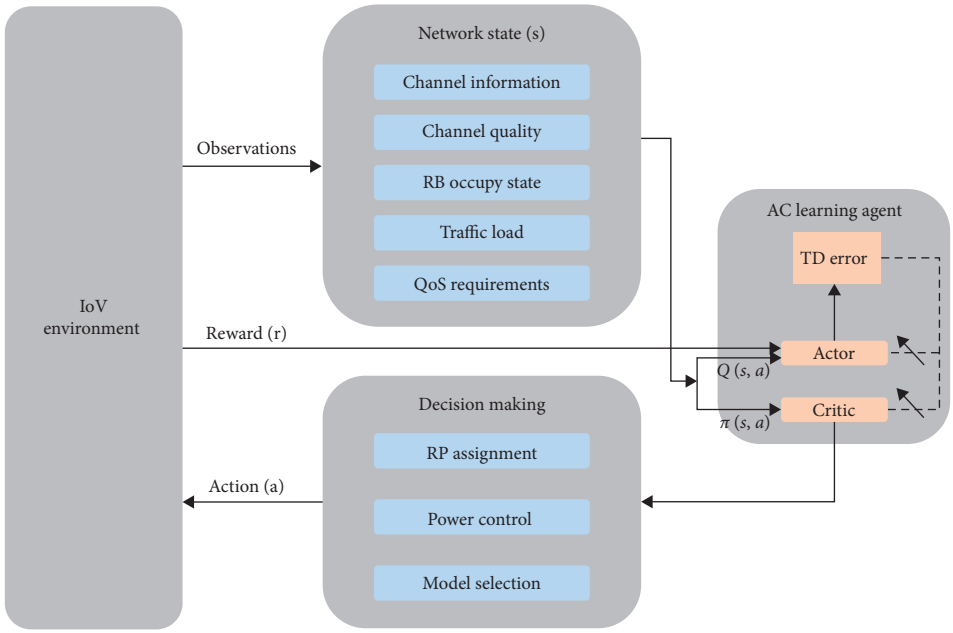


Figure 12: AC learning algorithm for IoV resource management [60].

6.1. Intelligent Autonomous Driving. Machine learning plays a vital role in vehicle intelligent driving applications, making vehicles perceive and estimate to manage the vehicle driving system efficiently. ML makes the vehicles self-automated which will improve the society by reducing road accidents. In general, self-driving vehicles are very closely associated with IoV. The combination of the IoT with ML and smart computing will provide an intelligent driving system. Machine learning algorithms in self-driving enable IoV to predict the possible changes in the surrounding driving environment and provide different tasks i.e., object detection and identification, in addition to prediction of another vehicle's localization and movement [61]. Many ML algorithms can be used to provide the mentioned tasks. Regression algorithms provide a localization scheme to develop prediction and feature selection models for self-driving vehicles. Clustering algorithms provide a way to model approaches such as centroid-based and hierarchical for intelligent localization [61, 62]. Decision matrix algorithms will help identify, analyze, and rate the performance of relationships between sets of values and information for intelligent decision-making.

To enable self-driving vehicles, intelligent decision-making must process streams of observations coming from different vehicle devices, such as cameras, radars, LiDAR's, ultrasonic sensors, GPS units, and sensors. The information gathered by these sources helps the vehicle's ML-based computer to make driving decisions, as presented in the study proposed by Hussain (2020) [58]. Decision-making can take place by the modular perception-planning-action or by the End2End learning fashion. The modular perception-planning-action uses the AI and deep learning methodologies to make various learning and nonlearning-based components. End2End learning is based on deep learning which performs the direct mapping from sensory data to control commands. End2End learning can also be formulated as a backpropagation algorithm scaled up to complex models [63]. Such deep learning-based algorithms will be able to find a route between the vehicle start position and the desired location, which represents path planning. It is able to consider all possible obstacles present in the surrounding environment and find out a trajectory free of the collision route.

6.2. Deep Learning for Driver Safety and Assistance. Due to the increasing number of accidents and the urgent need to reduce road accidents and improve traffic safety, modern vehicles are equipped with sensors and connected to high-speed mobile communication networks. The vehicle sensors allow collecting a large amount of data used in vehicle safety analysis procedures. The data are analyzed in real time by AI algorithms in the autonomous driving systems' applications to reach a high level of safety through several designs. It enables the designing of the road safety index and its prediction of parameters such as street engineering, human behavior, and traffic flow. The description of road safety by deep learning will predict the real-time road safety index based on the deep dense neural network. Moreover, ML helps to learn the association between visual entities and city characteristics to estimate road safety based on image processing [64]. The extraction of associations between captured pictures and estimated road safety with multiple cross-domain factors can achieve high prediction accuracy of the road safety index (SI). The real-time road safety index estimation will enhance vehicle safety. The road SI can be defined as a number used to inform the public about the area's safety, as published by Abdallah in 2018 [65]. The safety index can be calculated based on the traffic accident rate per 100,000 inhabitants Ra as follows:

$$SI = (1 - Ra) * 100. \quad (4)$$

Driver Assistance Systems (ADAS) are quickly being established for self-directed vehicles which are considered driver safety and assistance methods. ML and embedded computing are considered the main driving factors enabling the development [65]. ML will enable driver assistance systems to perceive obstacles, objects, lanes, pedestrians, and other cars and predict obstacle trajectories and targets. ML helps to detect and to track the obstacle to avoid collision and for path planning. Vehicle camera-based deep learning improves quality enhancement and cost reduction of blind spot prediction rather than radar [66], as proposed by Ball and Tang (2019). It uses a lightweight and computationally effective Neural Network (NN).

6.3. ML in Smart Transportation. Intelligent transport is one of the most important vehicle Internet applications, as it covers several applications including improving track, parking lots, avoiding and detecting accidents, and other applications related to infrastructure. ML technologies serve to develop advanced models of ITS. In general, traffic congestion is one of the most important problems faced by transportation systems in urban areas, especially in cities that contain high vehicle density [67]. The use of ML with smart transport systems provides optimization for traffic network configuration. Another smart transportation application in modern cities is parking. Smart cities try to find an intelligent parking method to provide reservation services and select parking for vehicles. IoT and ML technologies enable free parking methods. ML helps manage parking for different drivers. It can classify parking according to drivers' requirements, i.e., regular drivers or those with special needs. The IoT helps to exchange the mapping of parking information to the vehicles or for mobile users through cloud servers [68]. Moreover, IoT will improve traffic monitoring, live location streaming, and vehicle performance monitoring.

Since the IoV network consists of multiple types of smart vehicles, transport data processing of these numerous vehicles in real time requires an intelligent schedule and data processing mechanism [68]. Distributed systems provide an efficient and fast method for such a situation. This needs to deal with big transportation data collected from heterogeneous sources of database solutions. ML-based SQL database enables smart database queries and flow data processing. ML will allow the balance between the algorithms' accuracy and the size of the data and determine the circumstances in which it becomes useful to implement distributed systems. For transportation route optimization, ML provides reliable predictions to make routing decisions [69]. ML enables a clear understanding of available route options, associated energy, and environmental costs in real time. ML provides predictability of changes that can help convoy operators choose vehicles and methods that save fuel costs, while maximizing efficiency [70].

7. Secure Vehicular Network towards 6G

Cyber-physical security is one of the hot research areas in the Internet of Things, which is also a thoughtful subject in vehicular communication. The attack and malicious activities of vehicular networks cause thoughtful damage that threatens passengers' safety in vehicles and affects network performance. The vehicular systems usually need many severe strains in the ML-based security scheme. The precise restraints of the vehicular system can be presented as follows.

7.1. Vehicle Speed. One of the most critical parameters of the vehicular system is the high mobility of vehicle nodes and the network dynamics. Communications between nodes regularly go down, making the system security and authentication quite hard [71]. The system traffic is abruptly flapping with the rapid change and dynamic of the network topology which seriously disturbs the intrusion detection and security algorithm and schemes of the packets [72]. Additionally, the vehicle speed results in the random mobility of vehicles, which delays and disturbs the performance of the security and authentication data exchange.

7.2. Diversity Framework. The Vehicule-to-Vehicle (V2V) topology is structured with different nodes and a dynamic network, which is implanted with various network resources. The main challenge is to study the different resources among vehicles to guarantee security and authentication schemes. As an instance, in the storage-constrained, processing, and energy vehicular system, the challenge is to optimize and capitalize the security guarantee which can be resolved by neural network or fuzzy logic or game theory [73].

7.3. Network Size. The volume of the Internet of vehicular nodes grows rapidly. Thousands of nodes in vehicles are probable to be linked to the huge IoT network in near future [74]. Though, no present worldwide group or body offers security for such a huge rapidly changing network. Additionally, the growing volume of the Internet of Vehicular network grows both the processing and network protocol that lead the growing security protocol traffic, which grows the error detection and network latency.

7.4. Confidentiality Obligation. Always there is a compromise between privacy and security in the Internet network, especially in IoT and IoV. In IoV, vehicle nodes' authentication and confidentiality are usually designed and modeled by security algorithms. Without data privacy assurance, the security scheme is usually hungry for resources to distinguish and differentiate anomaly and error from flow data [75].

7.5. Fast Response Prerequisite. Unlike legacy networks, the Internet of vehicular networks needs an instantaneous handle to cope with the rapid and dynamic change network, i.e., news broadcasting, fast rescue, and avoidance of accidents. Such network needs fast response requirements for the Internet of vehicular network such as low-latency communication channels and real-time prevention of attack techniques [76]. To overcome these problematic issues, many solutions and mechanisms have been anticipated in the literature. Interfering and sniffing are two main vulnerabilities in IoV; Elliptic Curve Digital Signature Scheme (ECDSA) and vehicular Public Key Infrastructure (PKI) have been suggested to be the two primary intrusion techniques to guarantee privacy during communication from vexatious activities [46, 76].

Nevertheless, the vexatious activities include the pseudo spoofing, the wormhole attack, the packet drop attack, the Denial-of-Service (DoS) attack, the spurious data intrusions, and the reiteration intrusions that can counterfeit identity, broadcast junk packets, and kidnapping vehicle nodes to penetrate and stealth elliptic curve digital signature algorithm and public key infrastructure [77]. Many vexatious attitude recognition techniques have been invented [71, 73]. Some works suggest that infiltration and attacks can be drawn with machine learning techniques from rapid and high-diversity network traffic [75, 78]. The attack prevention mechanism for legacy networks is typically achieved by determining the normal state from flooding-emulated packets. In the highly rapid Internet of the vehicular network, the present emulated packets are not useful for innovative intrusions in such a dynamic environment. The legacy machine learning technologies, such as association rules, autoregressive, and classification, are widely used in the abovementioned works for intrusion detection. Recently, ML has proposed to be the promising machine learning and data mining utilities for enhancing the authentication, privacy, and attack prevention performance on the Internet of vehicular networks.

7.6. Authentication Technology in the IoV. Authentication algorithms for IoV networks are considered essential for network and communications' security purposes. Several studies dealt with the concept of certificates to determine the identity of vehicles. Other studies have used anonymous credentials and designated an unknown identity IoV area for the vehicle that allows access to it safely with the possibility of hiding the information. A hiding vehicle information mechanism is adopted by specifying the unknown identity to achieve safety against any attack by any malicious vehicle that tries to steal vehicle information in the IoV network [79]. Despite this mechanism's efficiency in maintaining the confidentiality of vehicle information, it is facing a delay issue in processing identity and is wasting a large amount of identity storage resources. Finding an effective anonymous authentication method in IoV, while reducing its computational cost is a big challenge. AI mechanisms can provide solutions to improve the anonymous authentication system by reducing account costs through a contextual tracking mechanism to manage IoV network vehicles and units on the roadside. Liu et al. (2018) proposed a mechanism based on the safe communication between vehicles and units on the side of the road using machine learning technology [80]. The study relies on creating a Certified Short Signature Model (CLSS) that works with the regional management strategy to design an anonymous and efficient anonymous authentication scheme for IoV. The study achieves a highly efficient model in terms of the interaction between vehicles and roadside units compared to traditional plans. The proposed CLSS scheme is secure under adaptively chosen message and ID attacks in the random oracle model.

On the contrary, issuing identity certificates to enable privacy protection is more efficient. Still, it is the responsibility of the RSU, which increases the operating cost and causes the greater consumption of network resources to operate and configure RSU. In addition, the resource management processes related to vehicle verification and requesting authentication take a lot of time [80]. Aggregated authentication technology can reduce time delay, but it also contains frequent authentication problems and requires a large amount of authentication information. An authentication protocol is known as Distributed Aggregate Privacy-Preserving Authentication (DAPPA), proposed by Zhang (2017), enables to authenticate the vehicles in the vehicular network by providing a multiple trusted authority based on an identity-based aggregate signature mechanism [81]. The aggregation of vehicle signatures in one of the verified messages reduces storage needs and resource management costs. A smart adaptive data aggregation study by Islam et al. (2016) presents a method to enable data communication between distributed mobile vehicles on vehicles unknown to other vehicles or IoV locations [82]. The adaptive data aggregation depends on machine learning to analyze data and extract information for the drivers, enabling fully automated switching of different vehicle sensors and data fusion processes' adaptation.

In general, the AI technologies in IoV develop secure management processes and achieve intelligent, fast authentication and progressive authorization. AI enables to provide a light authentication scheme in addition to a comprehensive authentication and authorization system. Machine and deep learning can enhance IoV security by investigating valuable information and providing self-adaptation for certification and authorization. In IoV channel communication, the Support Vector Machine (SVM) enables developing a lightweight authentication system to identify vehicles based on their pseudorandom arrival in the time domain or the frequency range via multiple sensors. Hasan et al. (2020) provided a fast authentication mechanism for large-scale IoV that depends on the identical unique Pseudo-Random Binary Sequence (PRBS) for vehicle access time slots or access frequencies [83]. This mechanism can be used to verify the authentication of vehicles to the IoV network base station access. Figure 13 shows the possible lightweight authentication scheme based on SVM for IoV secure authentication.

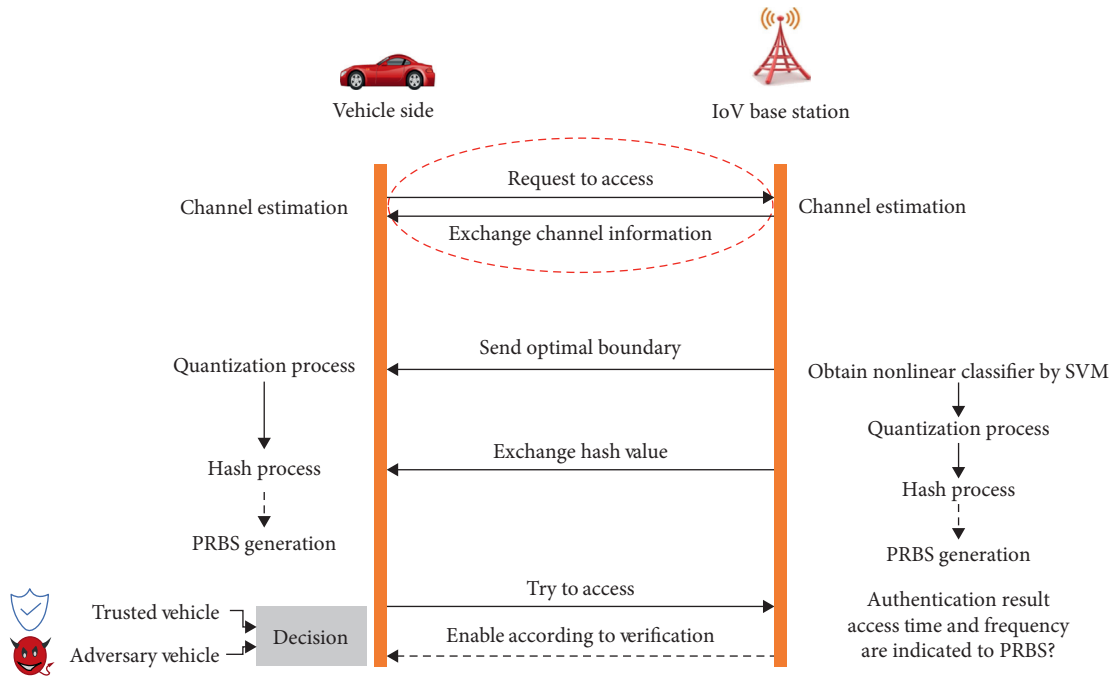


Figure 13: IoV authentication scheme based on the SVM.

The decisions taken by the SVM are a function of the support vector machine network. The output decision is according to several linear combinations of the middle layer nodes. The layer nodes correspond to the inner product between the input sample and support vector that enables the selection of the most suitable optimized communication mode [83]. This module provides a fast authentication scheme by directly specifying access times or frequencies and the progressive protection of trusted communications without the need for a complex account. The PRBS between each vehicle and the gateway can be obtained by utilizing their unique features by the SVM. The PRBS between each vehicle and the IoV base station can be obtained by utilizing their unique features based on the SVM. The selected features are measured by channel information estimation. The separation between the dense data and sparse data is enabled by an SVM-based quantization technique. This technique reduces the wrong decisions by diminishing the measurements near the boundary [84]. The lightweight authentication scheme based on the SVM ensures higher similar binary sequences acquired on both IoV base stations and vehicles because of the channel reciprocity.

7.7. Fog-Based Identity Authentication. In IoV applications, the fog-computing concept reduces the burden on the traffic control center. All computing components in IoV, i.e., the roadside and vehicle units, are well suited to the concept of fog computing and enable the communication and interaction between vehicles and clouds [85]. Due to several fogs' possibility, the vehicle identification certification system's security is essential to enhance the security issues related to the fog nodes. The fog-based identity authentication scheme presented by Song (2020) provides two authentication levels: vehicles outside the fog and the other for the security monitoring of the rest of the vehicles. The scheme uses deep learning for security monitoring to conduct real-time security in the IoV [86].

A reliable and secure IoV fog mechanism based on machine learning develops access security authentication and security timing detection for vehicles that need to join IoV. When using vehicle safety certification and timing detection mechanisms, it is very important to pay attention to the fog head replacement frequency to reduce resource consumption and time delay and detect the manufactured vehicles to access the fog legitimately [87]. AI-based coding algorithms provide intelligent solutions that ensure the exchange of information for vehicles that leave and join the IoV fogs. Machine learning technologies enable the detection of malicious vehicles that use legal personalities to join the fog. In general, most current fog-based IoV security mechanisms, such as authentication, encryption, and access control, are relatively weak solutions. AL and ML technologies provide a defensive scheme for the fog-IoV environment enabled to secure related operations such as activities monitoring, misuses identification, and threats and vulnerabilities detection in accessing processes [88–91].

Table 3 summarizes the key factors regarding secure IoV communications.

8. Future Directions and Potential Solutions

It is well known that artificial intelligence plays a vital role in most IoT applications that depend on perception and predictions of events. As one of these applications, IoV

Table 3: Summary for secure IoV communications.

Year	Source	Approaches	Features	Advantages	Challenges	Citations
2020	IEEE	Fog-based identity authentication (FBIA)	Fog-based identity authentication scheme and deep learning	IoV real-time security monitoring	Dual authentication levels for access authentication and vehicles' timing detection	Song et al. [86]
2020	IEEE	SVM-based classifier	Authentication scheme based on SVM	Secure access frequencies and progressive protection of trusted communications	Fast authentication mechanism for large-scale IoV	Hasan et al. [83]
2018	IEEE	Certificateless Short Signature Scheme (CLSS) and ML	Anonymous authentication scheme-based ML	Secure communication between vehicles and roadside units	Security under adaptively chosen message and ID attacks	Liu et al. [80]
2017	IEEE	Aggregate privacy-preserving authentication protocol; Multiplicative Secret Sharing (MSS) technique	Distributed aggregate signature mechanism	Secure vehicular network authentication and trusted authority	Trade-off between security and storage resource management	Memon et al. [81]
2016	Elsevier	Smart Adaptive Data Aggregation (SADA); machine learning-based data fusion and analysis	Adaptive data aggregation-based ML	Secure data exchange between vehicles	Fully automated switching to unknown vehicle	Islam et al. [82]

networks require the development of smart algorithms to manage intelligent technology, such as self-driving cars. Self-driving cars are a high-risk test for machine learning authorities, as well as a test case for social learning in technology management [92–96]. In IoV applications, the convergence between machine learning and the Internet of Things promises future progress in efficiency, accuracy, and improved resource management. The use of machine learning with IoV provides high performance in communication and computing to achieve efficient control, management, and decision-making processes [92, 97]. ML allows the extraction of big sensory data to get better insights into the range of problems associated with the IoV and the surrounding environment and the ability to make critical operational decisions. It also promises soon to upgrade vehicle networks' performance and make them more interactive with other things' Internet applications. Using ML in the IoV enables interaction between the cyber and physical components together and can significantly improve the efficiency and reliability of processes and systems [97]. Moreover, machine learning offers smart solutions to enhance decision-making in the event of cyber attacks.

ML provides solutions for many ITS applications, especially in 2D level realization and forecasting. However, it can develop AI techniques that can develop collaborative mobility applications based on the description of realistic 3D objects and 4D perception for autonomous driving [98]. For different IoV applications, such as driving managements, route, and localization prediction, smart ITS camera devices can create holograms to provide 3D object visualization. Due to the hybrid ITS context, the combination of data from different resources to improve 3D visualization accuracy is an exciting potential solution and critical future research direction. The 5G of the IoV network is expected to provide some AI technologies to provide network management completely smart and provide innovative services [89, 90]. However, the Sixth Generation (6G) is expected to pack machine learning techniques an essential role in its operation through self-reconfiguration on demand to ensure a doubling in network performance and service types [99]. ML techniques can provide the 6G network model that can rapidly respond to IoV management processes by learning in real-time the network's state.

9. Conclusion

Machine learning (ML) helps analyze big data in IoV networks, enabling intelligent forecasting and decision-making. Various potential applications have been indicated for the use of ML to improve the performance of IoV networks. ML technologies offer beneficial solutions in addressing congestion problems in high-density IoV networks to achieve quality services and experience. Moreover, the scope of employing machine-learning technology in network management and control, data flow, site forecasting, and resource tools across different layers of communication networks were discussed. In general, we find that, in most automated learning applications, performance depends on the amounts of data available and that must be large enough. Recently, parallel computing capabilities and machine learning methods have been developed to build smart integrated systems for IoV networks. This development can build intelligent systems with immense parallel processing capabilities and energy efficiency to prepare solutions for various operations associated with the IoV, such as multi-dimensional signal/image processing and wireless communications.

Data Availability

Data used to support the findings of this study are already available in the manuscript.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the research grant Universiti Kebangsaan Malaysia (UKM) under Grant nos. FRGS/1/2020/ICT03/UKM/02/6 and DIP-2018-040.

References

[1] W. Tong, A. Hussain, W. X. Bo, and S. Maharjan, "Artificial intelligence for vehicle-to-everything: a survey," *IEEE Access*, vol. 7, pp. 10823–10843, 2019.

[2] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial intelligence-enabled intelligent 6G networks," 2019, https://arxiv.org/abs/1912.05744.

[3] A. A. Eltahir, R. A. Saeed, A. Mukherjee, and M. K. Hasan, "Evaluation and analysis of an enhanced hybrid wireless mesh protocol for vehicular ad-hoc network," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, pp. 1–11, 2016.

[4] Y. Dai, "Artificial intelligence empowered edge computing and caching for internet of vehicles," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 12–18, 2019.

[5] H. Ji, "Artificial intelligence-empowered edge of vehicles: architecture, enabling technologies, and applications," *IEEE Acces*, vol. 8, pp. 61020–61034, 2020.

[6] A. H. Sodhro, Z. Luo, G. H. Sodhro, M. Muzamal, J. J. P. C. Rodrigues, and V. H. C. de Albuquerque, "Artificial Intelligence based QoS optimization for multimedia communication in IoV systems," *Future Generation Computer Systems*, vol. 95, pp. 667–680, 2019.

[7] M. B. Hassan, E. S. Ali, R. A. Mokhtar, R. A. Saeed, and B. S. Chaudhari, "NB-IoT: concepts, applications, and deployment challenges, book chapter (ch 6)," in *LPWAN Technologies for IoT and M2MApplications*, B. S. Chaudhari and M. Zennaro, Eds., Elsevier, Berlin, Germany, 2020.

[8] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for internet of vehicles," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 12–18, 2019.

[9] E. S. A. Ahmed and R. A. Saeed, "A survey of big data cloud computing security," *International Journal of Computer Science and Software Engineering (IJCSSE)*, vol. 3, no. 1, pp. 78–85, 2014.

[10] Z. K. A. Mohammed and E. S. A. Ahmed, "Internet of things applications, challenges and related future technologies," *WSN*, vol. 67, no. 2, pp. 126–148, 2017.

[11] H. Wu, "Developing vehicular data cloud services in the IoT environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587–1595, 2014.

[12] Z. E. Ahmed, M. K. Hasan, R. A. Saeed et al., "Optimizing energy consumption for cloud internet of things," *Frontiers of Physics*, vol. 8, p. 358, 2020.

[13] M. K. Hasan, A. F. Ismail, A.-H. Abdalla, H. A. M. Ramli, W. Hashim, and S. Islam, "Throughput maximization for the cross-tier interference in heterogeneous network," *Advanced Science Letters*, vol. 22, no. 10, pp. 2785–2789, 2016.

[14] A. H. Sodhro, "Artificial Intelligence based QoS optimization for multimedia communication in IoV systems," *Future Generation Computer Systems*, vol. 95, pp. 667–680, 2019.

[15] Y. Mao, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[16] J. Xu, "Joint service caching and task offloading for mobile edge computing in dense networks," in *Proceedings of the IEEE Conference on Computer Communications*, Honolulu, HI, USA, 2018.

[17] Y. Cao, "An EV charging management system concerning drivers' trip duration and mobility uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 596–607, 2016.

[18] S. Islam, A.-H. A. Hashim, M. H. Habaebi, and M. K. Hasan, "Design and implementation of a multihoming-based scheme to support mobility management in NEMO," *Wireless Personal Communications*, vol. 5, no. 2, pp. 457–473, 2017.

[19] N. S. Nafi, M. K. Hasan, and A. H. Abdallah, "Traffic flow model for vehicular network," in *Proceedings of the 2012 International Conference on Computer and Communication Engineering (ICCCE)*, pp. 738–743, IEEE, Kuala Lumpur, Malaysia, 2012.

[20] M. Abdallah, "Softwarization, virtualization, and machine learning for intelligent and effective V2X communications," 2006, https://arxiv.org/abs/2006.04595.

[21] Z. El-Rewini, "Cybersecurity challenges in vehicular communications," *Vehicular Communications*, vol. 23, 2020.

[22] H. M. Furqan, "Intelligent physical layer security approach for V2X communication," 2019, https://arxiv.org/abs/1905.05075.

[23] T. Zhou, R. R. Choudhury, P. Ning, and K. Chakrabarty, "P2DAP—Sybil attacks detection in vehicular ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 582–594, 2011.

[24] S. So, "Physical layer plausibility checks for misbehavior detection in V2X networks," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks, WiSec'19*, Miami, FL, USA, May 2019.

[25] W. Li and H. Song, "ART: an attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 960–969, 2016.

[26] J.-P. Monteuuis, "My autonomous car is an elephant": a machine learning based detector for implausible dimension," in *Proceedings of the Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*, Shanghai, China, 2018.

[27] M.-J. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS ONE*, vol. 11, no. 6, Article ID e0155781, 2016.

[28] M. Akhtaruzzaman, M. K. Hasan, S. R. Kabir, S. N. H. S. Abdullah, M. J. Sadeq, and E. Hossain, "HSIC bottleneck based distributed deep learning model for load forecasting in smart grid with a comprehensive survey," *IEEE Access*, vol. 8, pp. 222977–223008, 2020.

[29] S. Hu, "A fuzzy QoS optimization method with energy efficiency for the internet of vehicles," *Advances in Networks*, vol. 4, no. 2, pp. 34–44, 2016.

[30] S. Islam, A. H. Aisha-Hassan, R. A. Saeed et al., "Mobility management schemes in NEMO to achieve seamless handoff: a qualitative and quantitative analysis," *Australian Journal of Basic and Applied Sciences*, vol. 5, no. 6, pp. 390–402, 2011.

[31] C.-F. Lai, "A buffer-aware QoS streaming approach for SDN-enabled 5G vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 68–73, 2017.

[32] R. A. Saeed, R. Mokhtar, and S. Khatun, "Spectrum sensing and sharing for cognitive radio and advanced spectrum management," *ICGST International Journal on Computer Networks and Internet Research (CNIR)*, vol. 9, no. 2, pp. 87–97, 2009.

[33] H. Park and Y. Lim, "Reinforcement learning for energy optimization with 5G communications in vehicular social networks," *Sensor*, vol. 20, no. 8, p. 2361, 2020.

[34] E. Bozkaya and B. Canberk, "Software-defined management model for energy-aware vehicular networks," *EAI Endorsed Transactions onWireless Spectrum*, vol. 3, no. 11, Article ID 152099, 2017.

[35] Y. Zhao, "A survey of networking applications applying the software defined networking concept based on machine learning," *IEEE Access*, vol. 7, pp. 95397–95417, 2019.

[36] T. n. Nguyen, "The challenges in ML-based security for SDN," in *Proceedings of the 2nd Cyber Security in Networking Conference (CSNet)*, Paris, France, 2018.

[37] K. F. Hasan, "Cognitive internet of vehicles: motivation, layered architecture and security issues," in *Proceedings of the International Conference on Sustainable Technologies for Industry 4.0 (STI)*, Bangladesh, India, 2019.

[38] C. Chen, "A rear-end collision prediction scheme based on deep learning in the internet of vehicles," *Journal of Parallel and Distributed Computing*, vol. 117, pp. 192–204, 2017.

[39] L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: a deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10190–10203, 2018.

[40] Z. Chang, "Learn to cache: machine learning for network edge caching in the big data era," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 28–35, 2018.

[41] Z. Ning, "Deep reinforcement learning for vehicular edge computing: an intelligent offloading system," *Transactions on Intelligent Systems and Technology*, vol. 10, no. 6, 2019.

[42] H. Zhang, "Deep reinforcement learning-based offloading decision optimization in mobile edge computing," in *Proceedings of the IEEE Wireless Communications and Networking Conference(WCNC)*, Marrakesh, Morocco, 2019.

[43] J. Wang, "Vehicular edge computing: a deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4192–4203, 2018.

[44] H. Ye, "Machine learning for vehicular networks: recent advances and application examples," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, 2018.

[45] W. K. Lai, "A machine learning system for routing decision-making in Urban vehicular ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 3, Article ID 374391, 2015.

[46] K. Hamid, "Artificial intelligence and internet of things for autonomous vehicles," *Nonlinear Approaches in Engineering Applications*, Springer, 2020.

[47] J. Li, *Survey on Artificial Intelligence for Vehicles; Automotive Innovation*, Springer, Berlin, Germany, 2018.

[48] C.-Y. Fana, "Using machine learning to forecast patent quality–take "vehicle networking" industry for example," *Transdisciplinary Engineering: A Paradigm Shift*, vol. 5, 2017.

[49] J. Gu, "Introduction to the special section on machine learning-based internet of vehicles: theory, methodology, and applications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, 2019.

[50] M. Wang, "Machine learning for networking: workflow, advances and opportunities," *IEEE Network*, vol. 32, no. 2, pp. 92–99, 2017.

[51] J. de Hoog, "Improving machine learning-based decision-making through inclusion of data quality," in *Proceedings of the BNAIC/BENELEARN Computer Science*, Brussels, Belgium, 2019.

[52] J. Zerillil, "Algorithmic decision-making and the control problem," *Minds and Machines*, vol. 29, pp. 555–578, 2019.

[53] M. Usama, "Unsupervised machine learning for networking: techniques, applications and research challenges," *IEEE Access*, vol. 7, pp. 65579–65615, 2019.

[54] S. Petros, "A survey on machine-learning techniques for UAV-based communications; MDPI," *Sensors*, vol. 19, no. 23, p. 5170, 2019.

[55] S. Mozaffari, "Deep learning-based vehicle behavior prediction for autonomous driving applications: a review," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2019.

[56] X. Fan, "A deep learning approach for next location prediction," in *Proceedings of the 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design*, Nanjing, China, 2018.

[57] H. Jiang, "Trajectory prediction of vehicles based on deep learning," in *Proceedings of the 4th International Conference on Intelligent Transportation Engineering*, Singapore, 2019.

[58] F. Hussain, "Machine learning for resource management in cellular and IoT networks: potentials, current solutions, and open challenges," 2019, https://arxiv.org/abs/1907.08965.

[59] M. Chen, "Artificial neural networks-based machine learning for wireless networks: a tutorial," 2019, https://arxiv.org/abs/1710.02913.

[60] H. Yang, "Intelligent resource management based on reinforcement learning for ultra-reliable andLow-latency IoV communication networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4157–4169, 2019.

[61] R. Abduljabbar, "Applications of artificial intelligence in transport: an overview," *Sustainability*, vol. 11, no. 1, p. 189, 2019.

[62] Y. Xing, "Driver activity recognition for intelligent vehicles: a deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5379–5390, 2019.

[63] S. Grigorescu, "A survey of deep learning techniques for autonomous driving," 2020, https://arxiv.org/abs/1910.07738.

[64] Z. Peng, "Vehicle safety improvement through deep learning and mobile sensing," *IEEE Network*, vol. 32, no. 4, pp. 28–33, 2018.

[65] M. Abdallah, "Machine learning techniques in ADAS: a review," in *Proceedings of the International Conference on Advances in Computing and Communication Engineering (ICACCE-2018)*, Paris, France, 2018.

[66] J. E. Ball and Bo Tang, "Machine learning and embedded computing in advanced driver assistance systems (ADAS)," *Electronics*, vol. 8, no. 7, p. 748, 2019.

[67] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, "A review of machine learning and IoT in smart transportation," *Future Internet*, vol. 11, no. 4, p. 94, 2019.

[68] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: a survey of emerging trends," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3152–3168, 2019.

[69] A. J. Howard, "Distributed data analytics framework for smart transportation," in *Proceedings of the IEEE 20th International Conference on High Performance Computing and Communications*, Exeter, UK, 2018.

[70] I. Lana, "From data to actions in intelligent transportation systems: a prescription of functional requirements for model actionability," 2020, https://arxiv.org/abs/2002.02210.

[71] H. Nakayama, A. Jamalipour, and N. Kato, "Network-based traitor-tracing technique using traffic pattern," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 300–313, 2010.

[72] R. van der Heijden, "Security architectures in V2Vand V2I communication," in *Proceedings of the 20th Student Conference IT*, pp. 1–10, Enschede, The Netherlands, 2010.

[73] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, and Y. Nemoto, "Detecting black hole attack on AODV-based mobile ad hoc networks by dynamic learning method," *International Journal of Network Security*, vol. 5, no. 3, pp. 338–346, 2007.

[74] Y.-S. Shiu, S. Chang, H.-C. Wu, S. Huang, and H.-H. Chen, "Physical layer security in wireless networks: a tutorial," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 66–74, 2011.

[75] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "VANet security challenges and solutions: a survey," *Vehicular Communications*, vol. 7, pp. 7–20, 2017.

[76] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 85–91, 2007.

[77] H. Nishiyama, D. Fomo, Z. M. Fadlullah, and N. Kato, "Traffic pattern-based content leakage detection for trusted content delivery networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 301–309, 2014.

[78] H. Nakayama, S. Kurosawa, A. Jamalipour, Y. Nemoto, and N. Kato, "A dynamic anomaly detection scheme for AODV-based mobile ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 2471–2481, 2009.

[79] N. Nurelmadina, M. K. Hasan, I. Memon et al., "A systematic review on cognitive radio in low power wide area network for industrial IoT applications," *Sustainability*, vol. 13, no. 1, p. 338, 2021.

[80] J. Liu, Q. Li, R. Sun, X. Du, and M. Guizani, "An efficient anonymous authentication scheme for internet of vehicles," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, 2018.

[81] I. Memon, R. A. Shaikh, M. K. Hasan, R. Hassan, A. U. Haq, and K. A. Zainol, "Protect mobile travelers information in sensitive region based on fuzzy logic in IoT technology," *Security and Communication Networks*, vol. 2020, Article ID 8897098, , 2020.

[82] S. Islam, A.-H. Abdalla, and M. Kamrul Hasan, "Novel multihoming-based flow mobility scheme for proxy NEMO environment: a numerical approach to analyse handoff performance," *Scienceasia*, vol. 43S, no. 1, pp. 27–34, 2017.

[83] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman, and A. S. Ahmed, "Internet of things and its applications: a comprehensive survey," *Symmetry*, vol. 12, no. 10, p. 1674, 2020.

[84] C. Zhang, K. Chen, X. Zeng et al., "Misbehavior detection based on support vector machine and dempster-shafer theory of evidence in VANETs," *IEEE Access*, vol. 6, pp. 59860–59870, 2018.

[85] Z. Meng, "Security enhanced internet of vehicles with Cloud-Fog-Dew computing," *ZTE Communications*, vol. 15, no. S2, 2017.

[86] L. Song, G. Sun, H. Yu, X. Du, and M. Guizani, "FBIA: a fog-based identity authentication scheme for privacy preservation in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5403–5415, 2020.

[87] J. Yakubu, "Security challenges in fog computing environment: a systematic appraisal of current developments," *Journal of Reliable Intelligent Environments*, vol. 27, pp. 467–483, 2019.

[88] J. Pan, "Key enabling technologies for secure and scalable future Fog-IoT architecture: a survey," 2018, https://arxiv.org/abs/1806.06188.

[89] M. K. Hasan, A. F. Ismail, S. Islam, W. Hashim, M. M. Ahmed, and I. Memon, "A novel HGBBDSA-CTI approach for subcarrier allocation in heterogeneous network," *Telecommunication Systems*, vol. 70, no. 2, pp. 245–262, 2019.

[90] S. Islam, O. O. Khalifa, A. A. Hashim et al., "Design and evaluation of a multihoming-based mobility management scheme to support inter technology handoff in PNEMO," *Wireless Personal Communications*, vol. 114, no. 2, pp. 1133–1153, 2020.

[91] M. K. Hasan, M. M. Ahmed, A. H. A. Hashim, A. Razzaque, S. Islam, and B. Pandey, "A novel artificial intelligence based timing synchronization scheme for smart grid applications," *Wireless Personal Communications*, vol. 114, no. 2, pp. 1067–1084, 2020.

[92] J. Stilgoe, "Machine learning, social learning and the governance of self-driving cars," *Social Studies of Science*, vol. 48, no. 1, pp. 25–56, 2018.

[93] O. S. Al-Heety, Z. Zakaria, M. Ismail, M. M. Shakir, S. Alani, and H. Alsariera, "A comprehensive survey: benefits, services, recent works, challenges, security, and use cases for SDN-VANET," *IEEE Access*, vol. 8, pp. 91028–91047, 2020.

[94] S. H. Alrubaee, M. Ismail, M. A. Altahrawi, and B. B. Burhan, "Filter bank multi-carrier modulation technique for vehicle-to-vehicle communication," *Journal of Communications*, vol. 15, no. 7, 2020.

[95] A. Ghazvini, S. N. H. S. Abdullah, M. Kamrul Hasan, and D. Z. A. Bin Kasim, "Crime spatiotemporal prediction with fused objective function in time delay neural network," *IEEE Access*, vol. 8, pp. 115167–115183, 2020.

[96] M. Z. Ibrahim and R. Hassan, "The implementation of internet of things using test bed in the UKMnet environment," *Asia-Pacific Journal of Information Technology & Multimedia*, vol. 8, no. 2, pp. 1–17, 2019.

[97] E. Adi, A. Anwar, Z. Baig, and S. Zeadally, "Machine learning and data analytics for the IoT," *Neural Computing and Applications*, vol. 32, no. 20, pp. 16205–16233, 2020.

[98] T. Yuan, "Harnessing machine learning for next-generation intelligent transportation systems: a survey," in *Proceedings of the Computational Intelligence, Communication Systems and Networks (CICSyN)*, Tetova, Macedonia, 2019.

[99] Syed Junaid Nawaz, "Quantum machine learning for 6G communication networks: state-of-the-art and vision for the future," *IEEE Access*, vol. 7, pp. 46317–46350, 2019.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6673992, 12 pages
https://doi.org/10.1155/2021/6673992

Research Article

Fusion of Machine Learning and Privacy Preserving for Secure Facial Expression Recognition

Asad Ullah,[1] Jing Wang,[2] M. Shahid Anwar,[2] Arshad Ahmad,[3] Shah Nazir,[4] Habib Ullah Khan,[5] and Zesong Fei[2]

[1]*Department of Computer Science & IT, Sarhad University of Science and Information Technology, Peshawar, Pakistan*
[2]*School of Information and Electronics, Beijing Institute of Technology, Beijing, China*
[3]*Department of IT & Computer Science, Pak-Austria Fachhoschule: Institute of Applied Sciences and Technology, Haripur, Pakistan*
[4]*Department of Computer Science, University of Swabi, Swabi, Pakistan*
[5]*Department of Accounting & Information Systems, College of Business & Economics, Qatar University, Doha, Qatar*

Correspondence should be addressed to Jing Wang; wangjing@bit.edu.cn

Received 9 October 2020; Revised 8 December 2020; Accepted 18 January 2021; Published 30 January 2021

Academic Editor: Amir Anees

The interest in Facial Expression Recognition (FER) is increasing day by day due to its practical and potential applications, such as human physiological interaction diagnosis and mental disease detection. This area has received much attention from the research community in recent years and achieved remarkable results; however, a significant improvement is required in spatial problems. This research work presents a novel framework and proposes an effective and robust solution for FER under an unconstrained environment; it also helps us to classify facial images in the client/server model along with preserving privacy. There are a lot of cryptography techniques available but they are computationally expensive; on the other side, we have implemented a lightweight method capable of ensuring secure communication with the help of randomization. Initially, we perform preprocessing techniques to encounter the unconstrained environment. Face detection is performed for the removal of excessive background and it detects the face in the real-world environment. Data augmentation is for the insufficient data regime. A dual-enhanced capsule network is used to handle the spatial problem. The traditional capsule networks are unable to sufficiently extract the features, as the distance varies greatly between facial features. Therefore, the proposed network is capable of spatial transformation due to the action unit aware mechanism and thus forwards the most desiring features for dynamic routing between capsules. The squashing function is used for classification purposes. Simple classification is performed through a single party, whereas we also implemented the client/server model with privacy measurements. Both parties do not trust each other, as they do not know the input of each other. We have elaborated that the effectiveness of our method remains unchanged by preserving privacy by validating the results on four popular and versatile databases that outperform all the homomorphic cryptographic techniques.

1. Introduction

Facial expressions contain the most important nonverbal and rich emotional information in social communication [1]. People communicate with each other through verbal and nonverbal communications [2]. Nonverbal communication involves facial gestures, eye to eye contact, facial expressions, and paralanguage [3]. According to an earlier research, while communicating, 50 percent of the information is conveyed through facial expression, 40 percent through voice, and 8 percent through language. Apart from that, due to the rapid progression in technology, we spend most of the time on electronic devices that carry a variety of software interfaces that are tense, primitive, and nonverbal. Therefore, facial expression recognition should further improve to have a more natural and intelligent human–machine interaction.

Facial expression recognition is used in various domains like Intelligent Tutoring System (ITS), psychology, human-machine interaction, behavioral science, intelligent transportation, and interactive games [4]. It can help monitor the

abnormal expressions in the crowd at public places to avoid any crime. It can also be helpful in the service industry to timely capture the feedback of customers and it can provide timely treatment of patients by looking at the real-time expressions of the patient at the hospital. According to Ekman and Friesman [5], there are six basic expressions: happiness, surprise, disgust, fear, sadness, and anger (some researchers have termed neutral expression as the seventh expression). These expressions are conveyed almost among all species.

Facial expression recognition is widely studied by various researchers. Despite the available research, robust FER is yet an open and challenging task [6, 7] However, most of the recognition algorithms do not consider inter-class variations caused by the differences in facial attributes of the same individual. Hence, mostly, expression classification is done through facial expression information along with identity-related information [8, 9]. The main drawback it carries is that it affects the overall generalization capability of FER systems, thus resulting in degradation of performance on unseen identities [10]. An efficient FER system plays a vital role in the treatment of patients by observing their variable behavior patterns. Happiness expression depicts a healthy and positive mental state while sad and angry demonstrate an unhealthy mental state. Different mental diseases like autism or anxiety are detected due to the emotional conflicts of a particular patient. An important application of FER is E-health care; nowadays, almost 0.3 billion people are suffering from depression, which can also lead to suicidal tendencies if they are not treated timely and effectively [11]. In general, mental health treatment faces a lot of barriers like financial cost, social stigma, and shortage of accessible options. Normally, the clinical staff interviews a patient for identifying symptoms of depression via verbal and nonverbal indicators. Patients are asked to fill a questionnaire for the measurement of depression severity [12]. For timely detection of depression symptoms, an AI-based system will help in entrenching barriers for timely and effective treatment.

In this paper, we use a combination of different techniques to develop a robust model. Initially, we implement different preprocessing techniques to fine-tune and remove highly uncorrelated information in the images. Face detection is performed using facial attributes due to the following reasons: (1) The human face has a unique structure, with the most important local facial parts, such as eyes, mouth, nose, helping us to detect the face in an unconstrained environment. So, the partness map or the response map of five different parts is used in the method. (2) The face adheres to spatial arrangements like the hair being above the eyes and lips below the nose. Hence, the faceness score has been derived from the response configuration. (3) The face hypothesis is performed for the estimation of more accurate face locations. Our contribution is to introduce special attributes supervision to discover facial part responses. We adapt Deep Convolutional Generative Adversarial Network (DCGAN) for data augmentation. It helps us in the demonstration of realistic data augmentation and improvement in the generalization performance in the low-data regime.

For an accurate and robust FER, feature representation of the facial images is the most important step. A considerable amount of research has been done over local and global feature extraction [13]. Fan et. al [14] suggested a model, i.e., MRE-CNN, which aimed to enhance the learning power of the convolutional neural network by considering both the local and global features. Li and Deng [15] introduced the DLP-CNN framework in which the discrimination power of deep features is enhanced while maximizing the interclass scatter and by preserving the locality closeness. Still, they are unable to find the relative relationship between the local features. A face is composed of a certain structure where every part has a relative relationship with the other parts. To address this issue, we propose a method that is capable of spatial transformation due to the action unit aware mechanism and thus forwards the most desiring features for dynamic routing between capsules. Finally, the squashing function is used for classification purposes. We also faced the challenge of achieving classification while having client/server as mutually distrustful of disclosure of the private contents of the facial images and without presenting the result to the server. There are many practical and potential applications, but the main focus is to capture useful and discriminative features. The better feature representations can help to improve the overall efficiency of the system. An appropriate, flexible, and effective facial expression recognition system will add benefit to the industry. There are a lot of standard cryptophic techniques just like secure multiple-parties communication and homomorphic encryption, but they were computationally way too expensive. Thus, we have provided a practical solution to the aforementioned problems. We assess the effectiveness and performance of the introduced model on the Extended Cohn-Kanade, MMI, Oulu-CASIA, and Real-world Affective Faces (RAF) databases. Figure 1 shows some sample images from the CK + database.

The main contributions of this paper are as follows:

(1) We propose a network, which is capable of finding the active relationship between the features from different local regions. Spatial information is also introduced by having prior knowledge of the probability of an object's existence.

(2) Implementation of a simple FER without using the cryptographic techniques having high computational complexity.

(3) Simultaneously achieving the same classification accuracy as that of a conventional algorithm (non-privacy-preserving).

The organization of the next sections is as follows. In Section 2, we provide the problems with the existing methods. In Section 3, we elaborate on our novel architecture with the underlying information. Section 4 comprises the results and analysis. Finally, we provide the conclusion of our research and explain the direction for future work in the last section.

Figure 1: Different facial expressions' samples taken from CK + database.

2. Related Work

The main goal of FER is to capture the meaningful features that are discriminative and descriptive, and invariant to facial variations such as occlusion, illumination, pose, and other identity-related details. There are two main methods available for feature extraction: (1) handcrafted method and (2)deep-learning-based method. Nowadays, deep learning methods are gaining remarkable results. However, earlier, mostly facial expression recognitions were based on handcrafted/human-engineered features such as Histograms of Oriented Gradients (HOG) [16], *n*-dimensional scale-invariant feature transform (n: sift) [17], and Local Phase Quantization (LPQ) [18]. These methods are used for the extraction of global as well as detailed information of an individual face. However, the information obtained is from the overall facial region, and it ignores the expression changes in the local regions, which contain the eyes, nose, and mouth. These methods perform pretty well in a lab-controlled environment where subjects pose expressions under constant illumination, stable eye gaze, and head pose movement. Existing handcrafted approaches demonstrate comparatively less recognition accuracy. Efforts are exerted for manually extracting the desired discriminating features that are linked to expression changes. Considering in-the-wild scenarios deep learning methods for the robustness of facial expression recognition have been implemented [19–22]. However, deep representation is affected just because every facial attribute of a particular subject carries a hefty number of variations such as gender, ethnicity, and age of the particular posing expressions. It holds a very big disadvantage, i.e., the generalization capability for any model is highly and negatively affected; as a result of unseen objects, the performance of facial expression recognition is degraded. Although quite a lot of work has been done toward improving the performance of FER, alleviating the influence of inter-subject variations is still a challenge and an open area of research.

Several techniques have been implemented by reducing intra-class variations and by increasing the interclass differences, which further increases the discriminating property of the features extracted for FER in the real-time scenario [23]. Identity-Aware CNN (IACNN) proposed that by reducing the influence of identity-related information with the use of expression and identity-sensitive contrastive losses, the facial expression recognition performance can be enhanced [24]. The island loss has been proposed for extracting the effective discriminative features for FER [25]. Moreover, in [26], with the use of residue learning the person-independent expression representation has been learned. However, this technique was computationally costly, and due to the same intermediate representation used for the generation of neutral images for the same identities, it also was unable to disentangle the expression information from identity information. However, in [24], due to large data expansion caused by the compilation of training data in image pair forms, the effectiveness of contrastive loss is heavily affected [25]. Similarly, in [27], a fixed identity has been proposed for the transfer of facial expressions to fix the influence of identity relative information. The problem persists with the methods as the efficiency of FER depends on the expression transfer procedure. In short, it has been noticed that FER based on the deep learning methods has outperformed the traditional handcrafted methods. However, there is still a gap in deep learning because very few studies have employed facial depth images in the deep networks as an input. Compared with the existing models, the main goal is to design a network that can be fully adopted for the decomposition of the facial region, easy to implement, and is robust.

Different researchers have implemented different methods to ensure privacy. In [28], the privacy-preserving data classification was done with the use of Principal Component Analysis for feature extraction, and for classification, the nearest neighbor was used. However, it failed to perform in the presence of nonlinear facial variations. Fisher Linear Discriminant Analysis has been proposed in [29] and it had less error rate compared to PCA. However, it did not work well for maintaining the privacy of the discriminative features of a specific class in the multimodal class. Hence, LFDA was proposed to overcome the deficiencies in the FLDA. The work in [30] meets the privacy requirements by hiding the test image and achieving results using the Paillier Homo-morphic encryption [31]. In the research work in

[32], the author proposed EPOM that achieves the goal of secure integer number processing without resulting in privacy leakage of data to unauthorized parties. In [33], it has been proposed that subprotocols can dramatically reduce the number of messages exchanged during the iterative approximation process based on the coordinate rotation digital computer algorithm. Due to the large keys for the encryption as well as decryption, it involves computationally intensive operations such as a large number of exponents. Meanwhile, it also has a limited number of operations during the classification of data, which makes the client/server communicate even a lot more with each other. Hence, our proposed method can achieve true recognition rate even in the presence of the privacy protocol, which uses randomization and is capable of intense multiplication and addition.

3. Proposed Method

3.1. Preprocessing. Preprocessing is very important as it aims to capture the meaningful features, align, and normalize the most needed visual information conveyed by the facial image. Every real-time image is affected by nonlinear facial variations, i.e., varying illumination, the difference in the contrast between the foreground and background, and irrelevant head poses. Therefore, to get the maximum possible semantic meanings of the features for further training the deep neural network, we need to perform some preprocessing techniques. This step is used for the elimination of highly uncorrelated data in the image.

3.1.1. Face Detection. Face detection is one of the vital steps in the FER because of the excessive background, and there is still highly uncorrelated information in the image even taken from a few benchmark datasets. Most of the datasets have an almost frontal view and high-resolution images. So Viola and Jones algorithm [34] is used in most scenarios.

Faceness-net has been used in this paper. A full image is provided as an input image to the convolutional neural network for generation of partness map. The partness map is generated for different facial parts like eyes, nose, mouth, etc. Facial attributes are further categorized to distinguish it from other parts, just like how hair can be blond, black, wavy, straight, etc. Therefore, in the next stage, face proposals are much more refined, so that the usefulness of facial attributes are explored for learning an optimized and robust face detection. A CNN is trained over uncropped images and is used for obtaining face part detectors without any explicit part supervision. The faceness score is evaluated based on the face part responses and considering the spatial arrangements associated with them. After the generation of face proposals, a strong face detector is trained and it outperforms all other methods.

In Figure 2, the face is divided into five important parts, where eyes, nose, and hair are much more effective as compared to mouth and beard, which can be partially occluded. Therefore, the combination of facial parts gives much better results compared to individual facial parts.

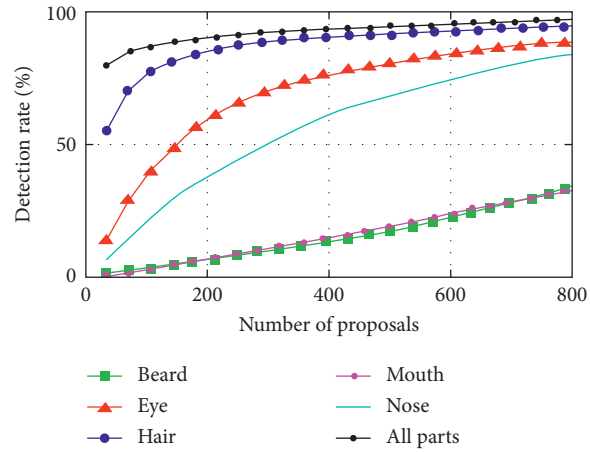


Figure 2: Impact of different facial parts to face proposal (individual or as a combination).

3.1.2. Data Augmentation. As far as the deep neural network is chosen for FER, data augmentation is used to produce much better results by providing a large amount of data. It is effective in the generalization capability of the model as many of the publicly available datasets are not large enough to validate the results more efficiently. Large training data yield to a well-trained model.

There are some standard methods of data augmentation like skewing, rotating, shifting, changing the color scheme, resizing the image, and enhancement of image noise [22]. To automatically learn the augmented data in the low-data setting, we have used Deep Convolutional Generative Adversarial Network (DCGAN). It is used for the alleviation of the overfitting problem over the on-the-fly data. The samples provided as input are randomly cropped from all the four sides and then a horizontal flip is performed for making a dataset ten times bigger than the original one.

3.2. Dual Enhanced Capsule Network (DE-Capsnet). The entire network has been shown in Figure 3, where the model is divided into portions. Firstly, we have to preprocess the images to avoid the uncorrelated information linked to the facial image. Then, we have two modules for further processing. In the first part, the box with the purple dashed line is attention aware of action units and consists of deep convolutional layers for the extraction of the enhanced features maps, and this has been termed as enhancement module 1. In the later part, with the use of dynamic routing, those enhanced feature maps are encoded between capsules, and the process of decoding is done by the fully connected layers (the process has been shown in the green dashed lines). At the end, the squashing function is used for the recognition of facial expressions.

VGG19 is used in enhancement module 1 because it is very much robust in object classification besides having a simple architecture. For a better understanding of the description, each stage is having multiple convolutional layers followed by a max-pooling layer. In the first 2 stages, each stage is having 2 convolutional layers. Whereas in the last 3

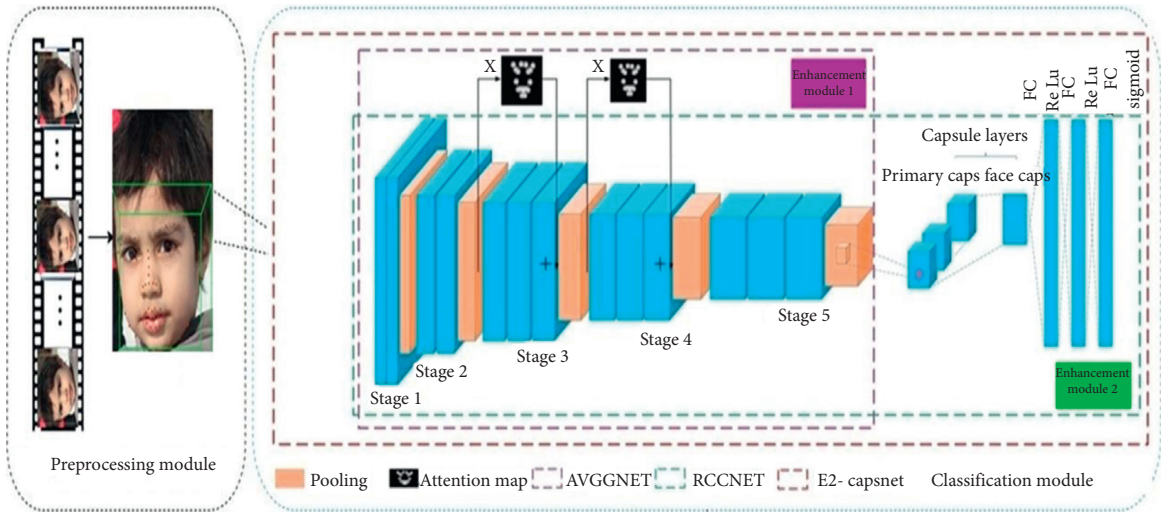


Figure 3: Overview of the proposed method.

stages, each stage is having 3 convolutional layers, respectively. We do not retain the last 3 layers as we have to get the feature maps.

To achieve the attention map, we have used the generation method by Li et al. [35]. Furthermore, we have made appropriate adjustments to the datasets used in our work for getting the key facial landmarks. Figure 3 shows the facial image with blue facial landmarks along with the attention map. Action unit's centres are obtained with key facial points by using scaled distance. To make sure that the scales must be the same among all the facial images, the facial images are resized. Hence, for making the shifting distance among the images as much adaptive as possible, the measurement reference is used to indicate the shift in the distance. To locate the action unit, centres of the inner corner distance have been used as scaled distance. For each action unit, the 7 pixels in the nearby area have been taken in the experiments, as a result, size of each action unit area is 15×15. $\mathbf{H_w}$ is assigned as the higher weight, which is the closest point to the action unit centre.

$$\mathbf{H_w} = 1 - 0.07\mathbf{m_d}. \tag{1}$$

The Manhattan distance is termed as $\mathbf{m_d}$ to action unit centre. Hence, those areas that are having higher values in the attention map correspond to the active areas of action units in facial images, and an attention map will further enhance them.

After the generation of attention maps, the maps are further forwarded to stage 3 and stage 4, as shown in Figure 3. The feature maps which are generated after the pooling layer of the second stage are multiplied with the attention map of the first stage, and after that being parallel with the convolutional layers of the third stage. Hence, the results obtained after the convolution are added element by element and then forwarded to the max-pooling layer of the current stage as an input. A similar operation is done at the fourth stage by jointly combining the convolutional layers with the attention map. Here, we explain the reason behind using attention maps; it is just because all areas are not equally important for facial expression recognition.

After the enhancement module 1, we get $512 \times 7 \times 7$ feature maps. For the dynamic routing the feature maps are further fed between primary capsule layers and face capsule layers. Three fully connected layers are used for decoding and reconstructing the facial image. The nonlinear function, i.e., the squashing function is used for facial expression recognition, which is defined in equation (2) as follows:

$$\mathbf{u_k} = \frac{\|\mathbf{j_k}\|^2}{1 + \|\mathbf{j_k}\|^2} \frac{\mathbf{j_k}}{\|\mathbf{j_k}\|}, \tag{2}$$

where $\mathbf{k}$ is used for the capsule, and $\mathbf{u_k}$ and $\mathbf{j_k}$ are output and input vectors, respectively. $\mathbf{L_m}$ is the minimizing margin loss and $\mathbf{L_r}$ is the reconstructing loss used for updating the parameters in the network. Total loss is defined as $\mathbf{L_t}$. Loss function expressions are defined in the equations (3)–(5), respectively.

$$\mathbf{L_m} = \mathbf{I_{cc}}\,\mathbf{max}\left(0, \mathbf{b}^+ - \|\mathbf{u_{cc}}\|\right)^2 + \lambda\,(1 - \mathbf{I_{cc}})\mathbf{max}\left(0, \|\mathbf{u_{cc}}\| - \mathbf{b}^-\right)^2, \tag{3}$$

$$\mathbf{L_r} = (\mathbf{f_c} - \mathbf{f})^2, \tag{4}$$

$$\mathbf{L_T} = (\mathbf{L_m} + 0.0005\mathbf{L_r})^2, \tag{5}$$

where $\mathbf{cc}$ is termed as the classification category and for that particular category the indication function is denoted by $\mathbf{I_{cc}}$. The upper and lower boundaries are represented by $\mathbf{b}^+$ and $\mathbf{b}^-$, respectively. The f represents the original image, whereas $\mathbf{f_c}$ represents the reconstructed image. This classification is based on one party; the training and testing phase is done by that party. However, we propose a method through which the server will be in charge of training, and testing will be done by both parties collaboratively.

3.3. Information Security. A security algorithm is information-secure in the sense that its security springs purely from scientific theory. The thought of information secure communication was initiated by the applied scientist, Shannon; he further added that the one-time pad system records excellent security subject to the subsequent two conditions [36]:

(1) The key that randomizes the information ought to be random and will be used one time

(2) The length of the key had to be as long as the length of the information

Even if any rule randomizes its parameter and the above conditions are satisfied, it is still hard to unmask the parameters even if an adversary is having exceptional computation power; e.g., if the random pace is the same as the message space, and is adequate to 1024–bits, then prior and posterior probabilities are the same, i.e., there is no particular advantage to urging posterior probability than prior probability.

3.4. Privacy Preserving Security. The main theme is to ensure secure operation between the client and the server. Both of them want to communicate with each other, and for that purpose need to compute $u^T v + p$. Where u is a vector known to client and v is a vector known to server with p being a scalar. However, only client will know the outcome of $u^T v + p$.

Where u input to client is composed of integers and v input to server is composed of floating points. Since we tend to perform integer random numbers, the process of conversion into integers is achieved by scaling the elements of the vector v, and it is approximated to the nearest integer. We use the scaling factor s that is large in $(u^T sv + s * p)$. First client adds random numbers in the vector v and server does few operations and returns the result to client. So, the operation is made valid by first scaling the scalar p and vector v by scaling the factor s, then the outcome is divided by that scaling factor.

So one thing is for sure, i.e., the server won't know anything about the client input and the same will be the case with the client. The client will just get to know about the result without having any knowledge about server vector and scalar. Hence, the above process is called a two-party protocol, which is completely information secure. Figure 4 demonstrates that any unknown input face image of any identity can be applied to synthesize a realistic equivalent face image of any other image.

3.5. Facial Expression Recognition Based on Privacy Preserving. The first step of every procedure is to mark the basic requirements and then fulfill them accordingly. The 3 requirements meeting with this process are as follows:

Requirement 1: without using more sophisticated public encryption system.

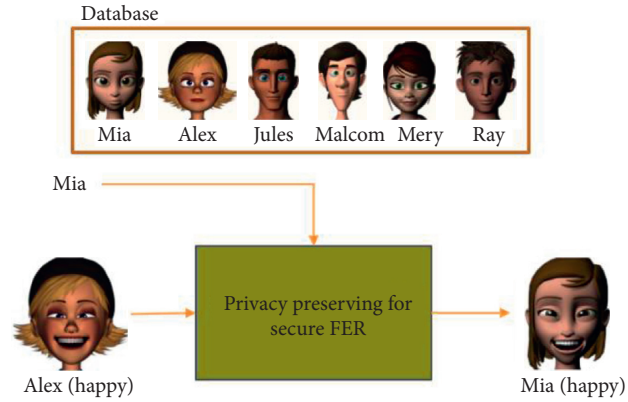


Figure 4: Application to an input face image of any identity to synthesize a realistic, expression-equivalent output face image of a target identity.

Requirement 2: hide a sample of client input data and server split result.

Requirement 3: hide server classification parameters and client will be unaware of means of database.

To explain this, let us break down the traditional assessment phase into four steps as follows:

Step 1. Find test image difference.

$$\text{test} = \widetilde{\text{test}} + \overline{b}, \tag{6}$$

where test is a difference test image, $\widetilde{\text{test}}$ is a testing image, and $\overline{b}$ is mean of database.

At the start, the client cannot send test image due to privacy issues. Therefore, the client only sends the image with the noise vector, $n \sim \in Z_n * 1$ having the same size as the test image. Since the server only receives the noise vector, it receives no information about the test image vector. So the difference noise vector is given as

$$n = \widetilde{n} + \overline{b}. \tag{7}$$

However, the difference between test image and noise image is just known to them. Let's represent it as

$$r = \widetilde{n} - \widetilde{\text{test}} = n - \text{test}. \tag{8}$$

Step 2. Illustration of the lower extremity difference.

$$a = B^T * \text{test}, \tag{9}$$

where B is the transformation matrix, a is a low dimensional vector corresponding to test. However, the server needs to project a low-dimensional vector with noise image given below:

$$\overline{a} = B^T * n. \tag{10}$$

Step 3. Euclidean distance calculation.

$$\overline{Ed_i} = \left\| \overline{a} - j_i \right\|_2^2, \tag{11}$$

where j_i is the training image (low dimensional) and $\overline{Ed_i}$ is the Euclidean distance and $i = 1, 2 \ldots..N$.

Step 4. Calculation of distance length to match the test image in the known section.

$$\overline{Ed_i} = Ed_i - m_i, \tag{12}$$

where the matching training image is denoted by m_i, but it is hard for the server to calculate the original distance Ed_i because the server doesn't know the r vector and the test image. So, in order to attain the matching image, the server will send all the $\overline{Ed_i}$, with a random number r_i for each $\overline{Ed_i}$ where

$$\widetilde{Ed_i} = \overline{Ed_i} + r_i = Ed_i + m_i + r_i. \tag{13}$$

Now the client can calculate the actual distance from the above equation as

$$Ed_i = \widetilde{Ed_i} - (m_i + r_i). \tag{14}$$

It is to make sure that only client knows the m_i; if the server gets to know about that, then he can calculate Euclidean distances between the provided test images and training images. Thus, the server will be able to find the expression corresponding with the test image and ultimately it will effect privacy.

3.6. Privacy Analysis. In this part of the research, we are interested in knowing whether our method is susceptible to any privacy leakage. Our method is based on the computation of both parties and therefore the only single possibility of privacy leakage can be the interaction between both the parties. To prove that our method does not leak unwanted information to a client or server, Goldreich's Privacy definition is used [37]:

3.7. Definition of Secured Privacy for Both Parties' Computation. The protocol we use for security should not disclose the hidden information to a third party (semi-honest) except the information that can be triggered by looking at the input and output of those parties.

Our primary purpose is to verify whether the proposed two-party calculation satisfies the definition of privacy or not. In the above four steps, it is clearly mentioned to the client and the server about their inputs and outputs. Therefore, we have to make sure that both of them don't infer other than the known inputs and outputs so that the proposed method would make sure that privacy is assured.

The client's ultimate goal is to make sure that the server is unaware of the test image and also just keep away the classification result. On the other side, the server had to keep the classification parameters away from the client. The client will just share the noise image initially, instead of sharing the true/original image; however, the size of both the images will remain the same. So, the server will know the size, and it will not be a privacy leakage. In return, the server also shares the random Euclidean distance obtained with the help of a random integer. Hence, information-theoretic security is achieved.

4. Results and Discussion

We have used four most popular databases for populating the results. These databases are CK+ [38], MMI [39], Oulu-CASIA [40], and RAF [23]. The RAF is used for large posed and real-world expressions, as the first three don't have large posed expressions. So to check the robustness of our method over large posed expressions, we have used the RAF data base.

4.1. Description of Databases. The Extended Cohn-Kanade database is the widest and the most popular database used in facial expression recognition. It contains 593 video sequences, which do vary from 10–60 frames with a shift from neutral to other expressions. There are a total of 123 subjects who performed different expressions, the ages of the subjects ranging from 18 to 30 years. Out of the 123 subjects, most of them are females. A total of 327 video sequences out of them are categorized into seven expressions. The core reason behind the algorithms not being uniform over CK+ is that it doesn't provide specific training, validation, and test sets.

The MMI database is laboratory-controlled and 75 subjects have performed 2900 expressions, both video sequences and static images with high resolution, out of which 326 video sequences are obtained from 32 subjects. The MMI database is different from CK+ as it uses onsets, offsets, and apex phases. In the sequences, the neutral expression is performed at the start of every sequence and reaches the peak and then returns back to the neutral expression. This database has very challenging conditions, i.e., it takes care of large inter-personal variations; every subject is performing different nonuniform expressions while wearing glasses, mustaches, etc.

The Oulu-CASIA database consists of 2880 images from 80 subjects for six expressions; most of them are males aged between 23 and 58 years. This database is specially designed to tackle the problem of illumination due to environmental changes. It consists of two different imaging systems; the first one is Near Infrared (NIR), whereas the second one is Visible Light (VIS). There are 3 different variable illumination scenarios: the first one is normal indoor illumination; the second one is used for weak illumination considering the scenario where just the computer display is on; and the third one is having all the lights off, i.e., dark illumination.

The Real-world Affective Faces Database is used, which consists of 29672 great, diverse real-world facial images. These images are downloaded from the Internet based on the approach of crowdsourcing; 40 annotators are used for independently labeling each image. This database consists of the large variability in different subjects' gender, age, ethnicity, varying lighting conditions, head pose, eye gaze, occlusions, and post-processing operations, which helps us to validate our network over versatile databases.

4.2. Implementation Details. The facial image is first preprocessed using face detection, data augmentation, and illumination normalization for fine-tuning of the image. The highly uncorrelated data are removed in order to process them further for a high-quality result. Then, the landmark detection is used to identify the key facial points. After that, VGG19 is used as a backbone of the network, where feature maps of $512 \times 7 \times 7$ are obtained after the 1st enhancement module. Then, $256 \times 6 \times 6$ feature maps are obtained from 2×2 convolutional kernels having the stride value of one; those feature maps are further forwarded to primary capsule layers with an 8D capsule and 32 convolutional layers. There are 3 routing iterations which are then executed between the primary capsule layers and the Face Capsule layers. Every expression is having 16D Capsules, where all the lower capsules forward information to the above capsule. Then with 3 fully connected layers, we use the squashing function for further classification. Adam optimizer is used for learning with a rate of 0.0001. The value of $\mathbf{b}^+$ is 0.9 and $\mathbf{b}^-$ is 0.1. Furthermore, the batch size is set to 16 and the maximum iteration is set to 300. Our whole network training is end to end.

In the Extended Cohn-Kanade database, we take the last frame to three frames and consider the first frame as a neutral expression for data selection. The subjects have been divided into a group of 10, and a 10-fold cross-validation is performed. Table 1 shows the average accuracy rates compared with other existing state-of-the-art methods. Our image-based method achieves the highest accuracy of 98.95 percent against sequence-based techniques that extract the features from a sequence of images or videos.

In the MMI database, we take three frames from the middle of each sequence that is associated with peak information and develop a dataset consisting of 624 images. Afterward, the data augmentation is performed and then distributed among 10 sets. For experimentation, the 10 cross-fold person independent validation is performed using the first frame, i.e., neutral expression, and it takes three peak frames from every frontal sequence. Table 2 shows the dominance in the average accuracy rates compared with other existing methods.

In the Oulu-CASIA database for training and testing, we use the last three frames from every sequence. A 10-fold cross-validation is performed just like CK+ in which based on the subject, each fold is completely disjointed with all the remaining folds. Table 3 shows the average accuracy rates, which outperform all novel methods. It achieves the highest accuracy of 91.2 percent.

Just like other databases in the RAF database, we perform a 10-fold cross-validation too. Table 4 shows the average accuracy rates of our method on the RAF database. We first obtained the true positives, false positives, true negatives, and false negatives, and then over 10 folds we calculated the F1 score and precision per class. Figure 5(a) shows the per-class precision and Figure 5(b) shows the per-class F1 score on the following databases.

4.3. Threats to Validation. There are a few factors that can enhance the robustness of facial expression recognition. While validating our approach, there are some limitations to the existing publicly available novel databases. The recognition of the expression with a closed mouth is less accurate as compared with the expression with an open mouth. Considering the agreement of facial expressions by face angles, we noticed that perceived arousal from the frontal face is more than compared with the shift in face angle. The happiness, disgust with closed mouth, and surprise remains unaffected with the face turned away. Furthermore, the

Table 1: The performance comparison of different approaches on the CK + database.

Method	Accuracy
LBP TOP [41]	88.99
HOG 3D [16]	91.44
MSR [42]	91.40
STM-explet [43]	94.19
DTAGN [44]	97.27
3D-CNN-DAP [45]	92.4
NMF-SSCCA [46]	97.3
FER-MPI-SFL (baseline) [47]	98.2
(Ours)	98.95

Table 2: The performance comparison of different approaches on the MMI database.

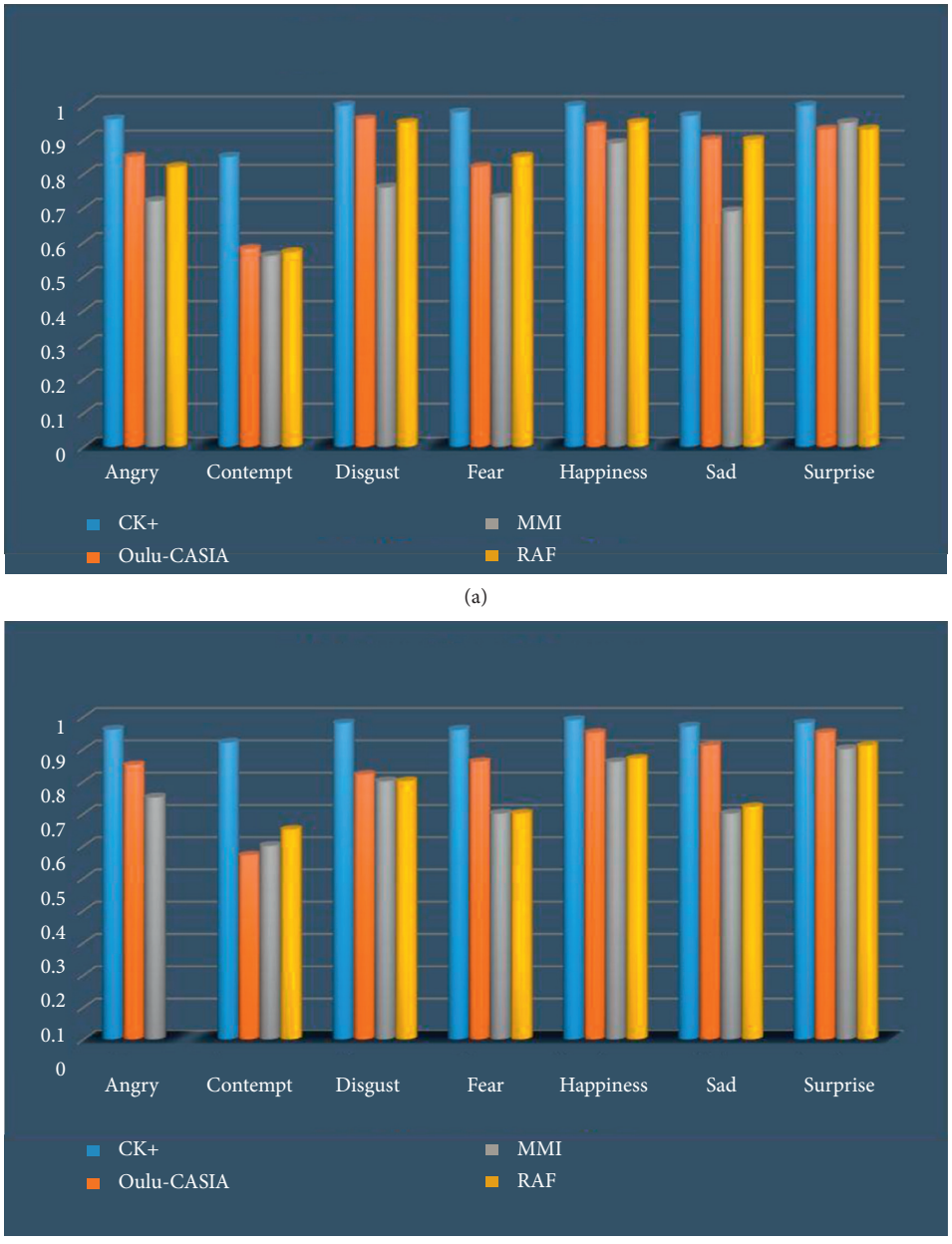
Method	Accuracy
LBP TOP [41]	59.51
HOG 3D [16]	60.89
CSPL [48]	73.53
STM-explet [43]	75.2
DTAGN-joint [44]	70.3
3D-CNN-DAP [45]	63.4
FER-MPI-SFL (baseline) [47]	83.1
(Ours)	89.31

Table 3: The performance comparison of different approaches on Oulu-CASIA database.

Method	Accuracy
LBP TOP [41]	68.1
HOG 3D [16]	70.6
STM-explet [43]	74.59
Atlases [49]	75.52
DTAGN-joint [44]	81.46
FN2EN [50]	87.71
PPDN [51]	84.59
FER-MPI-SFL (baseline) [47]	87.39
(Ours)	91.2

Table 4: The performance comparison of different approaches on RAF database.

Method	Accuracy
E2E-FC	23.99
AIR [52]	67.37
NAL [53]	84.22
IPA2LT (EM [54] + CNN)	85.30
IPA2(LTNET) (baseline) [55]	86.77
(Ours)	97.15



(a)

(b)

Figure 5: Performance metrics on four databases. (a) Per-class precision on four databases. (b) Per-class F1 score on four databases.

effective valence near the frontal is conveyed more by the full left-side profile rather than the full right side profile. It is because of this reason that the left hemiface observes a more spontaneous response than the right hemiface. The facial expression analysis can be enhanced by the facial motion information if the image is subtle or degraded. The dynamic neutral expression with the blinking of eyes or chewing is also a threat. Moreover, the dwell time is also a key factor; it takes more time over eyes than the mouth. However, the dwell time over the mouth of happy expression is relatively high. With an increase in the intensity, it can also be noticed that the accuracy is also increased, whereas the dwell time and round trip is decreased. Overall, the response time of females is faster than males even in a low-intensity environment. In the end, it was also concluded that the dwell time of the female eye is more than that of the male.

5. Conclusions

In this paper, we have introduced a state-of-the-art architecture that is robust and effective. A facial image is first preprocessed using different techniques to counter the problems of the excessive background, limitation of data, varying illumination, pose-variation, and occlusion. The facial image is fine-tuned and then forwarded to a dual enhanced capsule network that is capable of handling the

spatial transformation. It uses action units aware mechanism, which helps to locate the active areas, which can help in better facial expression recognition. The feature representation ability is enhanced due to multiple convolutional layers and it helps to capture the key information present in the particular structure of the face. We performed the privacy preservation with the help of a randomization technique, which added the benefit of less computationally expensive. It also performs secure communication between the two untrusted parties.

Different databases have different sets of pictures under varying conditions. As a result, class imbalance occurs due to the inconsistency in expression annotations. So a cost-sensitive layer can be enhanced for training the deep neural networks. Meanwhile, a powerful, deep neural network can be designed to have prior knowledge of the change in the local environment, which is capable of predicting specific parameters and inherently handling and recovering facial occlusions without any intervention. Furthermore, to improve the robustness of the FER, it can be fused with other models. The incorporation with other modalities like depth information from three-dimensional face models, neurosciences, cognitive sciences, infrared images, and physiological data can be a good future research direction.

Data Availability

All the data are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] N. Samadiani, G. Huang, B. Cai et al., "A review on automatic facial expression recognition systems assisted by multimodal sensor data," *Sensors*, vol. 19, no. 8, p. 1863, 2019.

[2] L. Zhang, B. Verma, D. Tjondronegoro, and V. Chandran, "Facial expression analysis under partial occlusion: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, p. 25, 2018.

[3] A. K. Vail, T. Baltrušaitis, L. Pennant, E. Liebson, J. Baker, and L. P. Morency, "Visual attention in schizophrenia: eye contact and gaze aversion during clinical interactions," in *Proceedings of the 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 490–497, IEEE, San Antonio, TX, USA, October 2017.

[4] A. Ullah, J. Wang, M. S. Anwar, U. Ahmad, U. Saeed, and Z. Fei, "Facial expression recognition of nonlinear facial variations using deep locality de-expression residue learning in the wild," *Electronics*, vol. 8, no. 12, p. 1487, 2019.

[5] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," *Journal of Personality and Social Psychology*, vol. 17, no. 2, p. 124, 1971.

[6] J. Kumari, R. Rajesh, and K. M. Pooja, "Facial expression recognition: a survey," *Procedia Computer Science*, vol. 58, pp. 486–491, 2015.

[7] A. Ullah, J. Wang, M. S. Anwar, U. Ahmad, U. Saeed, and J. Wang, "Feature extraction based on canonical correlation analysis using FMEDA and DPA for facial expression recognition with RNN," in *Proceedings of the 2018 14th IEEE International Conference on Signal Processing (ICSP)*, pp. 418–423, IEEE, Beijing, China, August 2018.

[8] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. O'Reilly, and Y. Tong, "Identity-free facial expression recognition using conditional generative adversarial network," 2019, http://arxiv.org/abs/1903.08051.

[9] H. Yang, Z. Zhang, and L. Yin, "Identity-adaptive facial expression recognition through expression regeneration using conditional generative adversarial networks," in *Proceedings of the 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 294–301, IEEE, Xi'an, China, May 2018.

[10] N. Van Quang, J. Chun, and T. Tokuyama, "CapsuleNet for micro-expression recognition," in *Proceedings of the 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pp. 1–7, IEEE, Lille, France, May 2019.

[11] W. H. Organization and others, "Depression: key facts," 2018.

[12] K. Kroenke, T. W. Strine, R. L. Spitzer, J. B. W. Williams, J. T. Berry, and A. H. Mokdad, "The PHQ-8 as a measure of current depression in the general population," *Journal of Affective Disorders*, vol. 114, no. 1-3, pp. 163–173, 2009.

[13] M. Zhu, D. Shi, and J. Gao, "Branched convolutional neural networks incorporated with jacobian deep regression for facial landmark detection," *Neural Networks*, vol. 118, pp. 127–139, 2019.

[14] Y. Fan, J. C. Lam, and V. O. Li, "Multi-region ensemble convolutional neural network for facial expression recognition," in *Proceedings of the 27th International Conference on Artificial Neural Networks*, pp. 84–94, Springer, Rhodes, Greece, October 2018.

[15] S. Li and W. Deng, "Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition," *IEEE Transactions on Image Processing*, vol. 28, pp. 356–370, 2018.

[16] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proceedings of the British Machine Vision Conference 2008*, Leeds, UK, September 2008.

[17] W. Cheung and G. Hamarneh, "*n*-SIFT: *n*-Dimensional scale invariant feature transform," *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 2012–2021, 2009.

[18] B. Jiang, M. F. Valstar, and M. Pantic, "Action unit detection using sparse appearance descriptors in space-time video volumes," in *Proceedings of the Face and Gesture 2011*, pp. 314–321, IEEE, Santa Barbara, CA, USA, March 2011.

[19] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen, "Deeply learning deformable facial action parts model for dynamic expression analysis," in *Proceedings of the Asian conference on computer vision*, pp. 143–157, Springer, Singapore, November 2014.

[20] B. K. Kim, H. Lee, J. Roh, and S. Y. Lee, "Hierarchical committee of deep cnns with exponentially-weighted decision fusion for static facial expression recognition," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pp. 427–434, ACM, Seattle, WA, USA, November 2015.

[21] H. W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pp. 443–449, ACM, Seattle, WA, USA, November 2015.

[22] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pp. 435–442, ACM, Seattle, WA, USA, November 2015.

[23] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2852–2861, Honolulu, HI, USA, July 2017.

[24] Z. Meng, P. Liu, J. Cai, S. Han, and Y. Tong, "Identity-aware convolutional neural network for facial expression recognition," in *Proceedings of the 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 558–565, IEEE, Washington, DC, USA, June 2017.

[25] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. O'Reilly, and Y. Tong, "Island loss for learning discriminative features in facial expression recognition," in *Proceedings of the 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 302–309, IEEE, Xi'an, China, May 2018.

[26] H. Yang, U. Ciftci, and L. Yin, "Facial expression recognition by de-expression residue learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2168–2177, Salt Lake City, UT, USA, June 2018.

[27] K. Ali, I. Isler, and C. Hughes, "Facial expression recognition using human to animated-character expression translation," 2019, http://arxiv.org/abs/1910.05595.

[28] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *Proceedings of the 9th International Symposium on Privacy Enhancing Technologies, PETS'09*, pp. 235–253, Seattle, WA, USA, August 2009.

[29] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[30] Y. Rahulamathavan, R. Phan, J. Chambers, and D. Parish, "Facial expression recognition in the encrypted domain based on local Fisher discriminant analysis," *IEEE Transactions on Affective Computing*, vol. 4, no. 1, pp. 83–92, 2012.

[31] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology EUROCRYPT 99*, pp. 223–238, Springer, Berlin, Germany, 1999.

[32] X. Liu, H. Robert, Deng, Kim-Kwang, R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, 2016.

[33] Y. Liu, Z. Ma, X. Liu, S. Ma, and K. Ren, "Privacy-preserving object detection for medical images with faster R-CNN," in *Proceedings of the IEEE Transactions on Information Forensics and Security*, Barcelona, Spain, October 2019.

[34] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, WI, USA, June 2003.

[35] W. Li, F. Abtahi, Z. Zhu, and L. Yin, "EAC-net: deep nets with enhancing and cropping for facial action unit detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 11, pp. 2583–2596, 2018.

[36] C. E. Shannon, "Communication theory of secrecy systems*," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[37] O. Goldreich, "Secure multiparty computation," 1998, http://www.wisdom.weizmann.ac.il/oded/pp.html.

[38] P. Lucey, J. F. Cohn, T. Kanade, and J. Saragih, "The Extended Cohn-Kanade Dataset (CK+): a complete dataset for action unit and emotion-specified expression," in *Proceedings of the Computer Vision and Pattern Recognition Workshops*, San Francisc, CA, USA, June 2010.

[39] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, Amsterdam, Netherlands, July 2005.

[40] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. Pietikäinen, "Facial expression recognition from near-infrared videos," *Image and Vision Computing*, vol. 29, no. 9, pp. 607–619, 2011.

[41] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, 2007.

[42] R. Ptucha and A. Savakis, "Manifold based sparse representation for facial understanding in natural images," *Image Vision Computing*, vol. 31, no. 5, pp. 365–378, 2013.

[43] M. Liu, S. Shan, R. Wang, and X. Chen, "Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, June 2014.

[44] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, CL, USA, December 2015.

[45] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen, "Deeply learning deformable facial action parts model for dynamic expression analysis," in *Proceedings of the ACCV*, Singapore, November 2014.

[46] A. Ullah, J. Wang, M. S. Anwar, U. Ahmad, J. Wang, and U. Saeed, "Nonlinear manifold feature extraction based on spectral supervised canonical correlation analysis for facial expression recognition with RRNN," in *Proceedings of the 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–6, Beijing, China, October 2018.

[47] W. Wu, Y. Yin, Y. Wang, X. Wang, and D. Xu, "Facial expression recognition for different pose faces based on special landmark detection," in *Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1524–1529, IEEE, Beijing, China, August 2018.

[48] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas, "Learning active facial patches for expression analysis," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2562–2569, IEEE, Providence, RI, USA, May 2012.

[49] Y. Guo, G. Zhao, and M. Pietikäinen, "Dynamic facial expression recognition using longitudinal facial expression atlases," in *Proceedings of the European Conference on Computer Vision*, pp. 631–644, Springer, Florence, Italy, October 2012.

[50] H. Ding, S. K. Zhou, and R. Chellappa, "FaceNet2ExpNet: regularizing a deep face recognition net for expression recognition," 2016, http://arxiv.org/abs/1609.06591.

[51] X. Zhao, X. Liang, L. Liu et al., "Peak-piloted deep network for facial expression recognition," in *Proceedings of the European Conference on Computer Vision*, Amsterdam, Netherlands, October 2016.

[52] S. Azadi, J. Feng, S. Jegelka, and T. Darrell, "Auxiliary image regularization for deep cnns with noisy labels," 2015, http://arxiv.org/abs/1511.07069.

[53] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," 2016.

[54] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 20–28, 1979.

[55] J. Zeng, S. Shan, and X. Chen, "Facial expression recognition with inconsistently annotated datasets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 222–237, Munich, Germany, September 2018.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8897098, 12 pages
https://doi.org/10.1155/2020/8897098

Research Article

Protect Mobile Travelers Information in Sensitive Region Based on Fuzzy Logic in IoT Technology

Imran Memon,[1,2] **Riaz Ahmed Shaikh**,[1] **Mohammad Kamrul Hasan**,[3] **Rosilah Hassan**,[3] **Amin Ul Haq**,[4] **and Khairul Akram Zainol**[5]

[1]*Department of Computer Science, Shah Abdul Latif University, Khairpur, Sindh, Pakistan*
[2]*Department of Computer Science, Bahria University, Karachi Campus, Sindh, Pakistan*
[3]*Network and Communication Technology Lab, Center for Cyber Security, The National University of Malaysia (UKM), UKM, Selangor 43600, Malaysia*
[4]*University of Electronic Science and Technology, Chengdu, Sichuan, China*
[5]*Digital Forensics Lab, Center for Cyber Security, The National University of Malaysia (UKM), UKM, Selangor 43600, Malaysia*

Correspondence should be addressed to Mohammad Kamrul Hasan; hasankamrul@ieee.org and Rosilah Hassan; rosilah@ukm.edu.my

Received 22 July 2020; Revised 16 September 2020; Accepted 21 October 2020; Published 18 November 2020

Academic Editor: Sajjad Shaukat

The Internet of Things (IoT) is susceptible to several identities, primarily based on attacks. However, these attacks are controlling for IoT due to extraordinary growth in consumers' density and slight analysis with low power access nodes. In this work, we explore the possible flaws associated with security for IoT environment insensitively meant for transfer conditions. We proposed a novel design aimed at detecting a spoofing attack that inspects the probability distributions of received power founded for the regions designed for mobile (moving) users. Additionally, we examine the influence on the Confidentiality Scope of targeted consumers in the absence and presence of observer. Our approaches were done through simulation results used for three diverse regions. Grounded on outcomes, we suggest an algorithm called MTFLA, which will guarantee detection and protection techniques intended to protect vastly sensitive areas, i.e., wherever the chance of an attack is maximized. We provide a comparison among various security algorithms prepared for the energy consumption of different patterns. Simulation results revealed that the proposed algorithm for protection (MTFL) is verified to be energy-proficient (secure garnering). It decreases the energy prerequisite for encrypting the data. We evaluated our techniques over simulation results for sensitive region information built on fuzzy logic.

1. Introduction

With the development of portable (Mobile) gadgets and applications, an intense growth in data rates demanded through consumers has tired a depiction of wireless communication. This can be accomplished with the help of installing small cells (or access nodes) in plenty of regions of the huge stream of traffic required to accomplish every consumer with high-frequency data necessity [1]. Minor cells have minor analysis (coverage) and little power access nodes and take applicability on behalf of indoor and outdoor scenarios. Consequently, IoT appears to be an auspicious (promising) approach and has provided the scholars with a new platform to explore the possible advantages of this technology [2]. In IoT, the consumers are nearer with their access nodes [3]. However, the identification cannot be accomplished infinitely; there has to be an essential constraint (limit) on the level of identification. IoT presents an innovative analysis (coverage) scenario and can be utilized by the cell users in homes, educational zones, roads, shopping malls, workplace buildings, and so forth. The Internet of Things (IoT) represents a major and significant component for the 4.0 industrial revolution, and its implementation requires extensive research to ensure

correct operation [4]. The overall structures and challenges of IoT are mainly in security. The adjacency among the consumers-BSs and the open nature of the wireless channel will produce security alarms for the consumers [5]. Thus, security is dominant in IoT. Consequently, IoT has several complex security challenges when achievements are being made to recover spectral adaptability by captivating into consideration deployment challenges [6]. The DTN gives consistent interchangeable widespread scope of systems that do not have great execution qualities. DTN can interconnect vehicles locally where current systems administration convention cannot arrive at the goal. For between-vehicle correspondences, there are various kinds of correspondence, for example, Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), and Vehicle-to-X (V2X) interchanges [7]; IEEE 802.11p supports those communications in outside situations. It characterizes upgrades to 802.11 essential to the help of Intelligent Transport System (ITS) applications. The innovation works on 5 : 9 GHz in different prompting situations to rapidly moving vehicles. There are various works for IOTs. In [8], the authors presented the Message Suppression Controller (MSC) for V2V and V2I correspondences. They thought about constraints to control the message concealment progressively. But still a secure parameter stays utilized to determine the term of message disguise. Towards taking care of this issue, the researchers presented Enhanced Message Suppression Controller (EMSC) [9] intended for Vehicular-DTN (V-DTN). The EMSC is an extended adaptation of MSC [10] and can also be exploited to be used for divergent system conditions. In any case, many control bundles were conveyed in the system. Security and confidently in IoT are important towards anticipating vindictive authorities undermining street wellbeing frameworks based upon the IoT structure, possibly making genuine disturbance traffic streams or wellbeing dangers. A few authors have proposed group head measurements which can help with recognizing malevolent vehicles and alleviating their consequence (effect) by rejecting their access to bunch (cluster) assets [11]. [12]. Security of the wellbeing messages may be accomplished by authentication [13]. Ensuring security for data transmission and storage became one of the biggest concerns and challenges of IoT [14]. To create the procedure of approval quicker, vehicles towards the correspondence scope of a Road Side Unit (RSU) can be gathered to a single bunch (cluster) and group head is chosen to verify every one of the vehicles accessible in the group. Arrangement of groups in a dynamic IoT and determination of bunch (cluster) head assumes a significant job remains chosen. In [15], a bunch (cluster) head choice measurement is processed dependent on vehicle bearing, level of availability, an entropy worth determination from the portability of hubs in the system, and a doubt level dependent on the unwavering quality of nodes' bundle handing-off. Vehicles are allotted verifiers, which are neighbors by lower doubt regard. Verifiers screen the system conducting the vehicle and affirm whether it is directing bundles and promoting portability and traffic data that is reliable with the verifier's perspective on the area. The doubt of an incentive for hubs that carry on anomalously is then naturally expanded, while it is diminished for hubs that perform dependably. Along these lines, the reliability of a hub is represented by the bunch (cluster) head choice procedure. The accompanying area is comprised of the investigations completed by different authors pursued by an examination of the current existing modules, the systems utilized by them, and their relative qualities and shortcomings. In [1], the authors center on taking care of the traffic and the executives issues just in urban territories. The strategy utilized by the authors to anticipate clog depends on on-course data and they separate the variables influencing traffic into two classes: physical volume of thick traffic implied for present day and outside happenings. The significant spotlight is on the last class. The key patterns that add to the traffic clog issues in India are evaluated in [2]. The settings and inadequacies of the current strategies and projects have been looked into and a lot of suggestions have been proposed to handle these difficulties. Just the climate information and variables influencing traffic and the executive issues are thought about in [3] and the framework utilized is Hadoop and Â library. To extend and explore traffic obstructing, a lot of exploratory foreseen atmosphere esteems have been used. Information perturbation techniques are utilized to deal with the characteristic proportion between the mining utility and protection assurance. Different strategies are utilized for information mining; however, the imperatives must be fulfilled. There have just been a lot of research studies that led to improving stopping proficiency at shut parking areas which are paid parking areas and bolster reservation of parking spots. There are online applications that give shrewd stopping administrations in shut parking garages. Be that as it may, writing is exceptionally rare as to another and basic kind of parking area, the open parking areas which do not bolster reservation, uninhibitedly accessible for a constrained measure of time, and are regularly put outside possessing a lot of room. Along these lines, there still exists an exploration hole to improve stopping proficiency at an open parking area. It is accepted that the driver conduct could be proficient whenever improved choice help to drivers is advertised. The areal size of the regularly enormous open parking areas could be decreased by improving stopping productivity alongside diminished clog and CO_2 emanations. In [16], the network disintegration-based methodologies were utilized, for example, sparsified solitary worth deterioration and particular worth decay; they are two of the most widely recognized strategies used to address the issues. In [12], Dvir and Vasilakos are acquainted with decrease usage and incorporate the presentation of SVD in applications, for example, content recovery frameworks. In [10], framework disintegration strategies on a psychological oppressor investigation framework have been proposed. In [13], deterioration techniques are involved and additionally utilized in a basic segment methodology proposed to partition information into numerous networks. In [15], Memon et al. proposed highlight determination and its essentialness for breaking down the information later on. In [16] and Liu et al.'s work, it was depicted that performing decay strategies and highlight choice is perhaps the best approach for order

and evacuating highlights have less contorted or annoyed qualities. In [6], secure multiparty calculation (SMC) information utilizes conventions to encode, for example, total security and association without uncovering delicate information to the experts. In [13], fuzzy logic and its enrollments capacity forced on genuine information have demonstrated a product increment. In [5], the authors have recommended the requirement for an intelligent urban transportation approach to stay away from specially appointed mediations to dispense with mayhem and perplexity. The authors have a progressively reasonable, down to earth, and comprehensive way to deal with the issue of urban traffic clog. The observing and displaying framework created in Netherlands for the forecast of traffic and direction in dealing with the equivalence are portrayed in [6]. The fuzzy AR system has been examined and handled by a bunch (cluster) calculation to foresee and oversee blockage in fast systems in [7]. A practical, continuous application to inform the explorers of the present traffic conditions on a given specific fix of the street has been proposed in [8]. The exhibition of the expectation calculation has been improved by utilizing Apache Spark and Hadoop structure. In [9], the authors have attempted to reclassify the worldwide parameters utilized for traffic expectations, for example, thickness and speed, to foresee traffic blockage all the more precisely under heterogeneous states of traffic. In [10], a system to foresee traffic particularly under Indian conditions has been created. Fuzzy logic (FL) is the logical fundamental method of thinking, which is rough instead of being precise. Several studies have been conducted regarding fuzzy logic methods including their usage in resource scheduling to improve the reliability of cloud computing [17] or in the medical area for diagnosing coronary heart disease [18], in addition to their implementation in hardware-based maximum power point tracking controller for PV systems [19]. The significance of FL stems from the fact that most methods of human thinking and particularly good judgment thinking are rough [16]. FL utilizes semantic factors to depict the control parameters. By utilizing generally basic phonetic articulations, it is conceivable to portray and get a handle on extremely complex issues. A significant property of the phonetic factors is the capacity of depicting uncertain parameters. As discussed previously, packet transmission inside the system is principally worked in two modes, that is, impromptu and foundation. In impromptu mode, there is no requirement for a focal facilitator though in foundation mode organizer incorporated methodology for transmission is utilized [8]. In many building applications, topologies are of prime significance. The essentials of these topologies are got from chart hypothesis, a part of science, comprised of many particular diagrams, for example, complete chart, work, and completely associated diagrams [16].

The key objective of this research is the following:

(1) To propose a mechanism that protects against interference

(2) To propose MTFLA that will ensure detection and protection mechanisms

The paper further discusses related work concerning different topologies such as the theoretical model, scalability model, and survivability models present in IoT communication technologies in Section 2, followed by the developed analytic model for probability distributions for different patterns, which is described in the proposed method in Section 3. Experiments and evaluations and conclusion are placed in Sections 4 and 5, respectively.

2. Related Work

A homogeneous node with equivalent energy is supplied with a clustering-based fuzzy logic. This article considers measurements as fuzzy inputs, for example, resting energy and the number of neighboring nodes. Nodes were most likely to remain selected as per the cluster's heads. When the heads of the cluster are decided and the number of control messages received decreases, the same heads of the cluster stay for another round as heads of the cluster, and the third round begins and new ones are selected [13]. The algorithm that clusters homogeneous nodes that have equal energy is provided in [20]. This article considers the number of neighbors resting energy and nodes as per fuzzy outputs. After selecting the heads of the cluster, each head of the cluster estimates its energy to determine how many rounds it can do. This is finished adaptively. Finally, after the end of the period of the cluster head, the selection is reheld and novel heads of clusters are selected. This article's technique differs in three areas from the previous two. First, clustering on heterogeneous nodes with inadequate energy is done in this article. Second, in the preceding methods, the fuzzy parameters were resting energy than the number of neighbors of a node, towards which the distance to the base station is added. Thirdly, as soon as the power of cluster head exceeds a constant threshold, elections are held and novel heads of clusters stay chosen. The suggested determination approach is only contrasted with those clustering heterogeneous nodes as reasonable evaluation by those clustering similarly energy homogeneous nodes. In what follows, algorithms are explained in the simulation section which are compared to the proposed method and some other related works. FBUC [20] is the first algorithm that was proved to be an enhancement on EAUCF [21]. Unlike EAUCF, at the very beginning of the clustering process, FBUC designates a threshold so that the system can decide which of the sensor nodes can be chosen for the impermanent cluster head based on fuzzy logic to develop the random figure in a way so that it can interact with its neighboring nodes. If the impermanent cluster head energy in the neighboring region is more supplementary than further sensor nodes, it will be named as the last head of the cluster. But if its energy is less than one of its neighbors' though, if there is a small amount of energy compared to individual neighbors, then it will be detached due to the list of impermanent cluster heads, resulting in another cluster head existence selected. Clearly, for each round, the clustering process is repeated causing the supplementary loss of the network's energy. FEMCHRP is presented in [22] as a fuzzy protocol. This approach chooses

as clusters diverse network zones to include all nodes in the clusters. Using fuzzy logic, it then selects each of the cluster heads. The existing energy and the length of the base station are the fuzzy engine's input criteria. That protocol permits the sink to use fuzzy logic to pick multiple cluster headers (CHL). Therefore, the maximum resting energy also distance to the leader of cluster heads of the base station remains chosen. Every leader of the cluster can send data either directly or through other head leaders of the cluster to the base station. Clustering here is done by centralized techniques of clustering. However, this increases the number of packets that had been acknowledged. Besides, holding and repeating two elections for every round, there will be a reduction in energy consumptions compared to that of before. Also, one clustering method benefiting from fuzzy logic was discussed in DUCF [23]. While that technique had been capable of selecting the best nodes in every round as cluster heads, holding elections in every round results in energy consumption reduction. Another fuzzy clustering method is the next IFUC algorithm. Clusters are formed and subsequently entirely the neighbors of every node are defined. For every cluster, the chances of each node being chosen as per cluster head will be recognized concerning inputs, for example, resting energy, distance from the base station, and nodes' degree, which is similar to that node's quantity of neighbors. Regardless of selecting the best nodes as per cluster heads, reiterating the process of clustering in every round increases the number of messages sent and received and subsequently increases the networks. In [23], the authors emphasize which methods to set each round's time dimension, prolong the network's lifetime, and increase the amount, which is referred to as the quantity of wireless figures packets sent to the sink node. A lifetime and throughput purpose associated with each round's time duration is deducted. To improve the performance of cluster-based wireless sensor networks, these functions can be used in terms of lifetime and throughput in which the interest-associated nodes individual store adjacent nodes in the incline direction to the sink, end only the nodes that have the greatest energy will be selected for next-hop data transfer when transmitting the data. In [24], the authors put onward minimal energy path that preserves the algorithm of topology control (MPTC). How does MPTC solve the delinquent of greater energy efficiency due to the closed regions that were discussed in SMECN [25]? Nevertheless it continues to maintain on slightest one lowest energy path across each pair of nodes in a communication network. The information confidentiality, accessibility, and transparency are included in the data security and privacy while ensuring that data are not accessible by any illegal party and illegal processes. To comply with regulatory and organizational policies, data privacy can be stated as the proper usage of information. The fuzzy set theory and fuzzy logic (FL) were developed in [26]. For data protection and processes, FL [5, 8, 9] was found to be useful. In modeling, control, decision-making, and automation, there are many successful applications of FL. For systems [22, 26–28], FL is also used to model uncertainties. Adequate results can be achieved by developing FL systems with few inputs. However, founding FL systems with numerous inputs was found to be difficult. Complexity difficulties arise as soon as the FL framework has several inputs [2, 9, 27, 29]. In FL systems, the scope of fuzzy knowledge base (FKB) increases as the number of inputs and input fuzzy sets increases [6, 8]. It has been found that HFL systems [7–9, 23–25, 27, 29] can overwhelm the problems associated with such a large FKB. HFL systems have been used in this research manuscript to improve the classification $o+$ data while simultaneously reducing fuzzy IF-THEN FKB rules of HFL data classification systems [1]. The usage of an HFL system recovers data security and management. In [26], by adding relay nodes, the researchers suggested a cost-effective and energy-efficient model for IoT. A model of energy-aware IoT design can be used to find the optimal number of relay nodes and their position. The proposed scheme, using the integer linear programming model, minimizes the energy consumption for both biosensor nodes and network nodes and also reduces installation costs. The researchers in [26] demonstrated that MNCs' placement would have a major impact on energy efficiency and IoT's lifetime work. The authors also suggested three separate routing schemes for the positioning of MNC by proper metric collection. They showed that, through effective MNC placement, the network lifetime could increase to 47 percent. The authors suggested an energy-efficient routing protocol for IoT, RE-ATTEMPT [15]. Biosensor nodes are located according to their level of energy. The biosensor nodes of high energy are deployed near the MNC, while MNC is located in the center of the human body. Emergency data is transmitted directly (single-hop) to MNC throughout the transmission of routine data through multihop communication. The authors proposed iMSIMPLE routing scheme for IoT in [22]. High performance, energy efficiency, and supported body posture movement were achieved through the proposed routing scheme. Multihop connectivity is used to improve the efficiency of power. Through the intermediate node (forwarder node), sensing data from biosensor nodes is transmitted to MNC. Forwarder node selection is based on the cost function. Co-LAEEBA is suggested to have a collaborative data routing system with limited route loss for the IoT [30]. Based on data priority, multihop and single-hop routing systems are used. A relay-based efficient cooperative networking system for IoT was developed by researchers in [31]. An evaluation method of energy efficiency (EE) and packet error rate (PER) were tested for separate relay nodes. For IoT, a routing algorithm has been proposed to enhance incremental cooperative critical data transmission in static IoT emergencies (EInCo-CESat). The proposed algorithm might have achieved enhanced network stability and reduced packet error rate (PER) and high throughput at high energy utilization costs [29].

3. Attack and Security Requirements

Three processes are needed to identify threats, challenges, and requirements to design and evaluate a new security model [5]. In this section, we reject safety regulations and attack situations in-vehicle networks as well as necessary.

3.1. Attack and Security Threats. The following attacking situations are believed to be plausible in this article: Bogus Message: the purpose behind this type of attack is to transmit incorrect information to the network.

Message alteration occurs when incorrect information is provided or when node-passing information is modified [16]. The requirement involved in this attack is message integrity.

Obstacles: mobile/immovable obstacles, as security threats, can form a NLOS case that blocks direct vehicle contact and prevents vehicles from testing their neighboring nodes properly [30].

3.2. Security Requirements. The purpose of this work is to design a scheme in IoT to provide a secure environment. The following criteria must be fulfilled by a program of secure messaging in an IoT:

Authentication: vehicle's replies to any incident should be based on authenticated communications. Therefore, first, it is necessary to authenticate the senders of the messages [26].

Message integrity: the integrity of the message should be checked as the message could be modified between the moment of sending and receiving, and it should be completely balanced to what it is received. In a broader sense, the message's credibility often requires equal reliability. That is why those messages produced in a closed space and time are more accurate. It should be noted that the dispatcher may be authentic though the message contains data that has been manufactured.

Privacy: the privacy message in IoT is decided by the request situation. Confidentiality can be achieved through the adoption of public or symmetric key encryptions to ensure communication security.

4. Proposed Method

The proposed model gets to the accuracy and reliability of a sender of the event messages by the execution of fuzzy logic. After getting an event message from encompassing vehicles, first, it checks the validation of the sender utilizing the confirmation module. It utilizes ID verification to assess the sender of the occasion message whether it is approved or not. At the same time, it checks the lifetime of the occasion message ascertaining the contrast between the aging time of the message incorporated into the occasion message and the present time. By performing fuzzy logic, it removes the precision level of the area of the occasion incorporated into the message on the off chance that it exists in the nearest mist hubs a while later. Next, it assesses the trustworthiness dependent on experience, credibility, and a precision level of an area, where experience and believability are needy upon past direct cooperation and area confirmation utilizing separation and time, separately. At last, in light of the seriousness level of trust esteem, the basic leadership module settles on occasion message whether it is adequate or not. Since the fuzzy logic is the primary methodology received in this work, a short portrayal of the technique and hidden explanations behind embracing this methodology are exhibited in the accompanying segment. Every module will be clarified and talked about in detail in this manner.

Why fuzzy logic? In contrast to old-style speculations, in the fuzzy hypothesis, every component can have a degree of enrollment. The fuzzy set hypothesis is additionally ready to reflect dubious and insufficient data by a characterized set participation as potential dissemination. Besides, it depends on the idea of guessing instead of exact conclusions. The fuzzy logic is progressively being embraced in a few applications in numerous ventures because of its capacities to manage estimation thinking. Plus, it is easy to get a handle on reasonably, tolerant of information imprecision, and adaptable, which is propelled by a characteristic language. Incorrectness, deficiency, and imprecision of the system data sent by every hub demonstrate that we can utilize the fuzzy logic hypothesis in-vehicle condition since it is a promising man-made brainpower innovation with solid execution in the basic leadership frameworks. Since an enormous number of terms are utilized for portraying, the radio sign is fuzzy [26] and as a result of the inalienable quality of fuzzy logic to handle vulnerability and imprecision, the fuzzy logic is received in this work.

4.1. Verification Module. In the proposed model, we suppose a module to check the sender's vital need for any security structure. Certain data associated with the transmitting center point are extremely basic in IoT. Such data can be ID information of the senders despite their features and regions. It is also essential to confirm all events, in which consumers are conversing or data is being swapped all through the framework. The level of endorsement of vehicles is checked by confirmation, which shields the IoT from Sybil attacks by giving a particular character to each vehicle. As a particular model, a vehicle may ensure that it is a lot of vehicles, which makes a dream that there is a blocked road. Obstruct avoiding can manage this fake information and foresees the mind flight. Outside procedures can be used by control checks to give certified and trustworthy confirmation to distinguish attacks. Such exterior systems can be ordinary law usage pros. In [30], it is demonstrated that approval ensures that the sender of a message is precisely recognized. The authors therein introduced ID confirmation, property approval, and zone affirmation to check the ID of the sender, properties of the sender, and the stated circumstance by the sender, independently. In the proposed arrangement, we use ID affirmation to survey the sender of the event message whether it is endorsed or not. ID affirmation empowers a vehicle to perceive the transmitter of a message in particular. This affirmation in like manner empowers a vehicle to be a bit of the framework. At the point when the ID check is executed keeping up a key good way from express attacks, for instance, emulate and fake centers, will be an essential task. Thus, the modernized support proposed by the IEEE 1609.2 standard [32] is grasped in this work. In this standard, the security organization relies upon elliptic twist cryptography (ECC), open key affirmations, and the all-inclusive community key establishment (PKI).

4.2. Lifetime Checking. As a result of the high conveying ability of vehicles and subsequently high exceptional lead, the lifetime of the message is a critical issue in IoT. Towards the day's end, fresh messages are stronger than old/ended messages in the vehicular condition. Note that the lifetime is the time intermission between the event time and end time of the event message. To oversee old/ended messages as abundance messages, the proposed structure first checks the lifetime of the event message. Thus, the structure determines the differentiation between the event (Time E), which is joined into the message, and the here and now (current). Moreover, dependent upon the sort of event message and the current situation with the vehicular condition, the edge time for the event message (Time edge) will be evaluated. For example, it should be set at a colossal motivator under small traffic circumstances or minimal under thick traffic conditions. In case the event message is exorbitantly old/slipped by, it will remain discarded. Else, it will be sent to the resulting stage to additionally be checked.

Definition 1 ((region k-anonymity) [44]). Accept that there occurs a flexible customer whose territory headings are $(K, K-1)$. If the customer and on any occasion other $k-N^{(a-1)}$ customer cannot be isolated by region's information after the theory for this customer, we can say that the k customers' zones satisfy region $k-1$ lack of clarity. The k customers' information shapes a customer mystery set. Note that the least rectangular region which joins all the zone k-anonymity customers is known as the region $k-1$ lack of definition region. It is not elusive in Figure 2 that the rectangular box is a baffling area with an obscure customer set where $N=12$. Formally, we use N to address a territory and $N-1$ the lack of clarity area. As such, as demonstrated by explicit benchmarks, $k-1$ can be divided into discrete rectangular lack of definition regions, addressed. It should be clarified that they are title disperse anonymity regions.

All the subhaziness areas of reliable mystery region in Figures 1 and 2 represent the status when the obscure territories are repartitioned, autonomously.

Definition 2 (central location of the anonymous region). The location of the two diagonals of a rectangular subanonymity region is said to be its central location, which is represented by coordinates. We will take the central location as a fake location to issue location service requests by replacing the subanonymity regions.

4.3. Sequence Estimation of Mobile Users. In an uplink transmission system, a subframe holds two training sequences, the 3rd and the 10th training sequences, used for frequency offset estimation. The signal received by the k subcarrier of mth training sequence of ith subframe is shown in the following formula:

$$\Gamma_i^{(u)} = \sum_{k-0}^{N_a-1} R_i(u) + (3,k)R_i^{(u)}(10,k). \tag{1}$$

The frequency offset estimation calculated by formula [7] is shown in the following formula:

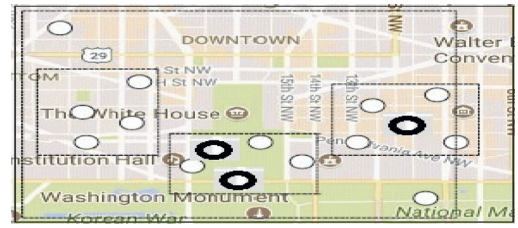


Figure 1: After the partition of the k-anonymity region (sensitive region).

Figure 2: Before partition of k-anonymity region.

$$\varepsilon_i^{(u)} = \frac{\arg\left\{\Gamma_i^{(u)}\right\}}{14\pi}. \tag{2}$$

The estimating range discussed previously is only ±0.07, which is a very small scale, while the algorithm described in literature [7] can only improve the accuracy of frequency offset estimation. The frequency deviation considered in the high-speed situation is roughly 0.107, which means that these methods that were discussed previously cannot reach the estimation range condition. Improved estimating method of frequency deviation is employed to enhance the estimation range, which is shown in the two following formulas:

$$f_i^{\wedge} = \arg\max\left\{M_{i,M}(\lambda)(f)\right\}, \tag{3}$$

$$M_{i,m}(\lambda)(f) = \frac{\sum_{k=0}^{N_{u-1}}\left|R_{i,m}^{S}(\lambda)(k)\right\|Z_{S_{i,m}}(\lambda,k)\Big|^2}{\sqrt{\sum_{k=0}^{N_{u-1}}\left|s_{i,m}^{(\lambda)}(k,f)\right|^4}}, \tag{4}$$

where $R_{(m)}^{S}(k)_m$ is received training sequence, and (k, f) is frequency-domain signal of m on subframe i.

$m(i)$ is training sequences hold frequency offset for subframe i, which is also the frequency-domain signal of m on subframe i; and f_i is the estimate of the residual frequency deviation of the preceding frequency offset estimation. Although the algorithm, which performs maximum likelihood calculation on a single training sequence, discussed in the literature [23] might obtain batter estimating performance and the range is larger in the high-speed motion scenario, its estimating accuracy is low. Meanwhile, this literature also draws on the idea of joint estimation to improve the estimating accuracy, which imposes the training sequence calculation on training sequence on one path and imposes phase, different computing on two training sequences on the other path and then carries out correlation calculation on the two paths. That is the idea of a united algorithm. According

to literature [27], the normalized frequency offset estimate is calculated out via frequency offset estimating, which is shown in formulas (5) and (6).

$$\varepsilon_i^{(u)} = \varepsilon_i^{\wedge(u)} + 15v_i^{2\wedge(u)}, \tag{5}$$

where $\epsilon_i^{(u)}$ is calculated out by phase, different computing of two training sequences.

$$v_i^{\wedge(u)} = \arg\min\left(\varepsilon_i^{\wedge(u)} + \frac{2v_i^{(u)}}{15}\right)\varepsilon_i^{\approx(u)^2}, \tag{6}$$

where $V_i^{(u)}$ is the estimation of the frequency offset achieved by the maximum likelihood calculation of single training sequence symbol and $v \in Z$, where Z is the set of all integers, the series of the maximum likelihood estimation algorithm [24, 25, 28, 33] of the method discussed in the literature is $-0.5 \le \varepsilon_i^{(u)} \le 0.5$, and the range of phase difference is $-0.07 \le \varepsilon_i^{(u)} \le 0.07$. Though the frequency deviation estimation range of this method is relatively large and the estimating accuracy is high, its high computational complexity is not easily employed in an actual communication system. Table 1 depicts fuzzy rules in the algorithm.

4.4. Time Estimation of Mobile Users. Using the technique of the cyclic prefix for frequency estimation, in the time-division mode of LTE uplink multiuser transmission system, all users' cyclic prefix and the corresponding data of its other part must be obtained. Obtaining a cyclic prefix is a method of extracting and reconstructing. And then dispose of the cyclic prefix received by receiving terminal through formula (7) to obtain phase shift owing to a frequency deviation:

$$\Gamma_I^{(u)} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1\mathrm{cp}} r_i\left(nT_S + T_0(m)\right)S_i(u)(m,n). \tag{7}$$

Therefore, the final frequency deviation estimate is shown in the following formula:

$$\varepsilon_i^{\Lambda} = \left(\frac{<\Gamma_i^{(u)} + v_{\mathrm{cp}}}{2\pi}\right) \times 1500, \tag{8}$$

where v_{cp} is an integer. The frequency offset range of cyclic prefix auxiliary is given according to formula (8).

(i) $0.5 \le V_i^{(u)} \le 0.5$, which reaches the requirement of a large frequency deviation estimating range in high-speed railway situation.

However, if there are huge delays in the channel or another influencing factor, the cyclic prefix may be somewhat mixed, which reduces the accuracy of the frequency offset estimation. Consequently, there is a new frequency offset estimating algorithm defined in literature [8], which reduces the impact of multipath in a Rayleigh channel situation. So, L defined as phase estimating length is shown in the following formula:

$$L(b) = \frac{1-b}{p}N_{\mathrm{cp},b} = [1, p-1], \tag{9}$$

Table 1: Fuzzy rules in algorithm.

Number of neighbors	Distance to BS	Remaining energy	Protection
Low	Low	Low	Medium
Low	Medium	Low	Medium
Low	High	Low	Very low
Low	Low	Medium	Medium
Low	Medium	Medium	Low
Low	High	Medium	Very low
Low	Low	High	Medium
Low	Medium	High	High
Low	High	High	High
Medium	Low	Low	High
Medium	Medium	Low	Medium
Medium	High	Low	Very low
Medium	Low	Medium	High
Medium	Medium	Medium	Medium
Medium	High	Medium	Low
Medium	Low	High	High
Medium	Medium	High	Medium
Medium	High	High	Low
High	Low	Low	Medium
High	Medium	Low	High
High	High	Low	Low
High	Low	Medium	Very high
High	Medium	Medium	High
High	High	Medium	Medium
High	Low	High	Low
High	Medium	High	Low
High	High	High	Very low

where p is phase estimating coefficient whose value is 16 when the cyclic prefix is relatively longer, or its value is 8 when the cyclic prefix is comparatively shorter. The range of the frequency offset estimate is $[-L, -1]$. Choose $L(b)$ and $L(b+1)$ to estimate frequency deviation starting from b. Then describe the two estimates computed as eb and eb + 1, where eb is estimating frequency deviation according to $L(b)$ which is defined as a cyclic prefix, which is shown in the following formula:

$$E(b) = \varepsilon^{\Lambda} - \varepsilon_{b+1}^{\Lambda\left(2/\varepsilon_b^{\Lambda^2}\right)}. \tag{10}$$

Plug eb and eb + 1 into formula (10), and if $E(b) \le e^{-4}$, it is considered as meeting the laboratory criteria in the phase estimating length. Then value L as $L(i)$, and let ε_b. Then, b will autoincrease 1, which may be calculated repeatedly. e^{-4} is the coefficient, which may be modified owing to the change of actual situation. Table 2 depicts fuzzy rules in MTFLA.

The method defined in literature [33] explains other paths interference to some extent.

4.5. Fuzzification Process. The AND logical operator is used for connecting input linguistic variables. The triangular and trapezoidal membership functions to map crisp (input) values to fuzzy sets is used by the proposed model. The fuzzy numbers H, A, and L represent High, Average, and Low correspondingly. Initially, the membership function of the fuzzy number H is shown as

Table 2: Fuzzy rules in MTFLA.

Distance to BS	Meet several mobiles	Protection
Low	Low	Very low
Low	Medium	Low
Low	High	Medium
Medium	Low	Low
Medium	Medium	Medium
Medium	High	Medium
High	Low	Medium
High	Medium	High
High	High	Very high

$$M_H(x) = \left\{ \begin{array}{ll} 0 & x < a_1 \\ \dfrac{x - a_1}{a_2 - a_1} & a_1 \le x \le a_2 \\ 1 & x > a_1 \end{array} \right\}. \quad (11)$$

Next, the membership function of the fuzzy number A is computed as

$$M_A(x) = \left\{ \begin{array}{ll} 0 & x \le b_1 \\ \dfrac{x - b_1}{b_2 - b_1} & b_1 < x \le b_2 \\ \dfrac{b_3 - x}{b_3 - b_2} & b_2 < x < b_3 \\ 0 & x \ge b_3 \end{array} \right\}. \quad (12)$$

Finally, the membership function of the fuzzy number L is derived by

$$M_L(x) = \left\{ \begin{array}{ll} 0 & x > c_1 \\ \dfrac{c_1 - x}{c_1 - c_2} & c_2 \le x \le c_1 \\ 1 & x < c_2 \end{array} \right\}. \quad (13)$$

Fuzzy-based clustering algorithms, after selecting multiple fuzzy inputs, calculate a convinced probability for every sensor node and formerly pick the nodes in the clusters by comparing these figures. The significant idea is that clustering steps are both fixed and replicated for every round from the start to the end of the lifetime of the network in the methods described above. On the other hand, repeating the clustering cycle for every round increases the number of messages and energy consumption due to the diverse arrangement of mobile nodes in networks of moving nodes, which eventually reduces network life. This paper grants a fuzzy logic-based algorithm for cluster nodes (M).

4.5.1. The MTFLA

(i) Determine the number of neighbors founded on the power of the acknowledged signal.

Table 3: Simulation parameters.

Parameter	Value
Total bandwidth	6 MH
Z location of the base station	(100, 100)
Number of nodes	100
Data packet size	6000 bits
Eelec	50 nJ/bit
εmp	0.0013 pJ/bit/m4
εfs	10 pJ/bit/m2
Initial energy	1 J

(ii) Specify constraints, for example, "residual energy," "distance from the base station," and "quantity of neighbors," for every node and transfer them to the implication engine.

(iii) Find the inference engine output (chance) and compare the probability of every node with those of its neighbors. Select a node by the maximum chance in each neighboring radius as the head of the cluster.

(iv) Transmit data of every node to the head of the cluster and from there to the base station. At the beginning of the 2nd round, the nodes after the preceding round are reselected as per heads of the cluster and there are no elections.

5. Experiments and Evaluations

5.1. Simulation Parameters. In this section, we give a comprehensive description of the simulation parameters and path losses information due to the deployment reason. Our proposed method is to protect the user information. We implemented our proposed method using Matlab software and OPNET Modeler version 10.5 simulation software [34]. Moving through sensitive regions and communication accomplishes the accuracy with decreasing the range as shown in Tables 1 and 2. Table 3 shows the simulation constraints.

6. Results and Discussion

The proposed method has been analyzed and verified by several sensitive regions. We divided our evaluation into the three subparts based on areas of mobile users. The probability of coverage mobile users during moving on the smart environment shows that region 1 has low protection because fewer mobile users meet inside the region, and medium security in the 2nd region meets the number of mobile users. Finally, the number of users communicating with each other gets high protection during the various moving regions shown in Figure 3. In the case of less use of communication in the sensitive region, there is a chance for more attacks because, with fewer users, the attacker can easily guess the user identity. Our proposed method builds the number of users communicating with each other while creating a sensitive region. Figure 4 shows distance varied with the users to increase communication range and high-level protection against the spoofing attack. For the number of users communicating with each other within the sensitive region, if the user has less coverage, this means distance is

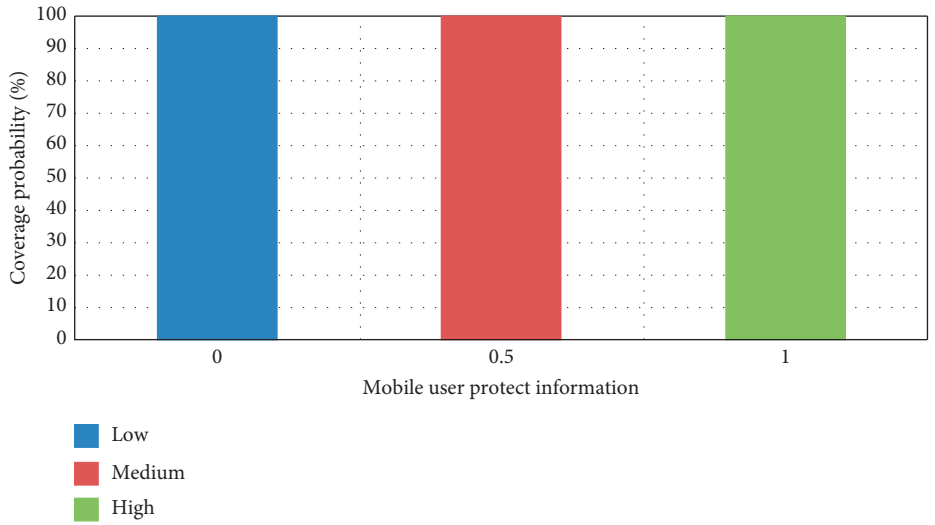


Figure 3: Regions attacks performance.

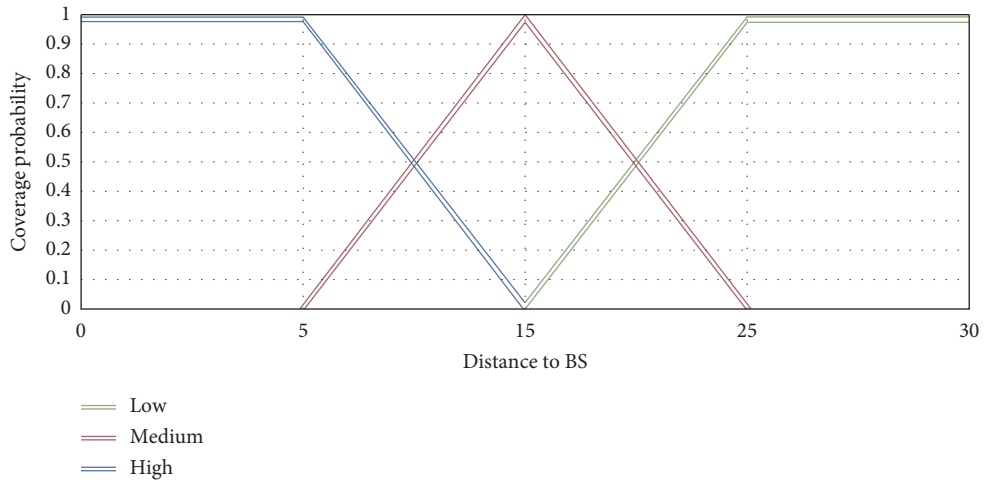


Figure 4: Distance to BS mobile user protection.

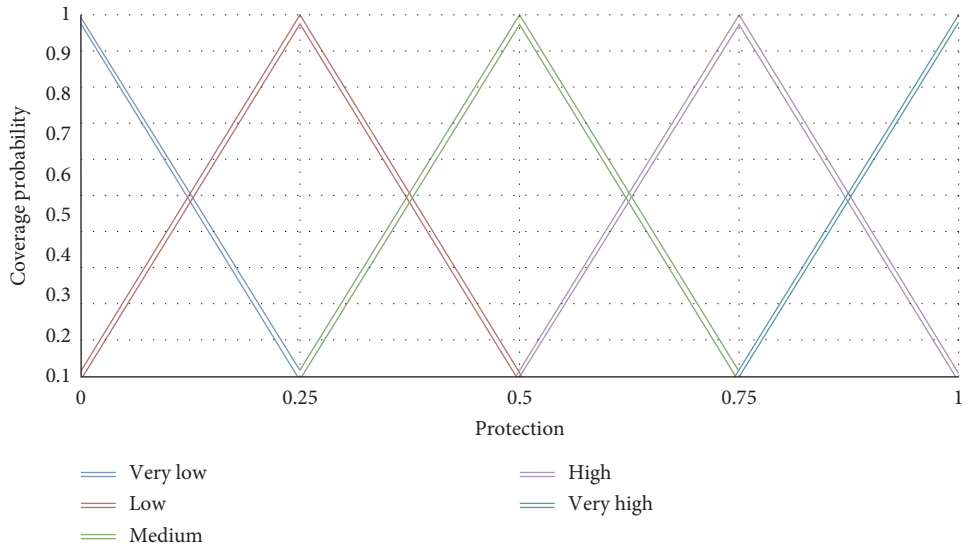


Figure 5: Output protection information of sensitive regions.

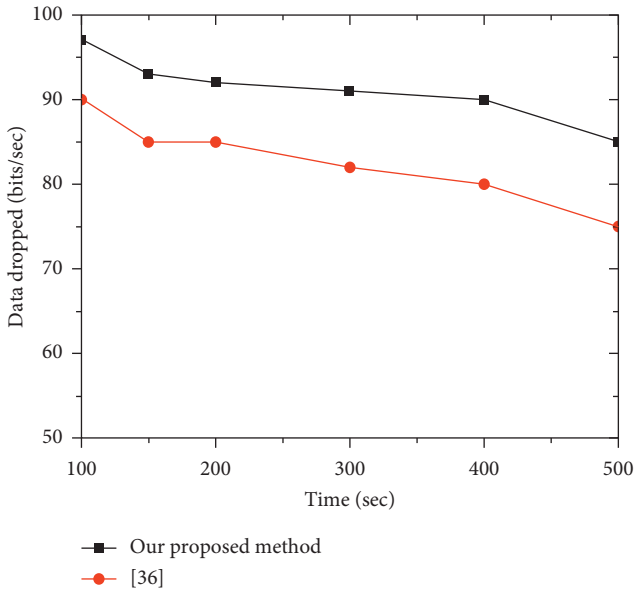


Figure 6: Packet dropped rate versus time (sec).

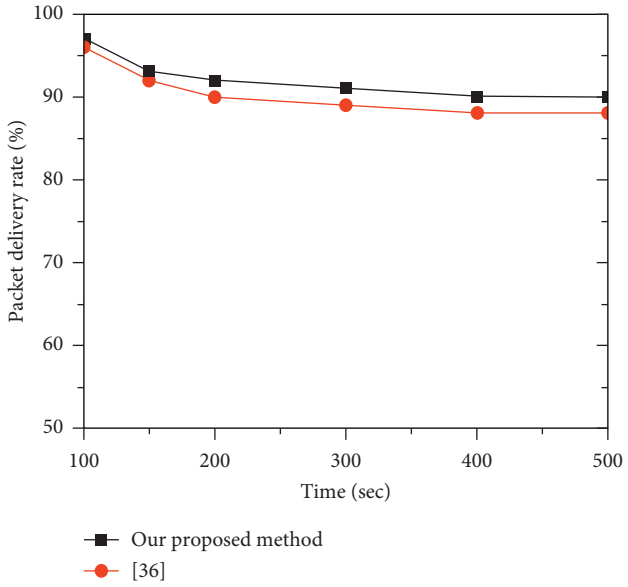


Figure 7: Packet delivery rate versus time (sec).

high to region, increasing chances of a spoofing attack. Figure 5 shows output protection-sensitive region information of the fuzzy laws of our method when the number of users increases coverage probabilities, raising a high level of privacy protection [35].

We compare our method with the existing method [36], and it has been found that the proposed method's packet dropped less than the existing method. This is because more users communicate with each other within the coverage sensitive region as shown in Figure 6. In Figure 7, we also compare our method with the existing method in terms of packet delivery, so our method's packet delivery ratio is more than 90%, but that of the existing method is less than 90. Finally, we compute the

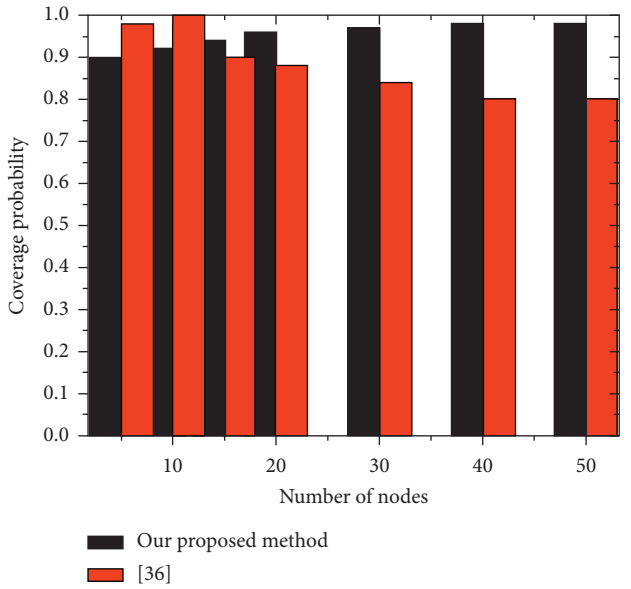


Figure 8: Sensitive region probability rate versus the number of nodes.

sensitive region probabilities in the number of nodes. Our method is more stable and has more protection due to the exposed region by comparison with the existing method as shown in Figure 8.

7. Conclusion

The objective of this paper is to propose a novel scheme for the protection against spoofing which makes use of probability distributions of received power signal based on the regions for the cell user. In addition to this, we inspect the influence on the targeted user's secrecy frequency in the absence and presence of the observer. We have evaluated our techniques via simulation outcomes for sensitive regions' information based on fuzzy logic. Grounded on results, we have suggested mobile travelers' fuzzy logic algorithm (MTFLA) to protect highly sensitive areas, that is, where the chance of attacks will be maximum, and provided comparisons with various security algorithms made for the energy consumption of diverse patterns. Based on simulation outcomes, it is concluded that our proposed algorithm for protection (MTFLA) is verified to be energy-proficient (secure harvesting) as it has decreased energy requirement for encrypting the facts resulting in low computational time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the support of Network Communication Technology (NCT) Research Groups, FTSM, UKM, for providing facilities for this research. This study was supported by the Fundamental Research Grant Scheme GGPM 2020-028, FRGS/1/2018/TK04/UKM/02/7, Dana Impak Perdana UKM DIP-2018-040, and GUP-2019-062.

References

[1] P. Verma and S. Sood, "Fog assisted-IoT enabled patient health monitoring in smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789–1796, 2018.

[2] S. Phoemphon, C. So-In, and D. T. Niyato, "A hybrid model using fuzzy logic and an extreme learning machine with vector paper swarm optimization for wireless sensor network localization," *Applied Soft Computing*, vol. 65, pp. 101–120, 2018.

[3] H. Mshali, T. Lemlouma, and D. Magoni, "Adaptive monitoring system for e-health smart homes," *Pervasive and Mobile Computing*, vol. 43, pp. 1–19, 2017.

[4] M. Z. Ibrahim and R. Hassan, "The implementation of internet of things using test bed in the UKMnet environment," *Asia-Pacific Journal of Information Technology and Multimedia*, vol. 8, no. 2, pp. 1–17, 2019.

[5] S. Sendra, L. Parra, J. Lloret, and J. Tomás, "Smart system for children's chronic illness monitoring," *Information Fusion*, vol. 40, 2017.

[6] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, pp. 69–72, 2005.

[7] F. Bashir, W.-S. Baek, P. Sthapit, D. Pandey, and J.-Y. Pyun, "Coordinator assisted passive discovery for mobile end devices," in *Proceedings of the 2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, vol. 15, no. 4, Las Vegas, NV, USA, January 2013.

[8] K. Zen, D. Habibi, A. Rassau, and I. Ahmad, "Performance evaluation of IEEE 802.15.4 for mobile sensor networks," in *Proceedings of the 2008 5th IFIP International Conference on Wireless and Optical Communications Networks (WOCN '08)*, pp. 1–5, Surabaya, Indonesia, May 2008.

[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[10] N. K. Walia, P. Kalra, and D. Mehrotra, "An IOT by information retrieval approach: smart lights controlled using WiFi," in *Proceedings of the 2016 6th International Conference—Cloud System and Big Data Engineering (Confluence)*, Noida, India, anuary 2016.

[11] R. Akhtar, S. Leng, I. Memon, M. Ali, and L. Zhang, "Architecture of hybrid mobile social networks for efficient content delivery," *Wireless Personal Communications*, vol. 80, 2015.

[12] A. Dvir and A. Vasilakos, "Backpressure-based routing protocol for DTNs," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 405-406, 2010.

[13] A. M. A. Abdo, X. Zhao, R. Zhang et al., "MU-MIMO downlink capacity analysis and optimum code weight vector design for 5G big data massive antenna millimeter wave communication," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7138232, 12 pages, 2018.

[14] A. S. Ahmed, R. Hassan, and N. E. Othman, "Improving security for IPv6 neighbor discovery," in *Proceedings of the 2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 271–274, Denpasar, Indonesia, August 2015.

[15] I. Memon, L. Chen, Q. Ali, H. Memon, and G. Chen, "Pseudonym changing strategy with multiple mix zones for trajectory privacy protection in road networks," *International Journal of Communication Systems*, vol. 31, no. 1, 2017.

[16] A. Mihovska and M. Sarkar, "Smart connectivity for internet of things (IoT) applications," in *Studies in Computational Intelligence*, pp. 105–118, Springer, Berlin, Germany, 2018.

[17] M. Zavvar, M. Rezaei, S. Garavand, and F. Ramezani, "Fuzzy logic-based algorithm resource scheduling for improving the reliability of cloud computing," *Asia-Pacific Journal of Information Technology and Multimedia*, vol. 8, no. 2, pp. 1–17, 2019.

[18] M. S. Mahdi, M. F. Ibrahim, S. Mahmood, P. Singam, and A. B. Huddin, "Fuzzy logic system for diagnosing coronary heart disease," *International Journal of Engineering & Technology*, vol. 8, no. 1, pp. 119–125, 2019.

[19] Subiyanto, A. Mohamed, and M. A. Hannan, "Hardware implementation of fuzzy logic based maximum power point tracking controller for PV systems," in *Proceedings of the 4th International Power Engineering and Optimization Conference (PEOCO2010)*, Shah Alam, Malaysia, June 2010.

[20] S. S. Amiripalli, A. K. Kumar, and B. Tulasi, "Introduction to TRIMET along with its properties and scope," *AIP Conference Proceedings*, vol. 1705, no. 1, 2016.

[21] C. Han, J. M. Jornet, E. Fadel, and I. F. Akyildiz, "A cross-layer communication module for the internet of things," *Computer Networks*, vol. 57, no. 3, pp. 622–633, 2013.

[22] D. Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future internet of things," *Architecting the Internet of Things*, pp. 1–24, Springer, Berlin, Germany, 2011.

[23] S. Lee, J. Lim, J. Park, and K. Kim, "Next place prediction based on spatiotemporal pattern mining of mobile device logs," *Sensors*, vol. 16, no. 2, 2016.

[24] S. Islam, O. O. Khalifa, A. H. A. Hashim, M. K. Hasan, M. A. Razzaque, and B. Pandey, "Design and evaluation of a multihoming-based mobility management scheme to support inter technology handoff in PNEMO," *Wireless Personal Communications*, vol. 114, pp. 1133–1153, 2020.

[25] M. K. Hasan, M. M. Ahmed, A. H. A. Hashim, A. Razzaque, S. Islam, and B. Pandey, "A novel artificial intelligence based timing synchronization scheme for smart grid applications," *Wireless Personal Communications*, vol. 114, pp. 1067–1084, 2020.

[26] D. Zhang, Y. Zhu, C. Zhao, and W. Dai, "A new constructing approach for a weighted topology of wireless sensor networks based on local-world theory for the Internet of Things (IOT)," *Computers & Mathematics with Applications*, vol. 64, no. 5, pp. 1044–1055, 2012.

[27] S. Lmai, A. Bourre, C. Laot, and S. Houcke, "An efficient blind estimation of carrier frequency offset in OFDM systems," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, 2014.

[28] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[29] L. Xao, L. Anfeng, L. Zhetao et al., "Distributed cooperative communication nodes control and optimization reliability for resource-constrained WSNs," *Neurocomputing*, vol. 270, 2017.

[30] M. Iffert, M. Kuenkel, M. Skyllas-Kazacos, and B. Welch, "Reduction of HF emissions from the TRIMET aluminum smelter (optimizing dry scrubber operations and its impact on process operations)," in *Essential Readings in Light Metals*, pp. 968–974, Springer, Berlin, Germany, 2016.

[31] V. Rohokale, N. Prasad, and R. Prasad, "A cooperative internet of things (IoT) for rural healthcare monitoring and control," in *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*, Chennai, India, February 2011.

[32] Q. A. Arain, D. Zhongliang, I. Memon et al., "Privacy preserving dynamic pseudonym-based multiple mix-zones authentication protocol over road networks," *Wireless Personal Communications*, vol. 95, pp. 505–521, 2016.

[33] L. E. Herrera, F. Calliari, D. V. Caballero, G. C. Amaral, P. J. Urban, and J. P. von der Weid, "Transmitter-embedded AMCC, LTE-A and OTDR signal for direct modulation analog radio over fiber systems," in *Proceedings of the 2018 Optical Fiber Communications Conference and Exposition (OFC)*, San Diego, CA, USA, March 2018.

[34] Documentation O. M. (2003). Opnet technologies. Inc.[Internet] http://www.opnet.com.

[35] I. Memon, H. Fazal, R. A. Shaikh, G. A. Mallah, R. H. Arain, and G. Muhammad, "Smart intelligent system for mobile travelers based on fuzzy logic in IoT communication technology," in *International Conference on Intelligent Technologies and Applications*, pp. 22–31, Springer, Bahawalpur, Pakistan, November 2019.

[36] S. Ghasemnezhad and A. Ghaffari, "Fuzzy logic based reliable and real-time routing protocol for mobile ad hoc networks," *Wireless Personal Communications*, vol. 98, no. 1, pp. 593–611, 2018.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8869688, 10 pages
https://doi.org/10.1155/2020/8869688

Research Article

An Improved Method to Evaluate the Synchronization in Neural Key Exchange Protocol

Yi Liang Han, Yu Li, Zhe Li, and Shuai Shuai Zhu

School of Cryptology Engineering, Engineering University of PAP, Xi'an 710086, China

Correspondence should be addressed to Yi Liang Han; yilianghan@hotmail.com

Received 3 September 2020; Revised 10 October 2020; Accepted 17 October 2020; Published 29 October 2020

Academic Editor: Amir Anees

The synchronization between two neural networks by mutual learning can be used to design the neural key exchange protocol. The critical issue is how to evaluate the synchronization without a weight vector. All existing methods have a delay in evaluating the synchronization, which affects the security of the neural key exchange. To evaluate the full synchronization of neural networks more timely and accurately, an improved method for evaluating the synchronization is proposed. First, the frequency that the two networks have the same output in previous steps is used for assessing the degree of them roughly. Second, the hash function is utilized to judge whether the two networks have achieved full synchronization precisely when the degree exceeds a given threshold. The improved method can find the full synchronization between two networks with no information other than the hash value of the weight vector. Compared with other methods, the full synchronization can be detected earlier by two communication partners which adopt the method proposed in this paper. As a result, the successful probability of geometric is reduced. Therefore, the proposed method can enhance the security of the neural exchange protocol.

1. Introduction

It is vital to ensure the information and communication security in the network society. There are a lot of techniques that can achieve the purpose. The most commonly used are symmetric cryptography and asymmetric cryptography. The efficiency of asymmetric cryptography is lower than the efficiency of symmetric cryptography. Hence, the asymmetric cryptography algorithm is used to construct a key exchange protocol, and the obtained key is used to encrypt and decrypt with symmetric cryptography algorithm. However, the algorithms of key exchange are mainly based on number theory, which requires massive computation and memory. Therefore, there is a novel approach to obtain a symmetric key by neural networks, which is called the neural key exchange. Two neural networks can achieve synchronization through mutual learning. They get the common input vector and send the output of their own to each other. Then, after synchronization, the equivalent weights of the two networks can be used as the symmetric key, which can be used to encrypt and decrypt in the later session.

One of the critical issues in neural key exchange is how to evaluate the degree of the synchronization of the two neural networks while with no weights information of the other party. The methods to calculate the cosine of the weights vector and the Euclidean distance of the weights vector cannot be used in practice because there is no information about the weights vector of the other party. Deolecki et al. [1] proposed a method to evaluate the degree of synchronization by calculating the frequency of the equivalent output in fixed previous learning steps of the synchronization process. However, there is a delay to find full synchronization between the two networks in their method, which means that the two networks still continue learning while they have achieved full synchronization. The active attacker will reveal extra information from the additional learning steps. Liu [2] proposed a method to evaluate the synchronization of neural networks by using hash function. Their method can find the full synchronization precisely. However, their method will increase the communication traffic and decrease efficiency. Exceptionally, if the hash function is used untimely, there will be substantial communication traffic. It is necessary to

find the full synchronization of the two networks early and finish the learning. The earlier the full synchronization is found, the less information the attacker can obtain.

The motivation of this work is to enhance the security and improve the efficiency of neural key exchange protocol by improving the method for evaluating the synchronization between two neural networks. In the existing synchronization evaluation methods, such as the hash function method and the frequency method, there is a large delay in the judgment of the full synchronization. Due to the large delay, the probability of the neural key exchange protocol being broken by the geometric attacker increases. The efficiency of neural key exchange protocol is also reduced by the large delay. What is more, the frequency method also has the possibility of misjudgment. When misjudgment occurs, both communication partners have to perform the neural key exchange protocol again. Consequently, the synchronization evaluation method plays an important role in neural key exchange. In this paper, an improved method to reduce the delay and misjudgment rate is proposed. The advantages of the hash function and the method based on the output of neural networks are combined. Therefore, the contributions of this paper are as follows:

(1) It proposes an improved method to evaluate the synchronization of the neural networks. The improved method can find full synchronization precisely and timely. The improved method can reduce the delay.

(2) It proposes an algorithm to search for the optimal parameter of the improved evaluation method. The binary search is also used. Thanks to the optimal parameters, the misjudgment rate of the improved method is reduced. The experimental results show that the delayed steps of the proposed method are lower than other methods, and the misjudgment rate is almost zero. The security of neural key exchange which adopts the improved method can be enhanced.

The remainder of this paper is organized as follows. The related work is given in Section 2. The neural key exchange protocol is presented in Section 3. The method proposed in this paper is presented in Section 4. The results and discussion of the proposed method are presented in Section 5. The conclusion is given in Section 6.

2. Related Work

Neural networks have been applied to cryptography widely [3–7]. A lot of work has been carried out for it. In this section, the development and current state of the neural key exchange are reviewed as follows.

Key exchange protocol based on neural networks was first proposed in [8] and implemented with simple parameters. The authors analyzed the security of the protocol with naive attackers. To accelerate the synchronization of neural networks, the authors also presented the bit-packages techniques and generalized the protocol. Klimov et al. [9] analyzed the protocol proposed in [8] and explained why the two communication partners could achieve full synchronization of weights. Their results show that the naive attacker cannot achieve full synchronization with the two communication partners even though the same structure of networks, input, and learning rules is used. Then, they proposed three types of attacks, which were the genetic attack, the geometric attack, and the probabilistic attack. The three types of attacks can break the protocol with simple parameters in [8]. However, the security of the neural key exchange protocol depends on the structure parameters of the neural networks. Hence, increasing the parameters can resist the genetic attack, geometric attack, and probabilistic attack. Shacham et al. [10] proposed a cooperating strategy, which can cooperate with the genetic attack, geometric attack, and probabilistic attack, so that the success probability of attackers is not affected by the depth of the synapse.

Ruter et al. [11] made a comparative analysis among genetic attack, geometric attack, and majority attack. The simulation results show that the neural key exchange protocol is a security with increasing the depth of the synapse. Besides, they proposed a method that queries were used instead of the random common input, which can improve the security of neural key exchange protocol. Santhanalakshmi et al. [12] applied the genetic algorithm in the synchronization of neural networks to search optimal weights as the initial weights. This method can accelerate synchronization by reducing learning time and steps. Later, the authors analyzed the performance of the protocol proposed in [12]. They analyzed the parameters of the genetic algorithm and tree parity machine [13]. Their results show that security can be improved by increasing the number of the hidden layer of the tree parity machine. Allam et al. [14] proposed an authenticated key exchange protocol by using a preshared key as the boundary of learning, which is called the neural cryptography secret boundaries (NSCB) protocol. The NCSB protocol adopted the dynamic learning rate and random walk learning rules. The advantage of NCSB is that it can improve the security of neural key exchange without reducing efficiency. Chourasia et al. [15] proposed a vectorized neural key exchange protocol. Pal et al. [16] proposed a new learning rule, which can accelerate the synchronization and increase the randomness of the key. Dong and Huang [17] generalized the tree parity machine by using the complex value. Édgar et al. [18] proposed a method to search the optimal structure of the tree parity machine, which makes the neural key exchange protocol more efficient and secure.

Ruttor et al. [19] adopted the cosine similarity to evaluate the degree of the synchronization between two neural networks. It can judge full synchronization precisely by using the weights of Alice and Bob. However, Alice or Bob has no information about the weights of the other party in practice. The Euclidean distance method which needs weight cannot be used in practice too. Doleci et al. [1] show that the frequency with which both communication partners have the same output can be used to evaluate the synchronization. The simulation results show that the relationships among the frequency method, cosine similarity, and Euclidean distant are very high. Gupta and Deshmukh [20] applied the

protocol proposed in [8] to encrypt and decrypt the image secret sharing. Sarkar [21] applied the session key generated by the synchronization of multilayer perceptron to wireless communications. Shishniashvili et al. [22] proposed a technique to use parts of the weight vector instead of the entire weight vector as the session key. The synchronization between two neural networks can also be used as error reconciliation in quantum key distribution protocols [23]. Niemiec [24] proposed a novel method that the tree parity machine is used to correct errors taken place during transmission in quantum key distribution protocol. The influence of parameter variation on security and the influence of different learning rules on efficiency are analyzed in [25].

These schemes introduced above mainly adopted the following four methods for evaluating the synchronization: the cosine similarity, the Euclidean distance, the hash function, and the frequency that the outputs of both partners are equal. The strengths and weaknesses of the four methods are listed in Table 1.

We analyzed the methods from four aspects: delay, extra traffic, misjudgment, and practicality. The indicator delay means that the method cannot find the full synchronization in the first place. Only after many steps of true synchronization have taken place, the synchronization is determined, and the neural key exchange is completed. The indicator extra traffic means that the communication traffic is increased by the delay. The indicator misjudgment means that the neural key exchange is completed, while the true synchronization is not achieved, which fail to exchange the key. The indicator practicality means that whether the method can be used in practice. In the cosine similarity method and Euclidean distance method, there is no delay, no misjudgment, and no extra traffic. Although they have so many advantages, their fatal disadvantage is that they are not practical. Because the weight vector of the other network is not available. The hash function method and the frequency method are practical. But there is also delay and extra traffic when the two practical methods are used. The key issue of the hash function method is when to use the hash function. If the hash function is used too early, it will lead to inefficiency. If the hash function is used too late, it will reduce the security of neural key exchange. There is also misjudgment when using the frequency method.

In conclusion, the evaluation of the synchronization of neural networks is a serious matter affecting the security of the neural key exchange. In this paper, we combined the advantage of the output of neural networks and the hash function. We presented a synchronization evaluation method which can improve the security of the neural key exchange.

3. Neural Key Exchange Protocol

There are two communication partners called Alice and Bob in the neural key exchange protocol. Both of them own a tree parity machine, which is a special feedback neural network [26]. The two tree parity machines can achieve full synchronization by mutual learning. And then, the identical weights of the tree parity machines can be used as a session key to encrypt/decrypt or as a seed to generate a secret key. The neural key exchange protocol contains five main phases, which are the initialization phase, calculating phase, updating phase, evaluating phase, and completion phase. The flowchart of the neural key exchange protocol is shown in Figure 1, and the detail of each phase is described as follows.

Table 1: The strengths and weaknesses.

Method	Delay	Extra traffic	Misjudgment	Practicality
Cosine similarity	×	×	×	×
Euclidean distance	×	×	×	×
Hash function	√	√	×	√
Frequency	√	√	√	√

3.1. Initialization Phase. Alice and Bob initialize their tree parity machine with the same parameters. Then, they randomly generate the weights vector and initialize their weights, respectively.

3.2. Calculating Phase. Alice and Bob receive the common input vector at each learning step. They calculate the output of the tree parity machine and send the result to each other.

3.3. Updating Phase. After receiving the output of another party, they compare whether the two outputs are equal. If equal, Alice and Bob update their weights according to the special learning rules, such as Hebbian learning rules, anti-Hebbian learning rules, and random walk learning rules. Otherwise, it goes to the calculating phase.

3.4. Evaluation Phase. If the weights are updating successfully, Alice and Bob need to judge whether they have achieved full synchronization. The evaluation algorithm of the synchronization degree between Alice and Bob is used here. If the full synchronization is achieved, the protocol goes to the next phase. Otherwise, it goes to the calculating phase.

3.5. Completion Phase. Alice and Bob generate their key according to the weights. They finish the process of neural key exchange.

4. Proposed Method

In this section, we propose an improved method to evaluate the synchronization between two neural networks and an algorithm of searching for the optimal parameter of the method. Both the improved method and the algorithm are described in phases. The algorithm for each phase is also presented.

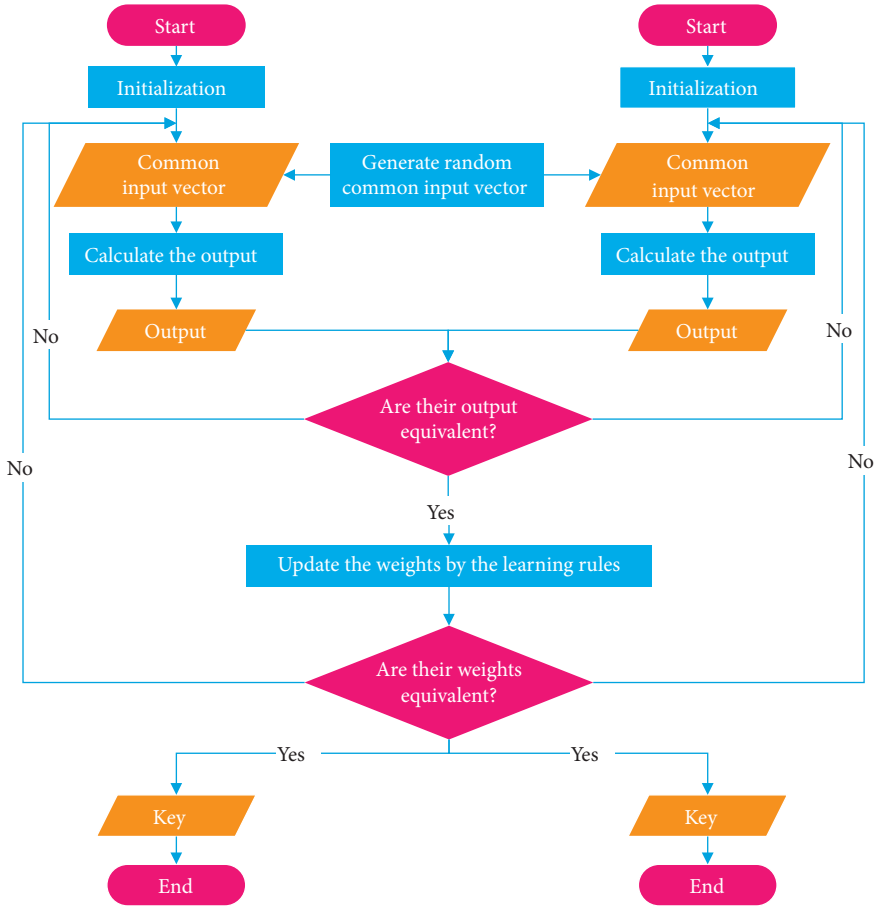


Figure 1: Neural key exchange protocol.

4.1. The Improved Method to Evaluate the Synchronization. There are four methods that can evaluate the synchronization, which are the cosines of the weight vector, the Euclidean distance of the weight vector, the frequency of the equivalent outputs, and the hash value of the weight vector. The shortcomings of the methods above are discussed in Section 1. And it is analyzed through simulation in Section 5. Therefore, it is important to evaluate the synchronization of the neural networks more accurately and timely. An improved method which can find the full synchronization in the first place is presented here. The parameters used in this paper are listed in Table 2.

The main idea of the improved method is judging the degree of synchronization roughly by calculating the frequency that both Alice and Bob have equal output. The hash values of the weights are calculated based on the degree. And then, whether the weights are equal is judged precisely by comparing the hash values. Hence, the process of evaluating the synchronization between Alice and Bob can be divided into two main phases, which are the rough evaluation phase and precise evaluation phase. The flowchart of the evaluating algorithm is given in Figure 2. The detail of each phase is described as follows.

4.1.1. Rough Evaluation Phase. Both Alice and Bob have no information about the weight vector of each other. The only information is the output delivered on a public channel. However, the frequency that Alice and Bob have the same output can be used to evaluate the degree of synchronization. The calculating algorithm is shown in Algorithm 1.

Table 2: Parameter description.

Parameter	Description
K	The number of hidden layer neurons
N	The number of input neurons of the perceptron
L	The depth of the synapse
s	Previous learning steps
W	The weight vector of neural networks
m	The number of learning steps in synchronization
τ_m^A	The result of Alice in m^{th} steps
τ_m^B	The result of Bob in m^{th} steps
a_m	The result of whether $\tau_m^A = \tau_m^B$
b_m	The average $a_m, a_{m-1}, \ldots, a_{m-s+1}$
t	The threshold of the degree of synchronization
sync	The status of synchronization
A	The communication partner Alice
B	The communication partner Bob
H	The hash function
h_A	The hash value of W of Alice
h_B	The hash value of W of Bob
left	The low limit of binary search
right	The high limit of binary search

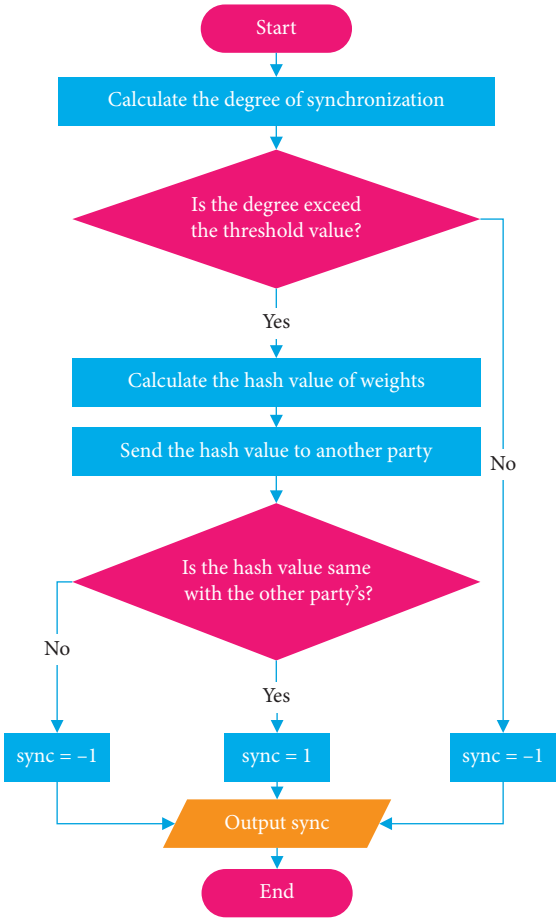


Figure 2: : The evaluation of the synchronization.

The Algorithm 1 accepts s, m, and a_m as the input and output the frequency b_m. The variable s is a natural number that should be set a proper value. The variable m is the learning steps. If Alice and Bob have the same output in m^{th} steps, then $a_m = 1$. If Alice and Bob have different outputs in m^{th} steps, then $a_m = 0$. If the learning steps m is less than s, the frequency is calculated by equation $b_m = (1/s)\sum_{j=1}^{m} a_m$. If the learning steps m is larger than s, the frequency is calculated by equation $b_m = (1/s)\sum_{j=m-s+1}^{m} a_m$.

Then, we compare b_m with the threshold value t. If $b_m < t$, the process of evaluation will be terminated and return the status sync with -1. sync $= -1$ means that Alice and Bob have not achieved full synchronization and will continue to learn with each other. sync $= 1$ means that Alice and Bob have made full synchronization and will stop the process of synchronization. If $b_m > t$, it will go to the next phase.

4.1.2. Precise Evaluation Phase. Only if the degree of synchronization exceeds the threshold value, it starts this phase. Then, Alice and Bob evaluation whether they have achieved full synchronization with the help of the hash function. Here, we take Alice as an example to show how the full synchronization is judged. The other party, Bob, should perform the same operation as Alice. The evaluation algorithm is shown in Algorithm 2.

Input: s, m, a_m
Output: b_m
Step 1: **if** $s < m$, **then**
Step 2: calculate $b_m = (1/s)\sum_{j=m-s+1}^{m} a_m$
Step 3: **else**
Step 4: calculate $b_m = (1/s)\sum_{j=1}^{m} a_m$
Step 5: **end**

Algorithm 1: Rough evaluation.

Input: t, b_m, **W**, h_B
Output: sync
Step 1: **if** $b_m \geq t$, **then**
Step 2: calculate $h_A = H(\mathbf{W})$
Step 3: **if** $h_A = h_B$ **then**
Step 4: sync = 1
Step 5: **else**
Step 6: sync = −1
Step 7: **end**
Step 8: **else**
Step 9: sync = −1
Step 10: **end**

Algorithm 2: Precise evaluation.

Input: none
Output: left, right
Step 1: let $s = 25$, left = 25
Step 2: calculating the number of misjudgment
Step 3: **while** the number does not exceed 0
Step 4: left = s
Step 5: $s = s + 50$
Step 6: calculating the number of misjudgments
Step 7: **end**
Step 8: right = s

Algorithm 3: Boundary searching.

The Algorithm 2 accepts t, b_m, W, h_B as input and output the synchronization status sync. The variable t is a threshold value that should be adequately selected and set. The variable b_m is the output of Algorithm 1. The variable W here represents the weight vector of Alice. The variable h_B is the hash value of the weight vector of Bob. At the beginning of Algorithm 2, Alice compares the output of Algorithm 1 and the threshold value. If $b_m \geq t$, Alice calculates the hash value of its own weight vector and compares it with the hash value of the weight vector of Bob. If $h_A = h_B$, the status sync will be returned with 1. Otherwise, the status sync will be returned with -1. If $b_m < t$, the status sync will also be returned with -1.

While the precise evaluation phase is completed, the neural key exchange protocol obtains the status of

Input: left, right
Output: the optimal s
Step 1: **while** right − left < 2
Step 2: s = (left + right)/2
Step 3: calculating the number of misjudgment
Step 4: **if** the number exceeds 0, **then**
Step 5: left = s
Step 6: **else**
Step 7: right = s
Step 8: **end**
Step 9: **end**

Algorithm 4: Binary searching.

synchronization. Then, if sync = 1, which means that Alice and Bob have achieved full synchronization, the neural key exchanged protocol will be finished. If sync = −1, which means that Alice and Bob have not achieved full synchronization, the neural key exchange protocol will continue until the two communication partners have achieved full synchronization.

4.2. The Algorithm for Finding the Optimal Parameter. The optimal s must satisfy two conditions. First, there is no misjudgment. Second, the delay in finding the full synchronization is kept as small as possible. In the later section, we have shown that the number of delayed steps in finding full synchronization increases linearly with the increment of s, and the number of misjudgments decreases with the increment of s. Hence, the minimum s which makes no misjudgment occur can be used as the optimal s. Here, an algorithm for searching the optimal s is proposed. The algorithm contains two phases: boundary searching phase and binary searching phase. The detail of each phase is described as follows.

4.2.1. Boundary Searching Phase. At the beginning, we let s start with 25. And then, if the number of misjudgments for given s exceeds 0, s will be increased by 50 each time until the quantity equals 0. The detail can be seen in Algorithm 3. It accepts no input and output the parameters left and right, which will be used in the next phase.

4.2.2. Binary Searching Phase. Then, the binary searching method is adopted to select the optimal s. The detail can be seen in Algorithm 4. It accepts the parameters left and right as the input and output the optimal s. The purpose of the algorithm is to find the minimum value, which makes no misjudgment occur between left and right quickly.

5. Results and Discussion

In this section, we analyzed the method proposed in [1] and the method proposed in [2] through simulation experiments. And we also analyzed the performance of the method proposed in this paper. The comparative analysis between the proposed method and other methods has also been made. All the algorithms are implemented in python. The code environment used in this paper is shown as follows: the computer is Sugon W560-G30. The operating system is Ubuntu 20.04 LTS with 64 bits. The memory is 62.6 GiB. The CPU is Intel® Xeon(R) Sliver 4110CPU@2.10 GHz with 4 core and 32 threads.

5.1. The Optimal Parameter. The idea of the method proposed by Dolecki et al. [1] is to calculate the frequency that Alice and Bob have the same output in previous s steps. Here, we show that the misjudgment will occur if the parameter s is not chosen properly. This is because there is some probability that the case where all of the outputs are equal between Alice and Bob in previous s steps, though Alice and Bob have not achieved full synchronization. In addition, the less the s is, the larger the number of misjudgment is. For example, as s increases from 25 to 225, we simulated 10000 samples for different combinations of K, N, and L: 3-101-1, 3-101-2, 3-101-3, 3-101-4, 3-101-5, 3-101-6, and 3-101-7 (Figure 3). For each combination of K, N, and L, the number of misjudgment decreases with the increment of s. While for a fixed s, the number of misjudgment increases with the increment of L.

We also show that the method proposed by Dolecki et al. [1] has a delay in finding the full synchronization of two neural networks. Therefore, if the full synchronization is achieved at the current step, while the two communication partners have one or more output that is different in previous s steps, the frequency will be less than 1. Hence, the learning between Alice and Bob will continue until there is no different output between Alice and Bob in previous s steps. For example, for $K = 3$, $N = 101$, and $L = 3$, if $s = 75$, the steps using cosine of weight vectors are 558, while the steps using frequency are 603. There are 45 delayed steps (Figure 4(a)). If $s = 125$, the steps using the cosine of weight vectors are 397, while the steps using frequency are 496. There are 99 steps that delayed (Figure 4(b)).

We use the cosine of the weight vector to find full synchronization at the first place and record the current steps. We also record the steps when their method judges full synchronization. Then, we compare the two steps and calculate the delay. We set five different values for s: 25, 75, 125, 175, and 225. We set several different values for K, N, and L: 3-101-1, 3-101-2, 3-101-3, 3-101-4, 3-101-5, 3-101-6, and 3-101-7. We simulate 10000 samples and calculate the average delayed steps for each case. In Figure 5, when K, N, and s are fixed, the number of average delayed steps decreases with the increment of L. This is because of the misjudging that has been analyzed before. When K, N, and L are fixed, the number of average steps increase linearly with the increment of the parameter s.

Therefore, selecting an optimal s is very important for evaluating the degree of synchronization. We use the algorithm proposed in Section 3 to select the optimal s for each combination of K, N, and L. The results are shown in Table 3.

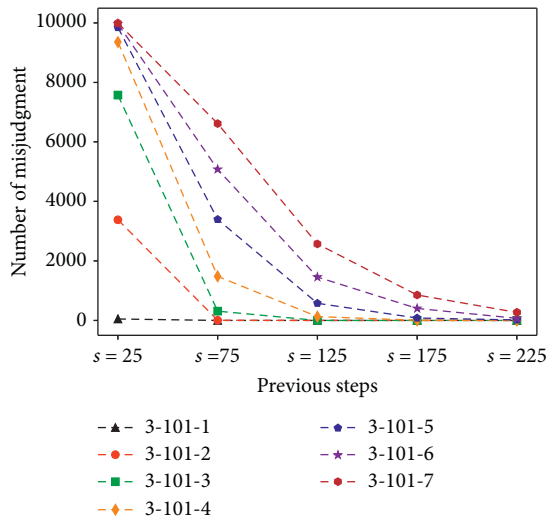


Figure 3: The number of misjudgment.

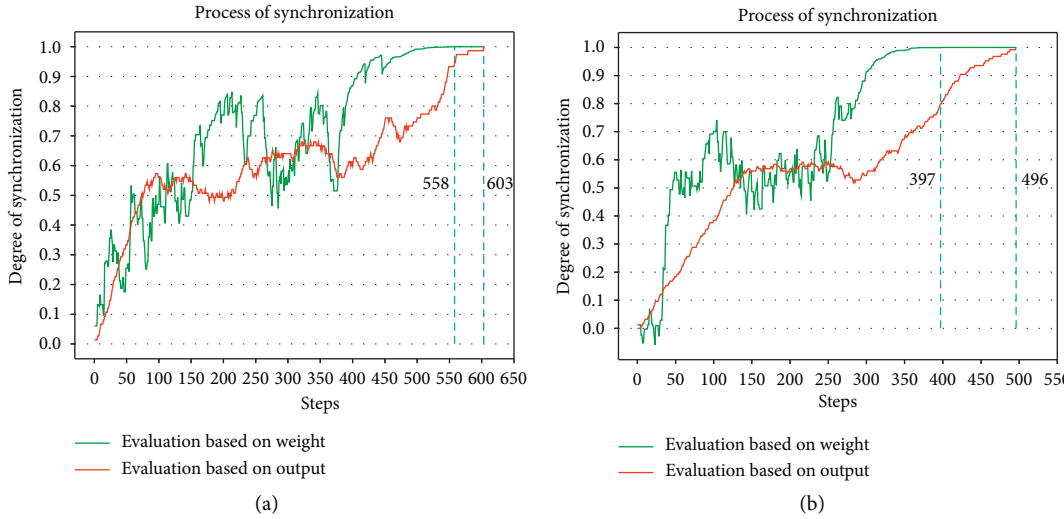


Figure 4: The delay using frequency ($K = 3$, $N = 101$, and $L = 3$). (a) $s = 75$; (b) $s = 125$.

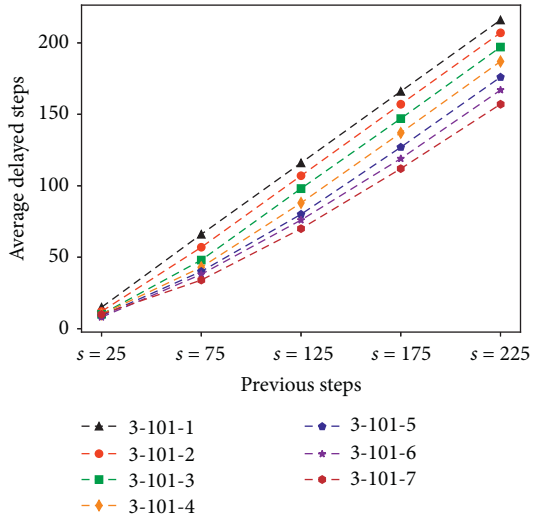


Figure 5: The average delayed steps.

5.2. Security Analysis. The delay in finding the full synchronization of neural networks may cause a security issue of neural key exchange. And the higher the number of delayed steps is, the less secure the neural key exchange is. That is, because the delayed steps can enhance the ability of the attacker. In this section, we made a comparative analysis of delayed steps and security among the method proposed in [1, 2] and this paper.

The evaluating method proposed in [1] is to use the hash function when the learning steps exceed the step threshold, which is the average learning steps of the fixed K, N, and L. The method proposed in this paper is to use the hash function when the rough degree of synchronization exceeds the degree threshold. The degree threshold means the average degree calculated by the rough evaluation phase while full synchronization occurs. In Table 4, the step threshold and the degree threshold were obtained by averaging 10000 simulations.

Table 3: The optimal s.

K	N	L	The optimal s
3	101	1	43
3	101	2	101
3	101	3	158
3	101	4	217
3	101	5	351
3	101	6	454
3	101	7	518

Table 4: The parameters used in the experiment.

K	N	L	Step threshold	Degree threshold
3	101	1	42.78	0.59
3	101	2	141.97	0.72
3	101	3	301.94	0.80
3	101	4	517.77	0.85
3	101	5	791.12	0.85
3	101	6	1106.95	0.89
3	101	7	1482.51	0.90

The delayed steps of each method are compared in Table 5. The parameter K is fixed as 3 and N is fixed as 101. The corresponding thresholds used in the simulation are presented in Table 4. The delayed steps are calculated by averaging 10000 simulations. However, the number of delayed steps in all three methods increases with the increment of L. The number delayed steps of the method proposed in this paper is the smallest among the three methods for each L (Table 5).

Then, we use the geometric attack to test the security of the neural key exchange protocol which adopts the evaluating method proposed in [1, 2] and this paper, respectively. We execute 10000 simulations for each method. The success probability of a geometric attack is shown in Table 6. For each method, the success probability decreases with the increment of L. When L increases to 4, the probability is very low. When L exceeds 4, the geometric attacker cannot break the neural key exchange protocol. For the fixed value of L, the probability of our method is the lowest, while the probability of the method in [1] is the highest. For example, the probability of our method is 25.52% lower than the method in [1] and 4.66% lower than the method in [2].

However, the probability gap between our method and the method in [1] decreases with the increment of L. The same phenomenon also occurs between the method in [2] and our method. This is also consistent with the conclusion in [9] that the success probability of geometric attack is reduced exponentially as L increases. The main reason is that the average delayed steps of each method grow more slowly than the average learning steps required for the two neural networks to achieve full synchronization. When the value of L is increased from 1 to 7 by 1 in each step, the corresponding average learning steps required are 42.87, 141.97, 301.94, 517.77, 791.12, 1106.95, and 1482.51, and the average delayed steps is showed earlier in Table 5. The average learning steps required for the two neural networks to achieve full synchronization go up exponentially as L increases (Figure 6).

Table 5: Comparison of the delayed steps.

Method	The delayed steps						
	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 6$	$L = 7$
Reference [1]	33.67	82.69	129.80	178.72	302.39	393.23	446.70
Reference [2]	12.34	33.41	64.27	99.00	139.42	177.20	225.26
Ours'	5.77	10.51	17.72	25.44	42.40	71.73	79.87

Table 6: Success probability of geometric attack.

Method	Probability						
	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 6$	$L = 7$
Reference [1]	53.05%	5.90%	0.48%	0.01%	0.0%	0.0%	0.0%
Reference [2]	32.19%	4.01%	0.37%	0.01%	0.0%	0.0%	0.0%
Ours'	27.53%	3.41%	0.32%	0.01%	0.0%	0.0%	0.0%

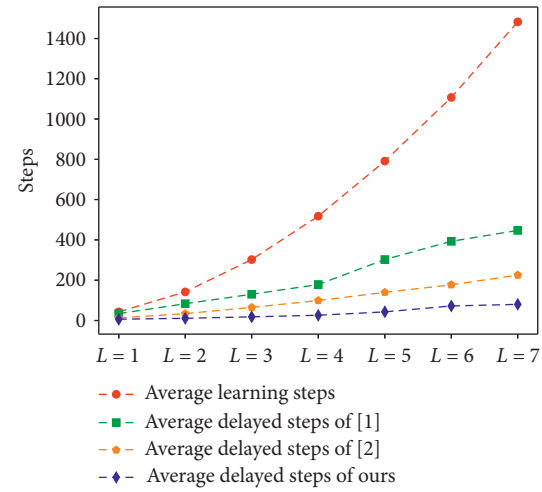


Figure 6: The average steps.

While the average delayed steps of each method only go up linearly as L increases. Since the average delayed steps grow more slowly than the average learning steps, the proportion of average delayed steps in average learning steps is decreasing. For example, when L is 1, the percentage of average delayed steps in the number of averaging learning steps in [1, 2] and methods are 78.54%, 28.78%, and 13.46%, respectively. When L is 5, the percentage of average delayed steps in the number of averaging learning steps in [1, 2] and methods are 38.17%, 17.62%, 5.36%, respectively. The impact of the delayed steps on the security of neural key exchange is decreasing. In Figure 6, our method has the slowest increase in the number of average delayed steps. While the number of average delayed steps in [2] is faster than that in our method and slower than that in [1]. This is why the success

probability of geometric attack of our method is lower than that in [1, 2].

6. Conclusion

The evaluation of synchronization is a significant part of the neural key exchange. The timing of finding full synchronization has a substantial impact on security. In this paper, an improved method to evaluate synchronization between two neural networks has been proposed. The combination of frequency that both the two communication partners have the same output and the hash function is used. The algorithm for finding the optimal critical parameter is also presented. The proposed method has been implemented and tested in the neural key exchange protocol. The experimental results show that the delay in finding full synchronization using the method in this paper is less than the method in [1, 2]. The neural key exchange protocol using the proposed method is more resistant to the geometric attack than them. Therefore, the security of neural key exchange protocol can also be improved by using the proposed method. However, there are still a few delays in finding full synchronization. In the future, two main aspects need to be further improved. One is to reduce the delay further. The other is to stop the long-time synchronization and start a new synchronization, which is a short-time synchronization with high probability.

Data Availability

The simulation data used to support this study are included within this article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61572521) and the Innovative Team Foundation of Engineering University of PAP (KYTD201805).

References

[1] M. Dolecki, R. Kozera, and K. Lenik, "The evaluation of the TPM synchronization on the basis of their outputs," *Journal of Achievements in Materials and Manufacturing Engineering*, vol. 57, no. 2, pp. 91–98, 2013.

[2] Y. L. Liu, "Schemes for neural synchronization based on tree parity machine," Chongqing University, Chongqing, China, 2014.

[3] B. ÖzÇakmak, A. Özbİlen, U. Yavanoğlu, and K. CİN, "Neural and quantum cryptography in big data: a review," in *Proceedings of the 2019 IEEE International Conference on Big Data*, pp. 2413–2417, Los Angeles, CA, USA, December 2019.

[4] M. M. Alani, "Applications of machine learning in cryptography: a survey," in *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, pp. 23–27, Kuala Lumpur, Malaysia, January 2019.

[5] Z. K. Abdalrdha, I. H. Al-Qinani, and F. N. Abbas, "Subject review: key generation in different cryptography algorithm," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 6, no. 5, pp. 230–240, 2019.

[6] D. Protic, "Neural cryptography," *Vojnotehnicki Glasnik*, vol. 64, no. 2, pp. 483–495, 2016.

[7] P. P. Hadke and S. G. Kale, "Use of neural networks in cryptography: a review," in *Proceedings of the 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, pp. 1–4, Coimbatore, India, February 2016.

[8] I. Kanter, W. Kinzel, and E. Kanter, "Secure exchange of information by synchronization of neural networks," *Europhysics Letters (EPL)*, vol. 57, no. 1, pp. 141–147, 2002.

[9] A. Klimov, A. Mityagin, and A. Shamir, "Analysis of neural cryptography," in *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 288–298, Queenstown, New Zealand, December 2002.

[10] L. N. Shacham, E. Klein, R. Mislovaty, I. Kanter, and W. Kinzel, "Cooperating attackers in neural cryptography," *Physical Review E*, vol. 69, no. 6, Article ID 066137, 2004.

[11] A. Ruttor, W. Kinzel, and I. Kanter, "Neural cryptography with queries," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 1, Article ID P01009, 2005.

[12] S. Santhanalakshmi, T. Sudarshan, and G. K. Patra, "Neural synchronization by mutual learning using genetic approach for secure key generation," in *Proceedings of the International Conference on Security in Computer Networks and Distributed Systems*, pp. 422–431, Thiruvanathapuram, India, March 2014.

[13] S. Santhanalakshmi, K. Sangeeta, and G. K. Patra, "Analysis of neural synchronization using genetic approach for secure key generation," in *Proceedings of the International Symposium on Security in Computing and Communication*, pp. 207–216, Kochi, India, August 2015.

[14] A. M. Allam, H. M. Abbas, and M. W. El-Kharashi, "Authenticated key exchange protocol using neural cryptography with secret boundaries," in *Proceedings of the 2013 International Joint Conference on Neural Networks*, pp. 1–8, Dallas, TX, USA, December 2013.

[15] S. Chourasia, H. C. Bharadwaj, Q. Das, K. Agarwal, and K. Lavanya, "Vectorized neural key exchange using tree parity machine," *Compusoft*, vol. 8, no. 5, pp. 3140–3145, 2019.

[16] S. K. Pal, S. Mishra, and S. Mishra, "An TPM based approach for generation of secret key," *International Journal of Computer Network and Information Security*, vol. 11, no. 10, pp. 45–50, 2019.

[17] T. Dong and T. Huang, "Neural cryptography based on complex-valued neural network," *IEEE Transactions on Neural Networks and Learning Systems*, p. 1, 2020.

[18] S. D. Édgar, F. Walter, and L. Edison, "On the development of an optimal structure of tree parity machine for the establishment of a cryptographic key," *Security and Communication Networks*, vol. 2019, no. 3, pp. 1–10, 2019.

[19] A. Ruttor, W. Kinzel, and I. Kanter, "Dynamics of neural cryptography," *Physical Review E*, vol. 75, no. 5, Article ID 056104, 2007.

[20] M. Gupta and M. Deshmukh, "Single secret image sharing scheme using neural cryptography," *Multimedia Tools and Applications*, vol. 79, no. 17-18, pp. 12183–12204, 2020.

[21] A. Sarkar, "Multilayer neural network synchronized secured session key based encryption in wireless communication," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 1, pp. 44–53, 2019.

[22] E. Shishniashvili, L. Mamisashvili, and L. Mirtskhulava, "Enhancing IoT security using multi-layer feedforward neural network with tree parity machine elements," *International Journal of Simulation Systems, Science & Technology*, vol. 21, no. 2, pp. 371–383, 2020.

[23] M. Mehic, M. Niemiec, H. Siljak, and M. Voznak, "Error reconciliation in quantum key distribution protocols," in *Proceedings of the International Conference on Reversible Computation*, pp. 222–236, Oslo, Norway, July 2020.

[24] M. Niemiec, "Error correction in quantum cryptography based on artificial neural networks," *Quantum Information Processing*, vol. 18, no. 6, p. 174, 2019.

[25] M. Niemiec, M. Mehic, and M. Voznak, "Security verification of artificial neural networks used to error correction in quantum cryptography," in *Proceedings of the 26th Telecommunications Forum (TELFOR)*, pp. 1–4, Belgrade, Serbia, November 2018.

[26] A. Ruttor, "Neural synchronization and cryptography," pp. 279–345, University of St Gallen, St. Gallen, Switzerland, 2006, Business dissertations.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8883884, 14 pages
https://doi.org/10.1155/2020/8883884

Research Article

The Effect of the Primitive Irreducible Polynomial on the Quality of Cryptographic Properties of Block Ciphers

Sajjad Shaukat Jamal,[1] **Dawood Shah**,[2] **Abdulaziz Deajim**,[1] **and Tariq Shah**[2]

[1]*Department of Mathematics, College of Science, King Khalid University, P. O. Box 9004, Abha, Saudi Arabia*
[2]*Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan*

Correspondence should be addressed to Sajjad Shaukat Jamal; shussain@kku.edu.sa

Received 23 May 2020; Revised 3 August 2020; Accepted 28 August 2020; Published 24 September 2020

Academic Editor: Tom Chen

Substitution boxes are the only nonlinear component of the symmetric key cryptography and play a key role in the cryptosystem. In block ciphers, the S-boxes create confusion and add valuable strength. The majority of the substitution boxes algorithms focus on bijective Boolean functions and primitive irreducible polynomial that generates the Galois field. For binary field F2, there are exactly 16 primitive irreducible polynomials of degree 8 and it prompts us to construct 16 Galois field extensions of order 256. Conventionally, construction of affine power affine S-box is based on Galois field of order 256, depending on a single degree 8 primitive irreducible polynomial over $\mathbb{Z}_2$. In this manuscript, we study affine power affine S-boxes for all the 16 distinct degree 8 primitive irreducible polynomials over $\mathbb{Z}_2$ to propose 16 different 8×8 substitution boxes. To perform this idea, we introduce 16 affine power affine transformations and, for fixed parameters, we obtained 16 distinct S-boxes. Here, we thoroughly study S-boxes with all possible primitive irreducible polynomials and their algebraic properties. All of these boxes are evaluated with the help of nonlinearity test, strict avalanche criterion, bit independent criterion, and linear and differential approximation probability analyses to measure the algebraic and statistical strength of the proposed substitution boxes. Majority logic criterion results indicate that the proposed substitution boxes are well suited for the techniques of secure communication.

1. Introduction

The exchange of digital data through the Internet has revolutionized the communication parameters over the years. But this rapid communication also provides opportunities to access this digital data illegally. For this reason, the security of this content on the Internet has become a serious challenge for the researchers of different fields. To counter the emerging challenges of security, cryptography and steganography are used to hide the secret information whereas watermarking is used for copyright protection. In this manuscript, we discuss cryptography and relevant aspects of this field. For convenience, cryptography is divided into two types named symmetric key cryptography and asymmetric key cryptography. In symmetric key cryptography, two parties share secret information and keys during encryption and decryption procedures. The private key is shared by both sender and receiver. In addition to this, block ciphers and stream ciphers are two main branches of symmetric key cryptography. In 1949, Shannon gave the idea of block cipher and some examples of block ciphers are Advanced Encryption Standard (AES) [1], Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), and many more [2, 3]. In AES, there is availability of three different key sizes such as 128, 192, and 256 bits, whereas in DES, the only available key size is 56 bits. The AES has 10, 12, and 14 rounds for key sizes of 128, 192, and 256 bits, respectively. All these rounds have four basic steps, that is, subbyte, shift row, mix column, and add round key. Subbyte is the step which substitutes the plaintext data with substitution box (S-box). This S-box is the only nonlinear part of block cipher used in different well-known cryptosystems. It is used to create confusion to make plaintext data obscure for any attacker and hence S-box is an integral part of any cryptosystem. S-box is a function which has input and output from the Galois field. The Galois field is a finite field having order 256 and denoted by $GF(2^8)$.

1.1. Related Work. S-box is used to create confusion as observed in AES, International Data Encryption Algorithm (IDEA), DES, and many more cryptosystems [4]. It is an established fact that the strength of block cipher depends on the standard and quality of S-box. Due to the necessary immersion of S-box to generate nonlinearity, intricacy persuades different researchers to design strong S-boxes to enhance the security level of cryptosystems. Among different available methods, the algebraic structure-based construction of S-boxes has much more attention. These S-boxes have strong cryptographic features and are robust against linear and differential cryptanalysis.

In the literature, different structural advancements are viewed to improve the quality of S-boxes. The algebraic complexity of AES S-box has been improved with the extension of this S-box, that is, affine power affine (APA) [5]. Furthermore, the symmetric group S_8 has also been applied to AES S-box to improve the quality and numbers of S-boxes [6]. Similarly, the application of transformation using binary gray codes on AES S-box gives Gray S-box [7]. In [8], S-boxes are constructed by using the projective general linear group (PGL). Moreover, the construction scheme of chaotic S-boxes using DNA sequence and chaotic Chen system is given in [9, 10]. Different analytical, algebraic, and chaos-based techniques for the construction of S-boxes are given in [11–16]. Conventionally, AES uses a polynomial of 8 terms which have all the required properties and improves the security for AES. But the Gray S-box has a 255-term polynomial. Moreover, residue prime, Xyi, and Skipjack S-boxes are frequently used for the encryption and decryption schemes [17, 18].

It is assumed that the model of Boolean functions and primitive irreducible polynomial has an impact on the strength of S-box. In [19], different primitive irreducible polynomials have been used to identify the effect of primitive irreducible polynomial. To investigate this fact, we want to study all the primitive irreducible polynomials to understand whether there is an impact of irreducible polynomial or not. Archetypally in the synthesis of an S-box, the numbers a, b, c, and d in affine transformation belong to Galois field $\mathrm{GF}(2^8)$. As the polynomial ring $\mathbb{Z}_2[x]$ has 16 primitive irreducible polynomials of degree 8, it shows that only 16 opportunities are available for constructing Galois fields $\mathrm{GF}(2^8)$. In this paper, we have constructed 16 different robust 8×8 S-boxes over the elements of these 16 irreducible polynomials. Firstly, we define 16 affine power affine transformations on these different Galois fields which can be given as $z \longrightarrow (az + b)o(cz + d)^{-1}$; here, for a, b, c, d values, we would be able to get 16 distinct S-boxes.

1.2. Motivation. Due to the role of S-boxes in cryptosystems, it is essential to explore all of its aspects. The motivation behind this work is to study all primitive irreducible polynomials and their role in the construction of S-boxes.

(1) The Mobius transformation used in a different construction of S-boxes has certain limitations and restrictions in its structure [7]. For example, the condition on the parameters, i.e., $a\,d - bc \neq 0$ $\forall a, b, c, d \in \mathrm{GF}(2^8)$ squeezes the remaining cases. Hence, there is a need for any other transformation.

(2) There are 16 primitive irreducible polynomials in the principal ideal domain $\mathbb{Z}_2[x]$ whose impact was not studied yet regarding their impression on analyses of S-boxes.

(3) By exploring all primitive irreducible polynomials, we have a better opportunity to obtain the cryptographically strong cryptosystems.

1.3. Our Contribution. In this manuscript, we studied all binary degree 8 primitive irreducible polynomials for the construction of S-boxes. The quality of the proposed work can be seen from the different security analyses and resistance against malicious attacks. This whole study can be summarized as follows:

(1) We constructed S-boxes associated with the 16 binary degree 8 primitive irreducible polynomials.

(2) The APA transformation is used in this work, which is bijective and has no restrictions on the parameters.

(3) To evaluate the strength of the proposed S-boxes, we have performed different analyses along with differential cryptanalysis. The outcomes of these analyses are compared with the well-known S-boxes.

The remaining part of the paper is planned as follows: Section 2 presents the preliminaries and construction scheme of the proposed S-boxes. In Section 3, algebraic and statistical analyses are calculated in detail. Section 4 presents definitions of the balanced Boolean function. Section 5 concludes the paper.

2. Primitive Irreducible Polynomials of Degree 8 and GF (2^8)

2.1. The Galois Fields $GF(2^8)$. We summarize here some well-known facts from the theory of rings and fields. Let R be a commutative ring with identity. A nonempty subset J of R is called an *ideal* of R if J is an additive subgroup of R and $aJ \subseteq J$ for every $a \in R$, where $aJ = \{aj \mid j \in J\}$. If, furthermore, there does not exist a proper ideal of R properly containing J, then we say that J is a *maximal* ideal of R Besides; R is said to be a *field* if each of its nonzero elements has a must inverse in R. If R is a field of prime characteristic p, then R is an extension of the prime field $\mathbb{Z}_p$. A polynomial $f(x) \in \mathbb{Z}_p[x]$ is said to be irreducible if it cannot be factored in $\mathbb{Z}_p[x]$ into two polynomials of strictly smaller degrees. The principal ideal,

$$\langle f(x) \rangle = \{h(x) f(x) \colon h(x) \in \mathbb{Z}_p[x]\}, \tag{1}$$

generated by a monic irreducible polynomial $f(x) \in \mathbb{Z}_p[x]$ is a maximal ideal in $\mathbb{Z}_p[x]$. If $f(x)$ is of degree m, then the quotient ring,

$$\frac{\mathbb{Z}_p[x]}{\langle f(x)\rangle} = \left\{g(x) + \langle f(x)\rangle \colon g(x) \in \mathbb{Z}_p[x]\right\}, \quad (2)$$

is an extension field of $\mathbb{Z}_p$ of degree m consisting of p^m elements. This field is called a *Galois field* and is denoted by $\mathrm{GF}(p^m)$ and is said to be the field extension of $\mathbb{Z}_p$ defined by the irreducible polynomial $f(x)$. A representative $g(x)$ of each element of $\mathrm{GF}(p^m)$ can be chosen to be of degree strictly less than m. If α is a root of $f(x)$ in an algebraic closure of $\mathbb{Z}_p$, then $\mathrm{GF}(p^m)$ is isomorphic to the field:

$$\mathbb{Z}_p(\alpha) = \left\{a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1} \colon a_i \in \mathbb{Z}_p, \forall i\right\}, \quad (3)$$

and so we can identify the two fields. Furthermore, if α is a generator of the cyclic finite multiplicative group of nonzero elements of $\mathbb{Z}_p(\alpha)$, then we say that $f(x)$ is *primitive*.

The Galois field $\mathrm{GF}(2^8)$ is particularly of specific interest in cryptographic applications, especially in S-boxes constructions. For our cryptographic purposes, we are interested in such a field whose defining irreducible polynomial is "primitive" (of degree 8, of course). It is well known that there are $(\varphi(2^8 - 1)/8) = 16$ such polynomials over $\mathbb{Z}_2$, for example, $p_1(x), \ldots, p_{16}(x) \in \mathbb{Z}_2[x]$, which we list in Table 1. In the following section, we construct 16 S-boxes out of the Galois fields corresponding to the aforementioned sixteen primitive irreducible polynomials.

2.2. The Proposed S-Box Construction Method. For each $i = 1, \ldots, 16$, consider the affine power affine map (APA):

$$S = A_1 \circ f \circ A_2 \colon \mathbb{Z}_2(\alpha_i) \longrightarrow \mathbb{Z}_2(\alpha_i), \quad (4)$$

where $A_1(x) = ax + b$ and $A_2(x) = cx + d$ $(a, b, c, d \in \mathbb{Z}_2(\alpha_i))$ are two affine maps with $a, c \neq 0$, and

$$f(x) = \begin{cases} x^{-1}, & \text{if } x \neq 0, \\ 0, & \text{if } x = 0. \end{cases} \quad (5)$$

Among other things, the map S, which is obviously bijective, was introduced by [5] to produce confusion in the scheme. For our S-boxes, we choose $a = 13$, $b = 14$ and $c = 102$ and $d = 210$. Figure 1 demonstrates the flow chart of the construction of the 16 different S-boxes. Moreover, the construction of S-boxes in correspondence to polynomial 1 (P_1) to polynomial 16 (P_{16}) is shown in Figure 1**.** All the S-boxes are given in Tables 2–17, corresponding to P_1 to P_{16}. These tables are before the conclusion section.

In the proposed work, we present an APA S-box corresponding to each $i = 1, \ldots, 16$ where the APA map S gives the 16×16 lookup tables. We, then, show that these S-boxes have strong cryptographic properties certified with the help of analyses such as nonlinearity, strict avalanche criterion (SAC), bit independent criterion (BIC), linear approximation probability (LP), and differential approximation probability (DP) [20].

Table 1: Primitive irreducible polynomials and their corresponding Galois fields.

Primitive polynomials $p_i(x)$& roots α_i	Galois field $\mathrm{GF}(2^8)$
$p_1(x) = x^8 + x^4 + x^3 + x^2 + 1; \alpha_1$	$\mathbb{Z}_2[x]/\langle p_1(x)\rangle$
$p_2(x) = x^8 + x^5 + x^3 + x + 1; \alpha_2$	$\mathbb{Z}_2[x]/\langle p_2(x)\rangle$
$p_3(x) = x^8 + x^5 + x^3 + x^2 + 1; \alpha_3$	$\mathbb{Z}_2[x]/\langle p_3(x)\rangle$
$p_4(x) = x^8 + x^6 + x^3 + x^2 + 1; \alpha_4$	$\mathbb{Z}_2[x]/\langle p_4(x)\rangle$
$p_5(x) = x^8 + x^6 + x^4 + x^3 + x^2 + x + 1; \alpha_5$	$\mathbb{Z}_2[x]/\langle p_5(x)\rangle$
$p_6(x) = x^8 + x^6 + x^5 + x + 1; \alpha_6$	$\mathbb{Z}_2[x]/\langle p_6(x)\rangle$
$p_7(x) = x^8 + x^6 + x^5 + x^2 + 1; \alpha_7$	$\mathbb{Z}_2[x]/\langle p_7(x)\rangle$
$p_8(x) = x^8 + x^6 + x^5 + x^3 + 1; \alpha_8$	$\mathbb{Z}_2[x]/\langle p_8(x)\rangle$
$p_9(x) = x^8 + x^7 + x^3 + x^2 + 1; \alpha_9$	$\mathbb{Z}_2[x]/\langle p_9(x)\rangle$
$p_{10}(x) = x^8 + x^7 + x^5 + x^3 + 1; \alpha_{10}$	$\mathbb{Z}_2[x]/\langle p_{10}(x)\rangle$
$p_{11}(x) = x^8 + x^7 + x^2 + x + 1; \alpha_{11}$	$\mathbb{Z}_2[x]/\langle p_{11}(x)\rangle$
$p_{12}(x) = x^8 + x^7 + x^6 + x + 1; \alpha_{12}$	$\mathbb{Z}_2[x]/\langle p_{12}(x)\rangle$
$p_{13}(x) = x^8 + x^7 + x^6 + x^5 + x^2 + x + 1; \alpha_{13}$	$\mathbb{Z}_2[x]/\langle p_{13}(x)\rangle$
$p_{14}(x) = x^8 + x^7 + x^6 + x^3 + x^2 + x + 1; \alpha_{14}$	$\mathbb{Z}_2[x]/\langle p_{14}(x)\rangle$
$p_{15}(x) = x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1; \alpha_{15}$	$\mathbb{Z}_2[x]/\langle p_{15}(x)\rangle$
$p_{16}(x) = x^8 + x^6 + x^5 + x^4 + 1; \alpha_{16}$	$\mathbb{Z}_2[x]/\langle p_{16}(x)\rangle$

3. Security Analysis

In this section, we present some algebraic and statistical analyses of S-box followed [21]. Such analyses indicate the strength of all the proposed S-boxes and give an idea for their application in image encryption and other modes of secure communication.

3.1. Nonlinearity. Nonlinearity analysis of a function f is the minimum hamming distance between the Boolean function $f : f \colon \{0,1\}^n \longrightarrow \{0,1\}$ and its all n-bit affine functions. In the truth table of Boolean function f, the nonlinearity of f represents the degree of dissimilarity between f and all affine function. If the function has high minimum hamming distance, it indicates it has high nonlinearity. It is an established fact that high nonlinearity provides resistance to any kind of linear approximation attacks [22, 23]. The calculated upper bound of nonlinearity is $M = 2^{m-1} - 2^{((m/2)-1)}$ so that, for $m = 8$, the optimal value of nonlinearity is 120. Table 18 shows the nonlinearity of 16 S-boxes corresponding to all primitive irreducible polynomials. From this table, it can be seen that the value of nonlinearity has not been affected due to background irreducible polynomial.

3.2. Strict Avalanche Criteria. In [24], Webster and Tavares introduced the strict avalanche criteria (SAC) on the concepts of completeness and avalanche. If a single input bit changes, the output bits change with almost 0.5 probability. It helps to show that the resulting output vector is highly random, and no single pattern can be predictable by minor variation in the input vector [25]. By seeing the performance indexes of S-boxes, the proposed S-boxes successfully satisfy SAC. Table 19 depicts the value of SAC for all the proposed 16 S-boxes. It shows that the maximum value of SAC is 0.562500 for the first 9 S-boxes including 11^{th}, 14^{th}, and 16^{th} S-boxes. Similarly,

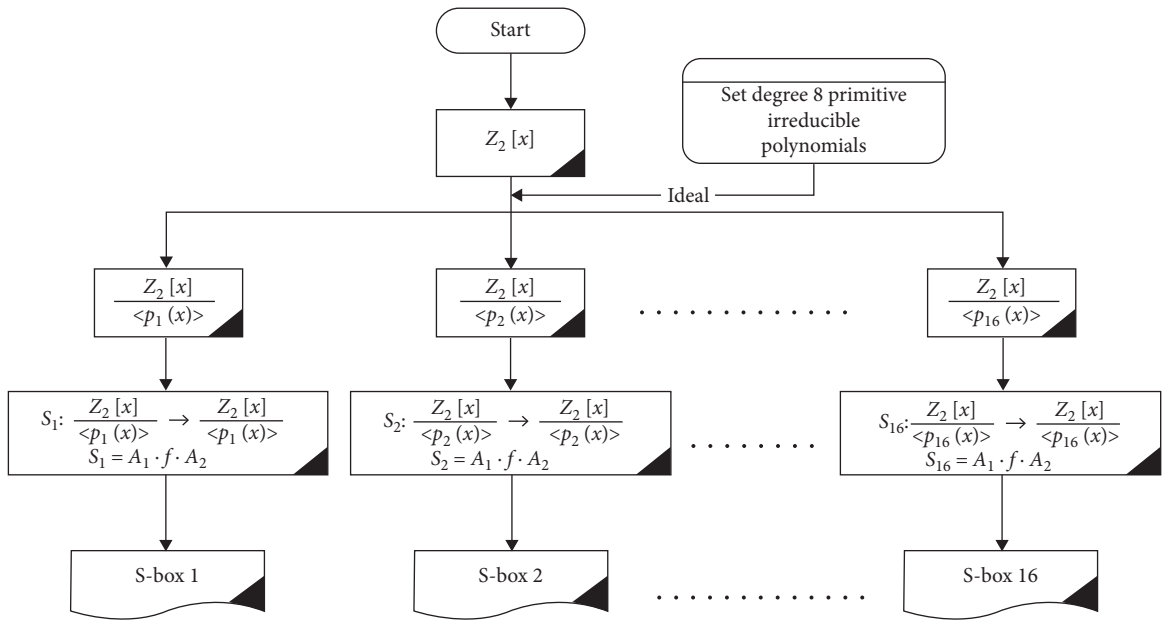


Figure 1: Flow chart for the construction of the proposed S-boxes and S-boxes corresponding to P_1 and P_{16}.

Table 2: S-box corresponding to P2.

Proposed S-box 1															
176	141	139	249	75	179	4	69	48	62	243	197	49	105	167	250
240	60	189	182	188	230	178	101	236	21	153	110	38	155	127	207
91	213	29	55	88	78	244	215	81	221	158	219	74	201	210	255
119	90	37	42	43	147	67	83	96	22	99	253	144	11	56	9
63	223	205	80	140	71	121	125	120	54	168	187	82	202	10	70
26	19	161	186	183	65	232	64	172	41	93	44	40	137	34	128
164	241	163	25	87	214	76	196	1	31	45	8	198	97	102	246
2	77	235	233	51	57	118	100	117	208	143	30	138	217	66	157
211	245	15	134	180	126	114	142	89	148	254	218	123	85	154	169
103	165	6	195	84	224	184	107	203	200	145	229	0	7	231	226
58	3	53	212	191	159	35	242	5	220	216	247	192	204	136	228
79	115	92	227	248	73	50	112	132	108	225	16	252	23	130	185
33	133	237	61	173	171	27	86	113	181	175	116	251	177	170	234
12	150	18	14	68	174	149	39	17	124	199	32	131	28	162	20
160	111	135	13	104	129	239	98	222	109	146	72	152	206	190	238
52	95	166	24	209	46	47	59	122	156	106	151	194	94	36	193

Table 3: S-box corresponding to P2.

Proposed S-box 2															
186	170	249	53	208	106	49	220	147	203	143	207	250	177	133	19
240	84	152	38	18	21	58	59	67	219	15	209	70	150	41	121
64	255	231	139	182	117	25	100	47	125	109	77	190	113	110	148
118	197	198	146	39	138	108	161	75	13	103	200	73	91	105	204
80	104	130	124	2	132	214	63	144	48	112	181	72	196	37	151
3	135	10	129	221	184	26	111	40	222	11	127	223	31	29	87
86	189	205	23	253	114	89	218	176	193	229	180	201	158	4	69
22	188	228	85	241	236	140	239	16	164	57	79	8	56	24	102
211	160	142	153	54	68	213	210	230	71	238	235	82	252	52	98
33	7	65	42	5	1	32	162	212	192	175	90	17	245	43	119
178	44	36	46	166	171	51	234	195	62	145	45	247	14	168	174
217	122	159	0	101	237	60	27	163	202	243	99	116	120	131	6
224	232	20	123	34	55	137	206	155	226	81	173	93	169	107	172
167	199	191	94	136	96	183	149	233	246	179	156	9	83	50	95
61	28	254	88	141	165	126	35	128	66	76	134	216	97	242	187
244	30	74	215	12	115	185	194	78	154	227	251	225	248	92	157

Table 4: S-box corresponding to P3.

Proposed S-box 3															
126	250	162	102	32	143	129	192	28	200	47	42	155	131	177	221
240	142	29	49	138	120	94	46	20	64	5	99	116	27	50	167
67	165	117	133	108	175	96	0	174	54	58	251	223	164	181	44
154	88	238	69	89	194	201	193	22	75	61	137	123	72	86	169
121	71	144	18	159	90	16	189	105	31	51	26	8	149	176	38
65	110	48	183	210	196	217	59	163	237	118	93	13	241	140	208
173	247	119	33	233	228	66	170	80	222	166	161	160	146	236	224
179	63	115	81	98	231	150	97	243	229	178	215	17	6	14	245
211	199	253	197	85	41	43	76	3	62	128	55	91	40	124	60
73	134	77	21	1	107	218	180	204	112	15	230	246	104	151	130
168	220	227	145	103	185	191	2	182	82	52	202	34	190	25	152
139	114	187	186	141	30	122	113	23	92	111	101	12	239	127	213
87	156	188	216	184	153	37	249	70	10	226	255	95	4	84	206
148	214	172	225	254	45	135	24	35	11	53	39	195	9	252	248
203	36	205	100	242	244	235	209	125	74	232	207	212	158	147	19
109	56	219	157	171	7	106	57	234	83	136	79	132	68	198	78

Table 5: S-box corresponding to P4.

Proposed S-box 4															
49	103	220	208	58	70	173	238	242	160	154	169	158	110	139	94
240	200	78	149	155	117	219	131	50	161	226	14	48	189	1	147
115	120	218	34	51	7	104	182	167	150	192	64	178	95	57	91
176	3	100	159	205	134	63	85	243	22	69	236	177	144	210	20
44	79	206	89	96	74	66	146	197	15	0	108	13	215	25	181
235	93	194	221	73	18	246	56	140	106	244	247	148	237	88	232
191	251	234	99	9	33	59	252	45	179	222	125	132	109	229	136
77	41	162	8	188	126	202	97	67	217	163	98	84	171	29	42
211	38	213	119	46	168	5	116	204	129	75	31	152	239	166	172
68	10	11	6	35	83	212	186	43	105	249	199	255	196	111	32
87	184	156	24	223	190	195	153	92	230	112	209	39	52	80	12
113	198	60	40	138	164	76	101	72	61	145	122	174	23	19	86
170	183	214	245	135	231	36	54	224	81	143	124	30	228	4	114
26	53	127	203	141	133	187	90	118	201	27	180	71	130	17	185
82	175	137	165	28	216	227	37	55	47	2	157	248	233	123	254
142	121	253	128	102	225	193	16	107	250	241	62	207	65	151	21

Table 6: S-box corresponding to P5.

Proposed S-box 5															
61	163	105	177	30	219	248	58	41	7	9	127	151	118	169	196
240	34	202	51	77	191	126	233	215	39	254	197	20	48	93	192
122	128	97	95	181	217	65	64	173	23	91	0	198	43	90	16
175	87	70	162	168	92	2	49	100	245	193	249	205	107	55	139
108	33	11	114	74	81	53	15	6	110	206	137	200	104	247	116
19	182	14	46	60	115	158	167	130	113	17	204	4	218	141	176
147	1	138	216	213	179	226	150	253	31	85	66	231	187	243	57
71	236	244	225	45	38	165	40	172	201	188	119	224	242	29	232
211	214	54	185	42	227	190	82	159	44	157	250	235	221	171	28
155	35	52	124	120	84	78	136	59	123	36	88	251	83	134	21
3	143	153	223	102	129	209	111	131	140	144	184	98	80	178	220
12	148	186	99	149	210	239	94	234	37	152	230	63	241	67	189
22	194	238	203	56	207	117	26	86	112	18	237	68	47	212	229
166	146	199	228	121	101	180	174	76	72	32	160	69	208	10	125
246	73	252	103	164	8	79	109	25	27	24	106	142	154	183	62
96	255	132	156	75	145	50	135	13	133	222	161	170	5	195	89

Table 7: S-box corresponding to P6.

Proposed S-box 6															
162	120	205	183	16	20	137	191	89	95	128	239	159	10	219	94
240	155	187	248	196	92	139	223	12	136	18	222	80	140	131	158
100	53	236	50	192	72	118	167	132	42	3	51	90	138	15	28
249	246	174	221	11	161	19	126	150	0	254	96	13	49	181	112
81	163	108	26	135	216	234	99	85	14	78	62	177	104	179	207
189	22	255	27	178	225	212	237	147	114	253	165	75	17	244	184
199	166	87	208	71	242	195	175	101	247	79	243	149	44	217	201
38	36	37	5	251	63	125	66	69	218	190	86	103	30	148	123
211	198	33	48	232	197	105	9	93	31	107	145	194	185	41	227
40	146	214	58	228	77	144	152	57	35	180	172	241	60	55	6
8	61	24	229	169	1	168	171	154	204	52	233	21	213	119	245
56	133	113	127	45	98	115	65	64	134	173	200	102	202	110	74
182	157	141	76	91	68	34	224	59	203	206	170	215	153	73	39
67	29	130	142	116	186	88	160	47	106	2	54	193	121	231	210
7	252	117	250	176	4	84	43	124	235	209	151	122	230	226	129
32	23	188	111	156	220	164	238	82	97	143	70	83	46	109	25

Table 8: S-box corresponding to P7.

Proposed S-box 7															
86	101	177	89	199	188	236	165	198	145	112	232	92	164	76	137
240	244	5	87	151	98	181	129	117	94	230	108	39	29	184	206
103	123	56	180	35	142	246	168	48	36	64	222	187	111	196	15
54	190	120	104	173	1	208	105	162	224	251	12	253	172	57	170
114	3	153	159	185	204	19	245	128	50	97	140	227	127	214	44
176	218	118	209	4	53	247	68	186	249	6	93	25	150	28	88
0	254	115	147	85	154	221	99	69	45	70	136	130	32	8	134
171	139	250	243	248	174	191	20	31	2	43	33	255	22	152	27
211	71	61	109	95	47	9	226	135	67	143	124	160	125	228	18
7	49	26	79	220	179	231	223	148	210	51	241	141	113	200	16
192	197	239	146	24	183	252	82	167	126	225	242	58	219	52	107
233	55	149	235	77	178	157	201	65	40	216	74	207	133	131	78
83	122	73	102	66	213	138	80	11	182	34	46	195	10	193	17
237	75	119	110	238	156	21	72	203	175	234	41	189	158	116	215
205	13	91	202	106	38	229	30	81	23	37	63	144	96	59	42
62	100	84	166	14	60	217	161	169	132	163	90	212	194	155	121

Table 9: S-box corresponding to P8.

Proposed S-box 8															
41	6	47	28	109	248	100	139	227	141	111	60	7	125	178	31
240	224	53	123	91	63	67	237	71	217	11	205	252	133	61	22
97	172	182	233	124	51	196	236	255	170	128	3	186	179	83	119
29	206	234	210	218	1	151	220	245	68	13	213	54	214	30	15
23	98	95	82	189	212	48	120	187	219	239	222	202	192	154	246
143	24	131	181	102	27	204	184	129	137	114	74	122	57	188	199
94	75	20	79	76	158	92	130	39	19	96	134	103	108	33	229
89	21	175	50	166	195	87	203	164	46	174	121	25	34	241	16
211	238	104	112	235	14	116	251	191	86	253	230	225	52	155	38
58	177	173	40	45	148	62	42	201	149	190	250	216	157	59	244
169	153	150	117	142	101	145	9	207	208	247	160	140	8	64	180
194	88	37	162	115	35	156	5	90	36	84	106	152	232	159	110
144	73	80	126	200	165	113	56	161	49	127	12	168	0	66	226
147	32	132	146	243	4	43	138	55	163	167	135	223	81	93	209
69	99	107	44	198	105	228	10	65	77	215	197	193	176	18	17
249	78	242	85	70	72	171	183	26	221	254	231	118	136	185	2

Table 10: S-box corresponding to P9.

Proposed S-box 9															
164	17	78	216	114	139	98	51	31	89	58	243	221	249	159	18
240	64	151	225	120	210	180	135	57	115	204	72	62	155	227	195
109	95	127	141	76	229	178	246	12	147	47	235	166	217	122	212
205	26	27	138	148	70	171	130	140	215	203	156	146	11	106	233
242	162	6	206	142	16	119	218	253	169	104	213	102	41	245	231
197	250	65	86	90	152	237	241	167	254	4	14	208	182	71	101
232	68	111	10	134	55	209	103	189	61	74	116	20	40	118	186
42	192	131	226	43	113	63	157	38	69	8	60	149	44	255	53
211	82	224	110	32	94	49	236	80	160	196	222	185	108	188	154
176	123	238	117	30	247	88	150	181	144	132	35	37	198	96	97
136	92	23	75	121	81	112	83	125	239	87	2	143	19	105	201
137	214	15	56	172	179	133	252	36	220	85	184	50	234	39	194
73	99	173	79	124	46	25	100	168	3	33	228	191	145	165	161
199	77	126	28	183	59	170	22	54	202	223	66	163	219	187	174
177	244	175	5	128	67	230	7	93	45	29	21	158	13	129	9
0	48	200	107	34	24	251	248	1	190	84	52	193	153	91	207

Table 11: S-box corresponding to P10.

Proposed S-box 10															
109	26	163	213	77	207	155	87	34	96	136	40	177	25	128	11
240	147	166	252	115	158	185	123	146	68	239	149	160	180	6	111
22	198	208	243	103	97	24	187	179	228	132	110	188	10	151	130
62	119	131	60	219	148	245	9	101	81	205	3	222	98	203	19
154	178	66	117	246	108	226	135	202	224	59	236	192	156	141	112
217	28	39	167	172	104	44	230	54	253	82	168	23	4	150	58
107	64	43	48	20	118	181	102	173	254	200	76	237	209	143	204
63	193	221	214	14	88	16	126	223	90	157	85	46	169	196	191
211	42	1	199	176	100	216	152	183	89	37	99	75	233	72	122
184	52	79	206	235	194	225	71	15	162	50	210	242	127	153	234
57	70	165	248	164	5	67	55	2	129	124	189	21	171	241	86
251	159	125	134	32	93	116	12	255	106	175	73	170	35	139	84
41	120	18	27	78	17	182	174	53	215	83	247	94	227	31	137
29	61	113	145	212	238	138	105	30	47	121	92	74	7	80	142
232	95	197	220	195	231	49	161	36	8	69	91	65	13	244	218
51	0	250	144	229	114	190	45	249	133	140	38	33	186	201	56

Table 12: S-box corresponding to P11.

Proposed S-box 11															
128	191	132	167	111	120	159	218	25	68	173	217	32	39	99	239
240	94	157	95	75	112	190	213	152	202	40	101	2	207	140	64
19	238	35	151	154	197	199	60	61	187	44	201	72	37	126	118
24	80	124	141	16	41	193	160	7	107	163	129	248	66	189	221
10	109	76	150	110	255	171	70	17	18	43	174	153	13	113	206
244	219	3	203	12	134	48	245	164	130	230	4	144	53	182	78
155	227	253	158	26	86	186	62	79	21	175	162	222	215	247	208
98	137	210	63	57	67	138	96	139	242	254	38	92	188	97	52
211	106	6	9	22	85	212	198	234	225	42	223	119	136	90	33
46	121	249	184	34	142	251	20	214	216	50	209	135	87	74	176
161	5	77	36	229	83	149	73	168	181	196	115	194	11	56	8
82	231	51	1	148	65	195	172	23	177	228	0	131	252	88	170
220	27	69	71	123	232	49	226	236	58	104	143	178	166	117	205
54	169	108	91	204	127	116	45	165	89	30	55	31	84	246	15
156	122	224	192	103	145	243	114	185	102	200	180	250	81	233	47
59	125	133	100	147	14	183	146	235	237	29	28	105	93	179	241

Table 13: S-box corresponding to P12.

Proposed S-box 12															
140	**249**	**110**	**3**	**253**	**6**	**19**	**160**	**102**	**247**	**235**	**219**	**17**	**166**	**184**	**34**
240	198	221	159	212	199	188	148	27	0	52	73	176	15	4	93
1	28	32	214	195	238	216	24	229	92	241	36	209	207	12	9
25	245	139	97	109	41	225	37	105	106	71	100	206	165	129	59
171	70	208	126	111	242	181	119	215	22	183	233	14	222	204	150
226	144	82	121	40	87	67	228	116	234	81	13	180	5	96	237
42	51	95	16	99	79	21	69	31	30	125	55	89	143	123	178
122	158	80	66	10	133	46	252	33	112	210	157	131	203	127	43
211	54	132	217	58	155	152	192	53	128	84	44	88	236	194	168
170	239	231	218	232	205	103	151	2	173	26	145	20	251	136	117
146	49	83	35	202	90	243	164	78	108	201	167	175	191	68	163
65	85	185	255	104	18	227	177	75	39	161	182	187	57	98	186
64	246	179	196	174	38	118	135	101	193	77	223	94	248	72	250
107	62	189	153	8	48	7	56	169	60	23	11	149	141	29	156
147	120	154	254	134	130	244	200	172	213	197	162	190	113	45	86
114	230	47	138	61	115	220	50	137	224	76	63	142	124	91	74

Table 14: S-box corresponding to P13.

Proposed S-box 13															
24	177	151	25	7	71	226	153	127	137	209	73	254	213	96	237
240	234	102	131	201	142	118	64	16	249	176	6	5	18	222	95
52	36	82	160	224	225	144	74	67	129	1	13	105	143	186	248
181	113	231	108	58	216	43	114	156	37	168	9	170	60	3	246
40	122	217	247	223	85	163	4	65	152	31	189	70	12	173	210
207	233	197	107	83	115	164	99	66	56	23	111	120	61	251	46
205	154	255	167	198	81	87	109	20	135	59	68	100	39	220	172
145	11	132	185	106	53	162	112	253	54	180	33	116	175	104	134
211	123	204	26	190	90	8	92	15	110	119	29	188	230	219	199
76	79	174	130	147	166	63	215	241	41	244	141	124	0	35	98
218	245	242	72	21	239	47	94	208	238	10	165	171	91	14	192
77	243	148	158	80	203	45	42	194	146	159	235	214	161	32	169
93	101	140	30	97	50	84	126	229	196	200	48	55	139	57	44
69	128	49	155	19	62	232	252	236	136	250	27	157	195	125	206
34	179	78	38	193	227	28	184	133	88	187	178	75	212	138	117
191	202	149	22	228	89	150	121	221	17	182	86	51	183	103	2

Table 15: S-box corresponding to P14.

Proposed S-box 14															
93	13	118	253	48	210	140	163	100	34	18	128	11	90	53	114
240	227	242	106	112	138	147	92	250	206	141	56	152	57	94	212
50	149	235	21	233	254	102	131	156	142	173	194	31	45	244	195
237	43	183	168	121	95	9	71	37	120	204	62	119	174	122	186
123	29	105	136	191	196	91	40	229	86	4	41	125	22	224	162
155	30	223	161	39	169	14	63	52	133	199	36	241	124	59	65
190	127	219	88	214	115	126	96	38	239	55	7	197	73	2	42
23	144	47	46	68	72	6	185	109	176	1	98	201	234	222	203
211	175	157	113	110	217	85	3	249	255	116	104	25	82	179	97
158	135	44	78	117	164	160	221	66	236	145	103	60	230	232	134
182	67	177	188	15	8	107	79	216	12	246	51	213	180	58	17
189	252	215	245	69	218	192	10	225	74	54	83	84	61	166	193
16	139	81	132	19	101	76	208	231	151	181	49	187	99	202	228
205	220	154	198	111	167	20	26	200	64	165	247	248	209	207	148
226	0	28	184	33	159	27	153	24	172	171	75	238	243	77	87
35	170	32	137	143	251	129	178	146	150	80	89	108	130	5	70

Table 16: S-box corresponding to P15.

Proposed S-box 15															
12	220	182	7	142	239	96	251	36	246	21	162	223	103	35	40
240	132	32	219	116	160	243	28	128	168	159	206	193	95	50	154
38	234	47	66	126	192	54	169	65	175	254	89	45	123	233	39
17	194	111	202	242	226	178	210	185	64	77	218	108	127	147	222
191	214	237	110	184	155	30	20	86	212	156	1	255	119	167	141
105	252	176	63	205	139	151	53	33	189	170	213	183	224	204	75
69	100	230	196	188	31	43	104	83	91	145	60	113	117	152	94
157	129	102	235	135	49	15	181	163	195	58	249	134	171	107	37
211	34	24	238	67	22	236	88	253	109	207	221	164	203	138	19
48	229	122	71	76	93	120	143	81	106	26	137	215	200	187	217
118	72	244	245	150	209	121	144	199	14	186	27	70	115	84	55
4	11	73	80	179	79	146	18	173	6	172	8	56	136	78	61
42	52	85	166	133	41	228	25	99	90	158	125	57	29	2	248
10	231	92	5	201	180	68	177	174	23	161	208	98	62	232	140
51	165	216	0	241	101	9	97	131	148	198	247	130	114	250	112
44	46	225	13	153	3	16	82	190	124	59	197	149	227	74	87

Table 17: S-box corresponding to P15.

Proposed S-box 16															
6	136	68	218	166	118	48	114	2	173	30	39	204	81	144	134
240	46	115	83	91	202	1	203	10	63	145	4	89	112	213	17
47	8	176	149	117	181	38	133	56	87	162	20	199	78	60	157
254	14	192	29	200	178	140	69	24	121	160	148	158	252	67	228
111	103	102	194	33	122	44	205	61	100	229	215	80	187	217	189
28	124	147	92	31	76	45	171	137	239	95	154	42	64	51	55
119	236	77	242	0	12	75	216	84	151	220	49	101	150	129	53
155	198	174	146	90	233	88	225	54	179	5	74	246	138	110	230
211	153	208	214	40	184	13	255	71	163	62	130	175	232	231	128
156	141	34	152	15	212	165	41	106	97	167	72	177	169	234	107
170	120	247	104	245	195	7	172	19	65	85	36	70	108	209	222
197	132	244	57	201	182	96	52	125	210	183	127	241	126	235	26
58	227	143	11	238	86	135	191	66	168	237	3	59	123	32	27
105	207	221	22	248	251	243	50	23	16	186	98	82	250	180	164
193	219	93	79	35	196	249	188	139	253	161	99	37	223	224	226
131	21	109	25	73	116	43	206	94	113	190	142	185	159	9	18

Table 18: Nonlinearity of the proposed S-box.

S-boxes	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	Average
S-box 1	112	112	112	112	112	112	112	112	112
S-box 2	112	112	112	112	112	112	112	112	112
S-box 3	112	112	112	112	112	112	112	112	112
S-box 4	112	112	112	112	112	112	112	112	112
S-box 5	112	112	112	112	112	112	112	112	112
S-box 6	112	112	112	112	112	112	112	112	112
S-box 7	112	112	112	112	112	112	112	112	112
S-box 8	112	112	112	112	112	112	112	112	112
S-box 9	112	112	112	112	112	112	112	112	112
S-box 10	112	112	112	112	112	112	112	112	112
S-box 11	112	112	112	112	112	112	112	112	112
S-box 12	112	112	112	112	112	112	112	112	112
S-box 13	112	112	112	112	112	112	112	112	112
S-box 14	112	112	112	112	112	112	112	112	112
S-box 15	112	112	112	112	112	112	112	112	112
S-box 16	112	112	112	112	112	112	112	112	112

Table 19: Strict avalanche criterion.

S-box	Max	Min	Average	Square deviation
S-box 1	0.562500	0.437500	0.496094	0.0172495
S-box 2	0.546875	0.453125	0.495117	0.0128725
S-box 3	0.562500	0.453125	0.494141	0.0152856
S-box 4	0.562500	0.453125	0.50708	0.0118748
S-box 5	0.546875	0.453125	0.503174	0.0153901
S-box 6	0.562500	0.453125	0.501709	0.016637
S-box 7	0.562500	0.453125	0.502441	0.0170951
S-box 8	0.562500	0.453125	0.503906	0.0165152
S-box 9	0.562500	0.453125	0.485596	0.0153978
S-box 10	0.546875	0.453125	0.509766	0.0123912
S-box 11	0.562500	0.4375	0.50415	0.0191487
S-box 12	0.5625	0.453125	0.501221	0.016475
S-box 13	0.546875	0.4375	0.500977	0.0127235
S-box 14	0.5625	0.453125	0.508301	0.0158654
S-box 15	0.546875	0.437500	0.498779	0.0143727
S-box 16	0.5625	0.437500	0.496582	0.0143171

Table 20: Bit independent criterion.

	BIC-SAC			BIC		
S-box	Min	Average	Square deviation	Max	Average	Square deviation
S-box 1	0.47461	0.50119	0.01132	112	112	0
S-box 2	0.49219	0.50600	0.00845	112	112	0
S-box 3	0.48047	0.50202	0.01015	112	112	0
S-box 4	0.47852	0.50656	0.01201	112	112	0
S-box 5	0.48438	0.50105	0.00924	112	112	0
S-box 6	0.47656	0.49819	0.00784	112	112	0
S-box 7	0.48633	0.50593	0.00925	112	112	0
S-box 8	0.48828	0.50251	0.00835	112	112	0
S-box 9	0.48828	0.50258	0.00644	112	112	0
S-box 10	0.48438	0.50739	0.00981	112	112	0
S-box 11	0.47656	0.49784	0.00950	112	112	0
S-box 12	0.47070	0.49679	0.01026	112	112	0
S-box 13	0.48242	0.50021	0.01085	112	112	0
S-box 14	0.49023	0.50718	0.00901	112	112	0
S-box 15	0.48438	0.50265	0.00883	112	112	0
S-box 16	0.48633	0.50544	0.00861	112	112	0

Table 21: Linear and differential approximation probability analysis.

S-box	Linear approximation probability		Differential approximation probability	
	Max count	LP	Max value	DP
S-box 1	144	0.0625	4	0.015625
S-box 1	144	0.0625	4	0.015625
S-box 2	144	0.0625	4	0.015625
S-box 3	144	0.0625	4	0.015625
S-box 4	144	0.0625	4	0.015625
S-box 5	144	0.0625	4	0.015625
S-box 6	144	0.0625	4	0.015625
S-box 7	145	0.0664	4	0.015625
S-box 8	144	0.0625	4	0.015625
S-box 9	144	0.0625	4	0.015625
S-box 10	144	0.0625	4	0.015625
S-box 11	144	0.0625	4	0.015625
S-box 12	144	0.0625	4	0.015625
S-box 13	144	0.0625	4	0.015625
S-box 14	144	0.0625	4	0.015625
S-box 15	144	0.0625	4	0.015625
S-box 16	144	0.0625	4	0.015625

Table 22: Comparison of the performance indexes of the proposed S-boxes with some standard S-boxes.

8×8 S-boxes	Nonlinearity	SAC	BIC	BIC-SAC	DP	LP
AES S-box	112	0.5058	112	0.504	0.0156	0.062
APA S-box	112	0.4987	112	0.499	0.0156	0.062
Gray S-box	112	0.5058	112	0.502	0.0156	0.062
Skipjack S-box	105.7	0.4980	104.1	0.499	0.0468	0.109
Xyi S-box	105	0.5048	103.7	0.503	0.0468	0.156
Residue prime	99.5	0.5012	101.7	0.502	0.2810	0.132
Reference [27]	106	0.4978	103.92	—	—	—
Reference [28]		0.505	—	—	—	—
Reference [29]	104	0.5241	103	0.50181	0.1625	0.0486
S-box 1	112	0.496094	112	0.50119	0.015625	0.0625
S-box 2	112	0.495117	112	0.50600	0.015625	0.0625
S-box 3	112	0.494141	112	0.50202	0.015625	0.0625
S-box 4	112	0.50708	112	0.50656	0.015625	0.0625
S-box 5	112	0.503174	112	0.50105	0.015625	0.0625
S-box 6	112	0.501709	112	0.49819	0.015625	0.0625
S-box 7	112	0.502441	112	0.50593	0.015625	0.0625
S-box 8	112	0.503906	112	0.50251	0.015625	0.0664
S-box 9	112	0.485596	112	0.50258	0.015625	0.0625
S-box 10	112	0.509766	112	0.50739	0.015625	0.0625
S-box 11	112	0.50415	112	0.49784	0.015625	0.0625
S-box 12	112	0.501221	112	0.49679	0.015625	0.0625
S-box 13	112	0.500977	112	0.50021	0.015625	0.0625
S-box 14	112	0.508301	112	0.50718	0.015625	0.0625
S-box 15	112	0.498779	112	0.50265	0.015625	0.0625
S-box 16	112	0.496582	112	0.50544	0.015625	0.0625

Table 23: MLC analyses for the proposed S-boxes.

S-boxes	Entropy	Correlation	Contrast	Energy	Homogeneity
S-box 1	7.2763	0.1032	8.8505	0.0175	0.4499
S-box 2	7.2763	0.1246	9.1517	0.0174	0.4530
S-box 3	7.2763	0.0895	9.5748	0.0175	0.4532
S-box 4	7.2763	0.1486	8.9132	0.0184	0.4579
S-box 5	7.2763	0.1277	9.2077	0.0195	0.4714
S-box 6	7.2763	0.1326	9.8130	0.0178	0.4532
S-box 7	7.2642	0.0904	10.0599	0.0177	0.4438
S-box 8	7.2763	0.1256	9.6635	0.0182	0.4509
S-box 9	7.2763	0.1099	8.9653	0.0179	0.4609
S-box 10	7.2763	0.0747	9.7106	0.0177	0.4453
S-box 11	7.2763	0.0927	9.7905	0.0182	0.4529
S-box 12	7.2763	0.0879	9.8720	0.0182	0.4531
S-box 13	7.2763	0.0868	8.5569	0.0188	0.4606
S-box 14	7.2763	0.0862	8.6308	0.0179	0.4510
S-box 15	7.2763	0.1245	9.3829	0.0184	0.4522
S-box 16	7.2763	0.0843	9.5898	0.0179	0.4533

the minimum value of SAC is 0.453125 for the first 10 S-boxes including 12^{th} and 14^{th} S-boxes. The average value of SAC lies in the interval [0.4856, 0.509766].

3.3. Bit Independent Criterion. Another algebraic criterion (BIC) is used to evaluate the strength of S-box, which is presented by Detombe and Tavares in [26]. In Table 14, the outcomes of BIC to SAC and BIC for the proposed S-boxes are given. The minimum BIC to SAC value is 0.47070 for 12^{th} S-box and the highest minimum value is 0.49219 for 2^{nd} S-box. The average BIC to SAC lies between 0.49679 and 0.50739. Similarly, the square deviation values for all the proposed S-boxes are given in Table 20. The maximum and average value of BIC is 112 for all S-boxes. It is depicted that the proposed S-boxes give the nearest best value of BIC analyses.

3.4. Linear Approximation Probability. Matsui defines the extreme value of the imbalance of an event as the linear approximation probability. It is notable that the parity of the input bits that is, the mask G_l, is equal to the parity of the output bits, i.e., the mask G_m. The linear approximation probability of a given S-box is defined in the following equation:

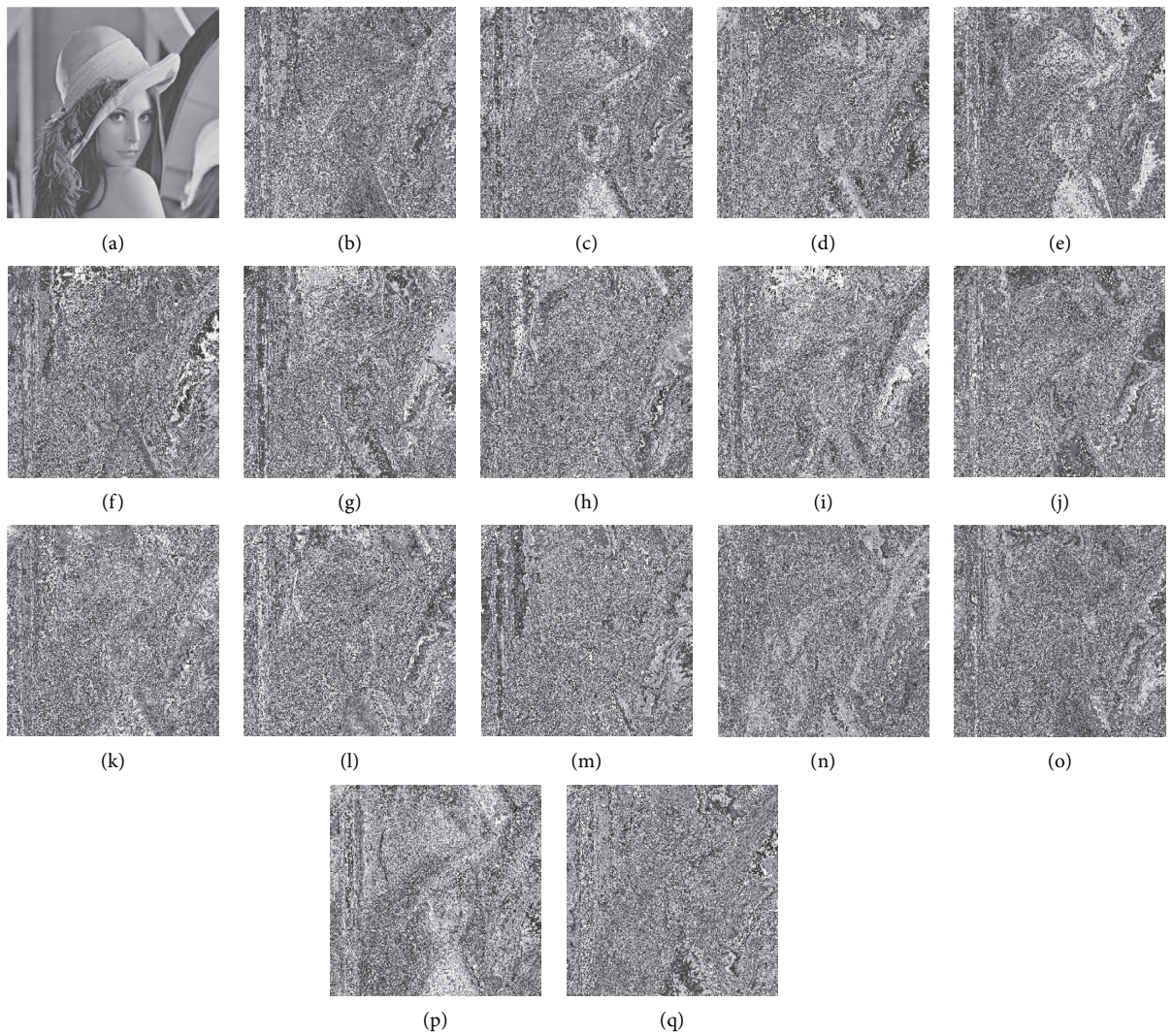


FIGURE 2: Original Lena image and the encrypted images using all 16 primitive irreducible polynomials S-boxes. (a) Lena image, (b) S-box 1, (c) S-box 2, (d) S-box 3, (e) S-box 4, (f) S-box 5, (g) S-box 6, (h) S-box 7, (i) S-box 8, (j) S-box 9, (k) S-box 10, (l) S-box 11, (m) S-box 12, (n) S-box 13, (o) S-box 14, (p) S-box 15, and (q) S-box 16.

$$\mathrm{LP} = \max_{G_l, G_m \neq 0} \left| \frac{\#\{l \in X \mid l \cdot G_l = S(l) \cdot G_m\}}{2^n} - \frac{1}{2} \right|, \tag{6}$$

where G_l and G_m are input and output masks, respectively, and the set "X" represents the set of all possible inputs; 2^n is the number of elements of X. The value of linear approximation indicates the strength of S-box against various linear attacks. In Table 21, the maximum count and the LP value for all proposed S-boxes is 144 and 0.0625. These values of LP of the proposed S-boxes are appropriate against linear attacks.

3.5. Differential Approximation Probability. The degree of differential uniformity is known as differential approximation probability (DP^s) of S-box. Mathematically, it can be given as

$$\mathrm{DP}^s(\Delta l \longrightarrow \Delta m) = \left[\frac{\#\{l \in X \mid S(l) \pm S(l \pm \Delta l) = \Delta m\}}{2^m} \right]. \tag{7}$$

Briefly, it can be explained as follows: an input differential Δl_i must be mapped to an output differential Δm_i uniquely for each i. Here, X represents all the possible input values and the number of its elements is given by 2^m. Table 21 depicts the results of DP, which include the maximum and DP value.

Moreover, Table 22 represents the values of proposed S-boxes along with AES, Skipjack, Xyi, APA, Gray, and residue prime S-boxes.

3.6. Statistical Analyses. To evaluate the visual strength of the substitution with the help of the proposed S-boxes, various statistical analyses are made on the host

and substituted images. In this proposed work, statistical analyses like homogeneity, entropy, contrast, energy, and correlation are used to evaluate the substitution ability of the 16 proposed S-boxes. These analyses are given as

$$\text{correlation} = \sum_{k,l} \frac{(k-\mu k)(l-\mu l)}{\sigma_k \sigma_l} p(k,l), \tag{8}$$

$$\text{contrast} = \sum_{k,l} |k-l|^2 p(k,l), \tag{9}$$

$$\text{entropy} = -\sum_{k,l} \text{pr}(p(k,l)) \log \Pr(p(k,l)), \tag{10}$$

$$\text{homogeneity} = \sum_{k,l} \frac{p(k,l)}{1+|k-l|}, \tag{11}$$

$$\text{energy} = -\sum_{k,l} p(k,l)^2, \tag{12}$$

where k, l give the row and column locations of an image. The pixel value at k^{th} row and l^{th} column is represented by $p(k, l)$ and $\Pr(p(k,l))$ is the probability of the image pixel. In equation (8), μ and σ are mean and standard deviation, respectively.

Correlation analysis helps to find the similarity between the host and substituted image. The correlation analysis provides the range which indicates the perfect, negative, and positive correlation. This is $[-1, 1]$ interval for correlation and value of 1 indicates the perfect correlation.

The randomness of the digital image can be calculated with the help of entropy. The higher value of entropy from the interval $[0, 8]$ represents the higher amount of randomness in a digital image. For any viewer, it is only possible with the help of contrast analysis to intensely recognize the objects in the texture of an image. With the help of contrast analyses, one can observe the maximum distinction in image pixels. The range of the contrast can be given by $[(\text{size}(\text{Image}) - 1)^2]$. For constant image, the value of contrast is zero. The goal of finding close distribution between the matrix and its diagonal is obtained in homogeneity analysis. The matrix used in this analysis is named gray level cooccurrence matrix (GLCM) and the range of homogeneity lies between 0 and 1. The range for energy analysis also lies in the interval [0, 1]. The results of Table 23 are obtained by applying these analyses on the original and encrypted images. For all the proposed 16 S-boxes, we calculated the values of the statistical analyses.

A 256×256 JPEG image of Lena is considered for MLC analysis. Figure 2 shows the results of image encryption with 16 proposed S-boxes.

4. Balanced Boolean Function

4.1. Balance Property. The imbalance of a Boolean function weak system against linear cryptanalysis highlights the importance of balance property. The balance property indicates that the higher the magnitude of a function's imbalance, the more the chances of a high probability linear approximation. A Boolean function $f: \mathbb{Z}_2^n \longrightarrow \mathbb{Z}_2$ is balanced. If the cardinality or Hamming weight of these two functions, that is, $\{x: f(x) = 0\}$ and $\{x: f(x) = 1\}$ is the same, then it is named the balance function.

4.2. Balance Property of the Proposed S-Box. All the Boolean functions $f_i, i = 1, 2, 3, \ldots, 8$ involved in proposed S-boxes are balanced just like the Boolean functions of AES, S_8, AES and other well-known S-boxes. The nonlinearity of the proposed S-boxes is equal to 112.

5. Conclusion

In this paper, a scheme for the synthesis of 8×8 S-boxes over 16 isomorphic Galois fields is presented. Here, we fixed all the parameters of affine power affine transformation, that is, a, b, c, d for 16 S-boxes. We have 16 primitive irreducible polynomials of degree 8 and they prompt us to construct 16 Galois field extensions of order 256. By using elements of the Galois field, corresponding to each different pair of the parameters, one can construct different S-boxes. These S-boxes obtained as a result of APA transformation which is bijective, pass nonlinearity test, and out bit independent criterion (BIC) which demonstrates that the existing S-boxes have high confusion producing capability. The evaluation of constructed S-boxes is done with some algebraic and statistical analyses. The results of these analyses highlight the characteristics of all the proposed S-boxes and later these S-boxes are equated with some of the existing S-boxes. In addition to this, we also ensured that all these constructed S-boxes are balanced that guarantee the strength of our S-boxes. Hence, we have concluded that a large class of S-boxes can be obtained by varying parameters of affine power affine transformations. These S-boxes can be used for secure communication.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

There are no conflicts of interest among the authors.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through research groups program under Grant no. R.G.P. 1/234/41.

References

[1] J. Daemen and V. Rijmen, "The design of Rijndael: AES,"in The Advanced Encryption Standard, Springer, Berlin, Germany,2002.

[2] R. Zimmermann, A. Curiger, H. Bonnenberg, H. Kaeslin, N. Felber, and W. Fichtner, "A 177 Mb/s VLSI implementation of the international data encryption algorithm,"

IEEE Journal of Solid-State Circuits, vol. 29, no. 3, pp. 303–307, 1994.

[3] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[4] National Bureau of Standards, *Data Encryption Standard*, Vol. 46, FIPS Publication, U.S. Department of Commerce, Washington, DC, USA, 1977.

[5] L. Cui and Y. Cao, "A new S-box structure named affine-power-affine," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 3, pp. 751–759, 2007.

[6] I. Hussain, T. Shah, H. Mahmood, and M. A. Gondal, "Construction of S8 Liu J S-boxes and their applications," *Computers & Mathematics with Applications*, vol. 64, no. 8, pp. 2450–2458, 2012.

[7] M. T. Tran, D. K. Bui, and A. D. Duong, "Gray S-box for advanced encryption standard,"in Proceedings of the 2008 International Conference on Computational Intelligence and Security, vol. 1, IEEE, Suzhou, China, December 2008.

[8] T. Shah and D. Shah, "Construction of highly nonlinear S-boxes for degree 8 primitive irreducible polynomials over $\mathbb{Z}2$," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 1219–1234, 2019.

[9] M. Khan, T. Shah, H. Mahmood, M. A. Gondal, and I. Hussain, "A novel technique for the construction of strong S-boxes based on chaotic Lorenz systems," *Nonlinear Dynamics*, vol. 70, no. 3, pp. 2303–2311, 2012.

[10] M. Khan and T. Shah, "An efficient construction of substitution box with fractional chaotic system," *Signal, Image and Video Processing*, vol. 9, no. 6, pp. 1335–1338, 2015.

[11] I. Hussain, T. Shah, H. Mahmood, and M. A. Gondal, "A projective general linear group based algorithm for the construction of substitution box for block ciphers," *Neural Computing and Applications*, vol. 22, no. 6, pp. 1085–1093, 2013.

[12] Y. Tian and Z. Lu, "Novel permutation-diffusion image encryption algorithm with chaotic dynamic S-box and DNA sequence operation," *AIP Advances*, vol. 7, no. 8, Article ID 085008, 2017.

[13] M. Khan, F. Masood, A. Alghafis, M. Amin, and S. I. Batool Naqvi, "A novel image encryption technique using hybrid method of discrete dynamical chaotic maps and Brownian motion," *PLoS One*, vol. 14, no. 12, Article ID e0225031, 2019.

[14] M. Khan and T. Shah, "A novel cryptosystem based on general linear group," *3D Research*, vol. 6, no. 1, 2015.

[15] D. Shah, T. Shah, and S. S. Jamal, "A novel efficient image encryption algorithm based on affine transformation combine with linear fractional transformation," *Multidimensional Systems and Signal Processing*, vol. 31, no. 3, pp. 885–905, 2020.

[16] Y. Naseer, D. Shah, and T. Shah, "A novel approach to improve multimedia security utilizing 3D mixed chaotic map," *Microprocessors and Microsystems*, vol. 65, pp. 1–6, 2019.

[17] K. E. A. Skipjack, "Algorithm," *Specifications Version*, vol. 2, no. 29, pp. 1–23, 1998.

[18] E. S. Abuelyman and A.-A. Sultan Alsehibani, "An optimized implementation of the S-box using residue of prime numbers," *International Journal of Computer Science and Network Security*, vol. 8, no. 4, pp. 304–309, 2008.

[19] S. Mahmood et al., "To study the effect of the generating polynomial on the quality of nonlinear components in block ciphers." *Security and Communication Networks*, vol. 2018, Article ID 5823230, 8 pages, 2018.

[20] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology—Eurocrypt'93*, pp. 386–397, Springer Berlin Heidelberg, Heidelberg, Germany, 1993.

[21] Y. Wang, Q. Xie, Y. Wu, and B. Du, "A software for S-box performance analysis and test," in *Proceedings of the 2009 International Conference on Electronic Commerce and Business Intelligence*, pp. 125–128, IEEE, Beijing, China, June 2009.

[22] M. A. Gondal, A. Raheem, and I. Hussain, "A scheme for obtaining secure S-boxes based on chaotic Baker's map," *3D Research*, vol. 5, no. 3, pp. 5–17, 2014.

[23] A. Belazi, R. Rhouma, and S. Belghith, "A novel approach to construct S-box based on Rossler system," in *Proceedings of the 2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 611–615, Dubrovnik, Croatia, August 2015.

[24] A. F. Webster and S. E. Tavares, "On the design of S-boxes," in *Advances in Cryptology—Crypto'85 Proceedings*, pp. 523–534, Springer Berlin Heidelberg, Heidelberg, Germany, 1985.

[25] F. Sattar and M. Mufti, "Spectral characterization and analysis of avalanche in cryptographic substitution boxes using walsh-hadamard transformations," *International Journal of Computer Applications*, vol. 28, no. 6, 2011.

[26] J. Detombe and S. Tavares, "On the design of S-boxes," *Advances in Cryptology: Proceedings of CRYPTO_92*, Springer Berlin Heidelberg, Heidelberg, Germany, Lecture Notes in Computer Science, 1992.

[27] A. K. Farhan, R. S. Ali, H. Natiq, and N. M. G. Al-Saidi, "A new S-box generation algorithm based on multistability behavior of a plasma perturbation model," *IEEE Access*, vol. 7, pp. 124914–124924, 2019.

[28] A. Farhan, R. Subhi, H. Rashed Yassein, and N. Al-Saidi, "A new approach to generate multi S-boxes based on RNA computing," *International Journal of Innovative Computing, Information and Control: IJICIC*, vol. 16, no. 1, pp. 331–348, 2020.

[29] D. Shah and T. Shah, "Binary galois field extensions dependent multimedia data security scheme," *Microprocessors and Microsystems*, vol. 77, Article ID 103181, 2020.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8867792, 10 pages
https://doi.org/10.1155/2020/8867792

Research Article

Towards an Improved Energy Efficient and End-to-End Secure Protocol for IoT Healthcare Applications

Arshad Ahmad,[1,2] **Ayaz Ullah**,[2] **Chong Feng**,[1] **Muzammil Khan**,[3] **Shahzad Ashraf**,[4] **Muhammad Adnan**,[5] **Shah Nazir**,[2] **and Habib Ullah Khan**[6]

[1]*School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China*
[2]*Department of Computer Science, University of Swabi, Anbar, Pakistan*
[3]*Department of Computer Science, University of Swat, Swat, Pakistan*
[4]*College of Internet of Things Engineering, Hohai University, Changzhou, Jiangsu, China*
[5]*Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan*
[6]*Department of Accounting & Information Systems, Qatar University, Doha, Qatar*

Correspondence should be addressed to Chong Feng; fengchong@bit.edu.cn

Received 22 July 2020; Revised 24 August 2020; Accepted 3 September 2020; Published 22 September 2020

Academic Editor: Amir Anees

In this paper, we proposed LCX-MAC (local coordination X-MAC) as an extension of X-MAC. X-MAC is an asynchronous duty cycle medium access control (MAC) protocol. X-MAC used one important technique of short preamble which is to allow sender nodes to quickly send their actual data when the corresponding receivers wake up. X-MAC node keeps sending short preamble to wake up its receiver node, which causes energy, increases transmission delay, and makes the channel busy since a lot of short preambles are discarded, as these days Internet of Things (IoT) healthcare with different sensor nodes for the healthcare is time-critical applications and needs a quick response. A possible improvement over X-MAC is that local information of each node will share with its neighbour node. This local information exchanged will cause much less overhead than in the nodes which are synchronized. To calculate the effect of this the local coordination on X-MAC in this paper, we built an analytical model of LCX-MAC that incorporates the local coordination in X-MAC. The analytical results show that LCX-MAC outperformed X-MAC and X-MAC/BEB in terms of throughput, delay, and energy.

1. Introduction

As with rapid development in IoT healthcare devices, a large amount of data is transmitted over limited energy resources, so there is a requirement for IoT healthcare environment which mostly consists of wireless sensor nodes that collect different data from the environment [1, 2]. MAC protocols with duty cycle mechanism have been established mainly for saving energy in IoT healthcare and wireless sensor networks (WSNs). IoT healthcare and WSNs have many numbers of nodes; each node has a transceiver and more than one sensor to monitor changes in the environment and send the sensed data to collection centre directly or through some middle nodes. As the events to inform infrequently occur, the node only wakes up periodically for a short time to send or receive data known as duty cycle mechanism. Node only wakes up for a short time and remains in the sleep mode most of the time. In the sleep mode, the node turns off transceiver to save energy by avoiding idle listening. Thus, a duty cycle interchanges an extended sleep time duration and a short wakeup time duration to save energy.

MAC protocols with duty cycle behaviour are broadly divided into two categories, asynchronous and synchronous. Nodes in asynchronous MAC protocols randomly wakeup and keep sending preamble until its corresponding receivers become awake. Synchronous MAC protocols from time to time exchange their schedule information with their neighbour nodes. The first synchronous MAC protocol was S-MAC [3], and there are many modifications of S-MAC

such as T-MAC [4], RMAC [5], DW-MAC [6], and SCP-MAC [7] with the main purpose being to save energy.

To increase energy efficiency, asynchronous MAC protocols consisting of B-MAC [8], WiseMAC [9], and X-MAC [10] use different method compared to synchronous MAC protocols. In asynchronous MAC protocols, each node is allowed to function independently from its neighbour nodes. The advantage of working independently is that the nodes do not periodically exchange their schedule information; however, sender nodes in asynchronous MAC protocols required continuing to send a long preamble until their corresponding receiver nodes wake up. Even the receiver detects the presence of preamble, but it has to stay awake, unable to reply to its sender node. The sender node at the end of the non-preemptive preamble sends the actual data. Although the asynchronous MAC protocols perform better, for types of networks with infrequent traffic, these protocols have the problem of long preambles duration of one duty cycle. Those WSNs which are less frequently sending data with long duty cycle will consume more energy as it required long preamble to wake up receivers.

B-MAC has many variants. X-MAC is one of the B-MAC variants with two advanced functions. The first function is that X-MAC allows receiver nodes to reply instantly to their sender nodes of the availability, even during the transmission of the preamble. X-MAC divides the long interruptible preamble into short preambles with a break in which the receiver which woke up can promptly respond, which is known as early acknowledgement (ACK). The other function is that X-MAC inserts the target address in each preamble, so the overhearing nodes problem is solved.

Although X-MAC reduced the duration of the long preamble, it still has the performance degradation in the crowded networks due to collisions. There is no mechanism in X-MAC to deal with collisions. As in the X-MAC, the sender node cannot identify the occurrence of the collision, so the nodes send preamble for the complete cycle once a collision occurred. There are possibilities of collisions at the retransmission because of using constant contention window. Note that, in binary exponential backoff (BEB) algorithm, the CW is double as collisions consecutively occur.

Nodes in IoT healthcare and WSNs are overcrowded, so the probability of collision is high, because of *spatially correlated contention (SCC)* [11], in which many nodes closely located sense the same event and concurrently start sending data. For solving the problem, we need to design and control SCC due to the fact that collisions can occur by reducing throughput and energy efficiency. For this problem, we proposed X-MAC/CA [12] and X-MAC/BEB [13] protocols; X-MAC/CA is X-MAC with collision avoidance (CA), and X-MAC/BEB is X-MAC with BEB algorithms to randomize the transmission time. As WSNs become congested, the BEB decrease the effect of collisions by dynamically randomizing the transmission time.

As for X-MAC and X-MAC/BEB, sender node continuously sends short preamble until its receiver node wakes up, which causes energy consumptions and makes the medium busy as many short preambles are discarded because the receiver node is not awake. The possible improvement over X-MAC and X-MAC/BEB is that each node will share its wakeup information with its neighbour node so that each node can know the wakeup time of its neighbour nodes, which is named as LCX-MAC.

The exchange of this wakeup information will cause very low overhead compared to the synchronization of the nodes used in other synchronized protocols, and this will reduce the problem of sending many numbers of sending short preamble, as now each node will only need to send one short preamble. This wakeup information share can help the neighbouring nodes to collaborate access to the shared medium, so this will solve the spatially correlated contention. This research aims to develop an energy-efficient and low latency X-MAC protocol for IoT healthcare and WSNs. To reduce the duration of short preambles of X-MAC/BEB, we will synchronize the wakeup time of the nodes between source and destination.

This research work is an extension of our previous work X-MAC/BEB. In X-MAC/BEB only BEB was added to X-MAC protocol, but in this research using X-MAC/BEB we will adjust the wakeup time of sender nodes according to their corresponding receiver nodes, as the sender nodes in X-MAC and X-MAC/BEB have the problem of sending the short preamble until their receiver replies giving more considerable delay and energy consumptions.

The X-MAC and our proposed X-MAC/CA and X-MAC/BEB are based on CSMA. In CSMA, each node tries to access the shared medium when the data ready to send makes the MAC protocols simple but leads to the "busy medium" as a node keeps sending short preambles to wake up its receivers nodes, which causes energy consumptions and keeps the medium busy as many of short preambles are discarded. The busy medium problem can also affect the result of applying a long duty cycle approach, as the sender node keeps sending many short preamble which increases energy consumptions to wake up its receiver nodes.

Busy medium by sending many short preambles has occurred because sender nodes have no information of their receiver nodes duty cycle information (X-MAC and X-MAC/BEB are asynchronous MAC protocols). The possible extension for this problem over X-MAC and X-MAC/BEB is that each node exchanges wakeup information with its neighbour nodes so that a sender node has wakeup information of its receiver node.

It is important to note that the exchanges of this wakeup information will cause very low overhead compared to nodes synchronization used in synchronized MAC protocols. Still, it will efficiently solve the busy medium problem as each sender node only sends one short preamble by knowing that its receiver node wakes up.

The exchange of this wakeup information allows the neighbouring nodes to coordinately access the shared medium and thus solve the problem spatially correlated contention. For example, each sender node can access medium according to its neighbour nodes access, which causes an increase in the throughput and reduction in energy consumptions. This exchange of neighbour information and how to coordinate are hot research issue that needs more research study.

The remainder of this paper is composed of the following. Section 2 is about related work and Section 3 introduces the LCX-MAC protocol. Section 4 describes Markov model of MAC protocols for WSNs and IoT healthcare. Section 5 presents the extended performance model of LCX-MAC, and Section 5 evaluates the performance of LCX-MAC with that of X-MAC/BEB and X-MAC. Finally, Section 6 concludes the paper.

2. Related Work

For better performance at MAC layer, there are many different extensions of X-MAC proposed such as DDC MAC [14], CL-MAC [15], EX-MAC [16], (VT) [17], RIX-MAC [18], TRIXMAC [19], and LA-MAC [20].

Dynamic Duty Cycle MAC (DDC-MAC) protocol is proposed for the IoT environment which adjusts the duty cycle (DC) ratio of the receiver node and the contention window (CW) size of the sender node according to the traffic congestion for various devices in the IoT. DDC-MAC only reduces transmission delay and energy consumption. Our proposed LCX-MAC not only decreases delay and energy consumption but also increases the throughput.

CLMAC was proposed for decreasing delay in multihop networks; CLMAC uses a synchronous function by including the receiver node address and rendezvous (RDV) point in the short preamble. All the neighbour nodes around the sender nodes should wake up which is shown by RDV point. All the neighbour nodes, including the receiver nodes after reading the preamble, schedule their next wakeup time according to the RDV point. To mitigate collision at RDV point, CL-MAC uses a backoff algorithm with constant CW. CL-MAC did not give any results on how the backoff algorithm has effects on throughput, latency, and energy consumptions.

EX-MAC also wants to reduce latency to save energy in multi-hop networks and IoT health by using reservation requests to synchronize middle nodes between a source and destination. However, EX-MAC has no mechanism for the collisions when the network is overloaded.

Virtual tunnel (VT) was proposed to reduce the end-to-end delay in multihop WSNs and IoT health by implementing synchronization at the neighbour's duty cycle (SND) mechanism. For this, VT inserts the schedule information in each preamble, so each node knows the next wakeup time of its neighbours. VT also does not provide a mechanism for the collisions problem.

There are receiver-initiated MAC protocols, RIX-MAC and TRIX-MAC. Both have the same working mechanism, by which sender nodes adjust wakeup according to their receiver wakeup time by early ACK. Both protocols clearly include the wakeup time of the receiver nodes by including the wakeup time into early ACK. However, it is needed that the nodes periodically synchronize clock time of nodes, which is very difficult in large WSNs and IoT healthcare has many nodes, giving clock drift problem.

LA-MAC lists sender nodes to wake up depending on the priority order of their corresponding receiver nodes wakeup time. LA-MAC also decreases the delay by allowing senders nodes to transmit more than one frame to their receiver. However, the X-MAC allows only two senders to send their data frames in one cycle when the destination node is the same receiver.

LA-MAC does not provide any mechanism on how it will work under overloaded WSNs to monitor collision.

There is another category of MAC protocols recently used for WSNs known as Quorum-system-based MAC protocols. QTSAC [21] MAC protocol is introduced for minimizing the latency and energy consumptions in WSNs. QTSAC is different from earlier Quorum based MAC protocols using two advanced features. First, it used more QTSs (Quorum time slots) for the nodes far from the sink node; second, QTSs are allocated only, when data is transmitted. However, using more number of QTSs will increase energy consumptions.

All these extensions of X-MAC protocol are proposed for wireless sensor networks. Still, our local coordination MAC with BEB and LCX-MAC not only is suitable for WSNs but also covers IoT healthcare solutions in which the data are very critical and time-sensitive.

For securing end-to-end communication between the source and destination nodes in WSNs, different approaches are used to identify the malicious nodes in the network. Security Disjoint Routing-Based Verified Message (SDRVM) [22] is proposed with different features. The first two sets are created based on remaining energy of a node, a data and v-message set and the second is based on the remaining energy; the node records the ID in the data packet with a specific probability. Third, the duty cycle of a node is adjusted and the energy of the node is divided into three levels. In a dataset, the duty cycle of the node is the longest and the node not in any dataset is the shortest duty cycle. Fourth, if a node has sufficient energy, then data is transmitted several times, and the v-message which is stored in the node is transmitted to the final destination node. However, using these different schemas will increase the energy consumptions of nodes.

3. LCX-MAC (Local Coordination X-MAC)

Figure 1 presents the flow chart of the LCX-MAC algorithm of the sender node. When a sender node wakes up and has some data to send, the node executes *preamble backoff* by randomly drawing an integer number i from the minimum CW (contention window) W_0 of $(0, W_0 - 1)$ and holds the transmission by i number time slots.

After counting to zero of the backoff timer and clear channel assessment (CCA), when the medium is free, then the sender starts transmitting short preambles and waits to receive ACK at the end of each preamble. When the medium is free, then the sender starts transmitting short preambles and waits to receive ACK at the end of each preamble. Once the sender node got ACK, it sends one data frame and goes to sleep mode. If the sender node does not get an ACK for the whole time period due to the collision, the backoff counter k increments. LCX-MAC tries to retransmit at the next cycle until either k reaches a predetermined maximum or the transmission succeeds. When the sender node receives early ACK, so the receiver node sends its wakeup time to its corresponding sender node. The sender node adjusts its

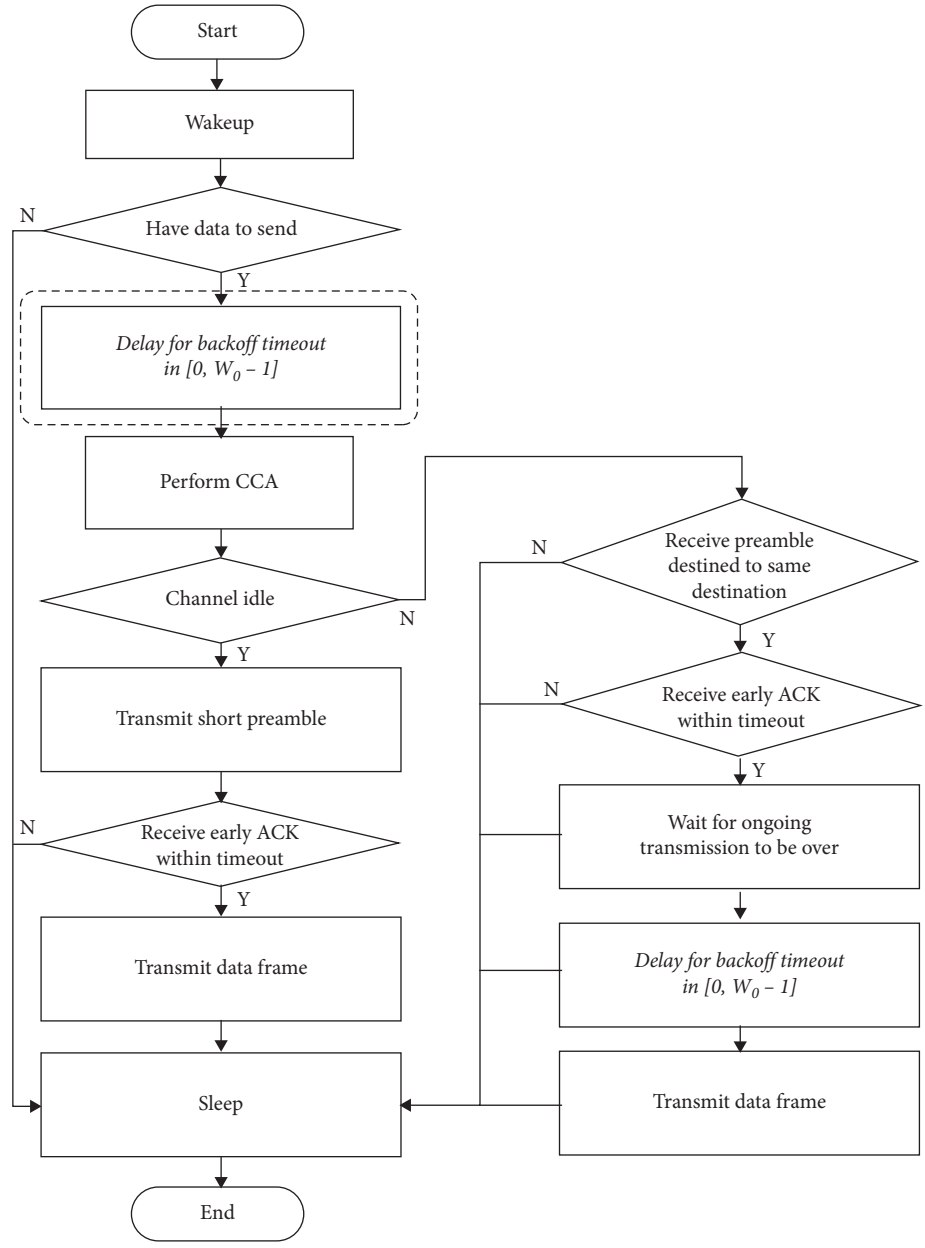


Figure 1: Sender side flow chart of LCX-MAC algorithm.

wakeup time according to its receiver node. At the end of the successful transmission, k decremented, unless $k = 0$.

When the medium is busy after counting to zero of the backoff timer, the sender node stays awake and receives the other node preamble; when it comes to knowing that destination address is the same, it wants to send data, so it stays awake for the current transmission to be finished. At the end of this current transmission, the other sender nodes use constant contention window backoff to avoid the collision because maybe more than one sender is waiting to send its data frame.

The behaviour of X-MAC/BEB and LCX-MAC is shown in Figure 2. Two sender nodes, S_1 and S_2, competing to win the channel send their data frames to their receiver R_1 and R_2, respectively. As sender node S_1 chooses by chance the randomly smaller backoff timer than S_2, sender node S_1 transmits three short preambles at the expiration of its backoff timer. As the sender node S_2 detects the medium busy at the expiration of its backoff timer, so it holds to send its short preamble to the next cycle. In the next duty cycle, sender node S_2 restarts its BEB algorithm. As the sender node S_2 is the only sender as shown in Figure 2, so sender node S_2 sends two short preambles at the expiration of its backoff timer. As for LCX-MAC, after exchanging wakeup information between the sender and the corresponding receiver, the sender sends only one short preamble, which results in saving energy and reducing delay.

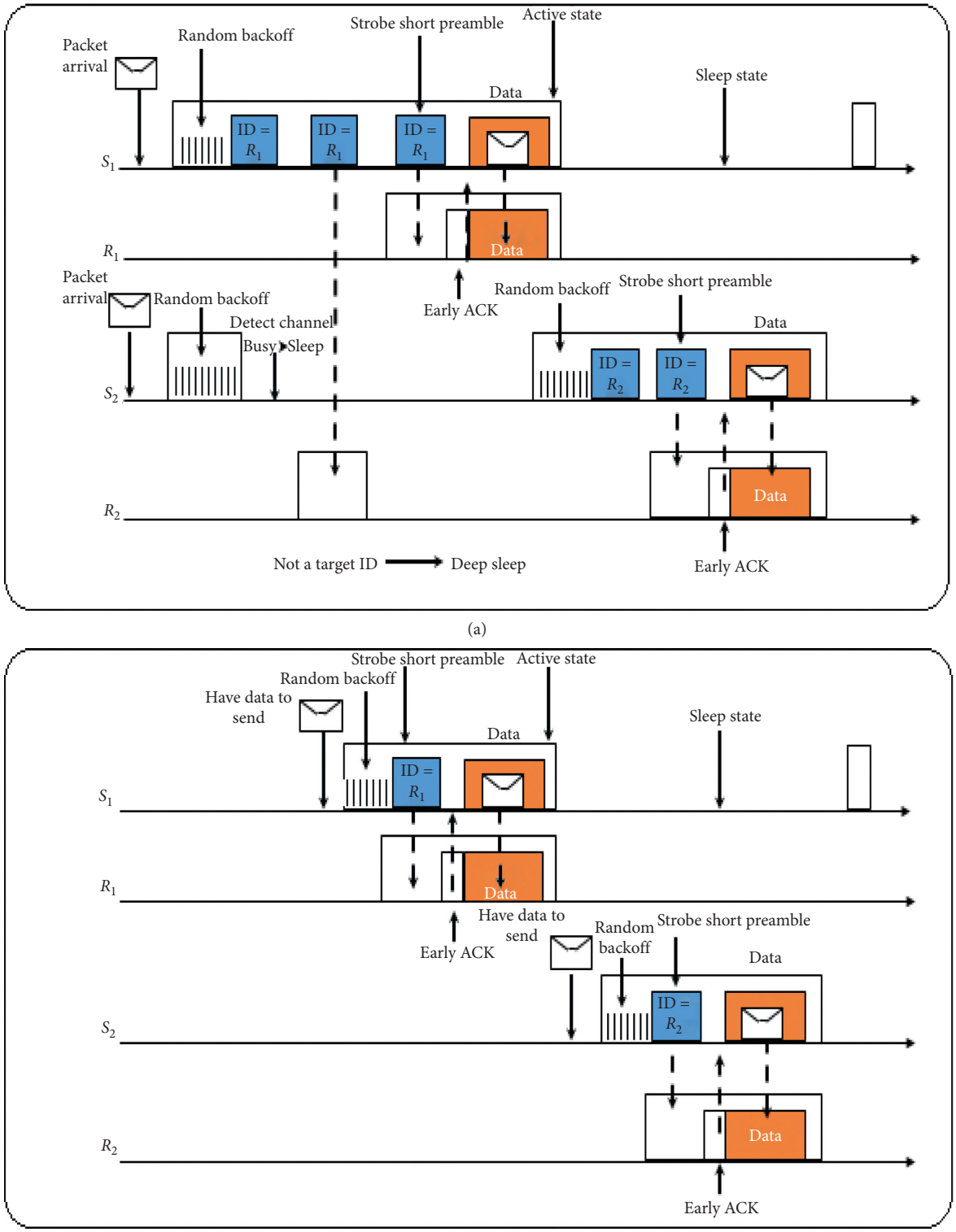


Figure 2: Comparing (a) X-MAC/BEB and (b) LCX-MAC in time domain.

4. Markov Model for Duty Cycle MAC Protocol

To investigate the performance of the MAC protocols with duty cycle behaviour, the authors [23] proposed a general mechanism consisting of two models: the first model for the duty cycle behaviour and the second model for the behaviour specific to any individual MAC protocol. The first Markov model is built according to the number of data frames stored in the buffer that reaches from the upper layer or set out to the underlying channel. The second model is built according to

the medium access behaviour of an individual MAC protocol to send data frames. Using these two models, we will get two broad equations with two unknown variables.

The first Markov model of duty cycle MAC protocols [23] describing transitions among states of a node is shown in Figure 3. A node's buffer state is represented as a number in a circle that denotes the number of frames buffered in the queue in MAC protocols. Events shown by arrows correspond to either a frame's departure to the link or a frame's arrival from the upper layer, respectively. In all the allowed transitions, this model makes some assumptions that only one frame will be transmitted in one cycle, while more than one frame can arrive within one cycle. Table 1 contains all the important notations.

The Markov model of a node queue has three types of probabilities. Probability of generating i-frame in a cycle is A_i, stationary probability of empty queue state is π_0, and a node can only transmit one data frame per cycle with probability p. Equation (1) is a set of equations, expressing $P_{i,j}$, a state transition probability, with A_i and p, where i and j are the number of frames in node queue in the earlier cycle and the number of frames in the present cycle, respectively. The first two equations in (1) described the transition from an empty queue to nonempty queue depending on the number of frames i taking place, where i-frames arrive in one cycle. For more details, refer to [23].

$$\begin{cases} P_{0,i} = A_i, & i = 0, 1, \ldots, Q-1, \\ P_{0,Q} = A_{\geq Q}, & \\ P_{i,i-1} = p \cdot A_0, & i = 1, \ldots, Q, \\ P_{i,j} = p \cdot A_{j-i+1} + (1-p) \cdot A_{j-1}, & i = 1, \ldots, Q,\ j = i, \ldots, Q-1, \\ P_{i,Q} = p \cdot A_{\geq Q-i+1} + (1-p) \cdot A_{\geq Q-i}, & i = 1, \ldots, Q, \\ P_{i,j} = 0, & i = 2, \ldots, Q,\ j = 0, \ldots, i-2. \end{cases} \tag{1}$$

The performance model in [23] builds (2) and (3) by relating $\pi = (\pi_0, \pi_1, \pi_2, \ldots, \pi_Q)$, the unique stationary distribution, and P the transition matrix. In (2) π_i is the stationary probability that a node has i-frames.

$$\sum_{s_i \in \widehat{S}} \pi_i = 1. \tag{2}$$

Equation (3) shows a condition of stable node. By knowing the data frame arrival information, we can combine three equations, (1)–(3), to remove $P_{i,j}$ and π_i by expressing only with A_i and p.

$$\pi P = \pi. \tag{3}$$

If the data frame arrival rate is known, then the equation for A_i, as with the Poisson process (4), is used to express π_0 with a function of p [23].

$$\pi_0 = f(p). \tag{4}$$

In (4), p is unknown variable for we required another function given in (5) which relates p to π_0. For (5), the second model for a MAC protocol is needed since p depends on the medium access behaviour. In the next section, we derived (5) involving p and π_0 according to the behaviour of LCX-MAC.

$$p = f(\pi_0). \tag{5}$$

5. Statistical Analysis

5.1. Throughput of LCX-MAC. To calculate the throughput of the LCX-MAC, (6), which is used in [23], is the ratio of the total data successfully transmitted in one cycle T to total cycle length T.

In (6), S is the size of a data frame, π_0 is the probability where a node have no frame in the queue, and P_{Succ} is for each node successfully transmitting a data frame.

$$\text{THR} = \frac{N \cdot (1-\pi_0) \cdot P_{\text{Succ}} \cdot S}{T}. \tag{6}$$

The only difference between X-MAC/BEB and LCX-MAC is that X-MAC/BEB on average sends $T/2$ time of the one cycle short preamble, while LCX-MAC only sends one short preamble.

Equation for X-MAC/BEB is as follows:

$$E_{\text{busy}}(W_k) = \sum_{n=0}^{\infty} \sum_{t_{\text{fr}}=0}^{T-1} \left(\left(\frac{T}{2} + t_{\text{Data}} \right) \cdot P_{\text{busy}}^{\text{succ}}(nT, t_{\text{fr}}, W_k) + T \cdot P_{\text{busy}}^{\text{coll}}(nT, n_{\text{fr}}, W_k) \right). \tag{7}$$

Equation for LCX-MAC is as follows:

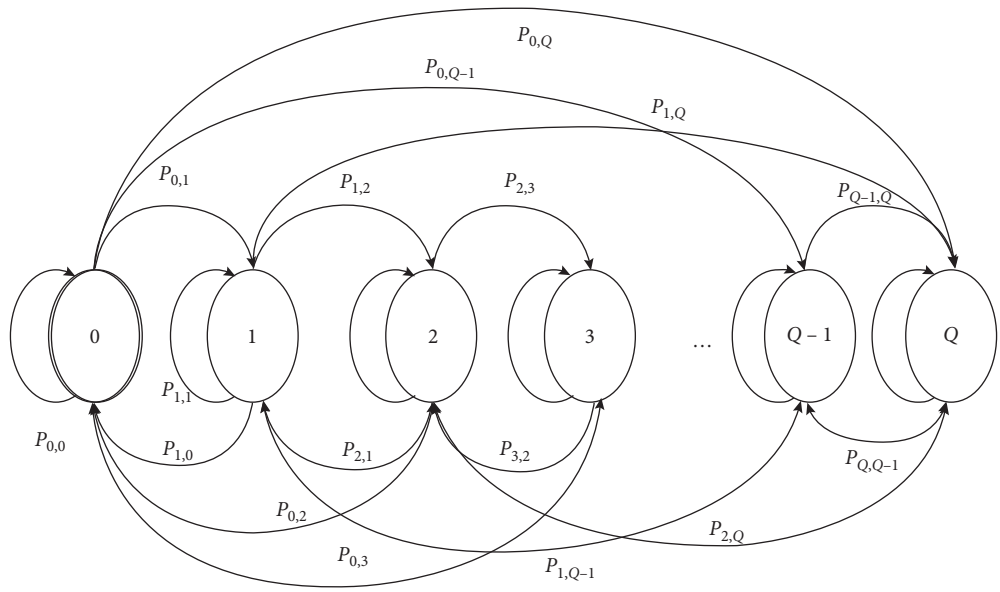


Figure 3: Markov model of a node queue with duty cycle.

Table 1: Notations.

Symbol	Description
N	Number of nodes
N_{ac}	Nodes wake up during T_a
n_i	Nodes wake up at the i^{th} time slot
a_i	Nodes have data to send
T	One cycle length
T_a	Time nodes wakeup can affect transmission at t
T_{un}	Time nodes cannot affect transmission at t
Q	Queue size
S	One data frame size
λ	Arrival rate of data frame
W_0	Initial CW size
W_m	Maximum CW size
m	Maximum backoff stages
τ	One time-slot duration
π_0	Empty queue stationary probability
p	Node transmission probability $p = P_{Succ} + P_{Coll}$
π_i	State I stationary probability
A_k	k frames are created in time T with probability $A_k = (e^{-\lambda T}(\lambda T)^k/k!)$
$A_{\geq k}$	No less than k frames are created in time T with probability $A_{\geq k} = 1 - \sum_{i=0}^{k-1} A_i$

$$E_{busy}(W_k) = \sum_{n=0}^{\infty} \sum_{t_{fr}=0}^{T-1} \Big(\big(t_{pre} + t_{Data}\big) \cdot P_{busy}^{succ}(nT, t_{fr}, W_k) + T \cdot P_{busy}^{coll}(nT, n_{fr}, W_k) \Big). \tag{8}$$

5.2. Delay of LCX-MAC. Equation (9) is used to calculate the time a frame generation at sender to its successful transfer at receiver is divided in three different types of delays, D_Q queuing delay, D_{CW} random backoff delay, and D_C contention delay.

Table 2: Network parameters.

Symbol	Value
Bitrate	250 kbps
T	20 μs
Λ	1 frame/s
T_{Active}	15 ms
t_{ACK}	1 ms
t_{pre}	3 ms
t_{DATA}	5 ms
T	50–300 ms
Q	10
S	50 bytes
Txp	59.1 mW
Rxp	52.2 mW

Equation for X-MAC/BEB is as follows:

$$D = D_Q + D_{CW} + D_C. \tag{9}$$

Equation for LCX-MAC is as follows:

$$D = D_Q + D_{CW}. \tag{10}$$

As in LCX-MAC, the sender node adjusts its wakeup time to its corresponding receiver, and there is no contention delay.

5.3. Power Consumption of LCX-MAC. Equation (11) is used to find the total energy used per cycle by each node and can be expressed as the addition of energies a node can spend at five different states. E_1 is a node that successfully transmits a data frame, E_2 is a node that successfully receives a data frame, E_3 is a node that has collision when transmitting a data frame, E_4 is a node that is a potential receiver to a data frame but failed because of

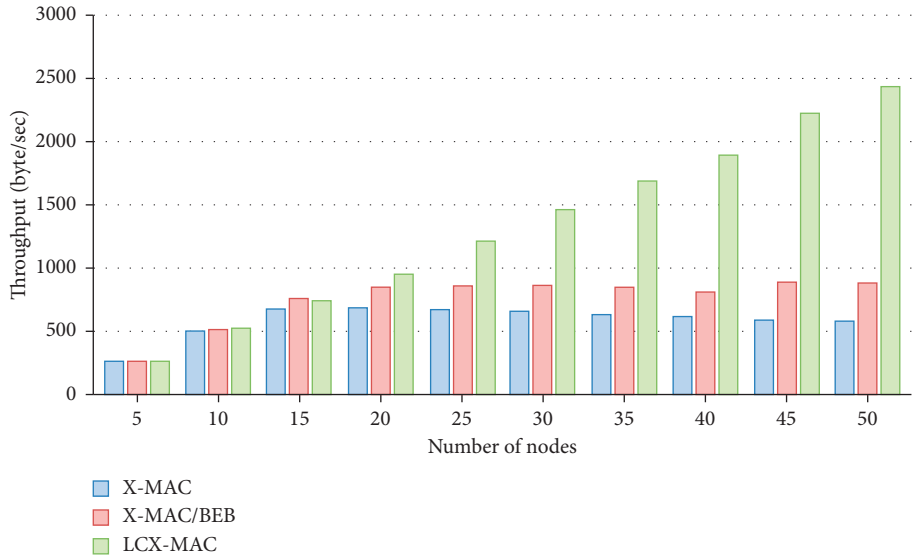


Figure 4: Throughput (bytes/sec) versus number of nodes. Comparing X-MAC, X-MAC/BEB, and LCX-MAC throughout.

collision, and, finally, E_5 stays awake for the entire active time. As the receiver node of LCX-MAC works as X-MAC, so the states E_2, E_4, and E_5 have the same energy consumptions from [23].

$$E = E_1 + E_2 + E_3 + E_4 + E_5. \tag{11}$$

The only difference between X-MAC/BEB and LCX-MAC that X-MAC/BEB on average $T/2$ time of the one cycle energy is consumed in sending short preamble and receiving early ACK, while LCX-MAC only sends one short preamble.

Equation for X-MAC/BEB is as follows:

$$E_1 = (1-\pi_0)\cdot P_{\text{Succ}}(W_k)\cdot\tau\cdot\left(\left(\frac{1}{m+1}\sum_{k=0}^{m}\left(\frac{W_k}{2}\right)\right)\cdot \text{rxp} + \frac{T}{2}\cdot\left(\frac{t_{\text{pre}}}{\left(t_{\text{pre}}+t_{\text{ACK}}\right)}\right)\cdot \text{txp} + \frac{T}{2}\cdot\left(\frac{t_{\text{ACK}}}{\left(t_{\text{pre}}+t_{\text{ACK}}\right)}\right)\cdot \text{rxp} + t_{\text{DATA}}\cdot \text{txp}\right). \tag{12}$$

Equation for LCX-MAC is as follows:

$$E_1 = (1-\pi_0)\cdot P_{\text{Succ}}(W_k)\cdot\tau\cdot\left(\left(\frac{1}{m+1}\sum_{k=0}^{m}\left(\frac{W_k}{2}\right)\right)\cdot \text{rxp} + \left(\frac{t_{\text{pre}}}{\left(t_{\text{pre}}+t_{\text{ACK}}\right)}\right)\cdot \text{txp} + \left(\frac{t_{\text{ACK}}}{\left(t_{\text{pre}}+t_{\text{ACK}}\right)}\right)\cdot \text{rxp} + t_{\text{DATA}}\cdot \text{txp}\right). \tag{13}$$

6. Evaluations

This section evaluates the throughput, delay, and per-frame energy consumption of LCX-MAC, X-MAC/BEB, and X-MAC by their analytical models. Table 2 contains the important parameters and values used in the mathematical calculations.

Figure 4 shows that LCX-MAC, represented by green bars, outperforms X-MAC/BEB shown by red bars and X-MAC as indicated by blue bars as the number of nodes increased.

Figure 5 shows that the delay of LCX-MAC is minimal because the only small amount of time is used to send one short preamble. In contrast, the X-MAC/BEB and X-MAC send on average half of cycle ($T/2$) time short preamble; that is why the delay of X-MAC/BEB and X-MAC went up when the number of nodes increased.

Figure 6 shows that $E_{\text{frame/node/s}}$ of LCX-MAC, X-MAC/BEB, and X-MAC falls as the number of nodes in the network increased because nodes have less chance to send data frames due to busy medium, leading to termination of data transmission, so the nodes stay in sleep mode most of the time.

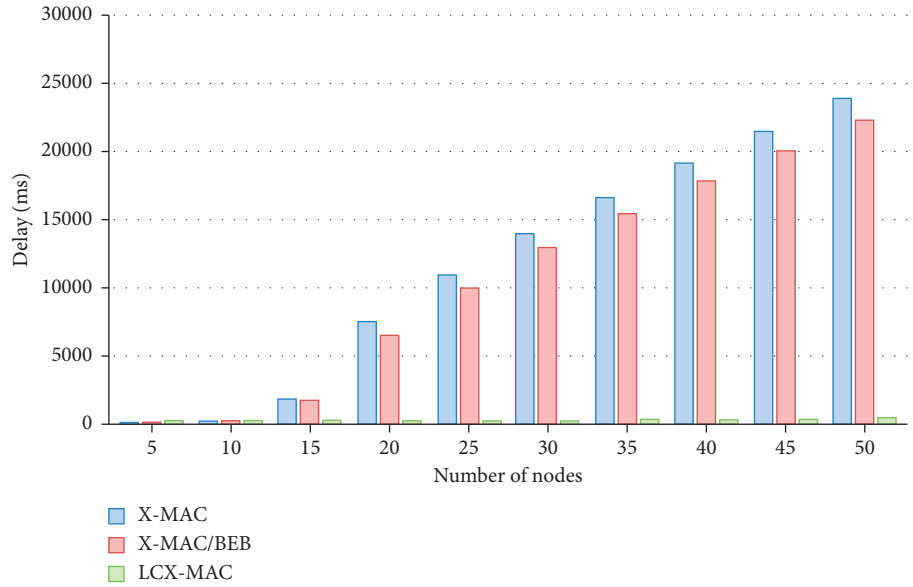


Figure 5: Delay (ms) versus number of nodes. Comparing X-MAC, X-MAC/BEB, and LCX-MAC delay.

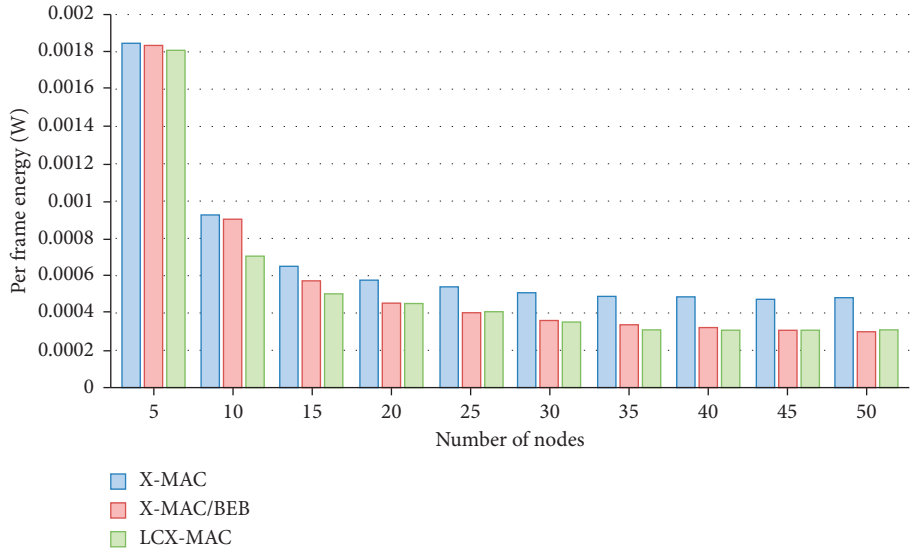


Figure 6: Energy ($E_{\text{frame/node/s}}$) versus number of nodes. Comparing X-MAC, X-MAC/BEB, and LCX-MAC per frame energy.

7. Conclusions and Future Work

We proposed LCX-MAC for IoT healthcare and WSNs, in which the sender nodes coordinate their wakeup time with their corresponding receivers to increase throughput, decrease delay, and reduce energy consumptions. On average, in half of the cycle of X-MAC and X-MAC/BEB, each node sends short preambles and listens for early ACK before sending the actual data. We also extend the model of X-MAC/BEB. The results show that LCX-MAC increases the performance of X-MAC/BEB when nodes in IoT and WSNs become populated. The throughput of LCX-MAC is increased by 130% compared to the throughput of X-MAC/BEB for the number of nodes 40. LCX-MAC is a kind of synchronous MAC protocol with much low overhead compared to the synchronous MAC protocols which gave minimal delay so that we will compare LCX-MAC with synchronous MAC protocols in our future works. The energy consumption of X-MAC/BEB and LCX-MAC is decreased by 40% compared to the energy consumption of X-MAC.

This paper presents single hope network topology. In the future, we will extend to multihop network topology as the nodes in the IoT environments are multihop. We will also do detail simulation works to validate the analytical model results in network simulator version 2 (ns2) [24].

Data Availability

The data used to support the findings of this study are included within the manuscript.

Conflicts of Interest

All authors declare that they have no conflicts of interest relevant to this research study in any form.

Acknowledgments

This research work was supported by the National Science Foundation of China (U1636203) and the National Key Research and Development Program of China (No. 2016YFB0801200).

References

[1] L. Fuhong, L. Qian, Z. Xianwei, C. Yueyun, and H. Daochao, "Cooperative differential game for model energy-bandwidth efficiency tradeoff in the internet of things," *China Communications*, vol. 11, no. 1, pp. 92–102, 2014.

[2] S. Huang, A. Liu, S. Zhang, T. Wang, and N. Xiong, "BD-VTE: a novel baseline data based verifiable trust evaluation scheme for smart network systems," *IEEE Transactions on Network Science and Engineering*, 2020.

[3] Y. Wei, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, pp. 1567–1576, New York, NY, USA, June 2002.

[4] T. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '02)*, pp. 171–180, Los Angeles, CA, USA, November 2003.

[5] S. Du, K. Amit, and D. B. Johnson, "RMAC: a routing-enhanced duty-cycle MAC protocol for wireless sensor networks," in *Proceedings of the 26th Annual IEEE Conference on Computer Communications (INFOCOM 2007)*, pp. 1478–1486, Anchorage, AK, USA, May 2007.

[6] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: a low latency, energy-efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proceedings of the Ninth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2008)*, pp. 53–62, Hong Kong, China, May 2008.

[7] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle mac with scheduled channel polling," in *Proceedings of the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 321–333, Boulder, CO, USA, November 2006.

[8] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 95–107, Baltimore, MD, USA, November 2004.

[9] Amre El-Hoiydi and J.-D. Decotignie, "WiseMAC: an ultra low power MAC protocol for multi-hop wireless sensor networks," in *Proceedings of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004)*, pp. 18–31, Lecture Notes in Computer Science LNCS 3121, Turku, Finland, July 2004.

[10] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 307–320, Boulder, CO, USA, November 2006.

[11] A. Bhatia and R. C. Hansdah, "A distributed TDMA slot scheduling algorithm for spatially correlated contention in WSNs," in *Proceedings of the IEEE International Conference on Advanced Information Networking and Applications Workshops (AINA)*, pp. 377–384, Barcelona, Spain, March 2013.

[12] A. Ullah, G. Kim, and J. S. Ahn, "Performance analysis of X-MAC protocol with collision avoidance algorithm," in *Proceedings of the 32nd Annual IEEE Conference on Computer Communications (INFOCOM 2013)*, pp. 164–212, Student Poster Session, Turin, Italy, April 2013.

[13] A. Ullah and J. S. Ahn, "Performance evaluation of X-MAC/BEB protocol for wireless sensor networks," *Journal of Communications and Networks (JCN)*, vol. 18, no. 5, pp. 857–869, 2016.

[14] G. Kim, J.-G. Kang, and M. Rim, "Dynamic duty-cycle MAC protocol for IoT environments and wireless sensor networks," *MDPI Energies*, vol. 12, no. 21, pp. 1–13, 2019.

[15] A. B. Nacef, S. M. Senouci, Y. Ghamri-Doudane, and A. L. Beylot, "A cooperative low power mac protocol for WSNs," in *Proceedings of the IEEE International Conference on Communications ICC 2011*, Kyoto, Japan, June 2011.

[16] S.-h. Hong and H.-k. Kim, "A multi-hop reservation method for end-to-end latency performance improvement in asynchronous MAC-based wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1214–1220, 2009.

[17] J.-H. Lee, J.-W. Kim, and D.-S. Eom, "A delay-tolerant virtual tunnel scheme for asynchronous MAC protocols in WSN," *Wireless Personal Communications*, vol. 70, no. 2, pp. 657–675, 2013.

[18] I. Park, H. Lee, S. Kang, and D.-S. Eom, "RIX-MAC: an energy-efficient receiver-initiated wakeup MAC protocol for WSNs," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 5, pp. 1604–1617, 2014.

[19] I. Park, J. Yi, and H. Lee, "A receiver-initiated MAC protocol for wireless sensor networks based on tree topology," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 950656, 10 pages, 2015.

[20] G. Corbellini, E. C. Strinati, and A. Duda, "LA-MAC: low-latency asynchronous MAC for wireless sensor networks," in *Proceedings of the IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 380–386, Sydney, Australia, September 2012.

[21] Y. Liu, K. Ota, K. Zhang et al., "QTSAC: an energy-efficient MAC protocol for delay minimization in wireless sensor networks," *IEEE Access*, vol. 6, no. 1, pp. 8273–8291, 2018.

[22] X. Liu, A. Liu, T. Wang et al., "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *Journal of Parallel and Distributed Computing*, vol. 135, pp. 140–155, 2020.

[23] O. Yang and W. Heinzelman, "Modeling and performance analysis for duty-cycled MAC protocols with applications to S-MAC and X-MAC," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 905–921, 2012.

[24] Network Simulator, ns-2, http://www.isi.edu/nsnam/ns/.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8886877, 14 pages
https://doi.org/10.1155/2020/8886877

Review Article

Multicriteria Decision and Machine Learning Algorithms for Component Security Evaluation: Library-Based Overview

Jibin Zhang[1], **Shah Nazir**[2], **Ansheng Huang**[1] and **Abdullah Alharbi**[3]

[1]*Materials Corporation of Petro China Southwest Oil & Gasfield Company, Chengdu 610017, China*
[2]*Department of Computer Science, University of Swabi, Swabi, Pakistan*
[3]*Department of Information Technology, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia*

Correspondence should be addressed to Ansheng Huang; huangansheng@petrochina.com.cn

Received 14 June 2020; Revised 1 July 2020; Accepted 3 July 2020; Published 10 September 2020

Academic Editor: Amir Anees

Components are the significant part of a system which plays an important role in the functionality of the system. Components are the reusable part of a system which are already tested, debugged, and experienced based on the previous practices. A new system is developed based on the reusable components, as reusability of components is recommended to save time, effort, and resources as such components are already made. Security of components is a significant constituent of the system to maintain the existence of the component as well as the system to function smoothly. Component security can protect a component from illegal access and changing its contents. Considering the developments in information security, protecting the components becomes a fundamental issue. In order to tackle such issues, a comprehensive study report is needed which can help practitioners to protect their system. The current study is an endeavor to report some of the existing studies regarding component security evaluation based on multicriteria decision and machine learning algorithms in the popular searching libraries.

1. Introduction

Technology has made life easier but has exposed several security issues. Over the last few years with the development of Internet, the number of attacks has increased. Technology plays an inevitable role in human life. The Internet of Things (IoT) enables communication with different devices. The smart devices are connected to communicate, process, compute, and monitor diverse real-time scenarios. The devices are normally heterogeneous and have low memory and short power for processing. The concept of Internet of Things came with the challenges of privacy and security, as the conventional security protocol does not fit the devices of IoT. The information security of an organization is highly dependent on different types of information of the organization. Manager of the information security is not only concerned with the relevant information but also with the interdependencies among this information. Individuals, government, and organizations are facing risks of information security. These risks can be damaged at a high level in terms of breach of confidentiality of sensitive data, financial loss, and loss of integrity and availability of data which is sensitive. Security of components plays an important role in the functionality of a system to run properly. Different studies are available for the security purpose [1–5]. The algorithms of ML have shown a considerable performance in different application fields such facial recognition, text recognition, spam detection, and so on. The applications of machine learning (ML) algorithms are obvious in different domain areas [2, 5–11].

The contribution of the proposed study is to present a comprehensive report on some of the existing state-of-the-

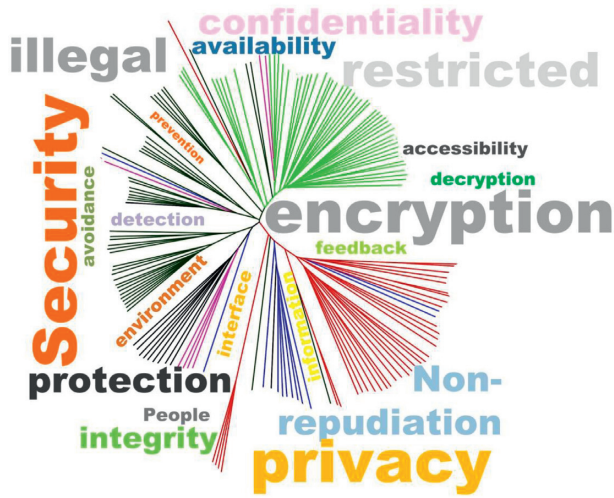


Figure 1: Relevant terminologies for security.

art research studies for component security evaluation based on multicriteria decision and machine learning algorithms. This study will support the researchers to extract the most useful insights of security to a particular domain to strengthen its existence and to avoid future hurdles.

The organization of the paper is as follows. Section 2 presents the related work to the current research, in particular to multicriteria decision and machine learning algorithm applications for component security evaluation. Section 3 briefly shows multicriteria decision and machine learning approaches to the security evaluation. Section 4 shows the library-based analysis of the existing literature from different perspectives in the most popular libraries.

2. Related Work

Different approaches are being proposed by researchers to tackle the issue of security from different perspectives. Saranya et al. [6] presented the study of comparisons of different machine learning algorithms for intrusion detection system with applications in different areas such as smart city, Interenet of Things, fog computing, big data, and so on. The KDD-CUP dataset was used to test efficiency and compared with existing available research. For the information security products of cloud computing, a test evaluation system is established [1]. The security identification has a significant role in the field like Internet of Things in smart city. Manjia Tahsien et al. [8] presented an overview of the IoT architecture with a detailed review on machine learning algorithms, significance of IoT security with diverse types of attacks. The study proposed a model of the associated information management factors for the information security of organization. Firstly, they surveyed 136 articles to identify the information security factors, and secondly, a series of interviews with 19 experts from the industry to evaluate the relevancy of these factors. In third step, a complete model was developed [3]. The authors [12] conducted a detailed survey of the state-of-the-art IoT security, deep learning, and big data technology.

Yuan and Luo [13] evaluated energy security of the Chinese provinces through analyzing the reasons and implementation of policy, with the help of MTGS and SPA-TOPSIS. Wijayarathna and Arachchilage [14] assessed the cognitive dimensions framework with the help of four security application programming interfaces, such as Bouncy Castle light weight Crypto API, Google Authentication API, OWASP Enterprise Security API, and Java Secure Socket Extension API. Wang et al. [9] presented a detailed overview of the security properties investigation of machine learning algorithms. They have analysed the security model of ML to build up a blueprint for multidisciplinary area of research. After that, the attack methods and the strategies of defense against them are discussed. The study presented an overview of the weaknesses and strengths of the available evaluation methods used for usability and security for the websites of electronic commerce (E-commerce). The evaluation models from 2000 to 2018 have been reviewed for E-commerce [4]. Many burning issues like untrustworthy information, insecure platform, malicious propagation, and illegal cheating exist. Security and trustworthiness play an important role for the communication among social interactions of sharing information and communication. Zhang et al. [15] proposed an approach for crowed assessing the security and trustworthiness of open social networks based on signaling theory.

Mao et al. [16] proposed a system for building security dependency to measure the significance of security of a system from a wide perspective of the system. The effect of small-world and power-law distribution for the degree for in- and out-degree in security dependency network was observed. Halabi and Bellaiche [17] proposed an approach for measuring performance and assessment of services of security for Cloud on the basis of set of assessment measures using Goal-Question-Metric. Cheah et al. [18] devised a systematic framework for security testing for interfaces of automotive Bluetooth and applied a tool of proof-of-concept

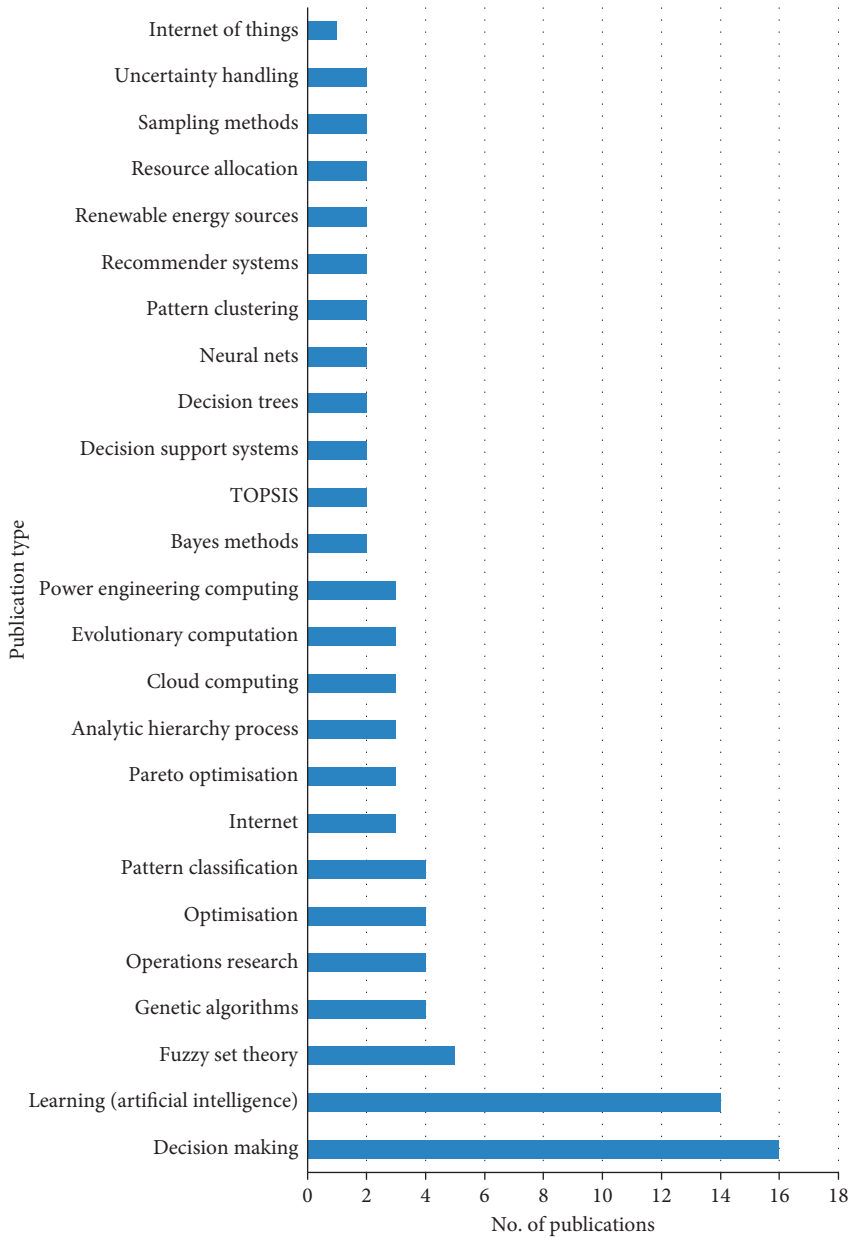


Figure 2: Publication type and total number.

to carry out testing on vehicle with the help of the proposed framework. Nazir et al. [19] proposed a methodology for evaluating the security of software components using the analytic network process. This technique works in the situation of complexity where dependencies exist among different nodes of network. Cherdantsevaet al. presented an evaluation of a reference model of information assurance and security for summarizing the information required by the information assurance and security community [20]. Jouini et al. proposed a quantitative approach to security risk for information systems which is extendable, systematic, and modular. The study aimed to effectively evaluate security threat in a comprehensive way [21]. The study considered an approach to attack of computer modeling and security assessment which is recommended to realize in advanced Security Information and Event Management (SIEM) systems. Subsorn and Limwiriyakul [23] examined the security of internet banking of 16 Australian banks for finding the shortcomings which were probably affecting the confidentiality of the bank customers. Furthermore, the study investigated 12 Thai commercial banks and compared the results with those of the previous research. Kotenko and Chechulin [24] presented a framework for security assessment and attack modelling in security information and event management system.

3. Multicriteria Decision Making and Machine Learning Algorithms for Security Evaluation

Several techniques are being practiced in the literature for security evaluation [25–27]. These techniques evaluate the security from different perspectives. A number of machine

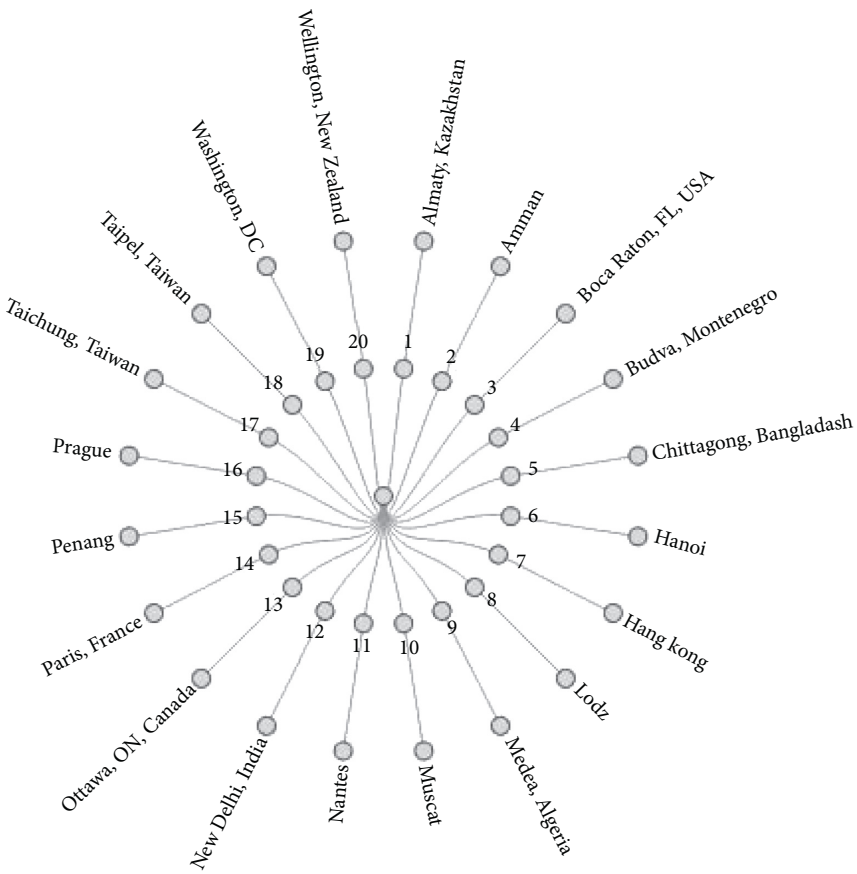


Figure 3: Places of conferences held.

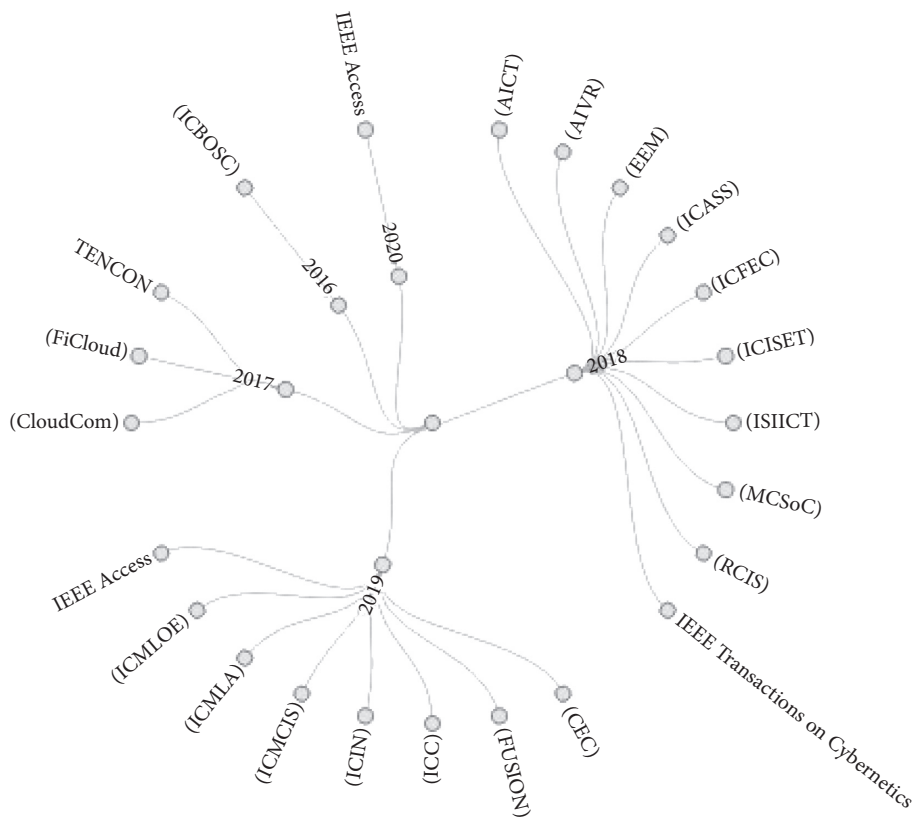


Figure 4: Name and year of conferences held.

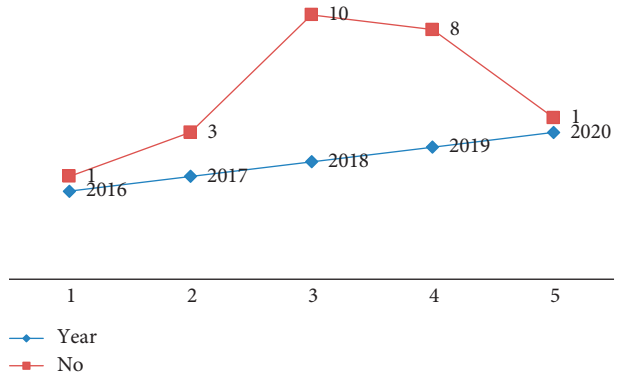


Figure 5: Total number of conferences held in the given year.

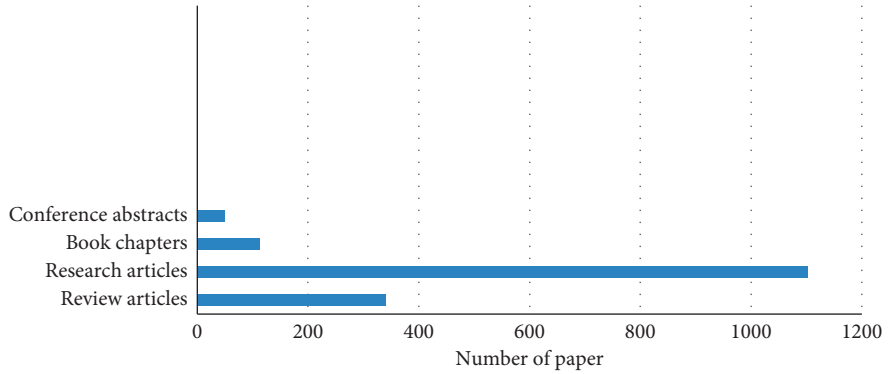


Figure 6: Article type total number of publication.

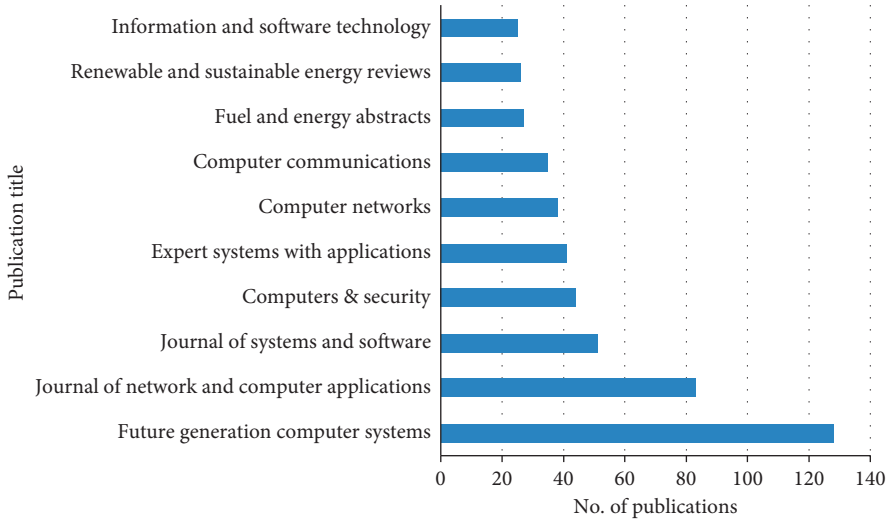


Figure 7: Publication title along with total number of publication.

learning algorithms are used for the detection of intrusion affecting the system of organizations. Shafiq et al. [7] proposed a novel framework and used BoT-IoT identification dataset and 44 features with the help of machine learning algorithm. After this, five effective machine learning algorithms are considered for the detection of anomaly and malicious traffic with performance of evaluation measures of machine learning algorithm. An approach of bijective softest and its algorithm is applied to find effectiveness of machine learning algorithm. Mohanta et al. [2] reported the technology of IoT and its applications in different areas. The security issues such as integrity, availability, and

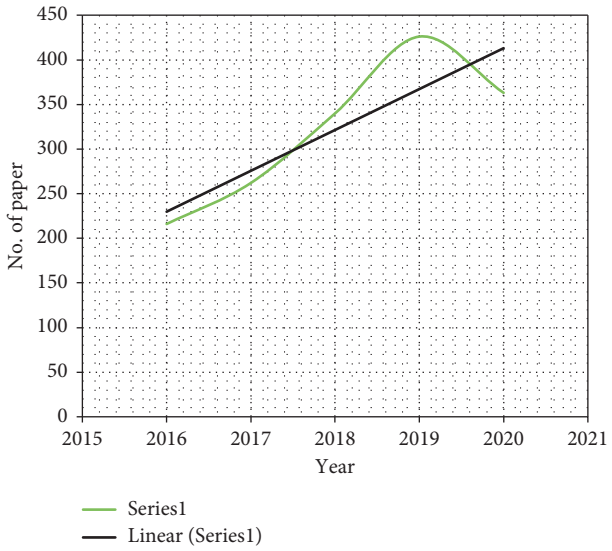


Figure 8: Total number of papers published in the given year.

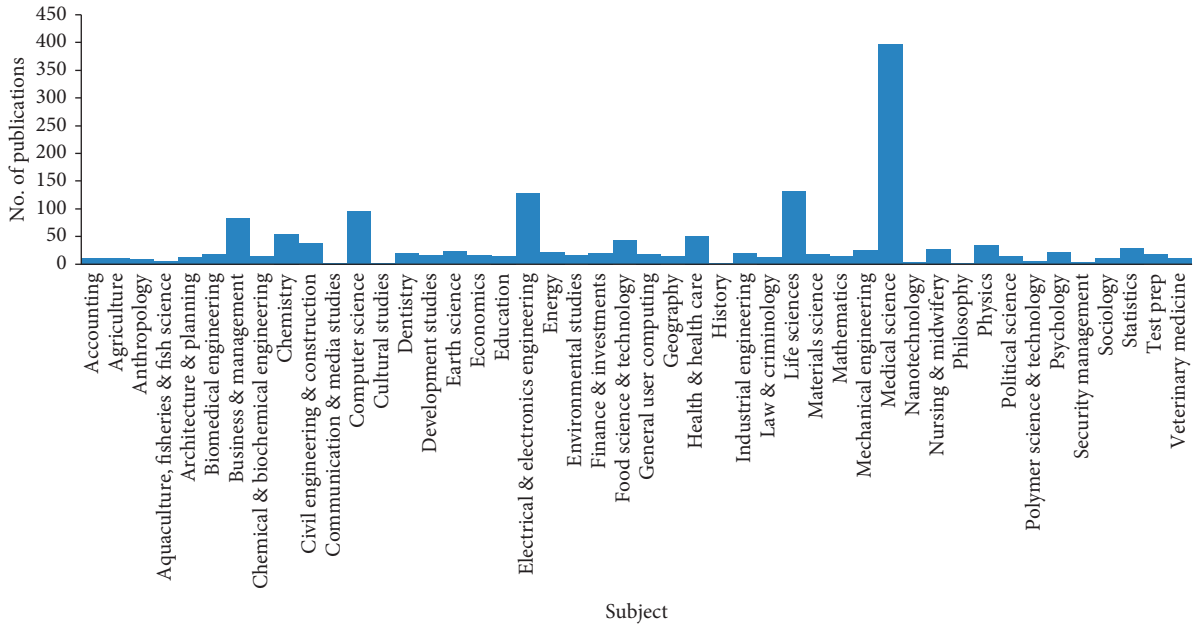


Figure 9: Subjects along with the number of publications.

confidentiality and the issues are discovered. The applications of artificial intelligence, machine learning, and Blockchain for the issues of security for IoT are studied. Marwan et al. [10] proposed a ML based approach to secure the processing of data based on cloud environment. The support vector machines and fuzzy c-means clustering were used to classify the pixels of images in an efficient way. To reduce the disclosure of medical information, the module of CloudSec into the conventional architecture of two-layered was incorporated.

Katzir and Elovic [5] presented the adversarial resilience based on supervised machine learning algorithm for detection systems. The study provides a definition of adversarial resilience with focus on system of multisensory fusion. Model robustness score was defined for evaluating the relative resilience of existing models, and then two

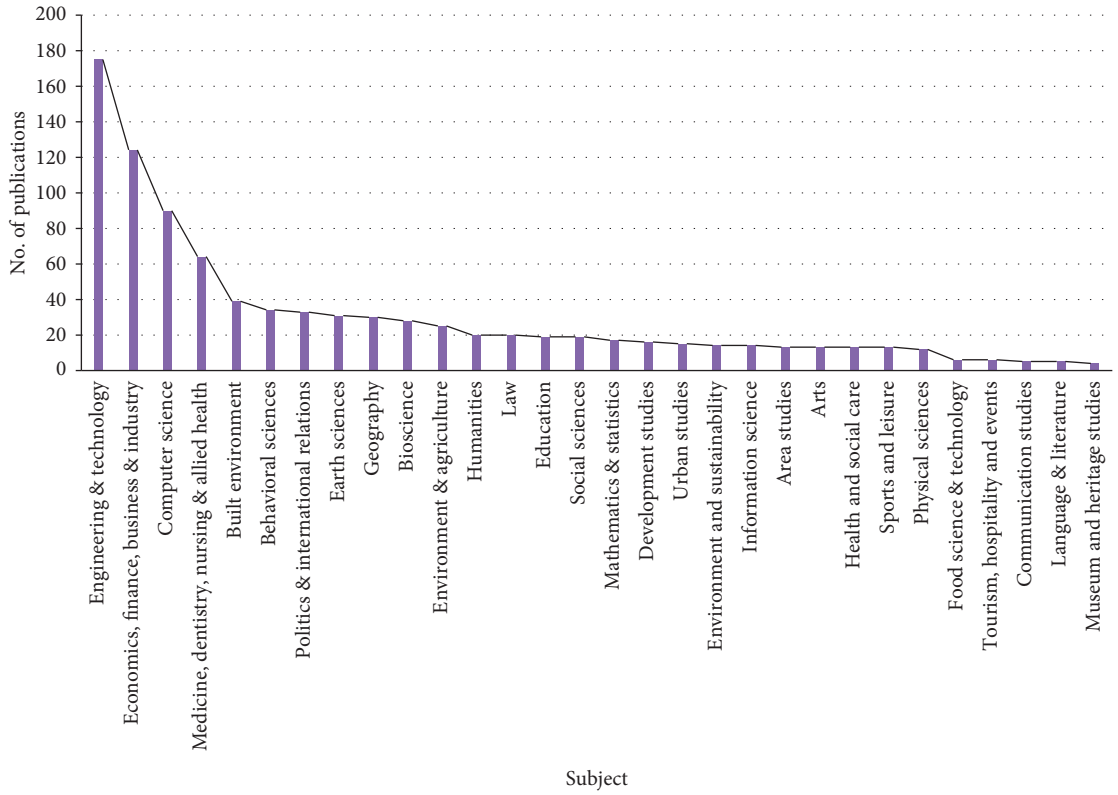


Figure 10: Subjects along with the number of publications.

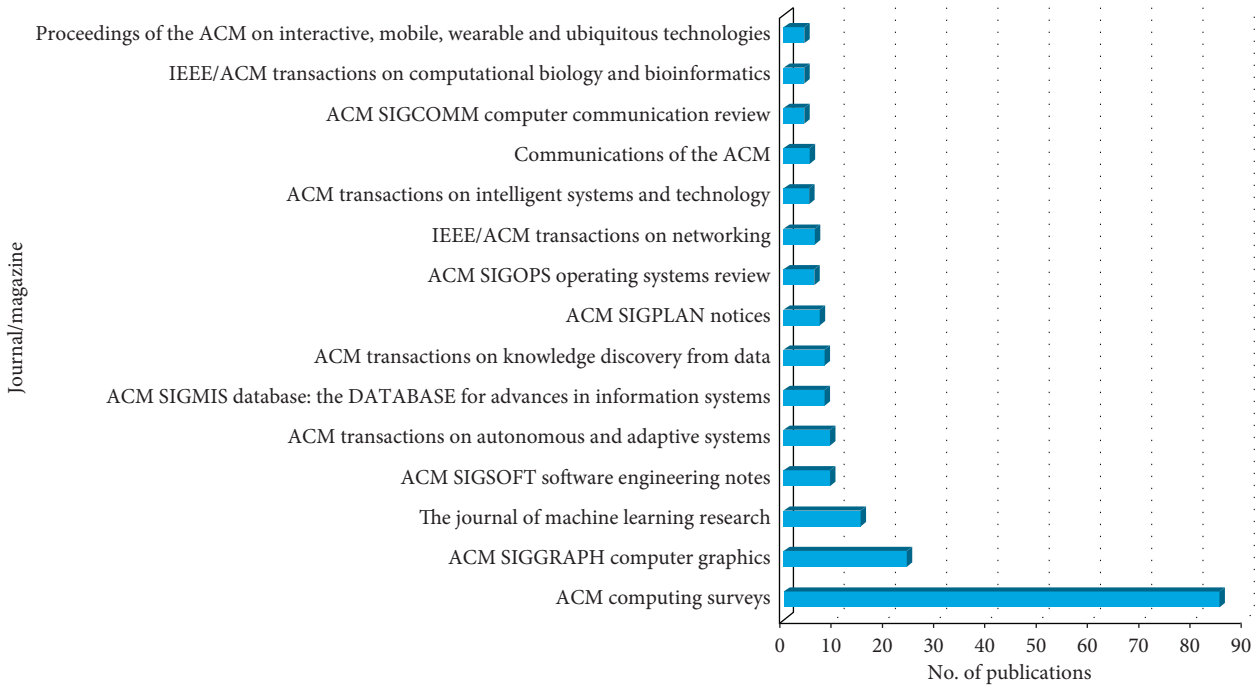


Figure 11: Journal/magazine name and number of publication.

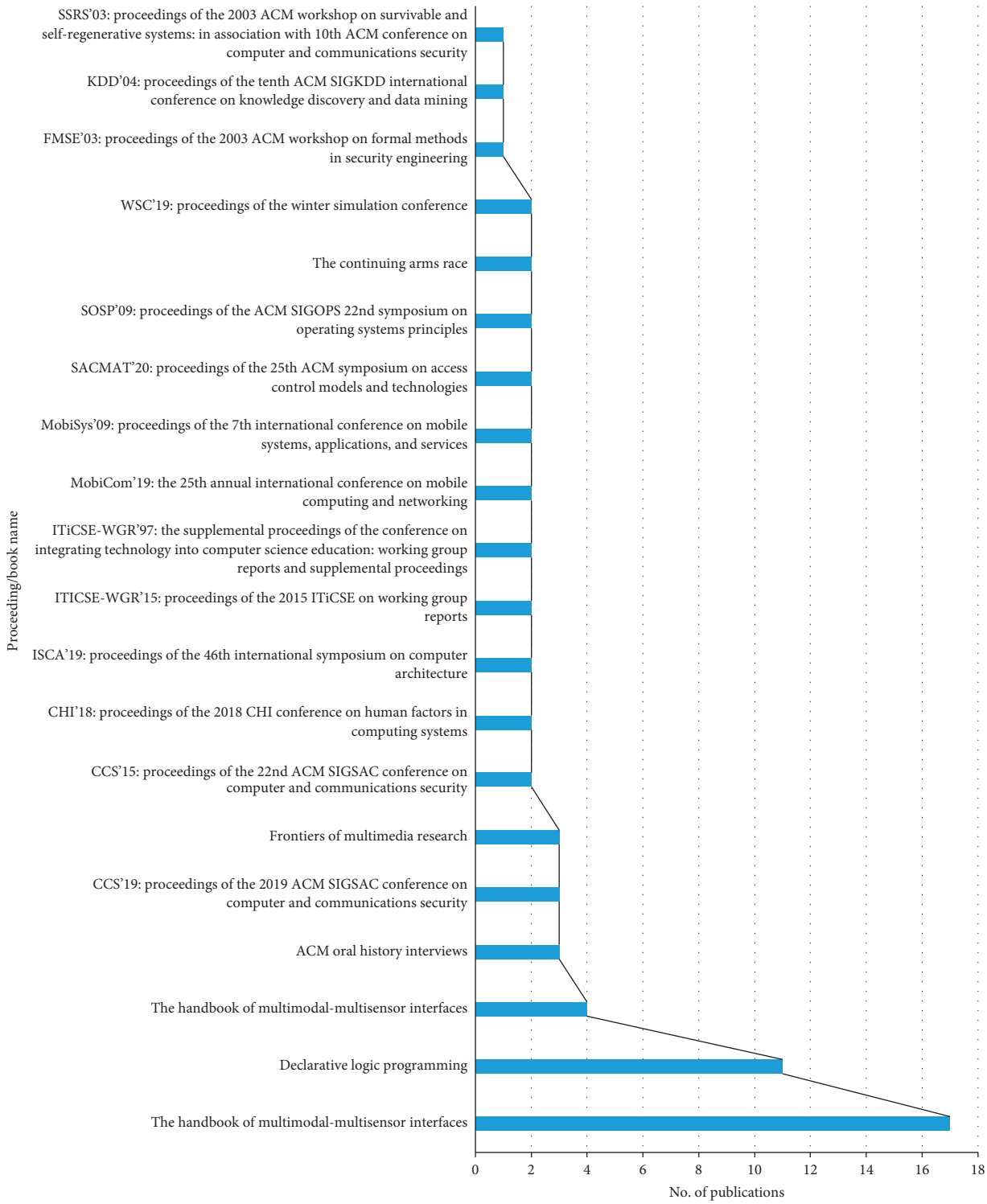


Figure 12: Proceeding/book name and number of publication.

novel feature selection algorithms for designing adversary aware classifiers were recommended. In network communication, one of the major concerns is the detection of intrusion. Different approaches are used for effective and efficient detection and prevention of intrusion and ensuring privacy and security. Four classifiers of machine learning algorithms that are, Naïve–Bayes, support vector machine, decision tree, and Random Forest using Apache Spark were used to evaluate the performance of intrusion detection in network [11]. Apart from this, several approaches exist for security evaluation such as analytic network process, analytic hierarchy process, fuzzy logic, IoT-based security evaluation, and feature-based birthmarks [19, 28, 29].

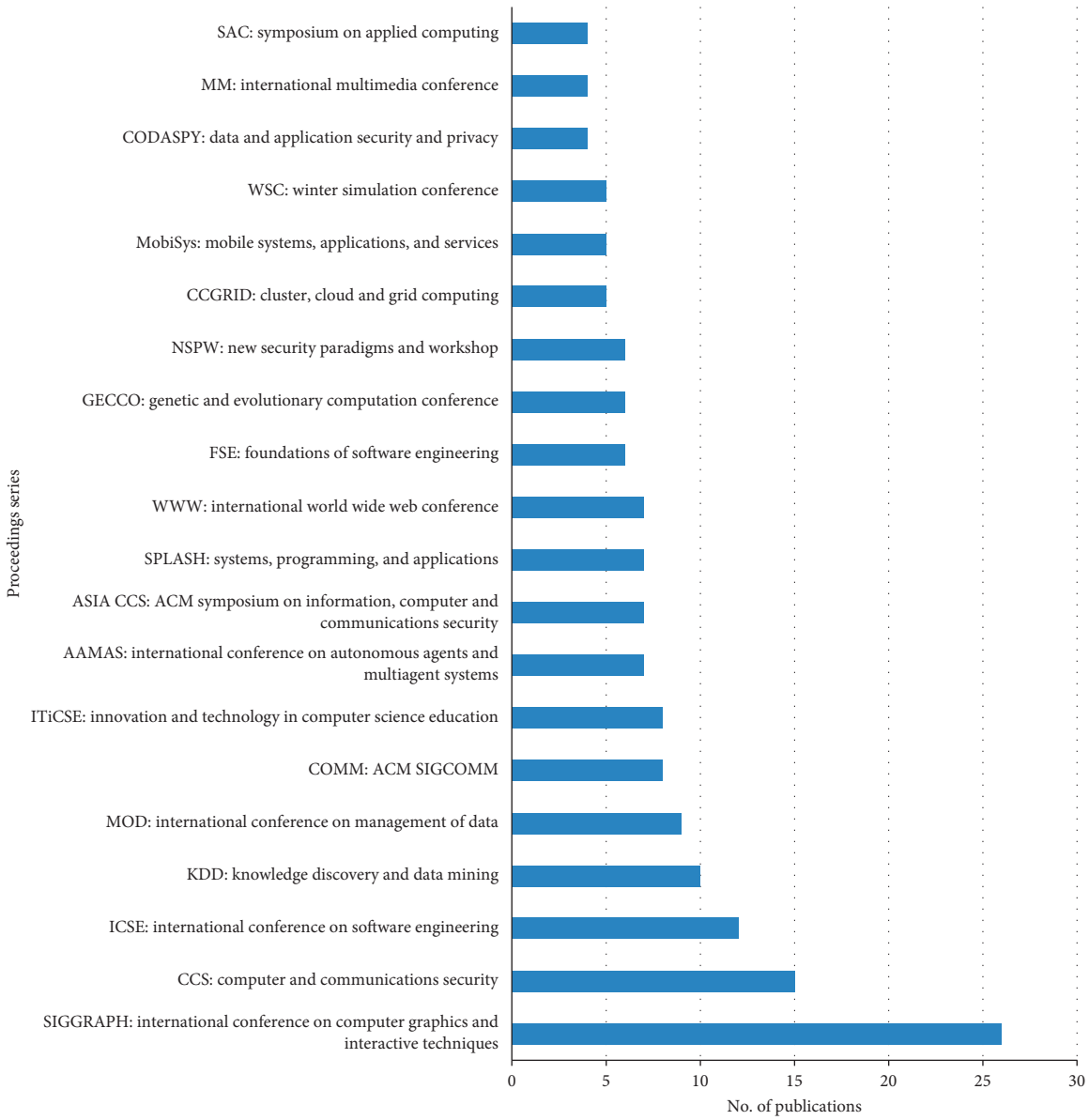


Figure 13: Proceeding series and number of publication.

4. Library-Based Search for the Existing Research

Before, data security was simply a specialized concern and specialized representatives were answerable for data security issues inside an organization. Thus, in previous years, there was a shift of paradigm from the official innovation master to the obligation of administration and a more business-centred view ensuring data security. Nowadays, security supervisors are completely capable to consider and react to data security issues. Due to the move from a specialized to an administration point of view, the examination concentration additionally changed from specialized setting to investigating the administration job. Supervisors must have the option to accept specialized dangers just as different elements like human conduct into record to take the privilege and powerful activities to moderate threats. Therefore, this examination has the reason to distinguish the key components and assess them and investigate between conditions to at last produce a thorough model to comprehend the security of data at multilevel nature and subsequently give high data security, the executive choices.

Multicriteria and machine learning algorithms plays an important role in security of information. Mostly, the security of the IoT devices is evaluated through machine learning algorithms. The purpose of this section is to identify the existing available research from different popular libraries in order to extract meaning insights for practitioners. These libraries mainly include ACM, Sciencedirect, IEEE, Springer, Wiley, and Tailor & Francis. The query was considered as collection of different words. An individual word shows more materials which is very difficult to analyze.

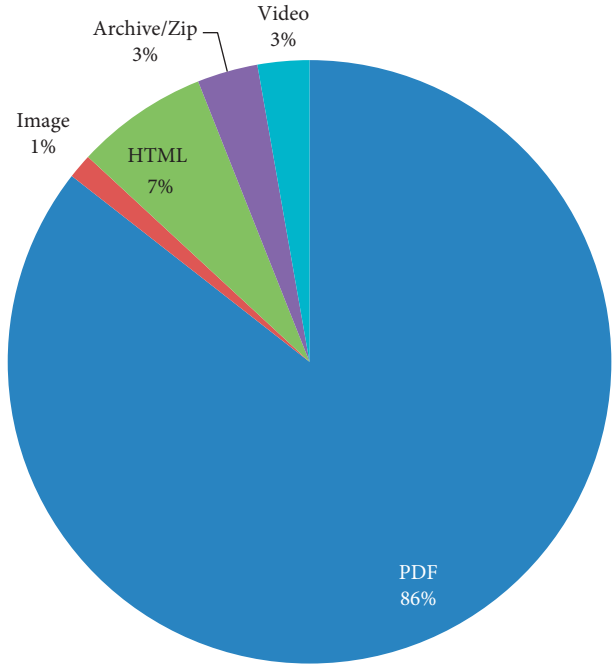


Figure 14: Media format and number of publication.

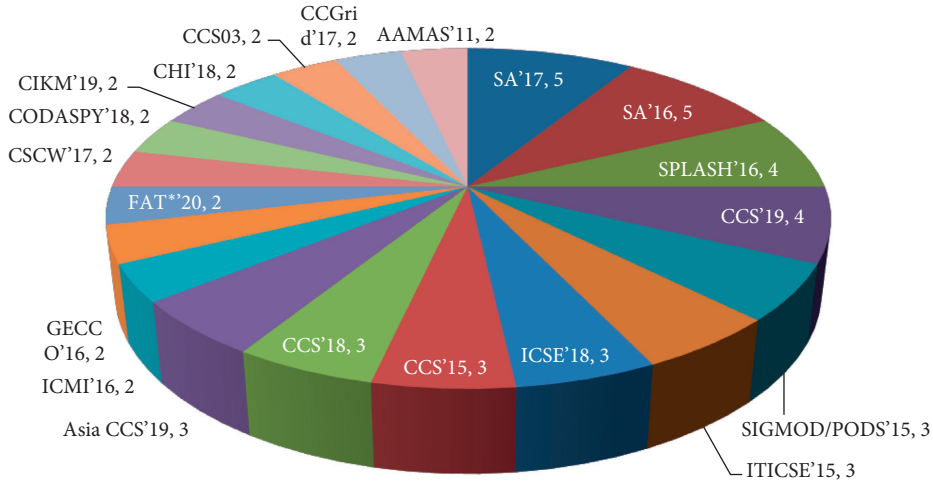


Figure 15: Conference event and number of publication.

So the query was considered as the collection of different words with the operator "AND" and "OR" to show all the relevant materials. The mentioned libraries were searched based on the following queries:

("*software component*" *OR* "*component of software*") *AND* ("*security evaluation OR security assessing*") *AND* ("*multi criteria decision*" *OR* "*multi-criteria decision*") *AND* ("*machine learning*"), *and/or* (*software component OR component of software*) *AND* (*security evaluation OR security assessing*) *AND* (*multi criteria decision OR multi-criteria decision*) *AND* (*machine learning*)

The reasons behind the two queries is that entering the first query gives less amount of materials while the second query gives huge amount of materials. The study attempts to select more articles to give more detail information to the research community. These libraries were searched from different perspectives and the details are given in the following subsections. Figure 1 shows the relevant terminologies to the security.

The following subsections briefly show the details of the search process in the selected famous libraries. The reason behind the selection of these libraries is that these are the most popular and well-known libraries. Googlescholar was not considered as there are more irrelevant materials and

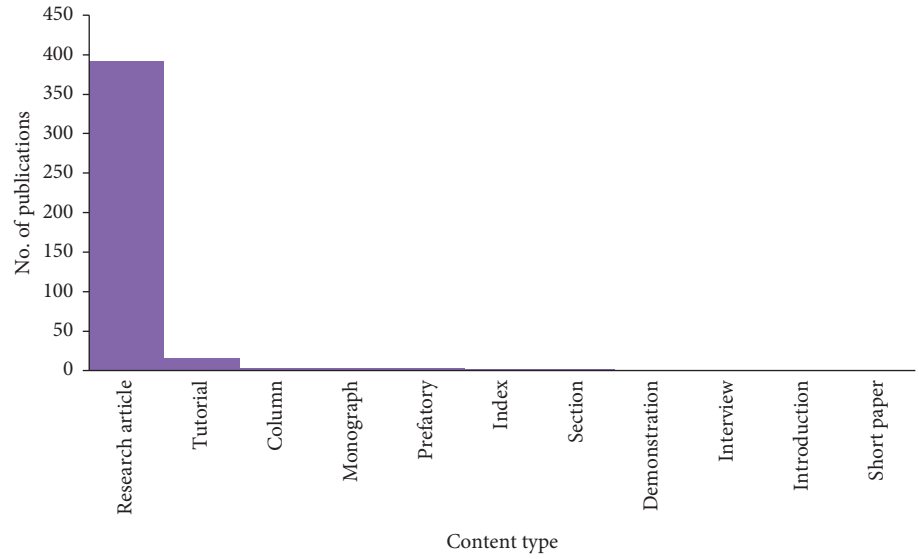


Figure 16: Content type and total number of publication.

there is no authenticity to the materials that is relevant or irrelevant. It shows all the available sources, which is then difficult to analyse.

4.1. Searching Process in IEEE. The IEEE library was searched to find the relevant information regarding the applications of machine learning and multicriteria decision regarding security evaluation. Figure 2 shows the publication type and total number of publications related to the given search. These publications are categorized into different areas such as decision making, learning, fuzzy theory, genetic algorithm, and operational research.

The search process was further explored to find more relevant information of these studies. Figure 3 shows the places of conferences held.

Figure 4 shows the year of conferences held.

Figure 5 shows the total number of conferences heldS in the given year.

4.2. Searching Process in Sciencedirect Library. After searching the library of IEEE, it was felt that the other famous libraries should also be searched to see the relevant materials published in the literature. Figure 6 shows the article type in the form of conference, journal, book chapter, and review articles along with the total number of publications.

The publications were then checked that which paper is published in which specific journal/conference. Figure 7 shows the title of publication where the paper is published along with the total number of papers.

The searched papers were checked to show the year of publication that a paper is published in which particular year. Figure 8 shows the total number of publications in the given year.

4.3. Searching Process in Wiley Library. The Wiley library was searched to find the relevant materials regarding particular search terms. This library does not contain more searching operations as compared to the other libraries. So, only the subjects related information along with the total number of publications is shown in Figure 9.

4.4. Searching Process in Tailor & Francis Library. The Tailor and Francis library was searched to get the most relevant information. Figure 10 shows the subjects of publication along with the total number of publication in the given library in which engineering and technology is on top followed by other disciplines.

4.5. Searching Process in ACM Library. The defined keywords were searched in the ACM library for obtaining relevant information. The ACM library contains several options to study the search results from different perspectives. These perspectives include the publication name where the paper is published, publication types, proceedings, media format, and many others. Figure 11 shows the journal/magazine name along with the total number of papers published for the search process.

Figure 12 shows the proceedings/book name along with the total number of publications in the ACM library.

Figure 13 shows the proceedings series along with the total number of publications.

The search process in this library was further explored to show the media format that which is the

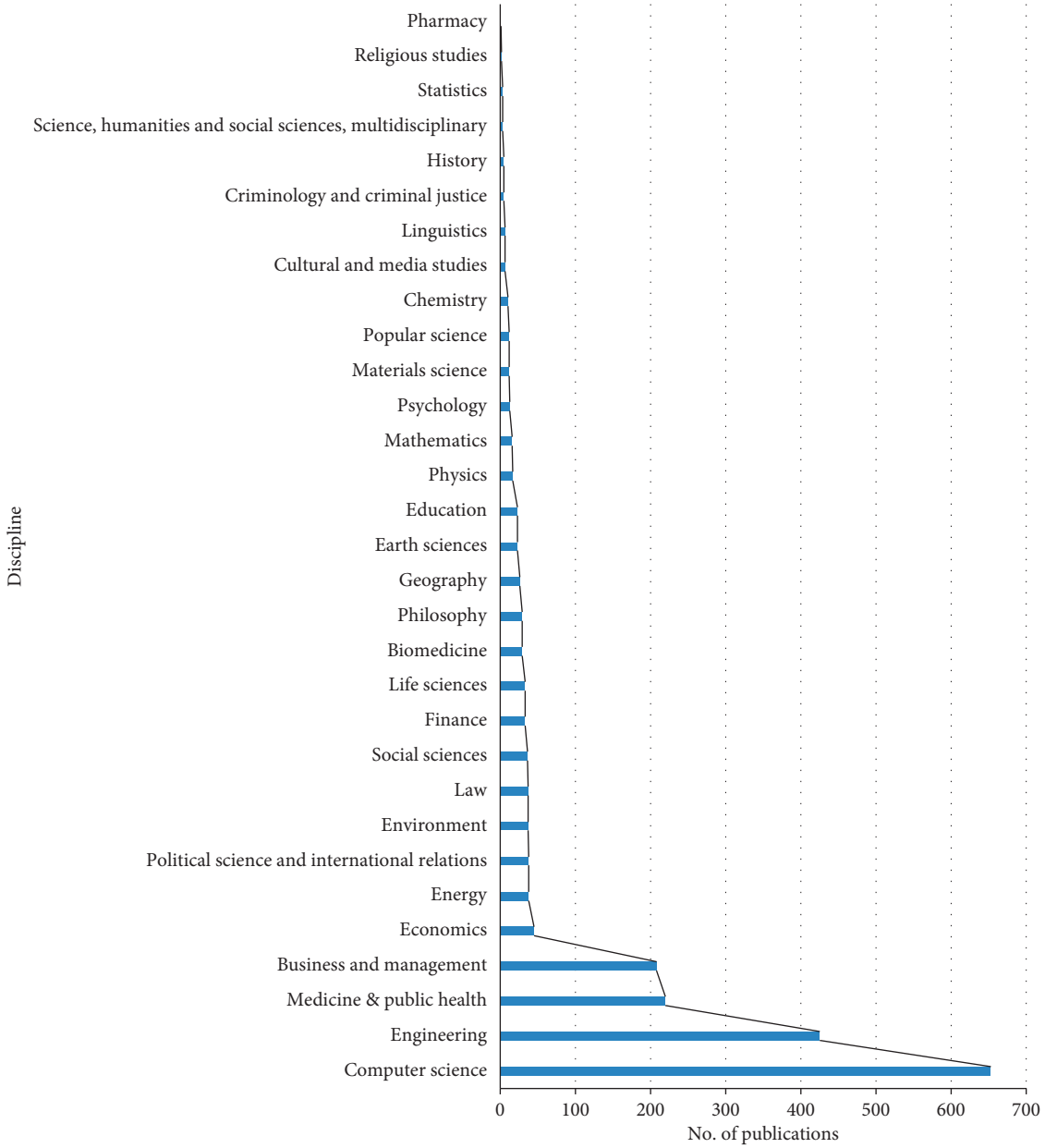


Figure 17: Discipline and total number of publications.

particular format of publication. The media format includes, PDF, image, HTML, Archive/Zip, and video. Figure 14 shows the media format of the publication in the ACM library.

Figure 15 shows the event of the conferences along with the total number of publications.

Figure 16 shows the content type along with the number of publications in the ACM library for the search process. The content types include research article, tutorial, column, monograph, prefatory, index, section, demonstration, interview, introduction, and short paper.

4.6. Searching Process in Springer Library. The Springer library was searched to show the relevant materials published for the given query and keywords. This library contains different options for searching a particular query of keywords. Figure 17 shows the discipline and total number of publications.

Figure 18 shows the content type and total number of publications in the Springer library.

Figure 19 shows publications type of all publications and total number.

5. Conclusion

Security of components plays an important role in a system to function properly. The components are reusable parts of a system which are reused to save time, effort, and cost of developments. Components can be reused as they are already tested, debugged, and experienced. Component

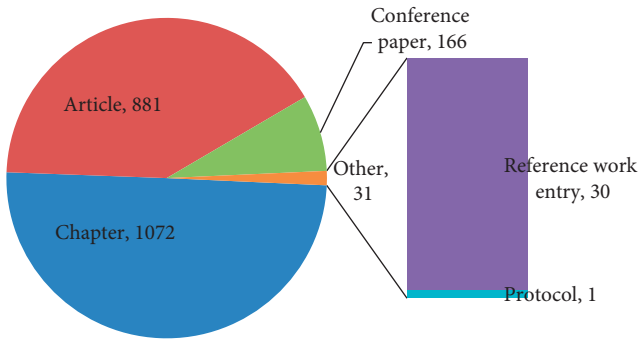


Figure 18: Content type and total number of publication.

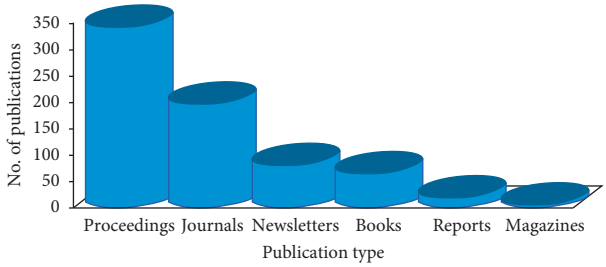


Figure 19: All publications and total number.

security can protect a component from illegal access, use, and change of its contents. Considering the developments in information security, protecting the components becomes a fundamental issue. To tackle this issues, a comprehensive study report is needed which can help practitioners to protect their system. The present study reports some of the available research regarding component security evaluation based on multicriteria decision and machine learning algorithms in the popular searching libraries. Different perspectives of the search process are shown to show the existence of the research related to the current research. Based on the available literature summarized in this paper, researchers can take help from it as evidence and can propose new ideas. In future, the proposed research can be extended to a more detailed analysis from different perspectives such as feature-based security evaluation and real-time security evaluation.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] H. H. Song, "Testing and evaluation system for cloud computing information security products," in *Proceedings of the 3rd International Conference on Mechatronics and Intelligent Robotics (ICMIR-2019)*, pp. 84–87, Kunming, Yunnan, China, May 2019.

[2] B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, "Survey on IoT security: challenges and solution using machine learning, artificial intelligence and blockchain technology," *Internet of Things*, vol. 11, Article ID 100227, 2020.

[3] R. Diesch, M. Pfaff, and H. Krcmar, "A comprehensive model of information security factors for decision-makers," *Computers & Security*, vol. 92, Article ID 101747, 2020.

[4] N. A. B. Mohd and Z. F. Zaaba, "A review of usability and security evaluation model of ecommerce website," in *Proceedings of the Fifth Information Systems International Conference 2019*, pp. 1199–1205, Surabaya, Indonesia, July 2019.

[5] Z. Katzir and Y. Elovici, "Quantifying the resilience of machine learning classifiers used for cyber security," *Expert Systems with Applications*, vol. 92, pp. 419–429, 2018.

[6] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. Ahamed Khan, "Performance analysis of machine learning algorithms in intrusion detection system: a review," *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020.

[7] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city," *Future Generation Computer Systems*, vol. 107, pp. 433–442, 2020.

[8] S. Manjia Tahsien, H. karimipour, and P. spachos, "Machine learning based solutions for security of internet of things (IoT): a survey," *Journal of Network and Computer Applications*, vol. 161, Article ID 102630, 2020.

[9] X. Wang, J. Li, X. Kuang, Y.-a. Tan, and J. Li, "The security of machine learning in an adversarial setting: a survey," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12–23, 2019.

[10] M. Marwan, A. Kartit, and H. Ouahmane, "Security enhancement in healthcare cloud using machine learning," *Procedia Computer Science*, vol. 127, pp. 388–397, 2018.

[11] M. Belouch, S. El Hadaj, and M. Idhammad, "Performance evaluation of intrusion detection based on machine learning using Apache Spark," *Procedia Computer Science*, vol. 127, pp. 1–6, 2018.

[12] M. A. Amanullah, R. A. A. Habeeb, F. H. Nasaruddin et al., "Deep learning and big data technologies for IoT security," *Computer Communications*, vol. 151, pp. 495–517, 2020.

[13] J. Yuan and X. Luo, "Regional energy security performance evaluation in China using MTGS and SPA-TOPSIS," *Science of the Total Environment*, Article ID 133817, vol. 696, pp. 1–11, 2019.

[14] C. Wijayarathna and N. A. G. Arachchilage, "Using cognitive dimensions to evaluate the usability of security APIs: an empirical investigation," *Information and Software Technology*, vol. 115, pp. 5–19, 2019.

[15] Z. Zhang, J. Wen, X. Wang, and C. Zhao, "A novel crowd evaluation method for security and trustworthiness of online social networks platforms based on signaling theory," *Journal of Computational Science*, vol. 26, pp. 468–477, 2017.

[16] W. Mao, Z. Cai, D. Towsley, Q. Feng, and X. Guan, "Security importance assessment for system objects and malware detection," *Computers & Security*, vol. 68, pp. 47–68, 2017.

[17] T. Halabi and M. Bellaiche, "Towards quantification and evaluation of security of cloud service providers," *Journal of Information Security and Applications*, vol. 33, pp. 55–65, 2017.

[18] M. Cheah, S. A. Shaikh, O. Haas, and A. Ruddle, "Towards a systematic security evaluation of the automotive Bluetooth interface," *Vehicular Communications*, vol. 9, pp. 8–18, 2017.

[19] S. Nazir, S. Shahzad, M. Nazir, and H. U. Rehman, "Evaluating security of software components using analytic network process," in *Proceedings of the 11th International Conference on Frontiers of Information Technology (FIT)*, pp. 183–188, Islamabad, Pakistan, December 2013.

[20] Y. Cherdantseva, J. Hilton, O. Rana, and W. Ivins, "A multifaceted evaluation of the reference model of information assurance & security," *Computers & Security*, vol. 63, pp. 45–66, 2016.

[21] M. Jouini, L. B. A. Rabai, and R. Khedri, "A multidimensional approach towards a quantitative assessment of security threats," in *Proceedings of the Procedia Computer Science the 6th International Conference on Ambient Systems, Networks and Technologies*, pp. 507–514, London, UK, December 2015.

[22] I. Kotenko and A. Chechulin, "Computer attack modeling and security evaluation based on attack graphs," in *Proceedings of the 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*, pp. 614–619, Berlin, Germany, September 2013.

[23] P. Subsorn and S. Limwiriyakul, "A comparative analysis of internet banking security in Thailand: a customer perspective," *Procedia Engineering*, vol. 32, pp. 260–272, 2012.

[24] I. Kotenko and A. Chechulin, "Common framework for attack modeling and security evaluation in SIEM systems," in *Proceedings of the IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber*, pp. 94–101, Besançon, France, 2012.

[25] M. Li, S. Nazir, H. U. Khan, S. Shahzad, and R. Amin, "Modelling features-based birthmarks for security of end-to-end communication system," *Security and Communication Networks*, vol. 2020, 2020.

[26] H. U. Rahman, A. U. Rehman, S. Nazir, I. U. Rehman, and N. Uddin, "Privacy and security—limits of personal information to minimize loss of privacy," in *Proceedings of the Future of Information and Communication Conference*, pp. 964–974, San Francisco, CA, USA, March 2019.

[27] B. A. Sassani, M. Alkorbi, N. Jamil, M. A. Naeem, and F. Mirza, "Evaluating encryption algorithms for sensitive data using different storage devices," *Scientific Programming*, Article ID 6132312, vol. 2020, pp. 1–9, 2020.

[28] B. Liao, Y. Ali, S. Nazir, L. He, and H. U. Khan, "Security analysis of IoT devices by using mobile computing: a systematic literature review," *IEEE Access*, vol. 8, p. 1, 2020.

[29] S. Nazir, S. Shahzad, S. Mahfooz, and M. N. Jan, "Fuzzy logic based decision support system for component security evaluation," *International Arab Journal of Information and Technology*, vol. 15, pp. 1–9, 2015.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8863345, 16 pages
https://doi.org/10.1155/2020/8863345

Research Article

Secure Framework Enhancing AES Algorithm in Cloud Computing

Ijaz Ahmad Awan,[1,2] Muhammad Shiraz,[3] Muhammad Usman Hashmi,[1] Qaisar Shaheen,[1] Rizwan Akhtar,[4] and Allah Ditta[5]

[1]*Department of Computer Science, Superior College, Lahore, Pakistan*
[2]*University of Engineering and Technology, Lahore, Pakistan*
[3]*Department of Computer Science, Federal Urdu University of Arts, Science and Technology, Islamabad, Pakistan*
[4]*School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, China*
[5]*Department of Information Sciences, Division of Science & Technology, University of Education, Lahore, Pakistan*

Correspondence should be addressed to Qaisar Shaheen; qaisar.shaheen2002@gmail.com

Received 17 June 2020; Revised 4 August 2020; Accepted 7 August 2020; Published 1 September 2020

Academic Editor: Umar M. Khokhar

The tremendous growth of computational clouds has attracted and enabled intensive computation on resource-constrained client devices. Predominantly, smart mobiles are enabled to deploy data and computational intensive applications by leveraging on the demand service model of remote data centres. However, outsourcing personal and confidential data to the remote data servers is challenging for the reason of new issues involved in data privacy and security. Therefore, the traditional advanced encryption standard (AES) algorithm needs to be enhanced in order to cope with the emerging security threats in the cloud environment. This research presents a framework with key features including enhanced security and owner's data privacy. It modifies the 128 AES algorithm to increase the speed of the encryption process, 1000 blocks per second, by the double round key feature. However, traditionally, there is a single round key with 800 blocks per second. The proposed algorithm involves less power consumption, better load balancing, and enhanced trust and resource management on the network. The proposed framework includes deployment of AES with 16, 32, 64, and 128 plain text bytes. Simulation results are visualized in a way that depicts suitability of the algorithm while achieving particular quality attributes. Results show that the proposed framework minimizes energy consumption by 14.43%, network usage by 11.53%, and delay by 15.67%. Hence, the proposed framework enhances security, minimizes resource utilization, and reduces delay while deploying services of computational clouds.

1. Introduction

It is observed that cloud technology is used in a number of architectures, services with further technologies, and various software design approaches [1]. Cloud service models include platform as a service (PaaS), software as a service (SaaS), and infrastructure as a service (IaaS). Architecture solutions for the public, private, community, and hybrid system depend on four cloud platform deployment models [2]. Advantages of cloud computing include flexibility, accessibility, and capacity when linked to traditional online computing or storage method [3]. However, a number of security concerns are associated with computational clouds including (i) privacy and security issues with cloud service providers and (ii) customer-related security issues [4]. In the literature, various types of attacks related to the strength of the AES (advanced encryption standard) algorithm have been proposed [5], for instance, different fault analyses which attack and introduce faults into the AES (advanced encryption standard) structure with the target to retrieving the secret information [6].

Furthermore, cloud computing standard can propose some feasible practices of service area, by means of computational resources on behalf of extraordinary performance in computing applications, telecommunication services, social networking, and web services [7, 8]. In addition, cloud storage in data centres is very valuable for users just before storing and

accessing their data distantly at any time without any further load [9, 10]. On the contrary, the main problem of cloud data storage is security. As a result, cloud data centres must have some mechanisms which are capable to ensure storage perfection and integrity of data that are stored on cloud [11].

Existing security systems employ one or two attributes at a time, i.e., low security and more time consumption to encrypt/decrypt the data. This makes the process more time-consuming and therefore increases the network use, power consumption, and delay in the network [12–16]. Cloud computing is that kind of platform which shares the data and resources efficiently, and therefore, security must be provided to the users as security is an important aspect of cloud computing. So, this is the responsibility of the cloud service providers to provide security with all attributes, such as less power consumption, delay of network, and time consumption [17–23]. Already, traditionally available methods are not able to quantify the security of cloud services effectively. Secure framework in cloud computing is a method that provides simplified management and accessing of computing resources, and a cost-effective approach is the need of the hour. The framework should use low power, time, and delay of network consumption with encryption and decryption that enhance the security of data in cloud computing.

The paper contributes towards the design of the security framework by implementing a new scheme of encryption/decryption. It also determines the serious components of the security framework within the cloud computing community. It would be helpful for those cloud users and cloud service providers who have similar requirements in terms of security during implementation. The framework helps in faster computing with lesser power consumption, network usage, and reduced network delay due to the smart algorithm. The framework employs a symmetrical encryption method to provide trust to users and enables trusted gateways. The proposed framework includes the key features including enhanced security and owner's data privacy. It modifies the 128 AES algorithm to increase the speed of the encryption process 1000 blocks per second by the double round key feature. However, traditionally, there is a single round key with 800 blocks per second. The proposed algorithm involves less power consumption, better load balancing, and enhanced trust and resource management on the network. The proposed framework includes deployment of AES with 16, 32, 64, and 128 plain text bytes. Simulation results are visualized in a way that depicts suitability of the algorithm while achieving particular quality attributes. Results show that the proposed framework minimizes energy consumption by 14.43%, network usage by 11.53%, and delay by 15.67%. Hence, the proposed framework enhances security, minimizes resource utilization, and reduces delay while deploying services of computational clouds.

The remainder of this paper is configured as the following sections: Section 2 details the literature review. Section 3 defines the framework architecture. Section 4 includes the experimental environment. Section 5 presents the performance results of both existing and proposed frameworks. Section 6 defines the forthcoming features associated with this paper exertion.

2. Literature Review

Several modifications were introduced in AES in order to enhance the performance speed and security by introducing some complexities in algorithms. These modifications are implemented on different software and hardware designs. However, preview framework security is always a concern due to some security constraints and problems with cloud computing. The security is provided to the information which is stored on the cloud by using cryptography algorithms. There are extensive security frameworks for cloud computing that uses enormous encryption techniques. Out of these, a few of them are presented here.

The security framework is based on the multicloud environment to store digital data at all. In order to prevent data disclosure, they practiced a segmentation approach to fragment the input appearance into several areas. The integrity of the outsourced clients' data helps to verify watermarking technique. Any accidental change to outsourced clients' data can be detected by the digital signature and watermarking methods [24]. This paper focuses on the computation of different methods which explain how to increase data security so that prevention from different security attacks and breaches can be made. Mitigation approaches used in this research on the HMAC (Hashed Message Authentication Code) were ECC and MD5. This proposed solution is based on different security levels; as a result, access control, authentication, confidentiality, integrity, and encryption are achieved in this work. The authors performed and checked the security solution in real-time as well as in real cloud computing environment and also concluded that the solution that is been provided has very low overhead for upload and download service time [17]. The framework presented in this study is more secure, and it provides more privacy to the data. This framework splits data into different blocks of bit. On every two blocks of bits, genetic algorithm is applied. Concluding output of each genomic algorithm procedure is a ciphertext along with two blocks of bits. Each ciphertext is stored on the cloud at a distinct location, and the location of the ciphertext is not secure. What makes it more secure from attackers to find the exact location of the ciphertext? The innovative security framework puts on a genetic algorithm on minor block size that increases the security. Furthermore, the framework uses the proficiency list aiming to secure and to access data [18].

In this paper, authors proposed a new framework that ensures the data security and integrity and also focused on the encryption and decryption approaches facilitating the cloud user with data security assurance. The proposed solution talked about the increased security along with the performance. Their solution has also included functioning of the forensic virtual machine, malware detection, and real-time monitoring of the system [25]. In this paper, the authors suggested a framework such that the objective is to store data in various clouds. The given framework is found based on 3DES and RSA encryption. On the contrary, this methodology is lacking in efficiency, privacy, and overload middleware through multiple functions [26]. In this paper, the authors studied, multilevel licensing framework approval

preservation cloud penetrating data. Safeguarding the familiar and delicate cloud data is obtainable by the three covers' framework. Those restrictions are being the security and privacy strategies, safety and approval policies which outcomes from the three films' security framework [19]. In this paper, the authors proposed quality metrics and details probe on instance cloud service broker frameworks are provided. These streak metrics help in enforcing standards on cloud service providers by using quality-based cloud service broker framework (QCSB). The algorithm and implementation of QCSB have been obsessing. At last, the authors concluded that the proposed material QCSB not only assists cloud computing to locate optimal CSP (cloud service provider) for cloud services but also affiliates candidate CSPs according to user quality preferences [20].

The complexity detects were an effect of dismiss logical purposes in the MixColumn conversion of AES. These reasonable tasks were eradicating in the modified version of AES. Afterward,on utilizing the modified AES, a 13.6% reduction in LUTs, 10.93% share discount, and a 1.19% reduction in interruption eating was attained. Likewise, the small dispersal rate met through the conservative AES at the initial nonentity, and important agenda sequences are spoken in [27]. In this research, they examined five metrics specifically: the graphic study, file size, radiance histogram, assessment by pixel, and show distance. In the file scopes, there were differences wherever it displays the regular worth of the fraction variations to −23.85% from the unique to the encrypt duplicate and −1.45% percentage worth from the innovative to the decrypt duplicate [28]. This paper showed an overview of the latest research studies that are going on in fog computing and the IoT and its uses; it also enlightened the research gaps and directions for further future research studies in the integration of fog computing and IoT (Internet of Things). A modern fog computing framework was presented [29]. The modified AES contained 10 series for encrypting, and the replacement and addition processes of the columns have been substituted by the line change and pixel standard summary. These processes not only decrease the spell complication of the algorithm but also improve the dispersal aptitude to the CCAES (combining the chaos and AES) algorithm.The encrypted descriptions by the CCAES algorithm remained unaffected to the variance occurrences. The project algorithm is protected alongside the entropy occurrences. The simulation consequences illuminate that the minor deviations in the unique appearance and consequences in the important fluctuations in the encrypt duplicate and the innovative appearance cannot be retrieved [30]. This paper described the CloudSim simulator counting its architecture, aces, convicts, and CloudSim forms. Likewise, it characterized exactly how to practice CloudSim demonstration and replication in the cloud environment. Furthermore, it also describe the way to calculate approximate presentation limits like regular reversal time, amount, implementation period, types pan and entire conclusion period, etc. [31].

This paper reported dissimilar data safety and privacy security concerns in a cloud calculating environment and suggested a technique for dissimilar security services such as verification, approval, and privacy along with observing in suspension. Cloud computing plans a different technique for obtaining cloud data in the actual environment. 128 bit AES encryption is recycled for privacy, genuineness, and contact controller [32]. In the future work, load balancer by means of My Load Balancer optimization method has been compared with the two greatest well-known weight balancer techniques, i.e., Round-Robin and Supper Present Implementation Freight, also recognized as Active Monitoring Load Balancer. All such Java-based virtual techniques are used to create Cloud analyst toolkit. Graph procedures have been recycling to prove the comparative analysis [33]. The procedure of cryptography involves two main methods which are encryption and decryption. In the encryption method, a basic manuscript is converted to an innovative text which the others cannot deliver and understand additional than the receiver. Blowfish and AES procedures are exploited for executing a hybrid approach connected to cryptography. This consequence in a cryptograph text which can merely be decrypted by the receiver this one [34]. In this paper, obtainable low-control AES architecture by exploiting humble shift catalogues and variation for key/data stored to decrease journey magnitude and control consumption. A low-power method, called clock gating is used to control exchangeable on S-box[35]. In the present study, Abikoye et al.'s modified AES algorithm [13] is presented which is also used in applications to make a comparison. K-L Tsai et al. presented the modified AES-based algorithm for power reduction in IoT using cloud computing applications [14]. In this paper, similarly, VM (virtual machine) allocation policy is used for security which is almost similar to the technique used in the previous work [36].

In general, the main purpose of all research studies related to the subject areas is to investigate the possible ways to improve the security of cloud computing services. Therefore, in this work, a secure framework has been proposed for securing confidential tasks being stored in cloud systems using AES encryption methods. Finally, a comparison of the results obtained through this proposed framework and traditional framework work formulated in the past is made which showed significant improvement of cloud computing using the proposed framework. The differences between our modified AES and previously developed or modified AES in the JAVA cipher-based security framework have been discussed in this manuscript. It is pertinent to mention here that our trust-based framework blocks the suspicious users from the network and maintains a queue for such users to protect the trusted users.

3. Architecture of the Proposed Secure Framework for Cloud Computing (SFCC)

The architecture of the proposed secure framework for cloud computing (SFCC) is presented in Figure 1.

Framework of secure cloud computing is proposed on the security architecture shown in Figure 1, which describes the information for each component and their applications which are required for secure technologies to operate between components in cloud computing. This framework acts in the

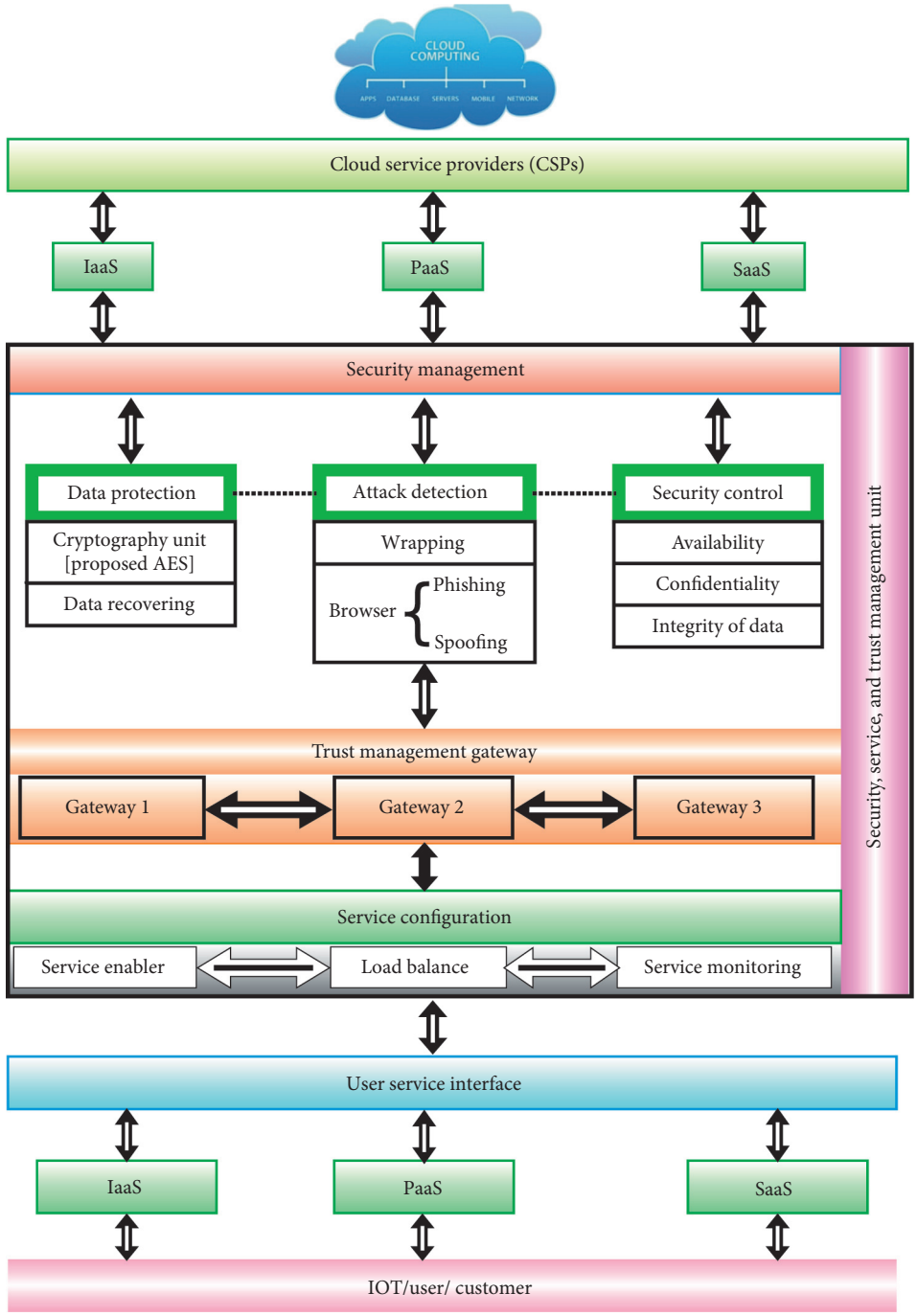


Figure 1: Secure framework for cloud computing (SFCC).

following conditions checking security, privacy, load balancing, and trust. When the user directs a demand to the cloud benefactor, it responds to the user's request and passes the data through framework gateways. The proposed framework includes the following components:

Cloud service provider (CSP) layer: the CSP controls the important sources and ability in construction and calculates the dispersed cloud storage servers processes and directs the live obscure work out method. Its main component is software as a service (SaaS); this is a model in which end users are provided software application (as a service). Platform as a service (PaaS): this model proposed an atmosphere for requests. Development tools that are essential for advanced applications are also provided in this model. Infrastructure as a service (IaaS): this is a platform that offers compulsory properties such as physical machines, virtual machines, and virtual storage.

Security service trust management unit: security service trust management controls all the units which include

security management; trust management gateways also control the service configuration, respectively. Further details of all units are described in the following.

Security management layer: the security management factor offers security and privacy details and implementation functionality. Security service has the following modules and their details.

Security control unit: availability is the percentage of time a customer can access the service. Confidentiality (authentication, authorization, and identification) is an integral component of security. It ensures that the information stored on the cloud is protected against the unintended or unauthorized access. Identification user is typically skilful by retaining usernames and passwords after utilizing web browser in order to admit in Cloud. Integrity of data security control is responsible for maintaining the accuracy of data computation that is coming from the combination of different files and is also responsible for its delivery.

Attack detection unit: ultimately, slightly usual activities that hover the cloud security necessities (e.g., integrity, confidentiality, and availability) are measured to be occurrences. Wrapping is when the attacker attacks by wrapping the communication between two people, while the users do not know this and think data are still coming from the actual root. Unethical browsing is to find bad actions happening, for example, phishing and spoofing and changing browser certificates.

Data protection unit: proposes the AES algorithm to enhance the data security by means of cryptography techniques using AES ciphers as they can encrypt 128 bits' data blocks within 1000 blocks per second with the double round key feature with less power consumption, load balancing, trust, and resource management on the network efficiently. We have used symmetric identification for security, i.e., the same key for encryption and the same key for decryption as identification of data streams in the form of security. It provides greater efficiency for software as well as hardware. The advantage of using symmetric key is to secure a large amount of data. Data recovery: if data is lost in a disaster that ithas a capability to regain it or restore it.

Trust management gateway layer: for the fourth layer, trusted gateways are implemented. These gateways get the encrypted data and decrypt only if the trusted source is connected with a valid internet protocol address of a given domain. These gateways support the issues of trust. There are three gateways in which two are in an alternative manner. In case of the normal gateway is being attacked and misused, other safe gateways shall be chosen to ensure data communication.

Service configuration layer: the service enabler makes provision for personalized cloud service using the user's profile for integration and interoperation. Load balancing can be implemented on hardware, software, or a combination of both. It is important in this configuration that all instances of identity server share the same directory server. Service monitoring: an automatic facility-checking system to assure an extraordinary level of facility presentation and obtainability.

User service interface layer: this layer provides different services to select the user via the internet: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).

Service configuration layer: the last unit for the user, IOT, and customer to send and receive data.

4. Experimental Setup and Implementation of SFCC

The SFCC can be implemented in real time. The results gathered from the simulations are very accurate. These results are theoretically consistent. Everything is implemented accordingly. Codes are very consistent with real-time mechanisms. The SFCC is developed using CloudSim and iFogSim simulators on the Eclipse integrated development environment. CloudSim is a very well-known and popular among simulators for cloud-based applications. It is responsible for the simulation and events handling at cloud. Some libraries are used for different purposes. Libraries used are JavaScript object notation (Json) data saver, common math, and JFreeChart.

The developed simulation comprises SFCC. The proposed framework is generic so that anyone could put one's idea or logic in this simulation and get the required results. It helps the user to test different scenarios under the proposed algorithm. The simulation has the ability to store and generate a large amount of data. It allows a user to measure the factors such as encryption, description, power consumption, network usage, delays, trusted devices, and service management. The advanced encryption standard for encryption and decryption for data protection is used. The comparison of the algorithm with the previous unmodified algorithms is discussed in later sections. The characteristics of the layers and devices are described in Tables 1–11.

4.1. Components. Data centre refers to on-premise hardware, while the cloud refers to off-premise computing. The cloud stores your data in the public cloud, while a data centre stores your data on your hardware. Data centre configuration is displayed in Table 1.

Infrastructure as a service (IaaS): this is a platform that offers compulsory resources such as physical machines, virtual machines, and virtual storage. Infrastructure-as-a-service configuration is displayed in Table 2.

Software as a service (SaaS): this is a model in which end users are provided software applications (as a service). Software-as-a-service configuration is displayed in Table 3.

Platform as a service (PaaS): this model proposed an atmosphere for requests. Development and deployment tools that are essential to advance applications are also provided in this model. Platform-as-a-service configuration is displayed in Table 4.

Table 1: Data centre characteristics of cloud.

Name of the device	Cloud
Level	1
Uploading bandwidth	5000
Downloading bandwidth	12000
Million instructions per second	130.0
RAM	45000
Rate per processing usage/MIPS	100000

Table 2: Data centre characteristics of infrastructure as a service.

Name of the device	Cloud IAAS
Level	2
Uploading bandwidth	4000
Downloading bandwidth	5000
Million instructions per second	50000
RAM	40000
Rate per processing usage/MIPS	400.0

Table 3: Data centre characteristics of software as a service.

Name of the device	Cloud SAAS
Level	2
Uploading bandwidth	4000
Downloading bandwidth	5000
Million instructions per second	60000
RAM	40000
Rate per processing usage/MIPS	400.0

Table 4: Data centre characteristics of platform as a service.

Name of the device	Cloud PAAS
Level	2
Uploading bandwidth	4000
Downloading bandwidth	5000
Million instructions per second	60000
RAM	40000
Rate per processing usage/MIPS	50000

Table 5: Data centre characteristics of security management.

Name of the device	Security management
Level	4
Uploading bandwidth	5000
Downloading bandwidth	5000
Million instructions per second	40000
RAM	35000
Rate per processing usage/MIPS	600.0

Security management: the security management factor offers the security and privacy details and implementation functionality table. Security management configuration is displayed in Table 5.

Table 6: Data centre characteristics of gateway1.

Name of the device	Trusted gateway1
Level	3
Uploading bandwidth	3000
Downloading bandwidth	4000
Million instructions per second	30000
RAM	20000
Rate per processing usage/MIPS	1000.0

Table 7: Data centre characteristics of gateway2.

Name of the device	Trusted gateway2
Level	3
Uploading bandwidth	3000
Downloading bandwidth	4000
Million instructions per second	30000
RAM	30000
Rate per processing usage/MIPS	400.0

Table 8: Data centre characteristics of gateway3.

Name of the device	Trusted gateway3
Level	3
Uploading bandwidth	4000
Downloading bandwidth	4000
Million instructions per second	50000
RAM	34000
Rate per processing usage/MIPS	600.0

Table 9: Data centre characteristics of service configuration.

Name of the device	Service configuration
Level	1
Uploading bandwidth	5000
Downloading bandwidth	5000
Million instructions per second	100000
RAM	40000
Rate per processing usage/MIPS	500.0

Table 10: Data centre characteristics of service provider.

Name of the device	Service provider
Level	1
Uploading bandwidth	5000
Downloading bandwidth	5000 Gbits/sec
Million instructions per second	50000
RAM	20000 gb
Rate per processing usage/MIPS	100.0

Table 11: Virtual machine configurations.

Virtual machine number level	Virtual machine number	Processing elements	Bandwidth (uplink)	Latency input
Level 0	2	20000	800	10
Level 1	4	18000	1000	6
Level 2	6	16000	1200	8

Gateway devices at the second-last level of the hierarchy gateway devices are created. These gateway devices are part of the layer responsible for communicating with proxy servers and cloud devices. Here are the characteristics of the gateway devices. Gateway device configuration is displayed in Tables 6–8.

Service configuration: This facility modifies the cloud service using the user's profile byintegrating service enabler, load balancing, and service monitoring. Service configuration is displayed in Table 9.

Service provider: this is the last unit for users and customers to send and receive data. Service provider configuration is displayed in Table 10.

Virtual machines are created and allocated to hosts to support processing and offloading the modules to support the load balancing mechanism. These virtual machines come with the proposed strong encryption algorithm to support the security and trust feature. The virtual machine configuration is displayed in Table 11

The materials and methods section should contain sufficient detail to repeat all procedures. It may be divided into headed subsections if several methods are described.

4.2. Physical Topology of SFCC. The physical topology shows the pattern of nodes and devices in the network. Physical entities are created, and their competence, capability, and configurations are specified. These entities include sensors, actuators, gateways, and cloud VM (virtual machines). The links between these entities and their configuration are also established. Physical network topology is important to understand the pattern of the network, how various network devices are organized, and how they communicate with each other. These configurations and capacity determine the load a network can tolerate and the amount of data it can transfer. The physical topology is shown in Figure 2.

4.3. Explanation Topology. The computing mechanism of cloud always happens at the top. Cloud stays at the top to manage the lower-level architecture [37]. The three different types of cloud stay below the top layer and act as CSPs [38] according to customers' need. For the third layer, the virtual machine allocation policy mechanism is implemented to support data offloading and privacy for security [39] in the proposed system. Offloading the modules not only provides load balancing but also solves the security issues of cloud by providing a new layer on the hosts. Virtual machines are created and allocated to the hosts to support processing and offloading of the modules to support the load balancing mechanism. These virtual machines come with a strong encryption algorithm to support the security and trust feature.

The virtual machine requires some storage and processing capabilities similar to a host H in nature. Equation (1) represents the conditions for creating a virtual machine. The Vm size is always smaller than the available host H and storage S, where the number of Vms depends on the size of load (β).

If $H = \{H1, H2, H3, \ldots, Hn\}$ and $V = \{Vm1, Vm2, Vm3, \ldots Vm\ N\}$, then

$$\exists Vm \in H \cup S: Vm \propto \beta \text{ where } H \cap S \gg Vm,$$
$$: Vm1, Vm2, Vm3, \ldots, Vm < H1, H2, H3, \ldots, H,$$
$$\cdot \forall V\ \exists Vm1, Vm2, Vm, \ldots, VmN \in H. \quad (1)$$

Equation (1) represents how VM creation is carried out under various rules and conditions

For the fourth layer, trusted gateways are implemented. These gateways get the encrypted data and decrypt only if a trusted source is connected with a valid Internet protocol address of a given domain. These gateways support the issues of trust [40]. There are 3 gateways in which 2 are alternate manner. In case of a normal gateway is being attacked and misused, other safe gateways shall be chosen to ensure data communication, as shown in Figure 3.

Trusted gateways put the blacklist users into the blocked users' category to ensure the security and privacy of trusted users. The fifth layer is responsible for 3 functions. These functions include service monitoring, load balancing, and service enabling/disabling. The bottom-most layer is based on the users of cloud, and it represents the Internet-of-Things layer in the proposed system. This is how all the aforementioned proposed frameworks work. The trusted customer stays as long as a mediator (trustee) stays. And a mediator stays as long as the cloud service providers are trustable. The chain of trust can be seen in Figure 4 [41].

4.4. Changesin Traditional AES Algorithm. The high-level flow of the proposed AES algorithm in a standard way is presented in Figure 5.

4.4.1. Changes in the Traditional AES Algorithm vs. the Proposed Algorithm. The cloud computing confidentiality framework is presented in this paper. In this framework, data integrity mechanism is used to enhance the data security by the means of cryptography technique. The modified AES (advance encryption standard) ciphers as it can encrypt 128 bit data blocks within 1000 cycles with low power, time, and delay of network consumption. The other work of the frameworks is load balancing, trust, and resource management on the network efficiently.

We have used symmetric identification for security, i.e., the same key for encryption and decryption as identification of data streams in the form of security. The difference between the proposed and previously developed AES is that we have also encrypted 1000 blocks per second with the double round key feature. Previously developed AES uses a single round key with 800 blocks per second. The advantage of using symmetric key is to secure a large amount of data.

4.5. AES Substitution Box (S-Box). The primary stage to around, remains to organize a byte by byte replacement through a lookup table called a substitution box or simply

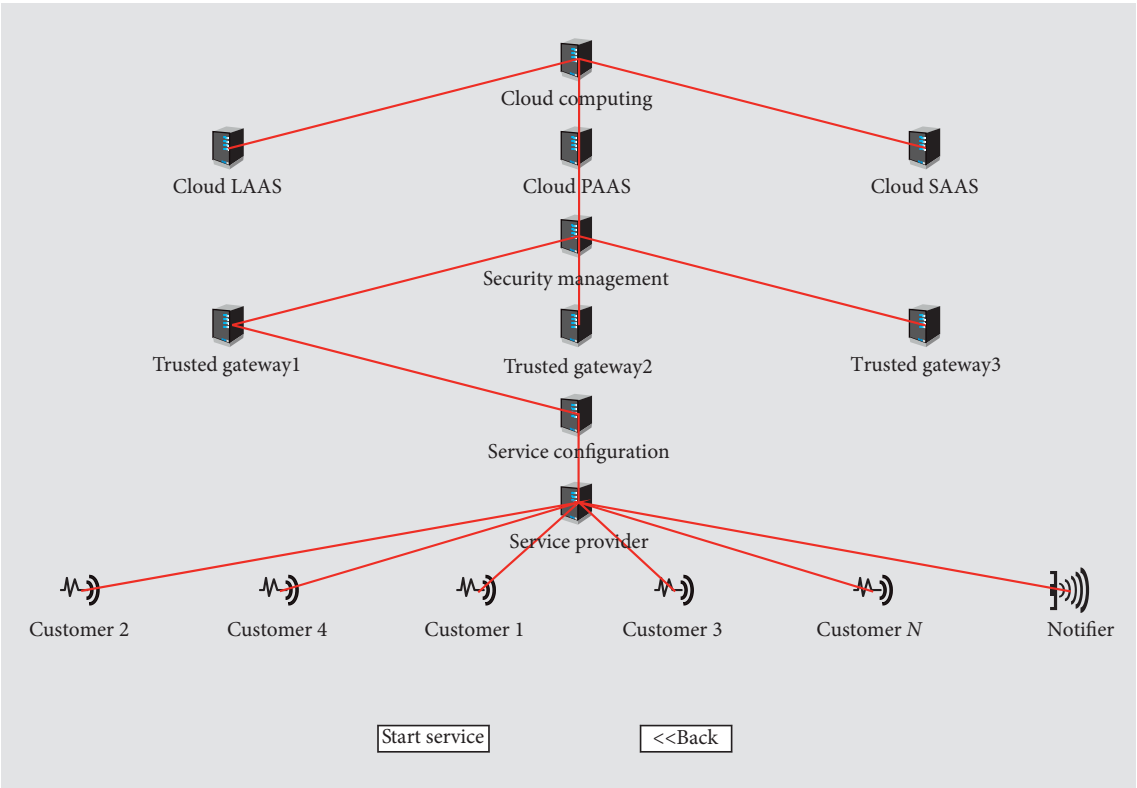


Figure 2: Physical network topology.

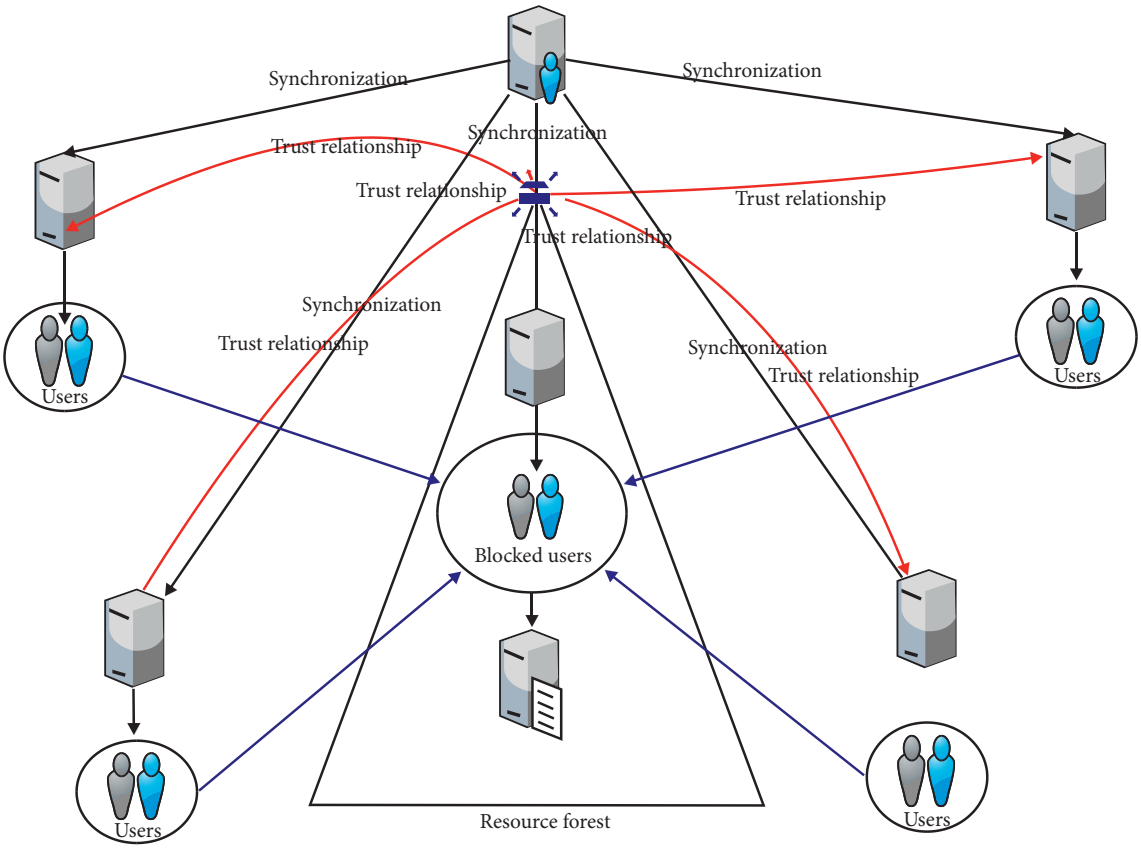


Figure 3: Trusted gateways.

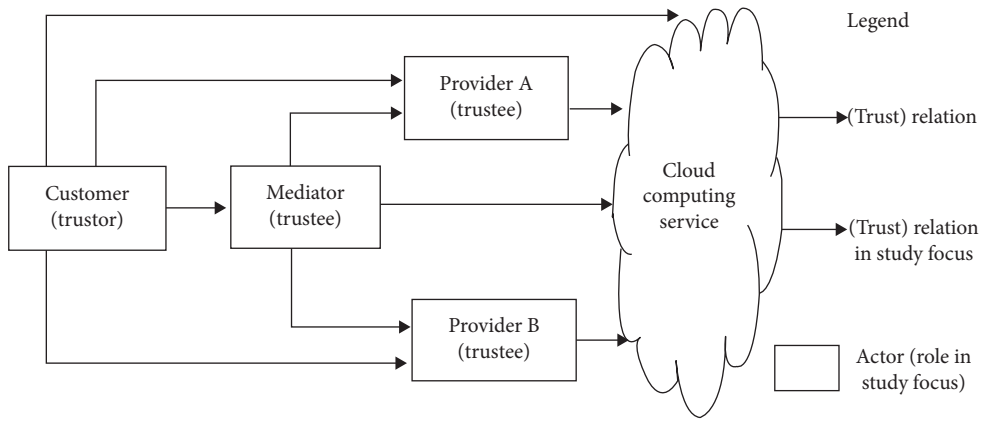


Figure 4: Mediator of cloud service providers' trusted chain.

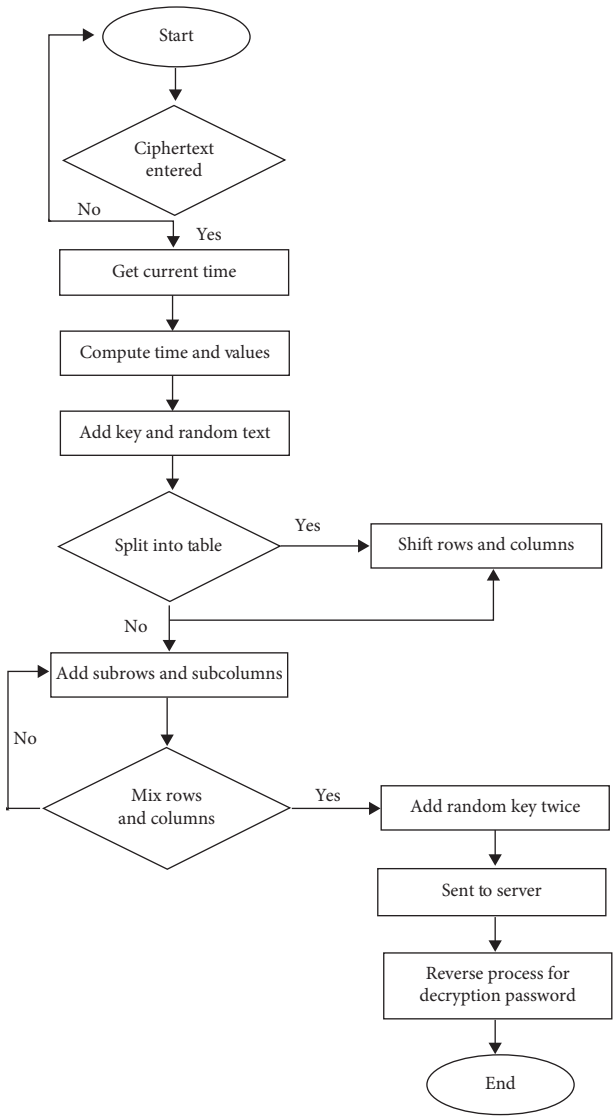


Figure 5: Flow diagram of the proposed algorithm.

S-box. The S-Boxescarry out one to one plotting for all byte values from 0 to 255 in 16×16 arrays. Replacement is a nonlinear conversion which achieves misperception of bits. A nonlinear revolution is vital for each current encryption algorithm and shown to be solid cryptographic original in contradiction to direct and disparity cryptanalysis. The S-box is shown Figure 6. All values are represented in hexadecimal notation [42]. The general substitution box for adding round keys is given in Figure 6.

Rows are represented by x, and columns are represented by y. The mixing process is done with XOR denoted by the symbol ⊕. The binary example in the following will illustrate the functionality of the XOR operator. The row and column mixing and shifting are done by Shift (x_row, y_clolumn) function. The transformed arrays x and y are converted into binaries using ASCII__ASCII 256 standard. Then, the XOR operator performs its ⊕ operation on the bits to generate the ASCII-(American Standard Code for Information Interchange-) generated ciphertext. The cryptographic technique used in SFCC is presented in the low-level language as follows:

$$CiT\,(\text{enc}) = \frac{1}{N}\sum_{i=0}^{1} X_r \oplus Y_c, \tag{2}$$

$$CiT\,(\text{dec}) = N\sum_{i=0} X_c \oplus Y_r. \tag{3}$$

⟶putfieldjavax.crypto.Cipher.spi: javax.crypto.CipherSpi

exec_0 [this] exec_2 [x__rows] (i)

⟶putfieldjavax.crypto.Cipher.provider: java.security.Provider

exec_0 [this] exec_3 [y__columns] (ii)

⟶putfieldjavax.crypto.Cipher.transformation: java.lang.String

exec_0 [this] (iii)

⟶getstaticjavax.crypto.CryptoAllPermission.INSTANCE:

javax.crypto.CryptoAllPermission (iv)

⟶putfieldjavax.crypto.Cipher.cryptoPerm: javax.crypto.CryptoPermission

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figure 6: Substitution box [42].

exec_0 [this]aconst_null (v)

⟶putfieldjavax.crypto.Cipher.lock:java.lang.Object return $\prod$; (vi)

The example of $\oplus$ is given as follows.

Let $X = 1110_2$ and $Y = 1001_2$. Then, XOR of X and Y is represented by Z:

$$Z = X \oplus Y = 0111_2. \tag{4}$$

$Z = 0111_2$ is the result of this operand. Table 12 displays the result in the tabular form.

Table 12: XOR operations.

X	Y	Z (result)
1	1	0
1	0	1
1	0	1
0	1	1

5. Results and Discussion

In order to check and test the efficiency of the proposed algorithm, a simple code is used. This test helped us to prove that the proposed AES algorithm is better than any other AES algorithm, and after implementation of AES and advanced AES code on hardware will reduce the execution time. The SFCC results as performed and the implementation of this security framework for cloud computing. The period actsas an energetic character during the peers of key, encryption, and decryption procedure. Altogether, inquiries remain complete on Intel(R) Core-i3 with CPU 2.27 GHz processor, 4 GB RAM on Windows 10 at the work framework by using CloudSim with iFogSim as simulators on Eclipse integrated development environment. CloudSim is a very well-known and popular among simulators for cloud-based applications.

Various parameters such as encryption, decryption, energy consumption, network usage, network delay, trusted devices, and service management devices are compared. The same algorithms are implemented in real-time applications to solve the aforementioned issues. The results gathered from the simulations are very accurate. Codes are very consistent with real-time mechanisms. The simulators are redesigned according to the application need. The implementation period is a basic of the spell that is required to change a basic text to an encryption manuscript and vice versa, while encryption time that is referred to the time taken to change a basic text to a ciphertext and decryption which is referred to the time required to convert a cipher text to a plain text are both predicted to be short in instruction to take rapid and approachable system. Moreover, this execution time somehow is contingent on the layout of the system used. Table 13 offers the execution period in milliseconds (ms), which is obtained by computing the average encryption/decryption time after encrypting/decrypting the input text in 0.5 MB sizes while using the same key run on 16, 32, 64 and 128 bytes.

Table 13 presents the execution time test results in milliseconds (ms), which are attained by computing the average encryption/decryption time after encrypting/decrypting the input text in 0.5 MB sizes of the while using the same key run on 16, 32, 64, and 128 bytes. The results of Figures 7–9 specify that the existing AES has a minor rise in the encryption and decryption time after matched to the existing AES algorithm. Table 13 presents the time comparison between existing AES and different proposed AES algorithms for a string key.

Table 13: Execution time test result [13].

Plain text size (bytes)	AES	Avrg. encryption time (ms)	Avrg. decryption time (ms)
16	Existing AES	0.1658	0.1789
	Proposed AES	0.1190	0.1481
32	Existing AES	0.2976	0.3114
	Proposed AES	0.2507	0.2839
64	Existing AES	0.4564	0.4626
	Proposed AES	0.3916	0.4590
128	Existing AES	0.6984	0.5911
	Proposed AES	0.6014	0.5805
0.5	Existing AES	2359.65	2269.32
	Proposed AES	2159.8	2207.1

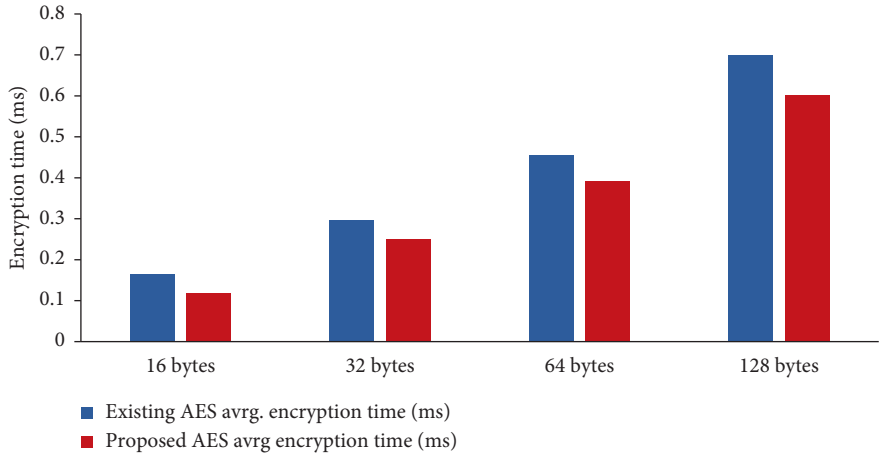


Figure 7: Encrypting time: existing AES vs. proposed AES.

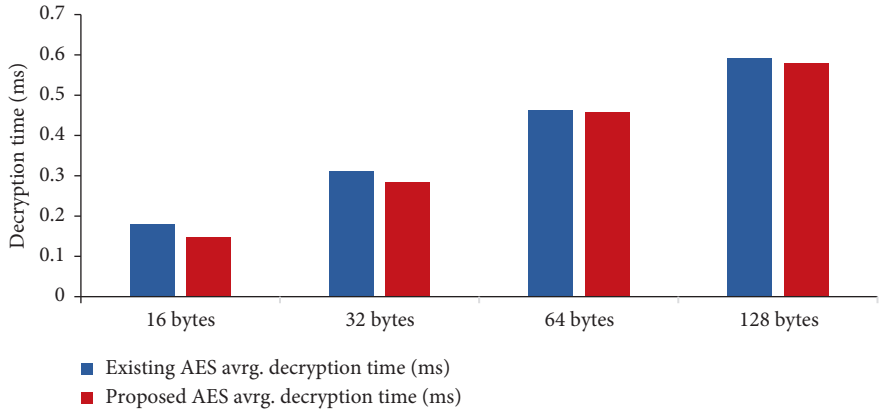


Figure 8: Decrypting time: existing AES vs. proposed AES.

5.1. Avalanche Effect. In cryptography, stuff called dispersal reproduces the cryptographic asset of an algorithm. If there is a small alteration in an input, the output changes meaningfully. This is also called the inundation effect. Avalanche consequence is leisurely by means of pretense reserve. Hamming reserve in material philosophy is the amount of variation. Playacting reserve is the amount of bit-by-bit XOR bearing in mind ASCII value as it develops informal to devise programmatically. A high gradation of dispersal, i.e., extraordinary avalanche consequence, is anticipated. Avalanche's conclusion reproduces the presentation of a cryptographic algorithm. The avalanche effect is described in Table 14.

The avalanche effect is described in Figure 10(simulation results from Table 14)

5.2. Comparative Analysis of Computed Results with the Existing Work. A comparative analysis of computed consequences with the current work is presented as

Table 14: Avalanche effect test result obtained after flipping a single bit in the plain text [13].

Execution program	Plain text	Secret key	Encryption and decryption time	Execution time
First time execution	I Love Unimorin!	H2 + 3S + MuePgIPK3h9SAHOtl6THtl8ak062IgB3ixEto	Encryption time Decryption time	0.05172414 0.03448276
Second execution	I Love Unimorin!	1mRVUf7lRS7W/K + BWFRkP3// KKjf0FtIaSnIGArvudY=	Encryption time Decryption time	0.06666667 0.044444446

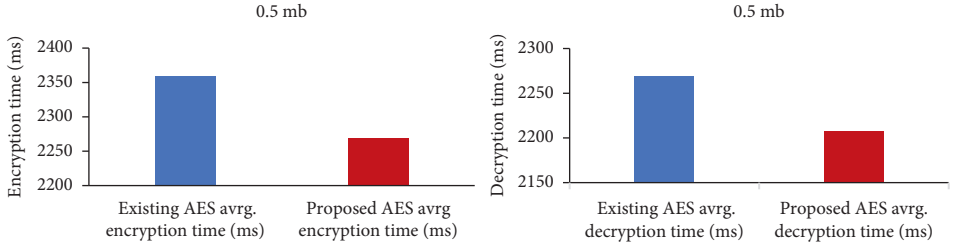


Figure 9: Encrypting and decrypting time: existing AES vs. proposed AES.

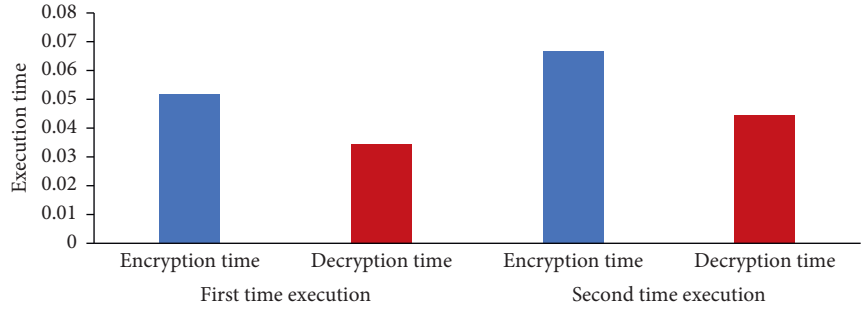


Figure 10: Avalanche effect test result.

follows. However, some researchers analyzed the performance of their advanced AES version. Meanwhile, many authors used encryption and description time as their performance metrics. The simulation environmental comparison between proposed AES and other AES using the CloudSim simulator is graphically represented in Figures 11 and 12.

5.3. Average Energy Consumed. By using the same technique described in [13], the energy consumption is being evaluated. These experiments shared that the proposed frameworks have 14% less energy consumption as compared to [13]. Actual cost taken is given by encryption and the average current that is used by every CPU clock cycle. Equation (5) is used to calculate energy cost per byte as well as various keys of AES encryption schemes:

$$: \sum E = E_c + \left(T_L - \frac{T_c}{T_u}\right) - \mathrm{P} * \mathrm{M}, \tag{5}$$

where C, L. and u represent the current, last, and updated, respectively.

The energy consumption E is the amount of work done on processing Mips M under a time frame T using power model P. The mathematical notation to represent the energy consumption is described in Figure 13.

5.4. Average Network Usage. Network usage is the overall network usage for the system. Network usage is represented in kilobytes. This parameter defines the usage of network resources. The length is reduced and approaches of requests to lower hierarchy by using service configuration so that the request could be processed in the lower hierarchy rather than sending it to cloud again and again. This algorithm reduces 3-hop communication to single-hop communication. Thus, low network usage is obtained through the proposed framework. The more the network is used, the more the expenditure. Efficient network topologies prefer to use minimal network. In these experiments, the network usage is evaluated using the same technique described in [13]. In the proposed framework, network resources are reduced by 11% as compared to [13]. The network while running the implemented encryption schemes is calculated using the following equation:

$$: \sum Nu = Ni + \frac{(L * D * B)}{T}, \tag{6}$$

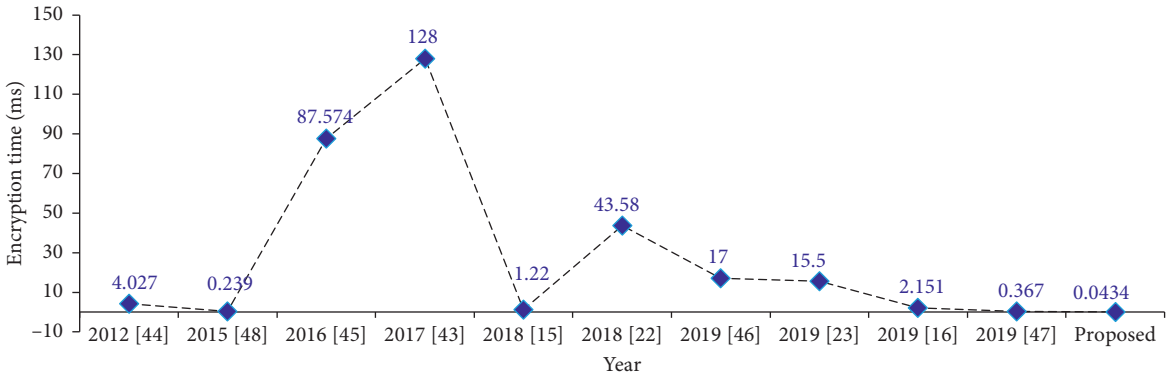


Figure 11: Encryption processing time factor in different AES.

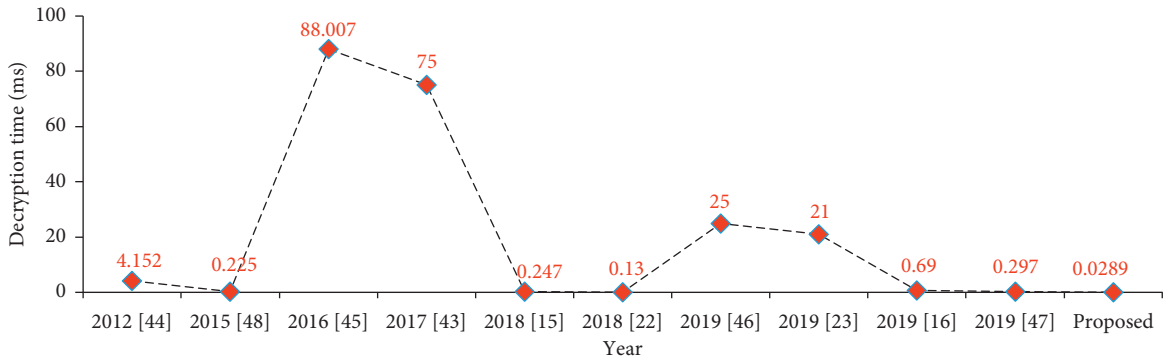


Figure 12: Decryption processing time factor in different AES.

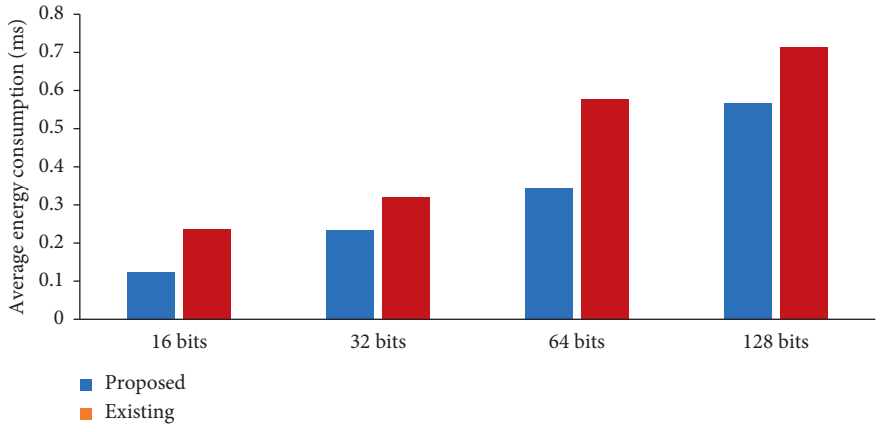


Figure 13: Energy consumption for different key AES encrypting and decrypting.

where *Ni* is the initial network usage (*Nu* at 0).

The network usage mathematical notation *Nu* is the number of bits *B* communicated in a certain time frame on devices under sets of data *D* with latency *L*. Simulation result is clear from Figure 14.

5.5. Average Networking Delay. In the calculation of testing and evaluating whether the data are secured, delay is likewise evaluated. In the local host cloud environment amount of consumers; the data traffic will develop tall, which will have influence on the scheme. In a real environment, numerous issues could cause delays, e.g., the size of the key and network speeds, which will cause suspensions and overcrowding. The larger numbers of key indicate increased delay due to the time when more data encrypt generate. When the key, it is originally split into dissimilar blocks formerly encryption. The scope of individual block may have contingent influence

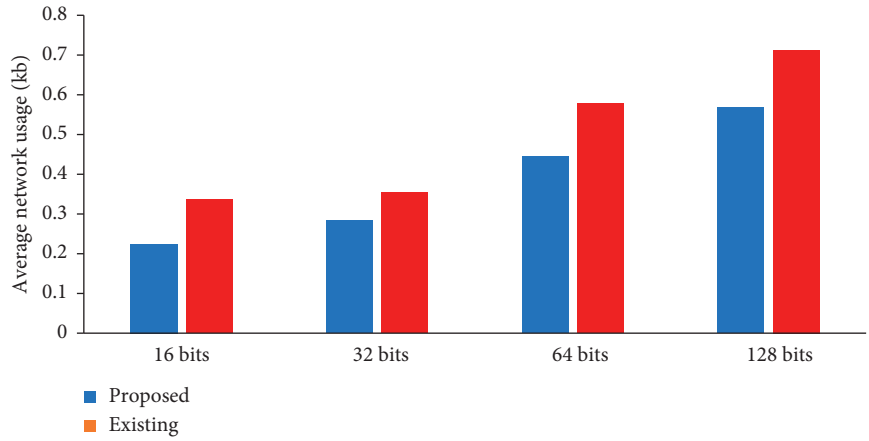


Figure 14: Network usage for different key AES encrypting and decrypting.

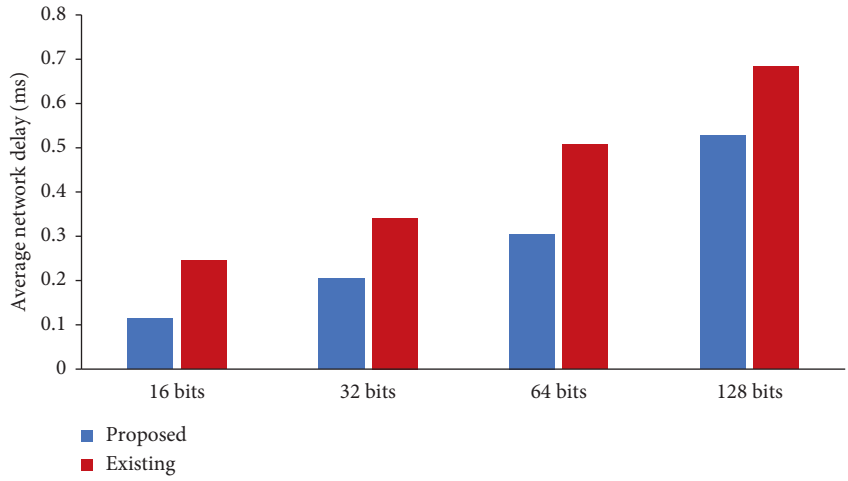


Figure 15: Networking delay for different key AES encrypting and decrypting.

on the scope. The delay comparison of the previous methodology [13] and the research shows that the significant differences in the delay indicate that the proposed framework is 15% better than the previous solution [13].

$$\sum D_n = B_s * \frac{L}{T} - B_d * \frac{L}{1} - \frac{T}{T_e}. \quad (7)$$

The delay D represents the time that the bits B take to reach a processing device from an end device under a certain latency L and connection time T. The observed delay is calculated using the equation. The mathematical notation to represent the delay is described below and by simulation result it is clear from figure. The delay calculation is shown in Figure 15.

6. Conclusion

To provide data confidentiality and information integrity of users' data in the cloud computing environment, an effective security framework is proposed that provides a mechanism through which communication is protected and unauthorized access is restricted. The proposed security framework allows cloud users to securely handle the privacy and integrity of data.

It also allows security, privacy, network usage, and storage in the cloud without depending on the plausibility of the cloud provider. The application of the AES algorithm provides a strong foundation that protects data stored in the cloud as well as authorizes access to data only on successful authentication and verification. The delays that occur in the actual environment vary in different situations all of which are not considered in this framework. Results show that the proposed framework minimizes energy consumption by 14.43%, network usage by 11.53%, and delay by 15.67%. Hence, the proposed framework enhances security, minimizes resource utilization, and reduces delay while deploying services of computational clouds.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G.S. Mahmood, J. H. Dong, and B. A. rahman Jaleel, "Achieving an effective, confidentiality and integrity of data in cloud computing," *International Journal of Network Security*, vol. 21, no. 2, pp. 326–332, 2019.

[2] S. Othman and A. S. Riaz, "A user-based trust model for cloud computing environment," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, 2018.

[3] A. Firman, A. N. Hidayanto, and P. Harjanto, "Critical components of security framework for cloud computing community: a systematic literature review," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 18, pp. 3345–3358, 2018.

[4] K. V. Pradeep, V. Vijayakumar, and V. Subramaniyaswamy, "An efficient framework for sharing a file in a secure manner using asymmetric key distribution management in cloud environment," *Journal of Computer Networks and Communications*, vol. 2019, Article ID 9852472, 8 pages, 2019.

[5] Dr. Ramalingam Sugumar and K. Arul Marie Joycee, "FEDSACE: a framework for enhanced user data security algorithms in cloud computing environment," *International Journal on Future Revolution in Computer Science & Communication Engineering*, vol. 4, no. 3, 2018.

[6] M. Kpelou and K. Kishore, "Lightweight security framework for data outsourcing and storage in mobile cloud computing," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, 2019.

[7] R. Ganga Sagar and N. Ashok Kumar, "Encryption based framework for cloud databases using AES algorithm," *International Journal of Research Studies in Computer Science and Engineering*, vol. 2, no. 6, 2015.

[8] J. R. Jain and A. Abu, "A novel data logging framework to enhance security of cloud computing," in *Proceedings of the SoutheastCon 2016,* IEEE, Norfolk, VA, USA, April 2016.

[9] J. Singh, "Framework for client side AES encryption technique in cloud computing," *IJIRMPS*, vol. 6, no. 5, 2018.

[10] J. Y. Gudapati Syam Prasad, S. sunil kumar, and A. Keerthi, "Integration of searching and AES encryption in cloud computing," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 4, 2019.

[11] I. A. Elgendy, W.-Z. Zhang, C.-y. Liu, and C.-H. hsu, "An efficient and secured framework for mobile cloud computing," *IEEE Transactions on Cloud Computing*, 2018.

[12] R. Saha, G. Geetha, G. Kumar, and T.-h. Kim, "RK-AES: an improved version of AES using a new key generation process with random keys," *Security and Communication Networks*, vol. 2018, Article ID 9802475, 11 pages, 2018.

[13] O. C. Abikoye, A. D. Haruna, A. Abubakar, N. O. Akande, and E. O. Asani, "Modified advanced encryption standard algorithm for information security," *Symmetry*, vol. 11, no. 12, p. 1484, 2019.

[14] K.-L. Tsai, Y.-L. Huang, F.-Y. Leu, I. You, Y.-L. Huang, and C.-H. Tsai, "AES-128 based secure low power communication for LoRaWAN IoT environments," *IEEE Access*, vol. 6, pp. 45325–45334, 2018.

[15] M.V. C. Suana, A. M. Sison, C. Aragon, and R. P. Medina, "Enhancement of advanced encryption standard (AES) cryptographic strength via generation of cipher key-dependent S-box," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 6, no. 4, 2018.

[16] S. NurRachmat, "Performance analysis of 256-bit AES encryption algorithm on android smart phone," *IOP Conf. Series: Journal of Physics: Conf. Series*, vol. 1196, 2019.

[17] J. Silki and V. Abhilasha, "An improved security framework for cloud environment using ECC algorithm," *International Journal for Research in Applied Science & Engineering Technology*, vol. 6, no. 1, 2018.

[18] A. Oussama and Z. Abdelha, "A security framework for cloud data storage (CDS) based on agent," *Applied Computational Intelligence and Mathematical Methods*, Springer, Berlin, Germany, 2019.

[19] H. J. Muhasin, R. Atan, M.A. Jabar, and S. Abdullah, "Cloud computing sensitive data protection using multi layered approach," in *Proceedings of the 2016 2nd International Conference on Science in Information Technology (ICSITech)*, pp. 69–73, Balikpapan, Indonesia, October 2016.

[20] K. Ravi and K. B. Rajesh, "Quality based cloud service broker for optimal cloud service provider selection," *International Journal of Applied Engineering Research*, vol. 12, no. 18, pp. 7962–7975, 2017.

[21] M. Adelmeyer, M. Walterbusch, B. Peter, and T. Frank, *Trust Transitivity and Trust Propagation in Cloud Computing Ecosystems*, Lawrence Erlbaum Associates, Mahwah, NJ, USA, 2018.

[22] F. Meng, R. Lin, Z. Wang, H. Zou1, and S. Zhou, "A multi-connection encryption algorithm applied in secure channel service system," *EAI Endorsed Transactions on Security and Safety*, vol. 5, no. 15, 2018.

[23] H. A. Al Essa and A. S. Ashoor, "Enhancing performance of AES algorithm using concurrency and multithreading," *ARPN Journal of Engineering and Applied Sciences*, vol. 14, no. 11, 2019.

[24] M. Marwan, A. Kartit, and H. Ouahmane, "A framework to secure medical image storage in cloud computing environment," *Journal of Electronic Commerce in Organizations*, vol. 16, no. 1, pp. 1–16, 2018.

[25] P. Sirohi and A. Agarwal, "Cloud computing data storage security framework relating to data integrity, privacy and trust," in *Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, pp. 4-5, Dehradun, India, September 2015.

[26] K. Subramanian, F. L. John, and F. L. John, "Dynamic and secure unstructured data sharing in multi-cloud storage using the hybrid crypto-system," *International Journal of Advanced and Applied Sciences*, vol. 5, no. 1, pp. 15–23, 2018.

[27] M. Edjie, D. L. Reyes, M. Ariel, Sison, and Dr.R. P. Medina, "Modified AES cipher round and key schedule," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 7, no. 1, March 2019.

[28] H. Talirongan, A. M. Sison, and R. P. Medina, "A new advanced encryption standard-butterfly effect in protecting image of copyright piracy," in *Proceedings of the Proceedings of the 6th International Conference on Information Technology: IoT and Smart City*, Hong Kong, China, December 2018.

[29] F. A. Hany, J. W. Robert, and B. W. Gary, "Fog computing and the internet of things: a review," *Big Data Cognitive Computer*, vol. 2, no. 2, 2018.

[30] A. Arab, M. J. Rostami, and B. Ghavami, "An image encryption method based on chaos system and AES algorithm," *The Journal of Supercomputing*, vol. 75, no. 10, pp. 6663–6682, 2019.

[31] M. Oqail Ahmad and R. Z. Khan, "Cloud computing modeling and simulation using CloudSim environment,"

International Journal of Recent Technology and Engineering (IJRTE) ISSN, vol. 8, no. 2, 2019.

[32] V. Surya, S. Ranichandra, and R. Ranjani, "Secure cloud storage using AES encryption," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 6, no. 6, 2018.

[33] A. Nair and S. S. SantoshAnand, "A performance booster for load balancing in cloud computing with my load balancer technique," *International Journal of Recent Technology and Engineering*, vol. 8, no. 1, 2019.

[34] D. Salama and A. Elminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *IJEIE*, vol. 8, no. 1, pp. 40–42, 2018.

[35] D.-H. Bui, D. Puschini, S. Bacles-Min, E. Beigné, and X.-T. Tran, "Ultra low-power and low-energy 32-bit datapath AES architecture for IoT applications," in *Proceedings of the 2016 International Conference on IC Design and Technology (ICICDT)*, pp. 1–4, Ho Chi Minh City, Vietnam, June 2016.

[36] H. Jia, X. Liu, X. Di et al., "Security strategy for virtual machine allocation in cloud computing," *Procedia Computer Science*, vol. 147, pp. 140–144, 2019.

[37] B. T. Spiers, M. Halas, R. A. Schimmel, and D. P. Provencher, "Secure network cloud architecture," U.S. Patent 8,984,610, United States Patent (Justia Patents), 2015.

[38] E. Bertino, F. Paci, R. Ferrini, and N. Shang, "Privacy-preserving digital identity management for cloud computing," *IEEE Data Engineering Bulletin*, vol. 32, no. 1, pp. 21–27, 2009.

[39] S. Yi, Li Cheng, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, pp. 37–42, ACM, Hangzhou, China, June 2015.

[40] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proceedings of the 2014 International Conference on Future Internet of Things and Cloud*, pp. 464–470, IEEE, Barcelona, Spain, August 2014.

[41] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. &Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[42] G. N. Selimis, A. P. Kakarountas, A. P. Fournaris, A. Milidonis, and O. Koufopavlou, "A low power design for sbox cryptographic primitive of advanced encryption standard for mobile end-users," *Journal of Low Power Electronics*, vol. 3, no. 3, pp. 327–336, 2007.

[43] M. A. FaiqaMaqsood, M. M. Ali, and M. Ali Shah, "Cryptography: a comparative analysis for modern techniques", (IJACSA)," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, 2017.

[44] R. Paul, S. Saha, S. Sau, and A. Chakrabarti, "Design and implementation of realtime AES-128 on real time operating system for multiple fpga communication," 2012, http://arxiv.org/abs/1205.2153.

[45] D. Lohit Kumar, Dr.A. R. Reddy, and S. A. K. Jilani, "Implementation of 128-bit AES algorithm in MATLAB," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 33, no. 3, 2016.

[46] Dr. N. Suba Rani, Dr. A. Noble Mary Juliet, and K. Renuka Devi, "An image encryption & decryption and comparison with text - AES algorithm," *International Journal of Scientific & Technology Research*, vol. 8, no. 7, 2019.

[47] O. I. Omotosho, "A review on cloud computing security," *International Journal of Computer Science and Mobile Computing, IJCSMC*, vol. 8, no. 9, pp. 245–257, 2019.

[48] L. R1 and H. S2 Mohan, "Implementation and performance analysis of modified AES algorithm with key-dependent dynamic S-box and key multiplication," *Computer Applications Research*, vol. 5, no. 3, 2015.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8829595, 14 pages
https://doi.org/10.1155/2020/8829595

Research Article

Evaluating Security of Internet of Medical Things Using the Analytic Network Process Method

Xucheng Huang[1] **and Shah Nazir**[2]

[1]*School of Finance, Shanghai Lixin University of Accounting and Finance, Shanghai 201209, China*
[2]*Department of Computer Science, University of Swabi, Swabi, Pakistan*

Correspondence should be addressed to Xucheng Huang; huangxucheng@lixin.edu.cn

Received 23 June 2020; Revised 2 July 2020; Accepted 24 July 2020; Published 1 September 2020

Academic Editor: Amir Anees

Internet of Medical Things (IoMT) plays an important role in healthcare. Different devices such as smart sensors, wearable devices, handheld, and many other devices are connected in a network in the form of Internet of Things (IoT) for the smooth running of communication in healthcare. Security of these devices in healthcare is important due to its nature of functionality and efficiency. An efficient and robust security system is in dire need to cope with the attacks, threats, and vulnerability. The security evaluation of IoMT is an issue since couple of years. Therefore, the aim of the proposed study is to evaluate the security of IoMT by using the analytic network (ANP) process. The proposed approach is applied using ISO/IEC 27002 (ISO 27002) standard and some other important features from the literature. The results of the proposed research demonstrate the effective IoMT components which can further be used as secure IoMT.

1. Introduction

Internet of Things has several applications in the daily life and has made life very easy. From industry to education, healthcare, and other places, the IoT is mostly used. Internet of Medical Things is the advanced version of IoT which has a key role in healthcare. Devices such as wearable, handheld, sensors, actuator, and others are connected for communication through Internet. For the smooth communication of these devices, security is important to run in an effective and efficient way. Security is the protection from unauthorized access of illegal users. In healthcare, the devices are sometimes connected through heterogeneous environment with the support of different IoT devices. So, the security evaluation is important for them to ensure that the communication is safe and secure. IoMT plays an important role in remote exchange data processes. The IoT devices have limited capabilities due to low processing, tiny memory, and limited storage, so implementing security will be a challenging task. The security and privacy in IoMT devices are vital due to a number of reasons as IoMT devices are ubiquitous and their applications are employed in health. For this purpose, reinforcing a security mechanism is indispensable to cope with these attacks, vulnerabilities, and security and privacy challenges. Security can be one of the important factors for IoHT [1–6].

The existing research regarding the security of IoMT covers different aspects. However, there is a lack of knowledge that how to evaluate the security of IoMT based on security attributes and features. So, to overcome this limitation, the proposed research presents the ANP approach for the evaluation of security of IoMT in term of the ISO/IEC 27002 (ISO 27002) standard, and some other important features identified from the literature. The ANP method incorporates the criteria given for achieving the goal based on the available alternatives. This method helps in situation when complexity arises.

The organization of the paper is as follows: Section 2 presents the related work to the security evaluation of IoMT, along with the existing approaches for security evaluations are discussed. In Section 3, the research method is briefly described. Section 4 concludes the paper.

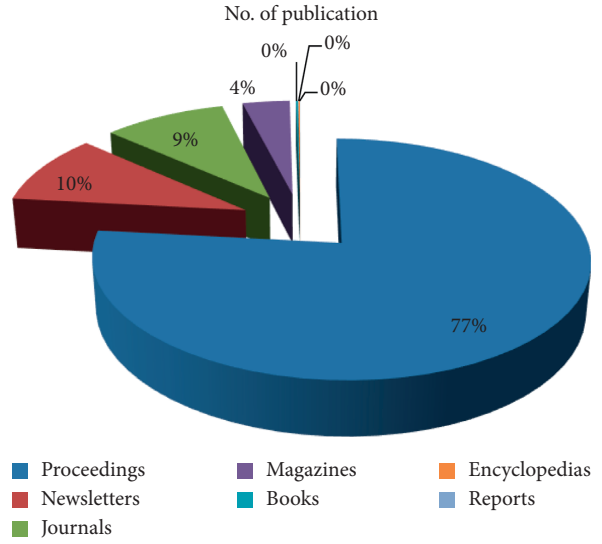


Figure 1: Type of publications along with the total number of papers.

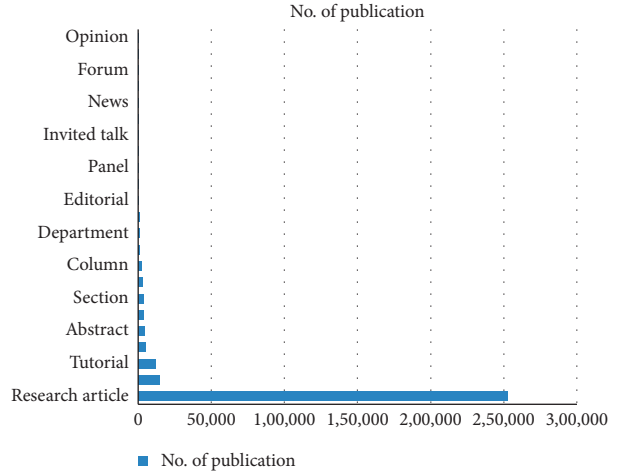


Figure 2: Content type along with the total number of publications.

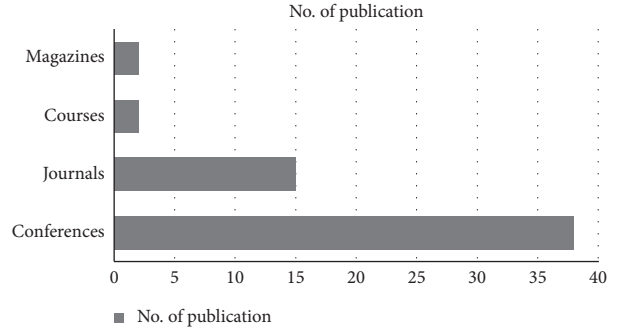


Figure 3: Type of publication with the total number of papers.

2. Related Work

Several approaches have been used by researchers for the evaluation security. The basic security requirements are defined in confidentiality, integrity, and availability (the CIA model) [7–12]. The IoMT devices are vulnerable to several threats of security, attacks, and vulnerabilities. IoMT devices suffer from enormous security threats due to low cost and power unlike traditional desktop and mobile devices. The malware can replicates itself by compromising the

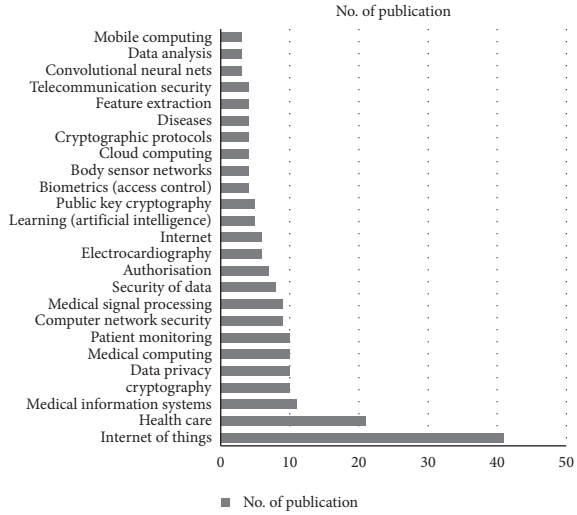


Figure 4: Publication topic with the number of papers published.

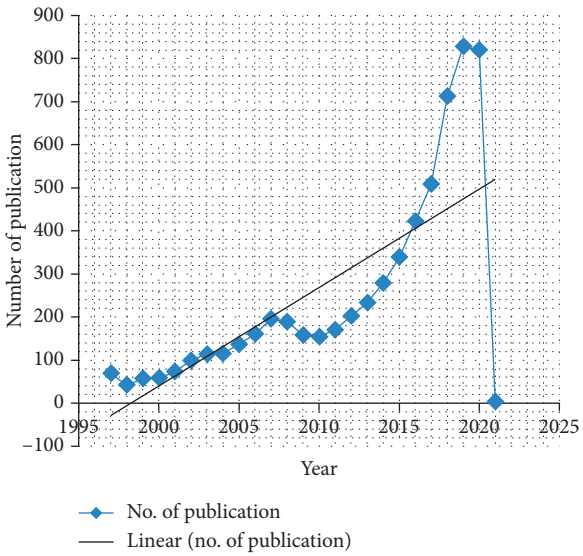


Figure 5: Number of papers published in the given year.

connection that links IoT devices [13]. Different frameworks, models, reviews, surveys, and analysis pertaining to the security of IoT-based systems for security analysis are used. Frustaci et al. [14] evaluated IoT security issues at three different layers of IoT such as perception, transportation, and application. Leister et al. [15] evaluated the security of IoT in e-health by presenting a scenario-based framework. Alrawi et al. [16] proposed component-based analysis such as IoT device, mobile application, communication channel, and cloud end points for the home-based IoT system. Tekeoglu and Tosun [17] presented a layer-based packet capturing framework for investigating security and privacy of IoT devices. Cherneyshev and Hannay [18] evaluated IoT security by using two smart TVs against the multisurface attacks. Ali and Awad [19] assessed the security of IoT smart home in terms of vulnerability. Mazhelis and Tyrväinen [20] evaluated IoT platforms from application provider perspectives. Apart from these approaches, several other approaches are being available in the literature [21–24].

Similarly, mobile computing services can be used in IoT by using services of mobile phones and apps or through the M-Health care system. The M-Health contributes to the IoT by furnishing various services such as compactness, IP connectivity, consumption of low power, and security [25]. Recently, many applications have been developed to deliver mobile-based services to the users in healthcare. The applications of smart phone enable the patients to know about their diseases after the analysis in the field of gynaecology and paediatrics [26].

The purpose of this section is to study the existing literature to know about the work done in the area of security evaluation. For this purpose, the popular libraries including ACM, IEEE, ScienceDirect, and Springer were searched. Different types of information were obtained, and the details

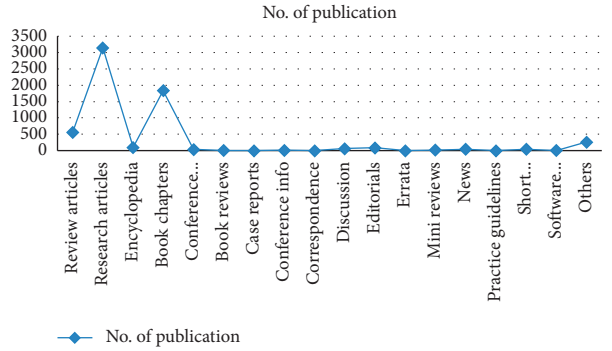


Figure 6: Number of publication and article type.

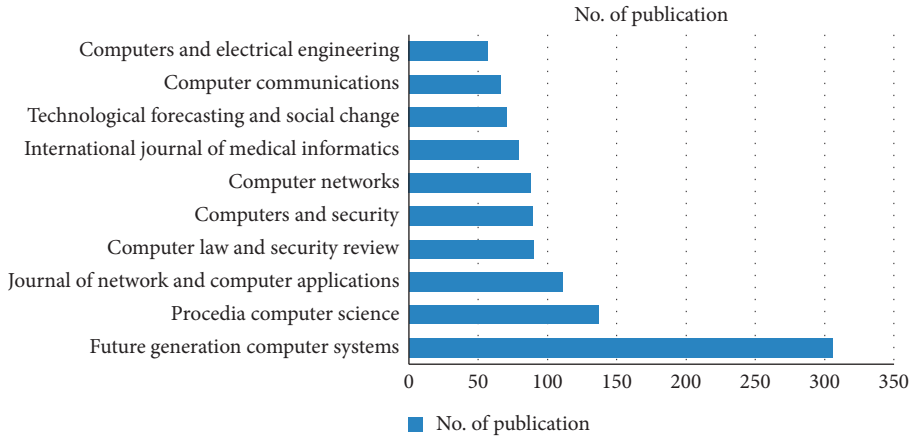


Figure 7: Publication title along with the number of papers.

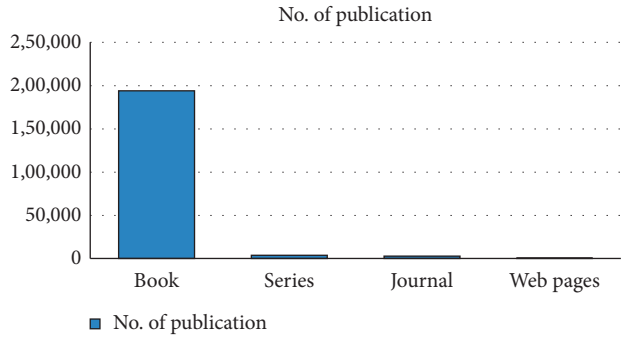


Figure 8: Number of publication with the type of publication.

are given in figures and tables in this section. Figure 1 shows the type of publication along with the total number of papers published in the ACM library.

Figure 2 shows the content type along with the total number of publications.

The purpose of searching different libraries was to know more about the research done in the area. For this purpose, the IEEE library was also searched. Figure 3 shows the type of publication along with the total number of papers published in the IEEE library.

Figure 4 shows the publication topic in the area along with the total number of papers published.

The library of ScienceDirect was also searched to know about the security-related work published in the area. Figure 5 shows the total number of publications in the given year in the ScienceDirect library.

Figure 6 shows the number of publications along with the type of publication.

Figure 7 shows the publication title along with the number of publications.

Figure 9: Publication topic along with the number of publications.

Table 1: Random consistency index.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

The value of CR should be less than 0.1.

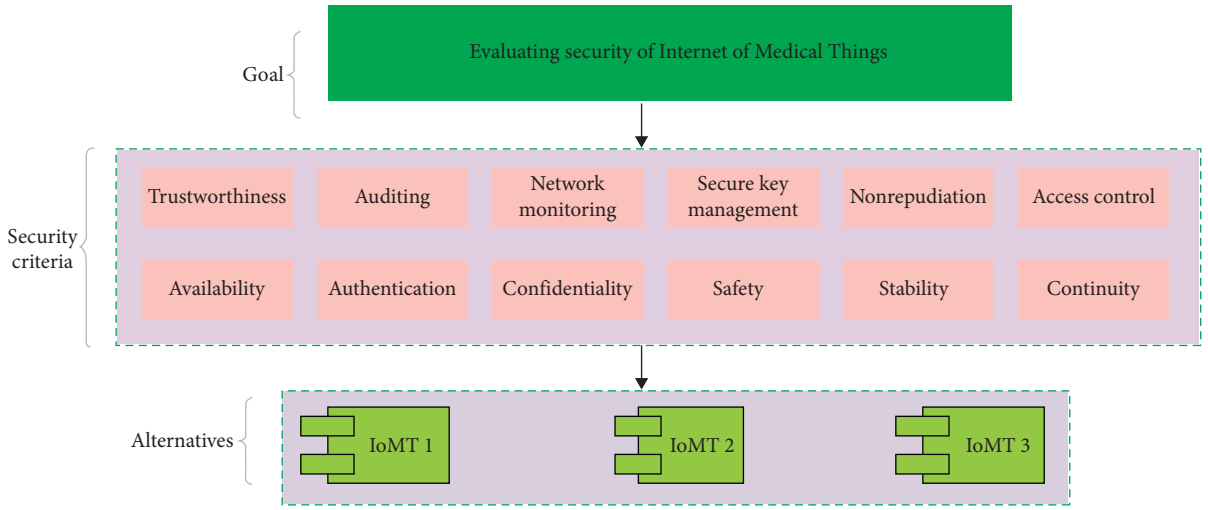


Figure 10: Goal, security criteria, and alternatives of the proposed research. The overall process of the proposed research is shown in Figure 11.

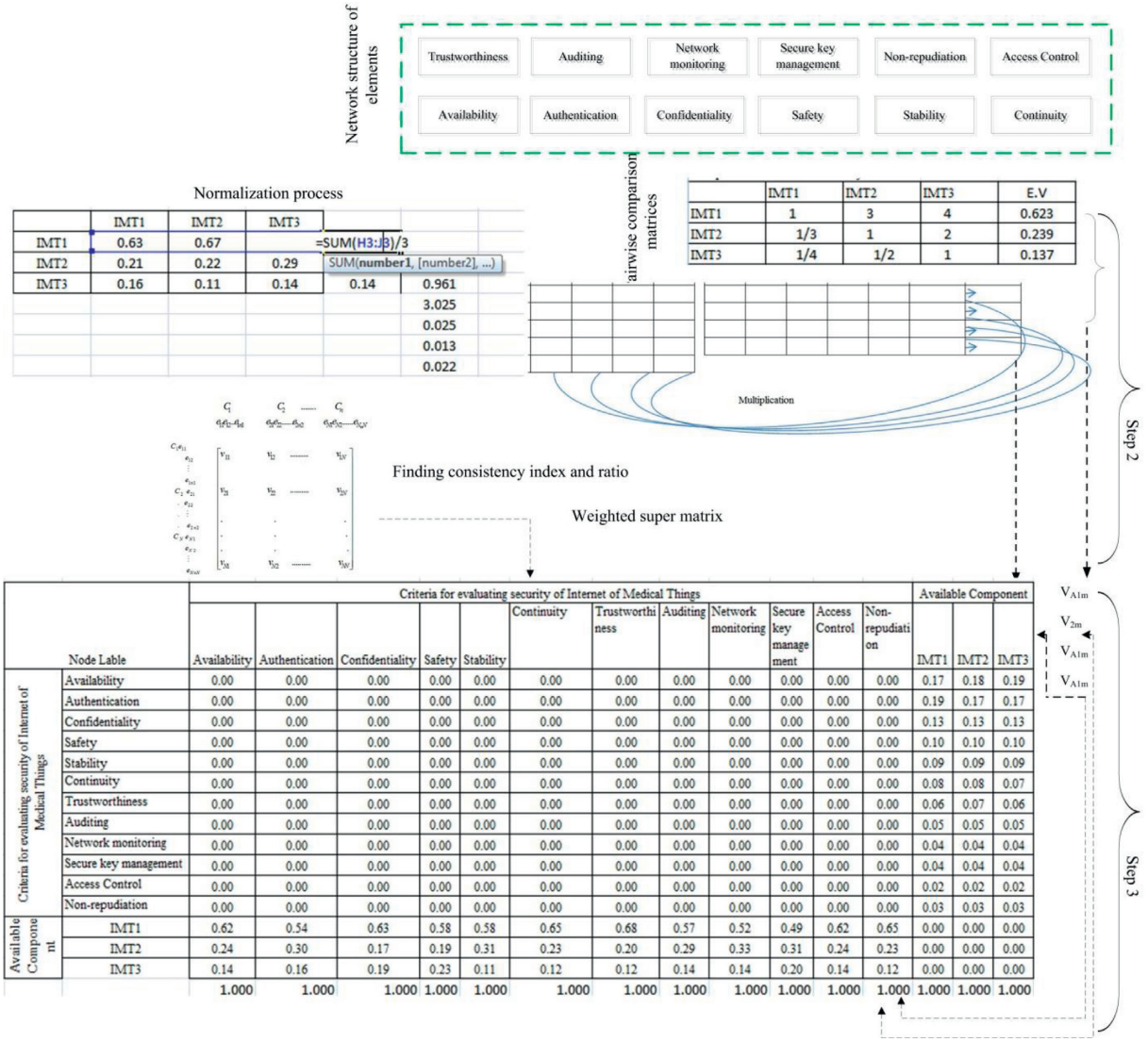


Figure 11: Process of the ANP for the proposed research.

Table 2: List of selected attribute.

Requirements	[7]	[8]	[9]	[10]	[11]	[12]	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[39]	[40]	[41]	[42]
Confidentiality	*	*	*	*	*	*		*	*								*	
Authentication	*		*		*		*			*	*	*	*	*	*		*	*
Integrity	*	*	*	*	*	*		*			*	*						
Availability	*	*	*	*	*	*		*							*			
Authorization				*	*		*	*			*					*		
Physical security						*			*					*				
Continuity						*												
Trustworthiness	*				*													
Auditing	*																	
Network monitoring														*				
Secure key management			*											*				
Access control	*				*		*			*	*		*		*		*	*
Nonrepudiation					*	*		*			*							

Table 3: Comparison with respect to IoMT1.

	Availability	Authentication	Confidentiality	Safety	Stability	Continuity	Trustworthiness	Auditing	Network monitoring	Secure key management	Access control	Non-repudiation	EV
Availability	1	1/2	2	3	4	2	3	2	7	2	9	5	0.167
Authentication	2	1	2	4	3	3	2	5	3	5	7	2	0.189
Confidentiality	1/2	1/2	1	3	2	2	2	2	7	2	7	5	0.130
Safety	1/3	1/4	1/3	1	1/2	2	3	5	3	5	3	5	0.104
Stability	1/4	1/3	1/2	2	1	2	2	2	3	2	7	2	0.090
Continuity	1/2	1/3	1/2	1/2	1/2	1	3	3	2	2	3	5	0.082
Trustworthiness	1/3	1/2	1/2	1/3	1/2	1/3	1	2	3	2	4	3	0.064
Auditing	1/2	1/5	1/2	1/5	1/2	1/3	1/2	1	2	3	3	2	0.051
Network monitoring	1/7	1/3	1/7	1/3	1/3	1/2	1/3	1/2	1	2	2	2	0.036
Secure key management	1/2	1/5	1/2	1/5	1/2	1/2	1/2	1/3	1/2	1	2	2	0.038
Access control	1/9	1/7	1/7	1/3	1/7	1/3	1/4	1/3	1/2	1/2	1	2	0.022
Non-repudiation	1/5	1/2	1/5	1/5	1/2	1/5	1/3	1/2	1/2	1/2	1/2	1	0.028

CR = 0.985.

Table 4: Comparison with respect to IoMT2.

	Availability	Authentication	Confidentiality	Safety	Stability	Continuity	Trustworthiness	Auditing	Network monitoring	Secure key management	Access control	Non-repudiation	EV
Availability	1	3	2	2	4	2	3	2	7	2	5	2	0.179
Authentication	1/3	1	2	4	3	3	2	5	3	5	7	2	0.167
Confidentiality	1/2	1/2	1	3	2	2	2	2	7	2	7	5	0.132
Safety	1/2	1/4	1/3	1	1/2	2	3	2	3	5	3	5	0.099
Stability	1/4	1/3	1/2	2	1	2	2	2	3	2	7	2	0.092
Continuity	1/2	1/3	1/2	1/2	1/2	1	2	3	2	2	3	5	0.080
Trustworthiness	1/3	1/2	1/2	1/3	1/2	1/2	1	2	3	2	4	4	0.068
Auditing	1/2	1/5	1/2	1/2	1/2	1/3	1/2	1	2	3	3	2	0.055
Network monitoring	1/7	1/3	1/7	1/3	1/3	1/2	1/3	1/2	1	2	2	2	0.035
Secure key management	1/2	1/5	1/2	1/5	1/2	1/2	1/2	1/3	1/2	1	2	2	0.039
Access control	1/5	1/7	1/7	1/3	1/7	1/3	1/4	1/3	1/2	1/2	1	2	0.023
Non-repudiation	1/2	1/2	1/5	1/5	1/2	1/5	1/4	1/2	1/2	1/2	1/2	1	0.031

CR = 0.994.

Table 5: Comparison with respect to IoMT3.

	Availability	Authentication	Confidentiality	Safety	Stability	Continuity	Trustworthiness	Auditing	Network monitoring	Secure key management	Access control	Non-repudiation	EV
Availability	1	3	2	3	4	2	5	2	3	2	7	5	0.192
Authentication	1/3	1	2	4	3	5	2	5	3	5	7	2	0.173
Confidentiality	1/2	1/2	1	3	2	2	2	2	7	2	7	5	0.132
Safety	1/3	1/4	1/3	1	1/2	2	3	5	3	3	3	5	0.098
Stability	1/4	1/3	1/2	2	1	2	2	2	3	2	7	2	0.089
Continuity	1/2	1/5	1/2	1/2	1/2	1	3	3	2	2	3	2	0.073
Trustworthiness	1/5	1/2	1/2	1/3	1/2	1/3	1	2	3	2	2	3	0.059
Auditing	1/2	1/5	1/2	1/5	1/2	1/3	1/2	1	2	3	3	2	0.054
Network monitoring	1/3	1/3	1/7	1/3	1/3	1/2	1/3	1/2	1	2	2	2	0.039
Secure key management	1/2	1/5	1/2	1/3	1/2	1/2	1/2	1/3	1/2	1	2	2	0.040
Access control	1/7	1/7	1/7	1/3	1/7	1/3	1/2	1/3	1/2	1/2	1	2	0.023
Non-repudiation	1/5	1/2	1/5	1/5	1/2	1/2	1/3	1/2	1/2	1/2	1/2	1	0.028

CR = 0.10.

Table 6: For availability.

	IMT1	IMT2	IMT3	EV
IMT1	1	3	4	0.623
IMT2	1/3	1	2	0.239
IMT3	1/4	1/2	1	0.137

CR = 0.022.

Finally, the library of Springer was searched for the detail information in the area. Figure 8 shows the number of publications with the type of publications in the Springer library.

Figure 9 shows the article topic along with the total number of publications.

3. Applications of the Analytic Network Process for Evaluating Security of Internet of Medical Things

The analytic network process has several applications in different areas [11, 24, 27–29]. The reason behind using this method was to evaluate the security of IoMT, as this method works very well in situation where complexity exists. In the proposed research work, the analytic network process approach is used for security evaluation of Internet of Medical Things. The ANP method incorporates the criteria given for achieving the goal based on the available alternatives. This method helps in situation when complexity arises. The method adopted the ISO standard of security along with the identified security features from the literature. The ANP method consists of three parts: (a) the goal, (b) criteria, and (c) alternatives. Details regarding the ANP can be found in [30]; however, the following are the main steps:

(a) A particular phenomenon is to be divided into subparts

(b) A qualitative scale of measure is applied while this can be converted into a quantitative scale between 1 and 9

(c) The pairwise comparison is done for all the criteria along with alternatives

(d) The relative importance is found by finding the principal eigenvalue and the related eigenvector of the comparison matrix

(e) The consistency of matrix is measured

Priority vector "w" is calculated as follows:

$$A_w = \lambda_{\max} w. \tag{1}$$

$\lambda_{\max}$ is the major eigenvalue of the matrix "A," and "w" is its eigenvector. The value of "λ" is obtained by summing the column of the original matrix multiplied by the normalized EV. The principal EV is obtained by the sum of all "λ".

The "consistency index (CI)" and "consistency random (CR)" of the pairwise comparison matrix are computed by the following equation:

$$C_i = \frac{\lambda_{\max} - n}{n-1},$$
$$CR = \frac{CI}{RI}. \tag{2}$$

The random consistency index (RI) table is given by Saaty and is shown in Table 1 [30].

(f) Construction of the supermatrix

(g) Conversion of the weighted supermatrix into the limit matrix for making the decision

(h) Deciding the most appropriate alternative from the limit matrix

Figure 10 shows the goal, criteria, and alternatives of the proposed research.

(i) Identification of attributes and scoring process: the process of identification of attributes was very tricky due to the reason that important attribute should be missed. For this purpose, the literature was searched and finally the attributes of the International Standard Organization (ISO) information security standard such as ISO/IEC 27000-series (ISO/IEC, 2018) along with 8 important attributes from the literature were identified. ISO/IEC 27000-series (ISO/IEC, 2018) is a well-known standard and widely accepted standard [12].

Table 2 shows the list of selected attributes.

After selecting the attributes for security evaluation, these attributes were shared with the experts in the field. The reason of sharing was to gather appropriate score for each component with respect to the defined attribute. Assigning the score to the relevant attribute was based on the expertise of the expert. Table 3 shows the comparison with respect to IoMT1.

Table 4 shows the comparison with respect to IoMT2.

Table 5 shows the comparison with respect to IoMT3.

Table 6 shows the comparison with respect to IoMT.

The rest of the calculations for the remaining attributes to IoMT were done the same as Table 6. After pairwise comparisons, all the calculations were brought together into the weighted supermatrix for the purpose to convert it into the limit matrix for decision-making about security evaluation. Table 7 shows the weighted supermatrix.

The weighted matrix was converted into the limit matrix by taking the power of the weighted matrix. This process was done till all the elements of each row become the same. The reason was to make decision based on the limit matrix. Table 8 shows the limit matrix.

Table 7: Weighted supermatrix.

Node label		Criteria for evaluating security of Internet of Medical Things												Available component		
		Availability	Authentication	Confidentiality	Safety	Stability	Continuity	Trustworthiness	Auditing	Network monitoring	Secure key management	Access control	Nonr-epudiation	IoMT1	IoMT2	IoMT3
Criteria for evaluating security of Internet of Medical Things	Availability	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.18	0.19
	Authentication	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19	0.17	0.17
	Confidentiality	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.13	0.13
	Safety	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.10	0.10
	Stability	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.09	0.09
	Continuity	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.08	0.07
	Trustworthiness	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.07	0.06
	Auditing	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.05	0.05
	Network monitoring	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.04
	Secure key management	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.04
	Access control	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.02
	Non-repudiation	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.03
Available component	IoMT1	0.62	0.54	0.63	0.58	0.58	0.65	0.68	0.57	0.52	0.49	0.62	0.65	0.00	0.00	0.00
	IoMT2	0.24	0.30	0.17	0.19	0.31	0.23	0.20	0.29	0.33	0.31	0.24	0.23	0.00	0.00	0.00
	IoMT3	0.14	0.16	0.19	0.23	0.11	0.12	0.12	0.14	0.14	0.20	0.14	0.12	0.00	0.00	0.00

Table 8: Limit matrix.

Node label		Criteria for evaluating security of Internet of Medical Things												Available component		
		Availability	Authentication	Confidentiality	Safety	Stability	Continuity	Trustworthiness	Auditing	Network monitoring	Secure key management	Access control	Non-repudiation	IoMT1	IoMT2	IoMT3
Criteria for evaluating security of Internet of Medical Things	Availability	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.00	0.00	0.00
	Authentication	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.00	0.00	0.00
	Confidentiality	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.00	0.00	0.00
	Safety	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.00	0.00	0.00
	Stability	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.00	0.00	0.00
	Continuity	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.00	0.00	0.00
	Trustworthiness	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.00	0.00	0.00
	Auditing	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.00	0.00	0.00
	Network monitoring	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.00	0.00	0.00
	Secure key management	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.00	0.00	0.00
	Access control	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.00	0.00	0.00
	Non-repudiation	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.00	0.00	0.00
Available component	IoMT1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.60	0.60
	IoMT2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.25	0.25
	IoMT3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.16	0.16

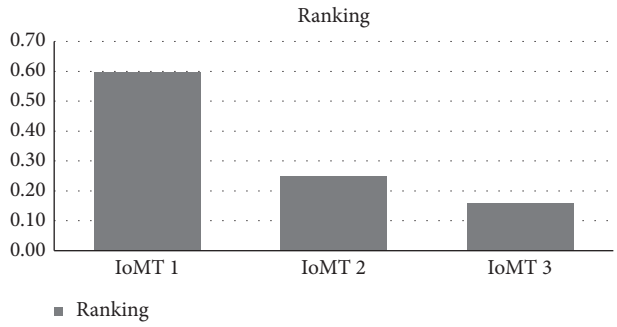


Figure 12: Ranking of IoMT.

Based on the limit matrix, we conclude that IoMT1 is the most secure component followed by IoMT2 and then IoMT3. Figure 12 shows the ranking of IoMT components.

4. Conclusion

The Internet of Medical Things is considered to be a significant part of healthcare which plays an important role. Communication among different devices such as smart sensors, wearable devices, handheld, and many other devices are connected in a network is possible due to the success of Internet of Things. For efficient and smooth running of healthcare, the security of different devices connected is mandatory. An efficient and robust security system is in dire need to cope with the attacks, threats, and vulnerability. The security evaluation of IoMT is an issue since the last few years. The proposed study is an endeavor toward the evaluation of the security of IoMT and using the analytic network process. The approach is applied using the ISO/IEC 27002 (ISO 27002) standard with the collection of some other important features from the literature. The results of the proposed research demonstrate the effective IoMT components which can further be used as secure IoMT.

Data Availability

No data were used to support the study.

Conflicts of Interest

The authors declare no conflicts of interest regarding this paper.

References

[1] X. Zhou, W. Liang, K. I.-K Wang, H. Wang, L. T. Yang, and Q. Jin, "Deep learning enhanced human activity recognition for internet of healthcare things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6429–6438, 2020.

[2] J. J. Kang, "Systematic analysis of security implementation for internet of health things in mobile health networks," in *Data Science in Cybersecurity and Cyberthreat Intelligence*, pp. 87–113, Springer, Cham, Switzerland, 2020.

[3] S. S. Rani, J. A. Alzubi, S. K. Lakshmanaprabu, D. Gupta, and R. Manikandan, "Optimal users based secure data transmission on the internet of healthcare things (IoHT) with lightweight block ciphers," *Multimedia Tools and Applications*, 2019.

[4] M. Asif-Ur-Rahman, F. Afsana, M. Mahmud et al., "Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4049–4062, 2019.

[5] J. J. P. C. Rodrigues, D. B. De Rezende Segundo, H. A. Junqueira et al., "Enabling technologies for the internet of health things," *IEEE Access*, vol. 6, pp. 13129–13141, 2018.

[6] A. M. Elmisery, S. Rho, and D. Botvich, "A fog based middleware for automated compliance with OECD privacy principles in internet of healthcare things," *IEEE Access*, vol. 4, pp. 8418–8441, 2016.

[7] S. Alam, M. M. R. Chowdhury, and J. Noll, "Interoperability of security-enabled internet of things," *Wireless Personal Communications*, vol. 61, no. 3, pp. 567–586, 2011.

[8] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom workshops)*, pp. 618–623, Kona, HI, USA, March 2017.

[9] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," in *Proceedings of the 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 336–341, Tokyo, Japan, December 2015.

[10] A. W. Atamli and A. Martin, "Threat-based security analysis for the internet of things," in *Proceedings of the International Workshop on Secure Internet of Things*, pp. 35–43, Wroclaw, Poland, September 2014.

[11] K. C. Park and D.-H. Shin, "Security assessment framework for IoT service," *Telecommunication Systems*, vol. 64, no. 1, pp. 193–209, 2017.

[12] R. Diesch, M. Pfaff, and H. Krcmar, "A comprehensive model of information security factors for decision-makers," *Computers & Security*, vol. 92, Article ID 101747, 2020.

[13] C. Hosmer, "IoT vulnerabilities," in *Defending IoT Infrastructures with the Raspberry Pi: Monitoring and Detecting Nefarious Behavior in Real Time*, pp. 1–15, Apress, Berkeley, CA, USA, 2018.

[14] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2483–2495, 2017.

[15] W. Leister, M. Hamdi, H. Abie, and S. Poslad, "An evaluation framework for adaptive security for the iot in ehealth," *International Journal on Advances*, vol. 17, 2014.

[16] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "Sok: security evaluation of home-based iot deployments," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pp. 1362–1380, San Francisco, CA, USA, May 2019.

[17] A. Tekeoglu and A. Ş. Tosun, "An experimental framework for investigating security and privacy of IoT devices," in *Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pp. 63–83, Vancouver, BC, Canada, November 2017.

[18] M. Chernyshev and P. Hannay, *Security Assessment of IoT Devices: The Case of Two Smart TVs*, SRI Security Research Institute, Edith Cowan University, Perth, Australia, 2015.

[19] B. Ali and A. Awad, "Cyber and physical security vulnerability assessment for IoT-based smart homes," *Sensors*, vol. 18, no. 3, p. 817, 2018.

[20] O. Mazhelis and P. Tyrväinen, "A framework for evaluating Internet-of-Things platforms: Application provider viewpoint," in *Proceedings of the IEEE World Forum on Internet of Things (WF-IoT)*, pp. 147–152, Seoul, Republic of Korea, March 2014.

[21] B. Liao, Y. Ali, S. Nazir, L. He, and H. U. Khan, "Security analysis of IoT devices by using mobile computing: A systematic literature review," *IEEE Access*, vol. 8, pp. 120331–120350, 2020.

[22] M. Li, S. Nazir, H. U. Khan, S. Shahzad, and R. Amin, "Modelling features-based birthmarks for security of end-to-end communication system," *Security and Communication Networks*, vol. 2020, pp. 1–9, Article ID 8852124, 2020.

[23] S. Nazir, S. Shahzad, S. Mahfooz, and M. N. Jan, "Fuzzy logic based decision support system for component security evaluation," *International Arab Journal of Information and Technology*, vol. 15, pp. 1–9, 2015.

[24] S. Nazir, S. Shahzad, M. Nazir, and H. U. Rehman, "Evaluating security of software components using analytic network process," in *Proceedings of the 11th International Conference on Frontiers of Information Technology (FIT)*, pp. 183–188, Islamabad, Pakistan, 2013.

[25] S. H. Almotiri, M. A. Khan, and M. A. Alghamdi, "Mobile health (m-health) system in the context of IoT," in *Proceedings of the IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pp. 39–42, Vienna, Austria, August 2016.

[26] Y. Karaca, M. Moonis, Y.-D. Zhang, and C. Gezgez, "Mobile cloud computing based stroke healthcare system," *International Journal of Information Management*, vol. 45, pp. 250–261, 2019.

[27] S. Nazir, S. Shahzad, Z. Hussain, M. Iqbal, and A. Keerio, "Evaluating student grades using analytic network process," *Sindh University Research Journal (Science Series)*, vol. 47, pp. 1–5, 2015.

[28] S. Nazir, S. Anwar, S. A. Khan et al., "Software component selection based on quality criteria using the analytic network process," *Abstract and Applied Analysis*, vol. 2014, pp. 1–12, 2014.

[29] S. Kheybari, F. M. Rezaie, and H. farazmand, "Analytic network process: An overview of applications," *Applied Mathematics and Computation*, vol. 367, Article ID 124780, 2020.

[30] T. L. Saaty, "Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process," *Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas*, vol. 102, no. 2, pp. 251–318, 2008.

[31] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2018.

[32] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," in *Proceedings of the IEEE World Congress on Services*, pp. 21–28, NewYork, NY, USA, June 2015.

[33] H.-J. Kim, H.-S. Chang, J.-J. Suh, and T.-S. Shon, "A study on device security in IoT convergence," in *Proceedings of the International Conference on Industrial Engineering, Management Science and Application (ICIMSA)*, pp. 1–4, Jeju Island, Republic of Korea, May 2016.

[34] L. M. R. Tarouco, L. M. Bertholdo, L. Z. Granville et al., "Internet of things in healthcare: Interoperatibility and security issues," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 6121–6125, Ottawa, Canada, June 2012.

[35] I. Alqassem and D. Svetinovic, "A taxonomy of security and privacy requirements for the Internet of Things (IoT)," in *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1244–1248, Selangor, Malaysia, December 2014.

[36] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of IoT systems: Design challenges and opportunities," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 417–423, San Jose CA, USA, November 2014.

[37] M. A. Razzaq, S. H. Gill, M. A. Qureshi, and S. Ullah, "Security issues in the Internet of Things (IoT): A comprehensive study," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 6, pp. 383–388, 2017.

[38] C. Lee, L. Zappaterra, K. Choi, and H.-A. Choi, "Securing smart home: Technologies, security challenges, and security requirements," in *Proceedings of the IEEE Conference on Communications and Network Security*, pp. 67–72, San Francisco, CA, USA, October 2014.

[39] S. Babar, P. Mahalle, A. Stango, N. Prasad, and R. Prasad, "Proposed security model and threat taxonomy for the Internet of Things (IoT)," in *Proceedings of the International Conference on Network Security and Applications*, pp. 420–429, Taganrog, Rostov Region, Russia, July 2010.

[40] B.-C. Chifor, I. Bica, V.-V. Patriciu, and F. Pop, "A security authorization scheme for smart home internet of things devices," *Future Generation Computer Systems*, vol. 86, pp. 740–749, 2018.

[41] S. Hameed, F. I. Khan, and B. Hameed, "Understanding security requirements and challenges in Internet of Things (IoT): A review," *Journal of Computer Networks and Communications*, vol. 2019, Article ID 9629381, 14 pages, 2019.

[42] A. Hinduja and M. Pandey, "An ANP-GRA-based evaluation model for security features of IoT systems," in *Intelligent Communication, Control and Devices*, pp. 243–253, Springer, Berlin, Germany, 2020.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8863617, 11 pages
https://doi.org/10.1155/2020/8863617

Research Article

Android Malware Detection Based on a Hybrid Deep Learning Model

Tianliang Lu, Yanhui Du, Li Ouyang, Qiuyu Chen, and Xirui Wang

College of Information and Network Security, People's Public Security University of China, Beijing, China

Correspondence should be addressed to Yanhui Du; duyanhui@ppsuc.edu.cn

Received 19 June 2020; Revised 22 July 2020; Accepted 30 July 2020; Published 28 August 2020

Academic Editor: Umar M. Khokhar

In recent years, the number of malware on the Android platform has been increasing, and with the widespread use of code obfuscation technology, the accuracy of antivirus software and traditional detection algorithms is low. Current state-of-the-art research shows that researchers started applying deep learning methods for malware detection. We proposed an Android malware detection algorithm based on a hybrid deep learning model which combines deep belief network (DBN) and gate recurrent unit (GRU). First of all, analyze the Android malware; in addition to extracting static features, dynamic behavioral features with strong antiobfuscation ability are also extracted. Then, build a hybrid deep learning model for Android malware detection. Because the static features are relatively independent, the DBN is used to process the static features. Because the dynamic features have temporal correlation, the GRU is used to process the dynamic feature sequence. Finally, the training results of DBN and GRU are input into the BP neural network, and the final classification results are output. Experimental results show that, compared with the traditional machine learning algorithms, the Android malware detection model based on hybrid deep learning algorithms has a higher detection accuracy, and it also has a better detection effect on obfuscated malware.

1. Introduction

Mobile phones have become increasingly important tools in people's daily life, such as mobile payment, instant messaging, online shopping, etc., but the security problem of mobile phones is becoming more and more serious. Due to the open source nature of the Android platform, it is very easy and profitable to write malware using the vulnerabilities and security defects of the Android system. This is the main reason for the rapid increase in the number of malware on the Android system. The malicious behaviors of Android malware generally include sending deduction SMS, consuming traffic, stealing user's private information, downloading a large number of malicious applications, remote control, etc., threatening the privacy and property security of mobile phones users.

The number of Android malware is growing rapidly; particularly, more and more malicious software use obfuscation technology. Traditional detection methods of manual analysis and signature matching have exposed some problems, such as slow detection speed and low accuracy. In recent years, many researchers have solved the problems of Android malware detection using machine learning algorithms and had a lot of research results. With the rise of deep learning and the improvement of computer computing power, more and more researchers began to use deep learning models to detect Android malware. This paper proposes an Android malware detection model based on a hybrid deep learning model with deep belief network (DBN) and gate recurrent unit (GRU). The main contributions are as follows:

(i) In order to resist Android malware obfuscation technology, in addition to extracting static features, we also extracted the dynamic features of malware at runtime and constructed a comprehensive feature set to enhance the detection capability of malware.

(ii) A hybrid deep learning model was proposed. According to the characteristics of static features and dynamic features, two different deep learning algorithms of DBN and GRU are used.

(iii) The detection model was verified, and the detection result is better than traditional machine learning algorithms; it also can effectively detect malware samples using obfuscation technology.

The rest of the paper is organized as follows. Section 2 gives an overview of the previous related work of malware detection and deep learning algorithms. Section 3 describes the extraction process of static and dynamic features of Android malware. Section 4 explains the malware detection process based on the hybrid deep learning model in detail. Section 5 evaluates our approach through experiments. Section 6 concludes the paper.

2. Related Work

Researchers are constantly improving and innovating the detection method of Android malware. Malware analysis technology is mainly divided into static analysis and dynamic analysis, and the detection method evolves from traditional machine learning to deep learning algorithms.

2.1. Android Malware Analysis Technologies

2.1.1. Static Analysis. The static analysis method refers to extracting malicious features through semantic analysis, permission analysis, etc., after decompiling the APK file. The major advantages of static detection are high efficiency and speed, but it is difficult to identify polymorphic deformation technology and code obfuscation [1, 2].

The required permissions of the APK can help gain awareness about the risks. Talha et al. [3] presented a permission-based APK auditor that uses static analysis to characterize and classify Android applications as benign or malicious. Rahul et al. [4] presented WHY-PER, a framework using natural language processing (NLP) techniques to identify sentences that describe the need for a given permission in an application description. To determine whether Android developers follow least privilege with their permissions requests, Felt et al. [5] built Stowaway, a tool that detects overprivilege in compiled Android applications. Arora et al. [6] identified the pairs of permissions that can be dangerous and proposed an innovative detection model, named PermPair.

APIs are also often used as key features in detecting Android malware [7]. API and permission based classification system were constructed as YARA Rule [8], and the API, class, and public methods of each application are extracted from AndroidManifest.xml, classes.dex and matched with YARA Rule. Chan and Song [9] proposed a feature set containing the permissions and API calls for Android malware static detection, and classifiers that used the proposed feature set outperform those only with the permissions.

2.1.2. Dynamic Analysis. As more and more Android malware avoid static detection through techniques such as repackaging and code obfuscation, dynamic analysis methods based on behavioral characteristics can solve this problem well [10]. Dynamic analysis refers to monitoring the behavior of Android application software when it is executed. The monitoring scope of most dynamic analysis methods is mainly access to sensitive data and API calls, etc.

Enck et al. [11] proposed TaintDroid, an efficient, system-wide dynamic taint tracking and analysis system capable of simultaneously tracking multiple sources of sensitive data. TaintDroid provides real-time analysis by leveraging Android's virtualized execution environment and has low CPU overhead.

Fu et al. [12] proposed ntLeakSemaic, a framework that can automatically locate abnormal sensitive network transmissions from mobile apps. Compared to existing taint analysis approaches, it can achieve better accuracy with fewer false positives.

Ali-Gombe et al. [13] proposed the DroidScraper system to recover important runtime data structures of application software by enumerating and reconstructing the objects in memory for mobile device forensics and postmortem analysis.

Malware attempts to evade detection by mimicking security-sensitive behaviors of benign apps and suppressing their payload to reduce the chance of being observed. Based on the contexts that trigger security-sensitive behaviors, Yang et al. [14] introduced AppContext, an approach of static program analysis that extracts the contexts of security-sensitive behaviors to assist app analysis in differentiating between malicious and benign behaviors.

2.2. Malware Detection Method

2.2.1. Machine Learning. Antivirus software can effectively detect Android malware, but it needs to manually extract the signature code and update it on the client side after obtaining the malware sample. This method has a high detection accuracy for known malware, but it also has certain limitations. For example, unknown malware that has not been seen before and malware processed by obfuscation techniques cannot be effectively detected. In order to improve the detection accuracy, in recent years, researchers have used machine learning algorithms to detect malware [15].

Kuo et al. [16] proposed an Android malware detection system which combines the machine learning methods (SVM or Random Forest) and hybrid analysis model, and the major feature combines the permissions characteristic and API.

To enhance security of machine learning-based Android malware detection, Chen et al. [17] developed a system called SecureDroid. They presented a novel feature selection method to make the classifier harder to be evaded and proposed an ensemble learning approach by aggregating the individual classifiers.

Combining of supervised learning (KNN) and unsupervised learning (K-Medoids), Arora et al. [18, 19] introduced a hybrid Android malware detection model using permissions and network traffic features. Awad et al. [20] proposed modeling malware as a language and assessed the

feasibility of finding semantics in instances of that language, and they classified malware-documents by applying the KNN.

In traditional machine learning algorithms, the SVM algorithm is often used for Android malware detection, and it has a good classification effect in many cases. Li et al. [21] studied an Android malware detection scheme using an SVM-based approach, which integrates both risky permission combinations and vulnerable API calls and uses them as features in the SVM algorithm.

2.2.2. Deep Learning. Traditional machine learning algorithms are usually shallow structures, so they cannot effectively characterize Android malware through correlation features. Therefore, researchers tried to distinguish Android malware using deep learning models. The deep learning model has a wide range of applications in image recognition, speech recognition, and natural language processing, and its strong fitting ability for nonlinear relationship makes it have a good application prospect in malware detection. Commonly used deep learning networks include stacked autoencoder [22], DBN [23], LSTM [24], and so on.

Deep learning demonstrated excellent performance in image recognition, so malware can be converted into images, and then deep learning algorithms are used for training and detection [25]. Cui et al. [26] converted the malware into grayscale images; then, the images were identified and classified using a convolutional neural network (CNN) that could extract the features of the malware images automatically. Depending on decompiling the Android APK, Zhao and Qian [27] innovatively mapped the opcodes, API packages, and high-level risky API functions into integrated three channels of an RGB image, respectively, and then used convolutional neural network to identify the malware family's features.

Pektaş et al. [28] proposed a deep learning Android malware detection method which examines all possible execution paths and the balanced dataset improves deep neural learning benign execution paths versus malicious paths. Yuan et al. [29] implemented an online Android malware detection engine based on deep learning.

In order to improve the detection accuracy and take advantage of various deep learning algorithms, some researchers have proposed malware detection models with a combination of multiple deep learning algorithms. Luo et al. [30] proposed an Android malware analysis and detection technology based on Attention-CNN-LSTM, which is a type of multi-model deep learning. Safa et al. [31] benchmarked deep learning architectures composed of recurrent and convolutional neural networks and developed an automatic feature extraction component and a hybrid CNN/RNN classification model.

3. Android Malware Features Extraction

The features extraction method combining dynamic analysis and static analysis is adopted, as shown in Figure 1. The static features are obtained by decompiling the APK file, including resource features and semantic features. The static features generate a binary feature vector through one-hot encoding. The dynamic features are obtained by monitoring the related API calls during the APK running process. For the dynamic features associated with the time series, the entity embedding method is used to generate feature vectors.

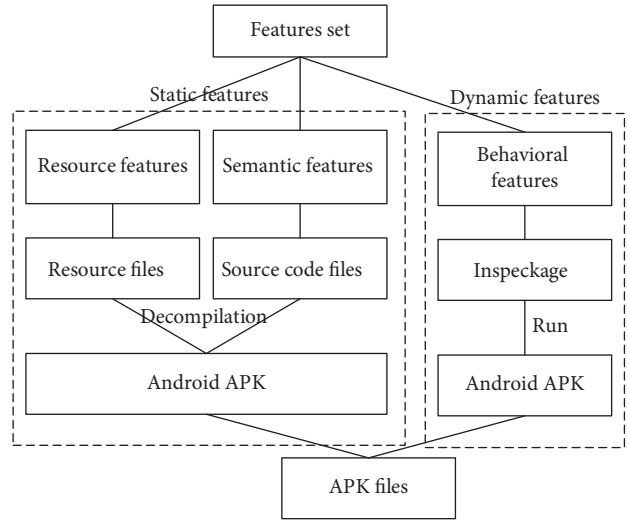


Figure 1: Android malware features extraction.

3.1. Features Extraction. Static features extraction is high-speed and consumes less system resources, which is suitable for large-scale features extraction, it cannot effectively detect obfuscated Android malware. Therefore, this paper uses a hybrid detection method of static analysis and dynamic analysis; a total of 351 features were extracted, including 303 static features and 48 dynamic features.

3.1.1. Static Feature Extraction. The extracted static features include resource features and semantic features.

(1) Resource Features. Resource features refer to features extracted from resource files stored in APK. The main basis for extracting resource features is the inconsistent structure and inconsistent logic of the APK. Inconsistent structure refers to the artifacts left behind by hiding malicious components, resulting in an abnormal structure of the APK file. Inconsistent logic refers to the fact that when a malicious software is repackaged as a benign application, it usually leaves traces. The types and quantity of resource features are shown in Table 1. A total of 124 resource features were extracted.

(2) Semantic Features. Semantic features are extracted from the APK code file. Common static features such as sensitive API and permission are also divided into semantic features. We propose some new semantic features, such as explicit intent and other features mined from meta information. The types and quantity of semantic features are shown in Table 2. A total of 179 semantic features were extracted.

Table 1: Types and quantity of resource features.

Types	Specific meaning	Quantity
Certificate	Timestamp	2
	Detail	8
Name	Package name length, entropy	14
	Package name similarity	7
Stealth	Embedded API	15
	Embedded API package name	4
	Embedded permission	26
	Embedded intention	13
Inconsistent	Inconsistent file suffix	4
	Inconsistent resource	5
Native code	System call	19
	Shared library	7

Table 2: Types and quantity of semantic features.

Type	Specific meaning	Quantity
DEX	API call	32
	API package	8
	String	13
Intent and permission	Intension	16
	Permission	67
Meta information	Services	6
	Content providers	3
	Broadcast receivers	2
	Activities	25
Escape	Encryption	2
	Reflection	3
	Dynamic code loading	2

3.1.2. Dynamic Features Extraction. Dynamic features are the behavioral characteristics of the Android application when it runs, such as data encryption and decryption, file reading and writing, network data transmission, call, SMS, geographic location, and access to sensitive information. These behaviors can represent the application's functions and intentions. A total of 48 dynamic features are extracted. The extraction of these dynamic features is mainly based on monitoring related API function calls. Each dynamic feature corresponds to several API functions, and the total number of API functions is 141. Some of the dynamic features and corresponding API examples are listed in Table 3.

We select the automatic test tool MonkeyRunner and the dynamic analysis tool Inspeckage to extract dynamic features.

MonkeyRunner is a test tool provided by the Android SDK. It supports writing test scripts to customize data and events and can simultaneously connect to multiple real terminals or emulators to trigger operations of the application software. It can better perform the functions of the Android applications.

The dynamic analysis tool Inspeckage is a simple application software that integrates the commonly used dynamic analysis functions, and a built-in web server can provide a friendly interactive interface for users. Inspeckage can not only obtain basic information such as permissions, components, shared libraries, UID, etc., but also view the behavior of the application in real time. It can customize the hooked API; that is, it can customize the dynamic behavior required for monitoring, and this is also the biggest advantage of the tool.

Table 3: Types and quantity of dynamic features.

Type	API examples	Quantity
Data encryption and decryption	doFinal setSeed	35
File reading and writing	getString putString	38
Network data transmission	Init connected getDate	28
Call	Listen getserviceStatus	13
SMS	createAppSpecificSmsToken getSmsManagerForSubscriptionId	15
Geographic location	addNmeaListener getAltitude getLongitude	12

3.2. Features Encoding

3.2.1. Static Features Encoding. For static features, most of them are binary features, and only a small number of features are discrete features, and there is no relationship between features. Therefore, the deep learning algorithm deep belief network is suitable for static features. Since the input of the deep belief network is a binary vector, one-hot encoding is used to encode the static features into binary vectors.

The process of one-hot encoding is to convert discrete features into corresponding binary sequences. For example, the discrete values of 0, 1, 2, 3 are encoded to binary sequences of 0001, 0010, 0100, 1000. For discrete features with large value range, the feature vector will be sparse if one-hot encoding is used. For this case, the value range can be finely classified to reduce the feature dimension after one-hot encoding. After one-hot encoding, all static features are concatenated into a binary vector, which is the input of the DBN.

3.2.2. Dynamic Features Encoding. After acquiring the dynamic features of the Android application, the dynamic features are formed into an operation sequence in chronological order. Because the dynamic features are correlated in time, the dynamic behavior of the Android application software can be better fitted through the recurrent neural network. The dynamic feature vector is the input of the GRU network.

Entity embedding is a method of data representation. It encodes structured discrete variables and tries to make the data representation retain the continuous relationship between data. In the implementation, the Keras library in Tensorflow is used to implement entity embedding. There are 48 discrete values ($c = 48$) for the dynamic behavior at a certain moment. And, the common choice of embedded size is to use $embedding\_size = (c + 1)/2$; therefore, the

embedding size is selected as 24; that is, the dynamic vector input to the GRU at each moment is 24 dimensions. Entity embedding is used to map the values of the discrete variables into a multidimensional space, generate dynamic feature vectors, and provide distance information of different dynamic behaviors in the multidimensional space.

4. Hybrid Deep Learning Model

According to the different characteristics of the static and dynamic features of the Android applications, a deep learning model based on a combination of deep belief network and gate recurrent unit is proposed. The advantage of using the DBN is that the learning speed of static features of Android applications is faster and the performance is better. Compared with the traditional RNN model, GRU can perform better in dealing with longer time operation sequences, with fewer parameters, faster training speed, and less data required to achieve good generalization effect. Therefore, the GRU neural network is more suitable for processing the dynamic features of Android applications.

The DBN-GRU hybrid model for Android malware detection is shown in Figure 2. The dynamic feature vectors and the static feature vectors are used to train the DBN and GRU, respectively, and the output vectors are input to the fully connected layer. The softmax function maps the output of multiple neurons to the interval of (0, 1) and outputs classification results in the form of probability. The softmax function is part of the back propagation neural network and is used to fine-tune the parameters of DBN and GRU.

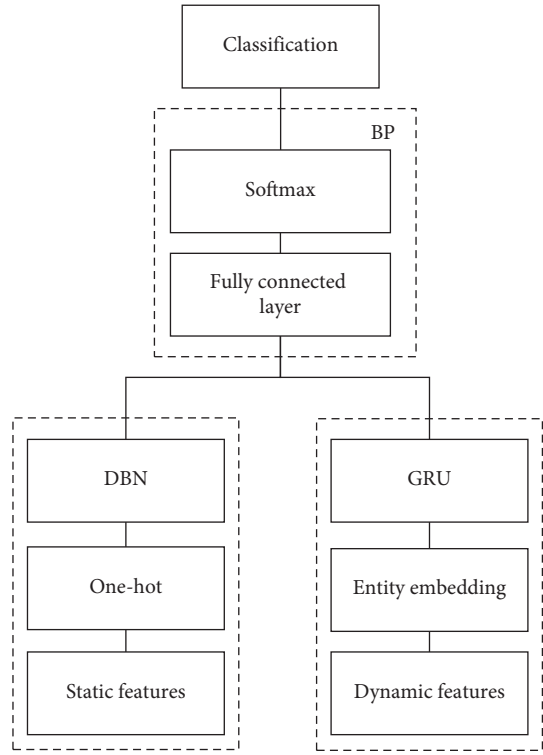


Figure 2: DBN-GRU hybrid model for Android malware detection.

4.1. Deep Belief Network. DBN is a widely used deep learning framework [32]. The deep belief network is divided into two parts. The bottom part is formed by stacking multiple restricted Boltzmann machines. The RBM of each layer is trained by contrastive divergence (CD) algorithm. The upper part is a supervised back propagation neural network, which is used to fine-tune the overall network. Since this paper uses a hybrid model of DBN and GRU, the BP neural network of the two is integrated and will be introduced in Section 4.3.

4.1.1. Network Structure. In Figure 3, V represents the visible layer, and H represents the hidden layer. In the stacked RBMs, except the last RBM, the hidden layer of each RBM is the visible layer of the next RBM. The weight matrix W_i is used to represent the mapping relationship between the visible layer and the hidden layer. V_0 is the initial feature vector of the first RBM, which is the initial input of the DBN network. The hidden layer H_2 of the last RBM represents the output, which is used for classification. In order to make each layer reach the local optimum, the weight matrix W_i needs to be trained by the CD algorithm.

4.1.2. Pretraining. DBN uses CD algorithm to optimize the parameters of each RBM during pretraining; Figure 4 shows the pretraining process. First, input the initial vector to the first RBM. In our model, the one-hot encoded Android static

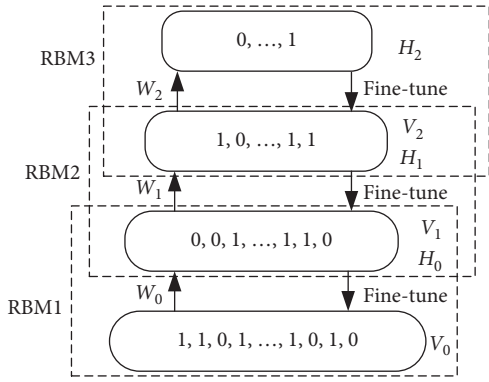


Figure 3: Structure of DBN.

feature vector is input to the DBN as the pretraining initial input.

CD algorithm is used to train the weight matrix W_0 of the first RBM, the bias vector A_0 of the visible layer, and the bias vector B_0 of the hidden layer. The output vector H_0 is obtained after the training, and then the output vector is input to the upper layer RBM as the input vector V_1 of the second RBM. The weight matrix W_1 and the bias vectors A_1 and B_1 of the second RBM are calculated, and the above process is repeated until all RBM training is completed. Pretraining makes the parameters of each RBM reach local optimum.

4.1.3. Parameter Selection. During the pretraining process, the effect of the pretraining is greatly related to the relevant

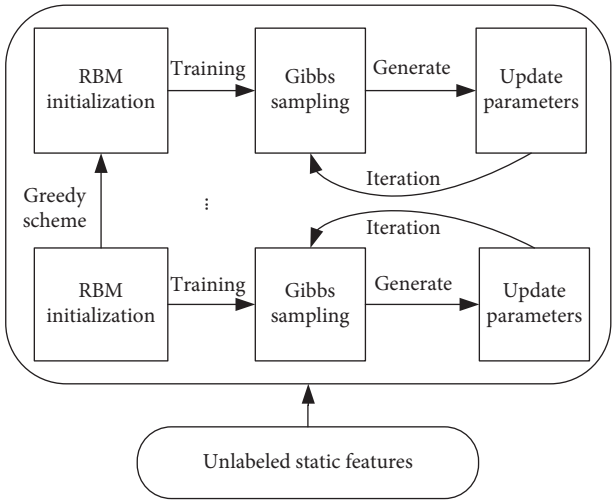


Figure 4: DBN pretraining process.

parameters. There are mainly three parameters for pretraining: the learning rate, batch_size, and epochs. Unlike the learning rate in the subsequent fine-tuning stage of the back propagation neural network, in the pretraining of the DBN, the learning rate usually does not need to be changed, but an appropriate value needs to be set. If the learning rate is too large, the model may fail to converge or even diverge, and if it is too small, the gradient descent can be slow.

Using the method proposed by Smith [33], the model is trained with learning rate range from small to large values, and then the change of loss is recorded. As the learning rate increases, the loss will gradually decrease and then increase, and the best learning rate can be selected from the area with the smallest loss. In this paper, when the pretraining learning rate is 0.05, the pretraining effect is better.

A large batch_size can reduce training time and improve stability, but as batch_size increases, the performance of the model will decrease. Therefore, an appropriate batch_size needs to be chosen. Considering the size of the dataset used in this paper, the batch_size is set to 256.

Finally, the epochs of pretraining: according to the value of the learning rate and the value of batch_size, the epochs are set to 10.

4.2. Gate Recurrent Unit. Recurrent neural network (RNN) can better deal with the time series data, so it is suitable for training the dynamic behavioral characteristics of Android applications. The traditional RNN has the problem of disappearing gradients, which is especially serious when the time series is long. To solve this problem, researchers made improvements to the RNN and got a variant, that is, GRU [34].

The GRU neural network is shown in Figure 5. According to the timeline, X^t is the input feature vector, which represents the dynamic behavior of the Android application at the current time. It can be seen that the input X^t can be remembered by the GRU through h^t. This is very suitable for dynamic features with temporal correlation. For example, if there is a correlation between the two dynamic

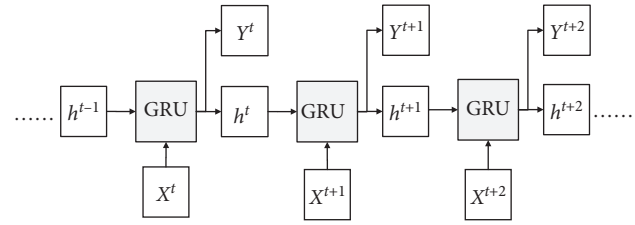


Figure 5: GRU neural network.

behaviors of reading address book and sending network data, it may be related to malicious behavior, and GRU can learn the link between these behaviors of the Android malware.

4.2.1. Internal Structure of GRU. The internal structure of the GRU is shown in Figure 6; X^t is the input of the current unit, Y^t is the output of the current unit, and h^t is the hidden state of the current unit. h^{t-1} is the hidden state output by the previous unit and passed to this unit. The hidden state contains information about the input of the previous unit.

The internal calculation process of GRU is as follows:

$\odot$ is an element-wise multiplication

$+$ is an element-wise addition

r is the reset gate, and z is the update gate

W^r, W^z, and W are weight matrixes

Activation function is tanh:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}},$$
$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{1}$$

Step 1. Calculate the states of gate r and gate z:

$$r = \sigma\left(W^r X^t + W^r h^{t-1}\right),$$
$$z = \sigma\left(W^z X^t + W^z h^{t-1}\right). \tag{2}$$

Step 2. Reset using reset gate r, and calculate $h\prime$:

$$h^{t-1'} = h^{t-1} \odot r,$$
$$h' = \tanh\left(W X^t + W h^{t-1'}\right). \tag{3}$$

Step 3. Update the memory. The closer the gate signal is to 1, the more information it memorizes, and the closer it is to 0, the more information it forgets:

$$h^t = z \odot h^{t-1} + (1 - z) \odot h'. \tag{4}$$

As mentioned above, combining X^t and h^{t-1}, the GRU will get the output Y^t of the current unit and pass it as the

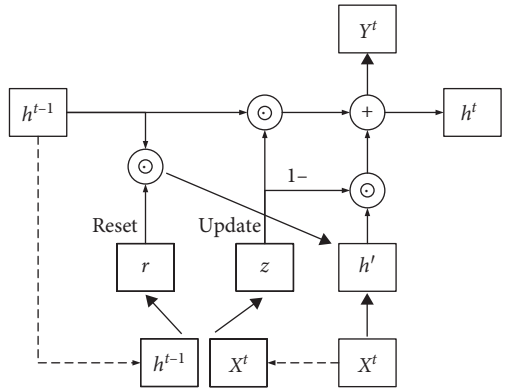


Figure 6: Internal structure of GRU.

hidden state h^t to the next unit, where Y^t and h^t are the same in value.

4.2.2. Number of Network Layers. Our model uses a single-layer GRU. If a second layer is added, it can capture higher-level correlation of dynamic behaviors theoretically, but this is based on the large number of Android applications and high-dimensional input vectors. For the input 24-dimensional dynamic feature vectors and the training samples that have just passed 10,000, the multilayer GRU neural network will not produce good results and may even cause overfitting.

4.2.3. Parameter Initialization. Parameter initialization refers to the process of initializing the weights and biases before the network training. The initialization of parameters is related to whether the network training can yield good results and converge at a faster speed. If the parameter is too small, the input signal of neuron will be too small, and the signal will slowly disappear after multiple layers. If the parameter is too large, the input signal will be too large, and the activation value is saturated causing the gradient to be close to zero.

The Gaussian distribution is used to initialize the weight matrix parameters. The parameters follow a Gaussian distribution with a fixed mean and a fixed variance. Assuming the number of neurons of a certain layer is n_i, the Gaussian distribution of the initialized weight matrix parameters has a mean value of 0 and a variance of $\sqrt{1/n_i}$.

4.3. Back Propagation Neural Network. Back propagation neural network uses supervised learning methods to compare the classification results with the labels of Android applications to fine-tune the hybrid deep learning model. This paper uses the BP neural network combined with the softmax classifier, and the gradient descent algorithm is applied to fine-tune both DBN and GRU networks.

4.3.1. Softmax. Softmax is used in the multiclassification process to map the output of multiple neurons to the (0, 1) interval for classification. Softmax is calculated as

$$S_i = P_i = \frac{e^{f_i}}{\sum_j e^{f_j}}. \tag{5}$$

where S_i represents the probability that the sample is divided into the i-th category. During classification, the category with the highest probability is selected as the classification result.

4.3.2. Fine-Tune the Parameters. The main function of the BP neural network is to fine-tune the parameters of the deep learning model, and the main basis in the fine-tuning process is to find the global minimum value of the loss function of the model. At this time, the parameters of the corresponding weight matrix are the global optimal. For the softmax classifier, the cross-entropy loss function is selected, as shown in

$$L = -\sum_i y_i \ln S_i, \tag{6}$$

where y_i represents the correct label value of each category corresponding to the sample. If it belongs to the i-th category, then, $y_i = 1$; otherwise, $y_i = 0$. Since only one label is 1, the other label is 0, so formula (6) is simplified to formula (7), and its gradient is calculated as

$$L = -y_i \ln S_i = -\ln S_i, \tag{7}$$

$$\frac{\partial L}{\partial f_i} = S_i - 1. \tag{8}$$

The cross-entropy loss function not only can effectively measure the similarity between the calculated value and the actual value, but also has a simple form and is easy to calculate and partial derivative, which is very convenient in the gradient calculation.

4.3.3. Optimization Algorithm. In the process of back propagation, the most commonly used optimization algorithm is to fine-tune the parameters by gradient descent. In order to solve the problems in the gradient descent method, an improved method mini-batch gradient descent is proposed, which can reduce the fluctuation of parameter update and finally get better results and more stable convergence. However, there are still some problems. For example, it is difficult to choose a suitable learning rate and it is easy to fall into the local optimum. Therefore, some algorithms for further optimization of gradient descent are proposed, such as Momentum, Adagrad, and RMSprop.

In this paper, the Adam (Adaptive Moment Estimation) algorithm is selected for fine-tuning. It is a combination of RMSProp algorithm and Momentum algorithm. The main feature of Adam is the adjustment strategy of the learning rate. The first moment estimation (the mean) and second moment estimation (the uncentered variance) of the gradients are used to adjust the learning rate of each parameter dynamically. The main advantage of Adam is that, after the bias correction, the learning rate of each iteration has a certain range, which makes the parameters relatively stable

and has low memory requirements. And in actual operation, the Adam algorithm is simple to use and does not require manual parameter adjustment.

5. Experimental Results and Analysis

5.1. Collation of Android Malware and Benign Samples. The dataset is divided into benign samples dataset and malware samples dataset. The type, source, and quantity of samples in the dataset are shown in Table 4. The total number of benign samples is 7,000, downloaded from the Google Play and APKpure mobile application markets through web crawler. The number of samples in the malware dataset is 6,298, all downloaded from public malware sharing websites. According to whether the samples use obfuscation technology, the malware samples dataset is divided into two parts: one part is the nonobfuscated malware dataset downloaded from VirusShare [35] and the other part is the obfuscated malware dataset downloaded from PRAGuard [36], obtained by obfuscating the MalGenome and the Contagio Minidump datasets with seven different obfuscation techniques.

5.2. Experimental Results and Analysis

5.2.1. Features with High Frequency of Use. The frequently used features between obfuscated malware samples and nonobfuscated malware samples are analyzed. Figures 7 and 8 show the frequently used top ten features of the two types of samples.

Permission-related features (such as Read_SMS, Write_SMS, etc.) of both sample types are used frequently, because permission features are difficult to be obfuscated, and obfuscating the permission features will destroy the inherent structure of APK. However, some sensitive API features (such as Telephonymanager_Getdeviceid, etc.) are used frequently in nonobfuscated malware samples but are used very rarely in obfuscated malware samples, which shows that the malware samples after obfuscation can avoid related detections when calling sensitive APIs.

It is worth noting that Stat_Cert_Diff, the top-ranked feature in the obfuscated malware samples, is a resource feature related to the certificate. It detects whether the time when the certificate is generated and the certificate is used to sign the APK is the same time. The frequency of this feature is high, which indicates that most of the obfuscated malware samples are generated through automatic repackaging. It also shows that the features extracted in this paper (such as Stat_Cert_Diff, Stat_Reflection) are very effective in detecting obfuscated malware samples.

5.2.2. Detection Effect of the Hybrid Deep Learning Model. Evaluate the detection effect of the hybrid deep learning model (DBN-GRU) on Android malware through the indicators of precision, recall, and accuracy; the results are shown in Table 5. Deep learning models (such as DBN, GRU, DBN-GRU) are significantly better than traditional machine learning models (such as SVM, Naïve Bayes, KNN). For deep learning models, the DBN-GRU hybrid model is superior to the separate DBN or GRU.

Table 4: Android malware and benign samples.

Type	Source	Number	Total
Benign	Google Play	5000	7000
	APKpure	2000	
Malware	VirusShare (nonobfuscated)	4038	6298
	PRAGuard (obfuscated)	2260	

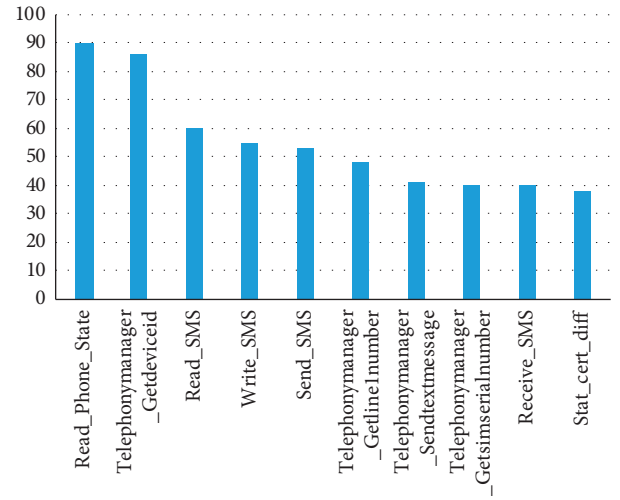


Figure 7: Top 10 features of nonobfuscated malware.

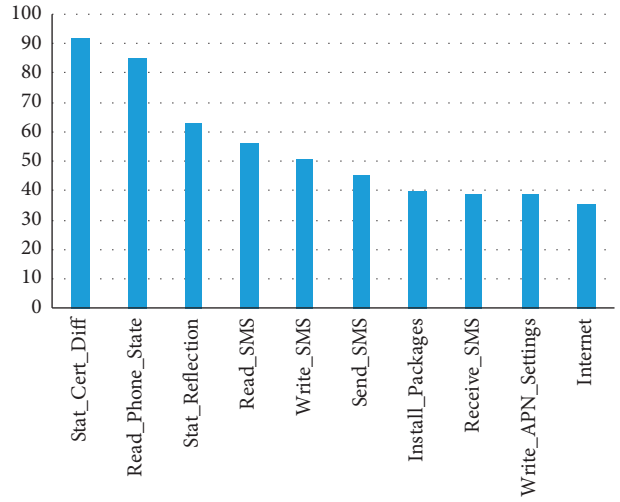


Figure 8: Top 10 features of obfuscated malware.

The DBN model uses static features. Although the antiobfuscation capability of the static features proposed in this paper has been greatly improved, it is still insufficient in capturing the dynamic behaviors of malware. The GRU model uses dynamic features, which has advantages in dynamic behavior analysis but is insufficient in the types of features. According to the experimental results, based on the combination of dynamic features and static features, the hybrid model of DBN and GRU can improve the detection ability and achieve a better detection effect.

Table 5: Detection effects of different algorithms.

	Benign samples		Malware samples		Accuracy
	Precision	Recall	Precision	Recall	
SVM	92.74	88.91	88.38	92.38	90.57
KNN	83.83	85.65	83.90	81.90	83.86
Naïve Bayes	86.54	78.26	78.45	86.67	82.27
DBN	95.98	93.49	93.06	95.71	94.55
GRU	93.78	91.74	91.16	93.33	92.50
DBN-GRU	**97.79**	**96.09**	**95.79**	**97.62**	**96.82**

5.2.3. Detection Effect of Different Training Datasets. Based on the collected samples, three sample datasets are constructed to evaluate the detection effect of the Android malware detection model on obfuscated samples and nonobfuscated samples. The composition of these three datasets is as follows:

Nonobfuscated dataset: benign samples + VirusShare

Obfuscated dataset: benign samples + PRAGuard

Mixed dataset: benign sample + VirusShare + PRAGuard

Each dataset is divided into a training dataset and a testing dataset, with 2/3 of the samples as the training dataset and 1/3 of the samples as the testing dataset.

Table 6 shows the detection accuracy of using different training datasets. Using the nonobfuscated dataset for both training and testing, an accuracy rate of 96.89% is obtained. Using the obfuscated dataset for both training and testing, an accuracy rate of 96.58% is obtained. In both cases, the detection accuracy is relatively high.

Then, analyze the situation when the training dataset and the testing dataset are different. Using the nonobfuscated dataset for training and the obfuscated dataset for testing, the accuracy rate drops significantly, being 89.51%. Using the obfuscated dataset for training and the nonobfuscated dataset for testing, the accuracy rate also drops to 92.32%. Using the mixed dataset for training, no matter whether the obfuscated dataset or the nonobfuscated dataset is used for testing, the accuracy rates are higher, 96.78% and 96.24%, respectively.

The experimental results show that the richer the sample types of the training dataset, the higher the detection accuracy. The mixed training dataset contains nonobfuscated malware and obfuscated malware. The detection accuracy of all types of testing datasets is high, which can meet the needs of malware detection in actual network.

5.2.4. Detection Effect of Repackaged Malware. 200 samples chosen from the malware sample dataset are repackaged. The process of repackaging is very simple; just uncompress and reassemble the application software without changing any functions. In that case, the MD5 or SHA hash value of the repackaged application software will be different from the original value. Then, using mainstream antivirus software and the DBN-GRU hybrid model proposed in this article for detection, the results are shown in Figure 9. It can be seen that the detection accuracy of the repackaged APKs by antivirus software has dropped significantly. This is because the antivirus software may use a hash check method to identify the known Android malware, due to the fact that the hash value of the repackaged APK has changed, causing the detection accuracy to decrease. The detection results of the DBN-GRU hybrid model are not affected by the application software repackaging, because repackaging does not affect the extracted static and dynamic features and the process of model training, which is the major advantage of the hybrid model in detecting Android malware.

Table 6: Accuracy of different training datasets.

Training dataset	Testing dataset	
	Nonobfuscated dataset	Obfuscated dataset
Nonobfuscated dataset	96.89	89.51
Obfuscated dataset	92.32	96.58
Mixed dataset	96.78	96.24

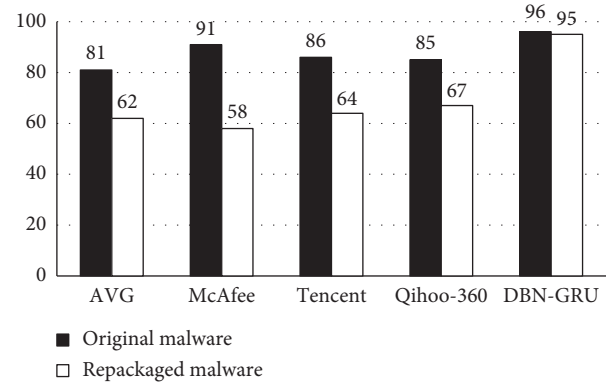


Figure 9: Detection effect of repackaged Android malware.

6. Conclusion

Due to the widespread use of obfuscation techniques in malware, the effect of traditional detection methods is greatly affected. This paper combines dynamic analysis technology and static analysis technology for Android malware detection and builds a hybrid deep learning model based on DBN and GRU. In order to deal with the obfuscation technology, new static features with strong anti-obfuscation capabilities are added, and the dynamic features of the application software at runtime are extracted to enrich the Android malware feature set. According to the different characteristics of static features and dynamic features, a hybrid deep learning model with DBN and GRU is used for learning, and the detection effect of this model is verified through comparative experiments.

Due to the limited computing resources of mobile devices, and the fact that deep learning is a compute-intensive task, the Android malware detection model proposed in this paper is suitable for running on high-performance computers. In order to solve this problem, the cloud antivirus technology is recommended, the mobile phone client is

responsible for uploading suspicious files, and the cloud-based server is responsible for sample analysis and detection.

The research has the following deficiencies and needs to be improved in future research work. First, the number of samples, especially malware samples, is not enough, and the representativeness of the obtained malware features is still not strong. It is necessary to constantly enrich the types and number of samples in the malware sample dataset; second, the calculating consumption of the hybrid model is larger than that of the separate model and the traditional machine learning algorithm, so further improvement and optimization is needed to reduce the time cost.

Data Availability

The Android malware samples used to support the findings of this study can be downloaded from VirusShare (available at https://virusshare.com) and Android PRAGuard (available at http://pralab.diee.unica.it/en/AndroidPRAGuardDataset).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Cryptography Development Fund of China (grant no. MMJJ20180108) and the Fundamental Research Funds for the Central Universities of PPSUC (grant no. 2020JKF101).

References

[1] K. Bakour, H. M. Nver, and R. Ghanem, "The Android malware static analysis: techniques, limitations, and open challenges," in *Proceedings of the 3rd International Conference on Computer Science and Engineering (UBMK'18)*, pp. 586–593, Sarajevo, Bosnia-Herzegovina, September 2018.

[2] K. Riad and L. Ke, "RoughDroid: operative scheme for functional android malware detection," *Security and Communication Networks*, vol. 2018, Article ID 8087303, 10 pages, 2018.

[3] K. A. Talha, D. I. Alper, and C. Aydin, "APK Auditor: permission-based Android malware detection system," *Digital Investigation*, vol. 13, pp. 1–14, 2015.

[4] P. Rahul, X. Xiao, W. Yang, W. Enck, and T. Xie, "WHYPER: towards automating risk assessment of mobile applications," in *Proceedings of the 22nd Usenix Security Symposium*, pp. 527–542, Washington, DC, USA, August 2013.

[5] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS'11)*, pp. 627–636, Chicago, IL, USA, October 2011.

[6] A. Arora, S. K. Peddoju, and M. Conti, "PermPair: android malware detection using permission pairs," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1968–1982, 2020.

[7] W. Wang, J. Wei, S. Zhang, and X. Luo, "LSCDroid: malware detection based on local sensitive API invocation sequences," *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 174–187, 2020.

[8] J. Park, H. Chun, and S. Jung, "API and permission-based classification system for Android malware analysis," in *Proceedings of the International Conference on Information Networking (ICOIN 2018)*, pp. 930–935, Chiang Mai, Thailand, January 2018.

[9] P. P. K. Chan and W. K. Song, "Static detection of Android malware by using permissions and API calls," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC 2014)*, pp. 82–87, Lanzhou, China, July 2014.

[10] H. Cai, N. Meng, B. Ryder, and D. Yao, "DroidCat: effective Android malware detection and categorization via app-level profiling," *IEEE Transactions on Information Forensics & Security*, vol. 14, no. 6, pp. 1455–1470, 2018.

[11] W. Enck, P. Gilbert, S. Han et al., "TaintDroid," *ACM Transactions on Computer Systems*, vol. 32, no. 2, pp. 1–29, 2014.

[12] H. Fu, Z. Zheng, S. Bose, M. Bishop, and P. Mohapatra, "Leaksemantic: identifying abnormal sensitive network transmissions in mobile applications," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2017)*, pp. 1–9, Atlanta, GA, USA, May 2017.

[13] A. Ali-Gombe, S. Sudhakaran, A. Case, and G. G. Richard, "DroidScraper: a tool for Android in-memory object recovery and reconstruction," in *Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses*, pp. 547–559, Beijing, China, October 2019.

[14] W. Yang, X. Xiao, B. Andow, S. Li, T. Xie, and W. Enck, "Appcontext: differentiating malicious and benign mobile app behaviors using context," in *Proceedings of the International Conference on Software Engineering (ICSE 2015)*, pp. 303–313, Florence, Italy, May 2015.

[15] M. A. Kadri, M. Nassar, and H. Safa, "Transfer learning for malware multi-classification," in *Proceedings of the 23rd International Database Applications & Engineering Symposium (IDEAS'19)*, pp. 1–7, Athens, Greece, June 2019.

[16] W. C. Kuo, T. P. Liu, and C. C. Wang, "Study on Android hybrid malware detection based on machine learning," in *Proceedings of the IEEE International Conference on Computer and Communication Systems (ICCCS 2019)*, pp. 31–35, Singapore, February 2019.

[17] L. Chen, S. Hou, and Y. Ye, "Securedroid: enhancing security of machine learning-based detection against adversarial android malware attacks," in *Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC 2017)*, pp. 362–372, Orlando, FL, USA, December 2017.

[18] A. Arora, S. K. Peddoju, V. Chouhan, and A. Chaudhary, "Hybrid Android malware detection by combining supervised and unsupervised learning," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom 2018)*, pp. 798–800, October 2018.

[19] A. Arora and S. K. Peddoju, "NTPDroid: a hybrid android malware detector using network traffic and system permissions," in *Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, pp. 808–813, New York, NY, USA, August 2018.

[20] Y. Awad, M. Nassar, and H. Safa, "Modeling malware as a language," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC 2018)*, pp. 1–6, Kansas City, MO, USA, May 2018.

[21] W. Li, J. Ge, and G. Dai, "Detecting malware for Android platform: an SVM-based approach," in *Proceedings of the*

IEEE International Conference on Cyber Security and Cloud Computing (CSCloud 2015), pp. 464–469, New York, NY, USA, November 2015.

[22] X. Yuan, J. Zhou, B. Huang, Y. Wang, C. Yang, and W. Gui, "Hierarchical quality-relevant feature representation for soft sensor modeling: a novel deep learning strategy," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 3721–3730, 2020.

[23] Y. Wang, Z. Pan, X. Yuan, C. Yang, and W. Gui, "A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network," *ISA Transactions*, vol. 96, pp. 457–467, 2020.

[24] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[25] F. Mercaldo and A. Santone, "Deep learning for image-based mobile malware detection," *Journal of Computer Virology and Hacking Techniques*, vol. 16, no. 2, pp. 157–171, 2020.

[26] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-g. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.

[27] Y. L. Zhao and Q. Qian, "Android malware identification through visual exploration of disassembly files," *International Journal of Network Security*, vol. 20, no. 6, pp. 1061–1073, 2018.

[28] A. Pektaş and T. Acarman, "Learning to detect Android malware via opcode sequences," *Neurocomputing*, vol. 396, pp. 599–608, 2020.

[29] Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: android malware characterization and detection using deep learning," *Tsinghua Science and Technology*, vol. 21, no. 1, pp. 114–123, 2016.

[30] S. Luo, Z. Liu, B. Ni, H. Wang, H. Sun, and Y. Yuan, "Android malware analysis and detection based on Attention-CNN-LSTM," *Journal of Computers*, vol. 14, no. 1, pp. 31–43, 2019.

[31] H. Safa, M. Nassar, and W. A. R. A. Orabi, "Benchmarking convolutional and recurrent neural networks for malware classification," in *Proceedings of the 15th International Wireless Communications & Mobile Computing Conference (IWCMC 2019)*, pp. 561–566, Tangier, Morocco, June 2019.

[32] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[33] L. N. Smith, "Cyclical learning rates for training neural networks," pp. 1–10, 2017, http://arxiv.org/abs/1506.01186v6.

[34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," pp. 1–9, 2014, http://arxiv.org/abs/1412.3555.

[35] VirusShare, 2020, https://virusshare.com.

[36] D. Maiorca, D. Ariu, I. Corona, M. Aresu, and G. Giacinto, "Stealth attacks: an extended insight into the obfuscation effects on Android malware," *Computers & Security*, vol. 51, pp. 16–31, 2015.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8830903, 10 pages
https://doi.org/10.1155/2020/8830903

Research Article

Convolution Neural Network-Based Higher Accurate Intrusion Identification System for the Network Security and Communication

Zhiwei Gu,[1] **Shah Nazir**,[2] **Cheng Hong**,[1] **and Sulaiman Khan**[2]

[1]*State Grid Quzhou Power Supply Company, Quzhou 324000, China*
[2]*Department of Computer Science, University of Swabi, Ambar, Pakistan*

Correspondence should be addressed to Zhiwei Gu; quzhouguzhiwei@163.com

Received 22 June 2020; Revised 17 July 2020; Accepted 25 July 2020; Published 28 August 2020

Academic Editor: Amir Anees

With the development of communication systems, information securities remain one of the main concerns for the last few years. The smart devices are connected to communicate, process, compute, and monitor diverse real-time scenarios. Intruders are trying to attack the network and capture the organization's important information for its own benefits. Intrusion detection is a way of identifying security violations and examining unwanted occurrences in a computer network. Building an accurate and effective identification system for intrusion detection or malicious activities can secure the existing system for smooth and secure end-to-end communication. In the proposed research work, a deep learning-based approach is followed for the accurate intrusion detection purposes to ensure the high security of the network. A convolution neural network based approach is followed for the feature classification and malicious data identification purposes. In the end, comparative results are generated after evaluating the performance of the proposed algorithm to other rival algorithms in the proposed field. These comparative algorithms were FGSM, JSMA, C&W, and ENM. After evaluating the performance of these algorithms and the proposed algorithm based on different threshold values ranging, L_p norms, and different parametric values for c, it was concluded that the proposed algorithm outperforms with small L_p values and high Kitsune scores. These results reflect that the proposed research is promising toward the identification of attack on data packets, and it also reflects the applicability of the proposed algorithms in the network security field.

1. Introduction

The technology is ever playing an important role in human life and made things easy. With the developments of technology, security remains one of the major concerns for communication and interaction [1–10]. Since the last few decades, the attacks on information security become raised and intruders are trying to capture ordination important information for their own benefits. Such attacks on network and information can drastically put the owner of information and network into big loss. The information security of an organization is highly dependent on different types of information of the organization [10–13].

Now a day, the communication is made through Internet of Things (IoT) and a number of devices are connected through a network. The smart devices are connected to communicate, process, compute, and monitor diverse real-time scenarios. The concept of IoT came with the challenges of privacy and security, as the conventional security protocol does not fit the devices of IoT. Different security approaches and measures are used to secure the information communication and to secure the network. This measure includes firewalls, logical access, control, authentication, identification, and encryption and decryption. To build a full-secure system is difficult to manage and none of these security measures alone can secure the communication inside network.

Keeping in view the severity of security, the proposed research has adopted convolution neural network (CNN) approach for intrusion detection. The CNN architecture is

capable of automatic recognition of data within an acceptable range. Whenever new data is fed to these algorithms, they learn and optimize their operations to improve performance, developing "intelligence" over time. The dataset used for the proposed research is available at UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Kitsune+Network+Attack+Dataset). The method shows success in identification of attacks on data packets for secure end-to-end communication.

The rest of the paper is organized as follows: Section 2 presents the related work to the current research and a systematic mapping of the similar work reported in the association of computation machinery (ACM) digital library. Section 3 briefly shows the research method followed for the development of an accurate intrusion detection system. Section 4 shows the results and discussions of the proposed research. The paper is concluded in Section 5.

2. Background Study

This section of the paper explains the relevant work reported in the proposed field and a systematic mapping to check the contribution of the work in the ACM digital library.

2.1. Related Work. Diverse approaches and techniques are used to tackle the issue of security from different perspectives. Kotenko and Chechulin [14] presented a framework for security assessment and attack modelling in security information and event management system. Subsorn and Limwiriyakul [15] examined the security of Internet banking of 16 Australian banks for finding the deficiencies which were probably affecting the confidentiality of the bank customers. Furthermore, the study investigated 12 Thai commercial banks and compared the results with the previous research. Kotenko and Chechulin [16] proposed a method for the attack of computer modelling and evaluation of security to realize in security information and event management system. The authors proposed a quantitative approach to security risk for information systems which is extendable, systematic, and modular. The study aims to effectively evaluate security threat in a comprehensive way [9].

Manjiatahsien et al. [17] presented an overview of the IoT architecture with a detailed review of machine learning algorithms, significance of IoT security with diverse types of attacks. The study proposed a model of the associated information management factors for the information security of organization. Firstly, they surveyed 136 articles to identify the information security factors, and, secondly, a series of interviews were held with 19 experts from the industry to evaluate the relevancy of these factors. In third step, a complete model was developed [18]. The security identification has significant role in the field like Internet of Things in smart city. The authors [19] conducted a detailed survey of the state-of-the-art IoT security, deep learning, and big data technology. Deep learning plays a key role from natural language processing to other recognition and security fields [20]. Zhang et al. [7] proposed an approach for crowed assessing the security and trustworthiness of open social networks based on signaling theory.

The authors [5] presented a detailed overview of the security properties investigation of machine learning algorithms. They have analysed the security model of ML to build up a blueprint for multidisciplinary area of research and, after that, the attack methods and discussed the strategies of defense against them. The study presented an overview of the weaknesses and strengths of the available evaluation methods used for usability and security for the websites of electronic commerce (e-commerce). The evaluation models from 2000 to 2018 have been reviewed for e-commerce [21]. Mao et al. [22] proposed a system for building security dependency to measure the significance of security of system from a wide perspective of the system. The effect of small-world and power-law distribution for the degree of in-and out-degree in security dependency network was observed. Nazir et al. [10] proposed a methodology for evaluating the security of software components using the analytic network process. This technique works in situation of complexity where the dependencies exist among different nodes of network.

2.2. Existing Approaches for Security. Information security plays a significant role in the functionality of a system to smoothly be functional. Data inside a network passes through different packets. Secure communication through these packets can further enhance the efficiency of a system to be reliable. Different approaches and methods are used to secure communication inside and outside the network. To know the details of the literature, the popular libraries were searched. The existing approaches along with their details in terms of years, type of publication, and the areas are given in the figures and tables in this section. Table 1 summarizes some of the techniques used in the literature for security purposes [40].

Table 2 shows the articles with references list proposed for the detection of the different types of malwares [40]. It also contains the information for the different types of techniques to address these certain types of malicious attacks.

Figure 1 shows the total number of publications within the selected range of the years (2016–2020 (a portion of 2020 is included in the systematic search process)). This figure also reflects the type of the research/articles reported during this specific range of the years.

The searched papers were checked to show the year of publication; that is, the particular year in which a paper is published (2016–2020 (a portion of 2020 is included in the systematic search process)). Figure 2 shows the total number of publications in the given year.

Figure 3 shows the journal/magazine name along with the total number of papers published for the search process in the ACM library.

Figure 4 shows publications type of all the publications in the ACM digital library. It also contains the information for a total number of publication type within the ACM digital library. The highest number of journal papers and proceedings represents the contribution of the work in the proposed field.

Table 1: Techniqueswise literature categorization.

Ref. no	Year	Description
[23]	2019	Risk assessment model for addressing the security issues in IoT ecosystem
[24]	2019	Threats and attack based analysis of IoT
[25]	2018	Architecture based analysis in light of security requirements
[26]	2018	Discussing the layer based security analysis of IoT
[27]	2018	Security analysis of mobile device-to-device network using Android operating system
[28]	2018	Security analysis of mobile health applications for testing functionality.
[29]	2018	Threat and attack based analysis of IoT
[30]	2018	Analysis of all security areas in IoT
[31]	2018	Study of the existing and proposed countermeasures in IoT based system
[32]	2017	Proposing a mobile application tool for analysis of IoT threats.
[33]	2017	Presenting a threat categorization based on security dimensions like integrity, confidentiality, etc.
[34]	2017	Proposing a classification model to analyse the relation between potential risk and potential vulnerabilities in home automation devices
[35]	2016	Security analysis of smart phone in IoT
[36]	2016	Analysis of identification of application, threats, and impacts in IoT
[37]	2016	Security issues and challenges of IoT and mobile computing
[38]	2015	Analysing the IoT security challenges, issues, and open problems
[39]	2015	Discussing security aims, goals, and vulnerabilities for IoT

Table 2: Malware detection techniques.

Ref. no.	Description
[41]	It uses four ways to detect malwares. It divides the applications into four types like malicious, benign, aggressive, and risky applications
[42]	Android analysis techniques for evaluating the effectiveness of Android intense
[43]	It uses ADA GRAD optimize algorithm for detecting malware pattern without manual intervention
[44]	It detects malware by using ensemble classifier for malware detection
[45]	It uses the machine learning algorithm which was presented by Waikato environment for knowledge analysis (WEKA)
[46]	It uses machine learning method for Android malware detection
[47]	It detects application features and decides whether malicious or not
[48]	It uses multiflow detection algorithm based on information flow analysis

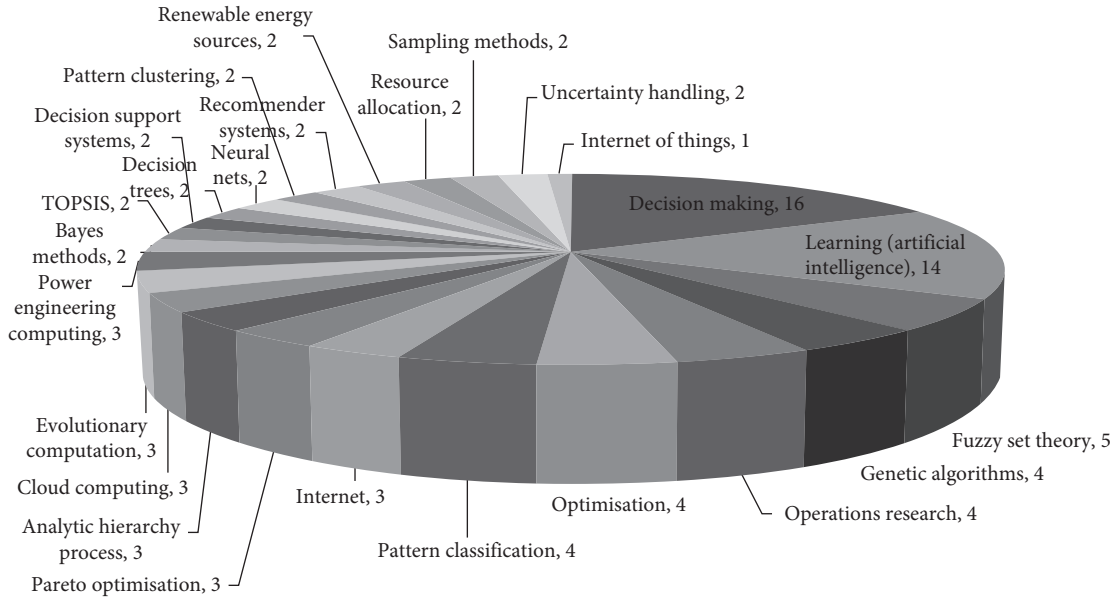


Figure 1: Publication type and total number.

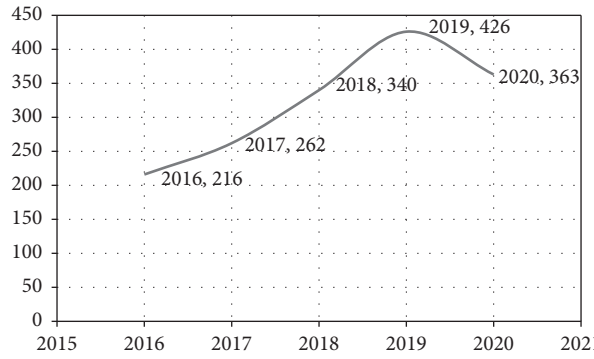


Figure 2: Total number of papers published in the given year.

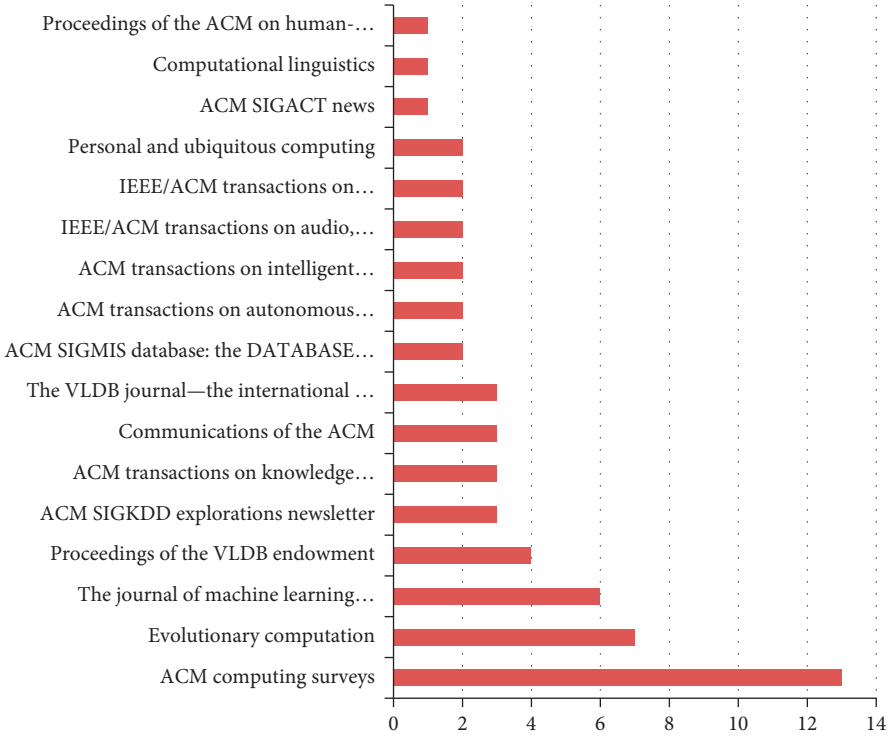


Figure 3: Journal/magazine name and number of publications.

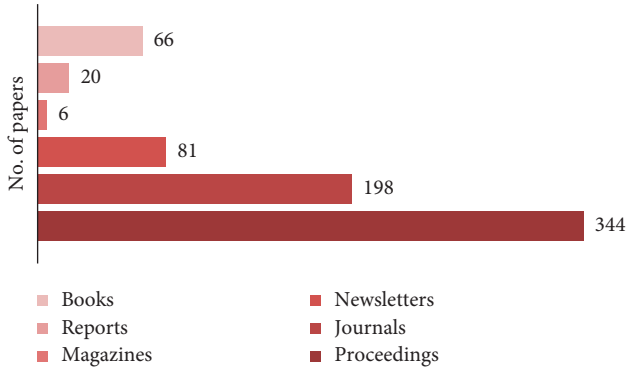


Figure 4: All publications and the total number.

3. The Proposed Methodology

The proposed model consists of an external library (a Kitsune network attack database) developed by Mirsky et al. [49]. This database is used for the simulation and experimental purposes. It consists of nine different attacks depicted in Table 1. It also contains the information about the number of packets selected for the training and test purposes. The experimental setup also contains the feature extractor and feature mapping section. To achieve this goal, the proposed research work uses convolution neural network (CNN) that acts as an automatic feature extractor and classification tool. CNN extracts the features and, based on these features, it generates the output in the form of anomaly detector. In our case, it generates two types of output classes as depicted in

$$\overrightarrow{O(x)} = \begin{bmatrix} \text{Benign} \\ \text{Malicious} \end{bmatrix}, \tag{1}$$

where $\overrightarrow{O(x)}$ represents the corresponding output. This output is generated in the form of malicious and benign data. Finally, the percentile score is generated based on the threshold, N_p norm values, and other parametric values explained in Section 4. Figure 5 shows the experimental setup.

A five-layered CNN architecture is used for the experimental purposes. It consists of an input and output layer and three hidden layers. A "relu" is used as an activation function. This architecture is tested for varying training and test sets. The CNN models are prominent in classifying spatial data.

4. Results and Discussion

The dataset used for the proposed experimental work is selected from the feature vector dataset (https://archive.ics.uci.edu/ml/datasets/Kitsune+Network+Attack+Dataset) developed by Mirsky et al. [49]. They developed this dataset after recording the network traffic on two different networks such as (a) a commercial IP-based camera video surveillance network on which they conducted 8 attacks that affect the availability and integrity of the video uplinks; (b) a noisier IoT network comprised of 9 IoT devices and 3 PCs; one of the devices was infected with the MIrai botnet attacks (malware). From each of these input vectors (in the dataset), we extracted a segment of consecutive packets. These packets are accordingly separated into training and test sets as depicted in Table 3.

Kitsune's developers mostly evaluate the deep learning based intrusion detection systems against a series of attacks based on different networks. In the case of the proposed study, accuracy of the system is dependent relative to the value of threshold, T. when deploying the system this threshold describes the boundary of decision and makes it a crucial parameter.

The following two metrics are followed to access the performance of a certain threshold parameter:

(a) False negative: the percentile of malicious data that is considered/classified as benign data

(b) False positive: the percentile of benign inputs that are considered/classified as malicious data

The false positives rate is associated with the network reliability, while the rate of false negatives accounts for the effectiveness of the network intrusion detection system (NIDS). Therefore, to achieve an ideal situation, both these parameters should be minimized. However, dealing with Kitsune settings, the value of T acts as a trade-off in between both false positives and false negatives parameters.

The functional range of the threshold values ranging from 0 to 15 is investigated for a given training and test set parameters as shown in Table 1. 100% false negatives are recorded for the false negatives on the given feature vector. Figure 6 shows the two threshold parameters versus the accuracy of the proposed system.

It can be observed from Figure 6 that, in the middle, both the parameters (false positives and false negatives) remain unchanged. Furthermore, it can also be concluded from Figure 6 that if we minimize one parameter, the other parameter significantly increases. Finally, the accuracy of the proposed system remains unchanged for a threshold value below 10 (which reflects that most of the data belongs to the benign inputs).

A receiver operating characteristic (ROC) is shown in Figure 7 to represent the effectiveness of the proposed algorithm for the Kitsune network attack dataset.

Two of the significant attacking objectives that are availability and integrity violation are in machine learning techniques. The violations of availability try to make benign traffic appear malicious.

The violations of integrity try to construct malicious traffic which escapes detection.

The network attacks containing the information differ from the images that are most commonly used in generic machine learning techniques.

One of the definitions for examples of adversarial, assisted by the architecture of Kitsune, is to adopt the features extracted as an indication of the difference be observed. So, the distance of L_P is adopted on the space feature between the perturbed input and original input as the distance metric. The L_0 norm correlates to altering a small number of extracted features, which might be a better metric than other L_P norms.

The proposed algorithm is also evaluated against generic NIDS to test the applicability of the proposed algorithm. These generic algorithms include Fast Gradient Sign Method (FGSM), Jacobian Base Saliency Map (JSMA), Carlini and Wagner (C&W), and Elastic Net Method (ENM). A description of these techniques is given as follows:

(i) FGSM**:** over the L1 norm, this technique is strictly optimal (i.e., it reduces the maximum perturbation on any input data (feature)) by selecting a single step to each element of $\sim x$ in the opposite direction to the gradient [50]

(ii) JSMA**:** this type of attack minimizes the L_0 norm by iteratively calculating a saliency map and then perturbing the feature that will have the highest effect [51]

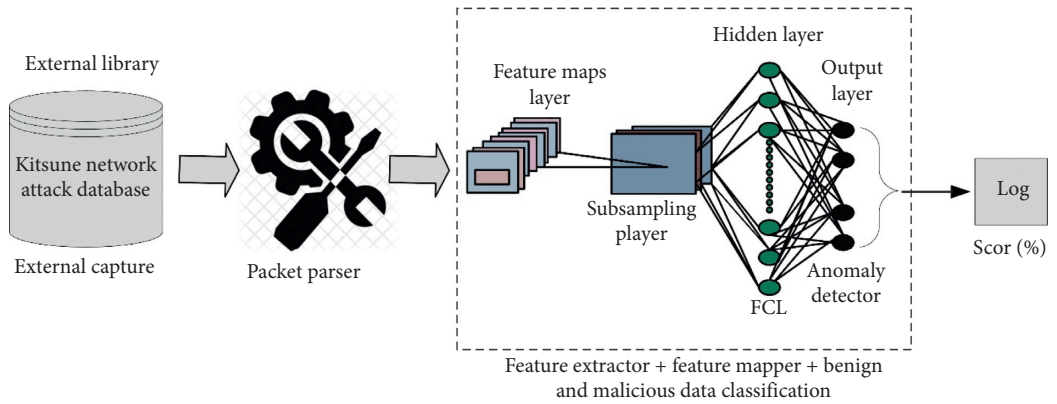


Figure 5: Experimental setup.

Table 3: Characteristics of the IoT network attack and commercial IP camera dataset [49].

S/no.	Type of attacks	Training set (packets)	Test set (packets)	Malicious test packets
(1)	OS Scan	7000	12,500	1499
(2)	Fuzzing SFuzz	1300	8900	1199
(3)	ARP MitM	7000	12,500	1499
(4)	Video Inj.	5000	8000	1199
(5)	Mirai	7000	8000	1199
(6)	SYN DoS	1300	8900	1199
(7)	SSDP Flood	7000	13,500	1499
(8)	SSL Reneg	7000	12,500	1499
(9)	Wiretap	1300	8900	1199

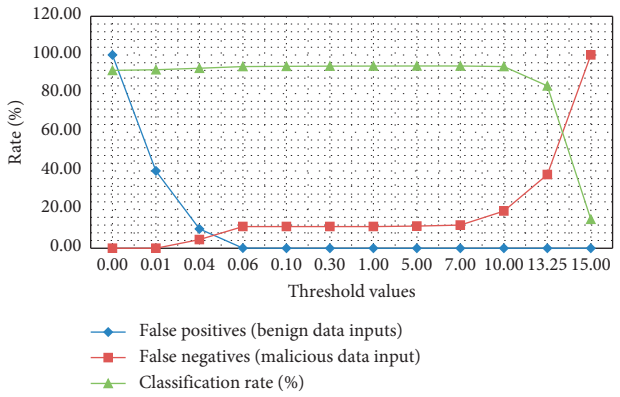


Figure 6: Threshold parameters versus the accuracy of the proposed system.

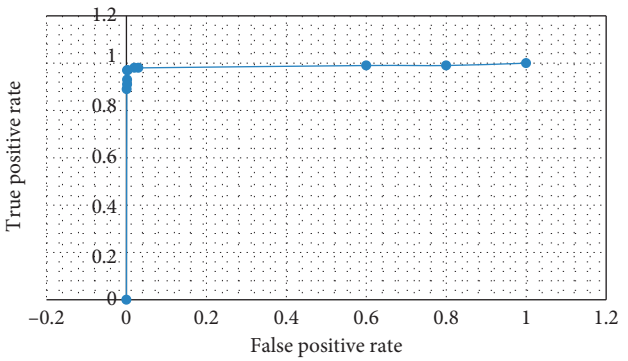


Figure 7: ROC curve for Kitsune.

(iii) C&W: Carlini and Wagner's adversarial framework, as discussed earlier, can either minimize the L_2, L_0, or L_1 distance metric [52]

(iv) ENM: **e**lastic net attack is an algorithm that restricts the total absolute perturbation across the input space. The ENM constructs the adversarial examples by expanding an iterative L_2 attack with an L_1 regularizer [53]

To check the validity of the proposed algorithm, the experimental results are carried out for the selected generic algorithms based on different threshold values ranging from 0.05 to 1 to test the Kitsune score. The experimental results are depicted in Table 4.

From Table 4, it is evident that our algorithm performs well compared to the other generic algorithms. The experimental results are carried out on the input vectors selected from the Kitsune network attack dataset as depicted in Table 1. The simulated results are shown in Figure 8.

For the same threshold values used in Table 2, the availability attacks on the Kitsune network are processed. Different training sets are selected for the simulation purposes as shown in Table 1. The input vectors (training sets) that yield closest output scores to the threshold were selected. Table 5 shows the experimental results. The normalizers were trained on benign inputs; several malicious input values would be normalized outside the typical range between 0.05 and 1.

Table 4: Integrity attacks on Kitsune network.

S/no.	Algorithms	Threshold value	Kitsune score (%)	L_p distances			
				L_0	L_1	L_2	L_∞
(1)	FGSM	1.0	100	100	102	9.7	1.5
(2)	JSMA	1.0	100	1.98	9.23	6.29	4.10
(3)	C&W	1.0	100	100	6.89	3.23	3.46
(4)	ENM	1.0	100	1.02	4.09	2.98	3.98
(5)	Our algorithm	1.0	100	0.87	3.33	1.09	3.45

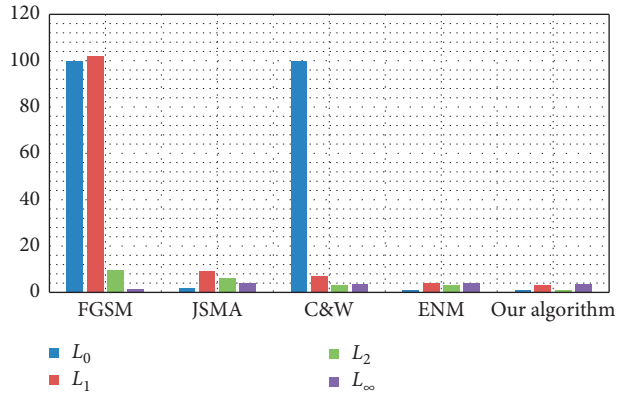


Figure 8: Integrity attack on Kitsune network.

Table 5: Availability attacks on Kitsune network.

S/no.	Algorithms	Threshold value	Kitsune score (%)	L_p distances			
				L_0	L_1	L_2	L_∞
(1)	FGSM	1.0	4	100	76.53	7.03	0.69
(2)	JSMA	1.0	0	—	—	—	—
(3)	C&W	1.0	100	100	21.32	7.98	5.46
(4)	ENM	1.0	100	7.85	20.09	7.48	3.68
(5)	Our algorithm	1.0	100	5.87	13.33	6.09	2.45

From Table 5, it is depicted that our algorithm outperforms for the availability attacks as well using the Kitsune network attack dataset. The comparative results are also shown in Figure 9. From Figure 9, it is concluded that our algorithm outperforms very well compared to the other generic algorithms in the proposed field.

To minimize the attacks on the Kitsune network, Cleverhans implementations are followed. These implementations use a simple gradient descent optimizer to minimize the function that is represented using

$$c.\max\{F(\vec{x})_i - Y, 0\} + \beta\vec{x} - \vec{x}_0 1 + \vec{x} - \vec{x}_0 2, \quad (2)$$

where $F(\vec{x})_i$ is the logit output of the target classifier, Y is the logit target output, and $\vec{x}_0$ is the original network input data. It can be seen that there are two regularization parameters, c and β. These parameters help in determining the contribution of the several metrics to the attacking algorithms, the success rate and L_1 distance with respect to changes in the regularization parameter, c.

The parameter, c, helps in determining the contribution of the adversarial misclassification objectives at the cost of diminishing the two LP normalization terms. For $\beta = 1$ and c the parametric values range from 0 to 500. And it is concluded from Figures 10 and 11 that 500 is the optimal parametric value for c that results in 100% success rate with a small perturbation. It can also be seen

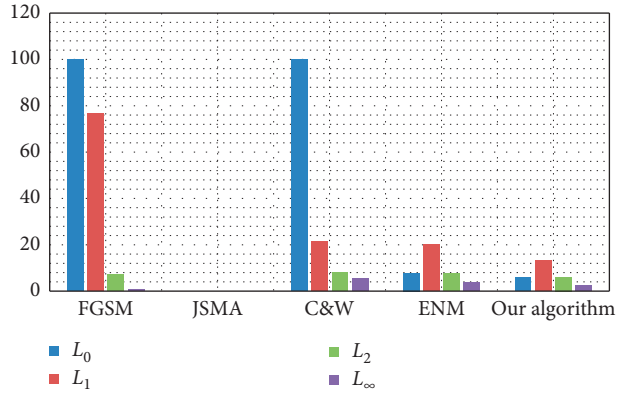


Figure 9: Availability attacks on Kitsune network.

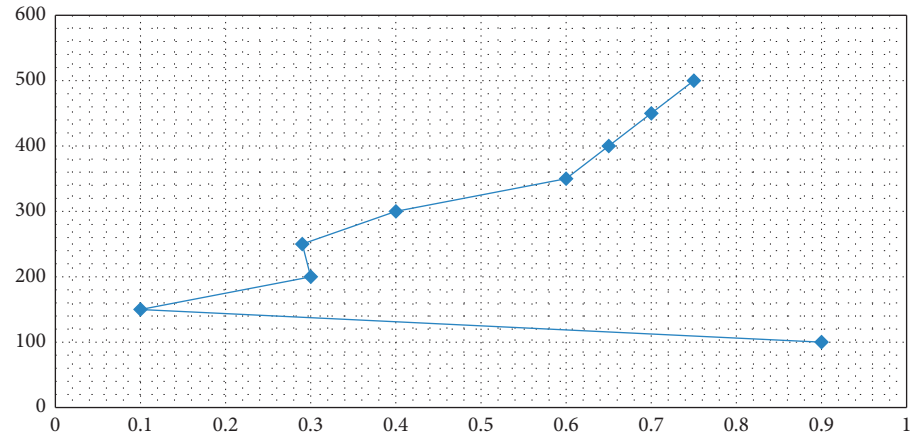


Figure 10: L_1 distance.

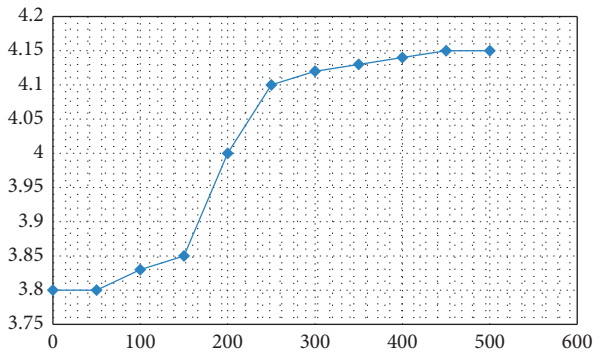


Figure 11: Success rate.

in Figure 10 that the generated L_1 distance does not directly correlate with the selection of parametric c values.

5. Conclusion

Security of components plays an important role in the functionality of a system to properly function. Different security approaches and measures are used to secure the information communication and to secure the network. This measure includes firewalls, logical access, control, authentication, identification, and encryption and decryption. A convolution neural network based approach is followed for the feature classification and benign and malicious data identification purposes. In the end, comparative results are generated after evaluating the performance of the proposed algorithm to other rival algorithms in the proposed field. These algorithms include FGSM, JSMA, C&W, and ENM. After assessing the performance of these algorithms and the proposed algorithm based on different threshold values ranging, L_p norms, and different parametric values for c, it was derived that the proposed algorithm outperforms with small L_p values and high Kitsune scores. These results show that the proposed research is capable of identifying intrusion and replicating the application of the proposed algorithms in the field of network security.

Data Availability

The proposed study has used the data avaliable online in the UCI Machine Learning Repository.

Conflicts of Interest

The authors declare no conflicts of interest regarding this paper.

References

[1] H. H. Song, "Testing and evaluation system for cloud computing information security products," in *Proceedings of the 3rd International Conference on Mechatronics and Intelligent Robotics (ICMIR-2019)*, pp. 84–87, Kunming, China, May 2020.

[2] B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, "Survey on IoT security: challenges and solution using machine learning, artificial intelligence and blockchain technology," *Internet of Things*, vol. 11, Article ID 100227, 2020.

[3] J. Yuan and X. Luo, "Regional energy security performance evaluation in China using MTGS and SPA-TOPSIS," *Science of the Total Environment*, vol. 696, Article ID 133817, 2019.

[4] X. Wu, S. Liu, Y. Sun, Y. An, S. Dong, and G. Liu, "Ecological security evaluation based on entropy matter-element model: a case study of Kunming city, southwest China," *Ecological Indicators*, vol. 102, pp. 469–478, 2019.

[5] X. Wang, J. Li, X. Kuang, Y.-A. Tan, and J. Li, "The security of machine learning in an adversarial setting: a survey," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12–23, 2019.

[6] M. Marwan, A. Kartit, and H. Ouahmane, "Security enhancement in healthcare cloud using machine learning," *Procedia Computer Science*, vol. 127, pp. 388–397, 2018.

[7] Z. Zhang, J. Wen, X. Wang, and C. Zhao, "A novel crowd evaluation method for security and trustworthiness of online social networks platforms based on signaling theory," *Journal of Computational Science*, vol. 26, pp. 468–477, 2018.

[8] Y. Cherdantseva, J. Hilton, O. Rana, and W. Ivins, "A multifaceted evaluation of the reference model of information assurance & security," *Computers & Security*, vol. 63, pp. 45–66, 2016.

[9] M. Jouini, L. B. A. Rabai, and R. Khedri, "A multidimensional approach towards a quantitative assessment of security

threats," *Procedia Computer Science*, vol. 52, pp. 507–514, 2015.

[10] S. Nazir, S. Shahzad, M. Nazir, and H. U. Rehman, "Evaluating security of software components using analytic network process," in *Proceedings of the 11th International Conference on Frontiers of Information Technology (FIT)*, pp. 183–188, IEEE, Islamabad, Pakistan, December 2013.

[11] M. Li, S. Nazir, H. U. Khan, S. Shahzad, and R. Amin, "Modelling features-based birthmarks for security of end-to-end communication system," *Security and Communication Networks*, vol. 2020, Article ID 8852124, 9 pages, 2020.

[12] H. U. Rahman, A. U. Rehman, S. Nazir, I. U. Rehman, and N. Uddin, "Privacy and security—limits of personal information to minimize loss of privacy," in *Lecture Notes in Networks and Systems*, pp. 964–974, Springer, Berlin, Germany, 2020.

[13] S. Nazir, S. Shahzad, S. Mahfooz, and M. N. Jan, "Fuzzy logic based decision support system for component security evaluation," *International Arab Journal of Information and Technology*, vol. 15, pp. 1–9, 2015.

[14] I. Kotenko and A. Chechulin, "Common framework for attack modeling and security evaluation in SIEM systems," in *Proceedings of the 2012 IEEE International Conference on Green Computing and Communications*, pp. 94–101, Besancon, France, November 2012.

[15] P. Subsorn and S. Limwiriyakul, "A comparative analysis of internet banking security in Thailand: a customer perspective," *Procedia Engineering*, vol. 32, pp. 260–272, 2012.

[16] I. Kotenko and A. Chechulin, "Computer attack modeling and security evaluation based on attack graphs," in *Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, pp. 614–619, Berlin, Germany, September 2013.

[17] S. Manjiatahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of things (IoT): a survey," *Journal of Network and Computer Applications*, vol. 161, Article ID 102630, 2020.

[18] R. Diesch, M. Pfaff, and H. Krcmar, "A comprehensive model of information security factors for decision-makers," *Computers & Security*, vol. 92, Article ID 101747, 2020.

[19] M. A. Amanullah, R. A. A. Habeeb, F. H. Nasaruddin et al., "Deep learning and big data technologies for IoT security," *Computer Communications*, vol. 151, pp. 495–517, 2020.

[20] S. Khan, H. Ali, Z. Ullah, N. Minallah, S. Maqsood, and A. Hafeez, "KNN and ANN-based recognition of handwritten pashto letters using zoning features," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 570–577, 2018.

[21] N. A. B. Mohd and Z. F. Zaaba, "A review of usability and security evaluation model of ecommerce website," *Procedia Computer Science*, vol. 161, pp. 1199–1205, 2019.

[22] W. Mao, Z. Cai, D. Towsley, Q. Feng, and X. Guan, "Security importance assessment for system objects and malware detection," *Computers & Security*, vol. 68, pp. 47–68, 2017.

[23] G. George and S. M. Thampi, "Vulnerability-based risk assessment and mitigation strategies for edge devices in the internet of things," *Pervasive and Mobile Computing*, vol. 59, Article ID 101068, 2019.

[24] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: internet of threats? A survey of practical security vulnerabilities in real IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[25] M. Ammar, G. Russello, and B. Crispo, "Internet of things: a survey on the security of IoT frameworks," *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2018.

[26] M. A. Khan and K. Salah, "IoT security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[27] K. Liu, W. Shen, Y. Cheng et al., "Security analysis of mobile device-to-device network applications," *IEEE Internet of Things Journal*, vol. 6, pp. 2922–2932, 2018.

[28] A. Papageorgiou, M. Strigkos, E. Politou, E. Alepis, A. Solanas, and C. Patsakis, "Security and privacy analysis of mobile health applications: the alarming state of practice," *IEEE Access*, vol. 6, pp. 9390–9403, 2018.

[29] R. Gurunath, M. Agarwal, A. Nandi, and D. Samanta, "An overview: security issue in IoT network," in *Proceedings of the 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 104–107, Palladam, India, August 2018.

[30] R. Román-Castro, J. López, and S. Gritzalis, "Evolution and trends in iot security," *Computer*, vol. 51, no. 7, pp. 16–25, 2018.

[31] X. Su, Z. Wang, X. Liu, C. Choi, and D. Choi, "Study to improve security for IoT smart device controller: drawbacks and countermeasures," *Security and Communication Networks*, vol. 2018, Article ID 4296934, 14 pages, 2018.

[32] A. Rodríguez-Mota, P. J. Escamilla-Ambrosio, J. Happa, and E. Aguirre-Anaya, "GARMDROID: IoT potential security threats analysis through the inference of android applications hardware features requirements," in *Applications for Future*, pp. 63–74, Springer, Berlin, Germany, 2017.

[33] F. Loi, A. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman, "Systematically evaluating security and privacy for consumer IoT devices," in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, pp. 1–6, Dallas, TX, USA, November 2017.

[34] M. Capellupo, J. Liranzo, M. Z. A. Bhuiyan, T. Hayajneh, and G. Wang, "Security and attack vector analysis of IoT devices," in *Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 593–606, Guangzhou, China, December 2017.

[35] M. H. Khan and M. A. Shah, "Survey on security threats of smartphones in internet of things," in *Proceedings of the 2016 22nd International Conference on Automation and Computing (ICAC)*, pp. 560–566, Colchester, UK, September 2016.

[36] J. Ahamed and A. V. Rajan, "Internet of things (IoT): application systems and security vulnerabilities," in *Proceedings of the 2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, pp. 1–5, Ras Al Khaimah, UAE, December 2016.

[37] A. Kamilaris and A. Pitsillides, "Mobile phone computing and the internet of things: a survey," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 885–898, 2016.

[38] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," in *Proceedings of the 2015 IEEE World Congress on Services*, pp. 21–28, New York, NY, USA, June 2015.

[39] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of things: security vulnerabilities and challenges," in *Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC)*, pp. 180–187, Larnaca, Cyprus, July 2015.

[40] B. Liao, Y. Ali, S. Nazir, L. He, and H. U. Khan, "Security analysis of IoT devices by using mobile computing: a systematic literature review," *IEEE Access*, vol. 8, pp. 120331–120350, 2020.

[41] V. G. Shankar, G. Somani, M. S. Gaur, V. Laxmi, and M. Conti, "AndroTaint: an efficient android malware detection framework using dynamic taint analysis," in *Proceedings of the 2017 ISEA Asia Security and Privacy (ISEASP)*, pp. 1–13, Surat, India, January 2017.

[42] A. Feizollah, N. B. Anuar, R. Salleh, G. Suarez-Tangil, and S. Furnell, "Androdialysis: analysis of android intent effectiveness in malware detection," *Computers & Security*, vol. 65, pp. 121–134, 2017.

[43] H. Liang, Y. Song, and D. Xiao, "An end-to-end model for android malware detection," in *Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 140–142, Beijing, China, July 2017.

[44] F. Ghaffari, M. Abadi, and A. Tajoddin, "AMD-EC: anomaly-based android malware detection using ensemble classifiers," in *Proceedings of the 2017 Iranian Conference on Electrical Engineering (ICEE)*, pp. 2247–2252, Tehran, Iran, May 2017.

[45] E. Gandotra, D. Bansal, and S. Sofat, "Zero-day malware detection," in *Proceedings of the 2016 Sixth International Symposium on Embedded Computing and System Design (ISED)*, pp. 171–175, Patna, India, December 2016.

[46] P. Palumbo, L. Sayfullina, D. Komashinskiy, E. Eirola, and J. Karhunen, "A pragmatic android malware detection procedure," *Computers & Security*, vol. 70, pp. 689–701, 2017.

[47] D. Li, Z. Wang, L. Li, Z. Wang, Y. Wang, and Y. Xue, "FgDetector: fine-grained android malware detection," in *Proceedings of the 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pp. 311–318, Shenzhen, China, June 2017.

[48] F. Shen, J. Del Vecchio, A. Mohaisen, S. Y. Ko, and L. Ziarek, "Android malware detection using complex-flows," *IEEE Transactions on Mobile Computing*, vol. 18, pp. 1231–1245, 2018.

[49] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune network attack dataset data set," 2019, https://archive.ics.uci.edu/ml/datasets/Kitsune+Network+Attack+Dataset.

[50] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, https://arxiv.org/abs/1412.6572.

[51] N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, Saarbrucken, Germany, March 2016.

[52] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (sp)*, pp. 39–57, San Jose, CA, USA, May 2017.

[53] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "EAD: elastic-net attacks to deep neural networks via adversarial examples," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, February 2018.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8865474, 9 pages
https://doi.org/10.1155/2020/8865474

Research Article

A Smart Agent Design for Cyber Security Based on Honeypot and Machine Learning

Nadiya El Kamel,[1] **Mohamed Eddabbah**,[2] **Youssef Lmoumen**,[1] **and Raja Touahni**[1]

[1]*Laboratoire des Systèmes de Télécommunication et Ingénierie de la Décision (LASTID), Département de Physique, Faculté des Sciences, Université Ibn Tofail, Kenitra, Morocco*
[2]*LABTIC Laboratory ENSA, Abdelmalek Essaadi University Tangier, Tangier, Morocco*

Correspondence should be addressed to Nadiya El Kamel; nadiya.elkamel@uit.ac.ma

Received 17 May 2020; Accepted 16 July 2020; Published 7 August 2020

Academic Editor: Sajjad Shaukat

The development of Internet and social media contributes to multiplying the data produced on the Internet and the connected nodes, but the default installation and the configuration of variety of software systems represent some security holes and shortcomings, while the majority of Internet users have not really set up safety awareness, leading to huge security risks. With the development of network attack techniques, every host on the Internet has become the target of attacks. Therefore, the network information security cannot be ignored as a problem. To deal with 0-day and future attacks, the honeypot technique can be used not only passively as an information system, but also to reinforce the traditional defense systems against future attacks. In this paper, we present an introduction of machine learning and honeypot systems, and based on these technologies, we design a smart agent for cyber-attack prevention and prediction.

1. Introduction

According to GDATA [1], the number of new attacks increases exponentially, each year, millions of attacks are detected (Figure 1), which involve more sophisticated and automatic analysis tools, since traditional tools are limited in the case of a huge quantity of information or when it is about new kinds of attacks. In fact, the main disadvantages of expert-based analysis are time consumption and the difficulty of classifying attacks [2].

The knowledge of opponent motivations, objectives, and techniques used to gain unauthorized access to the systems is the key not only to stop and protect systems from attacks but also to learn and predict new attacks that can hit our systems. Honeypots technology was deployed since 1992 [3], as a powerful information system, which consists of monitoring, detecting, and analyzing malicious activities, it is used to complement the traditional strategies such as intrusion detection systems (IDS) and log files, which are ineffective due to the huge quantity of information, false alarms, and the inability of detecting new attacks [4]. The honeypot is a security resource implemented for being probed, attacked, or compromised [4, 5], it was proposed to automatically consider any interaction detected as a malicious activity, while the administrator network uses the reports generated by the malicious source, to learn about the identity, motivations, and techniques used by the intruder to infiltrate the system.

The purpose of this paper is to show, firstly, the strength of using machine learning and honeypots, as solutions for the cyber security purpose, through some related works and by introducing these technologies. The second purpose of this work is to discuss a cyber security solution based on honeypot and machine learning techniques. Our main objective is to design an intelligent agent for predicting new attack profiles by analyzing, automatically, the gathered data via the honeypot, using a combination of machine learning algorithms. The objective of the algorithms combination is to represent the data with a lot of accuracy and build an efficient predictive agent for cyber security, especially for the future and 0-day attacks prediction.

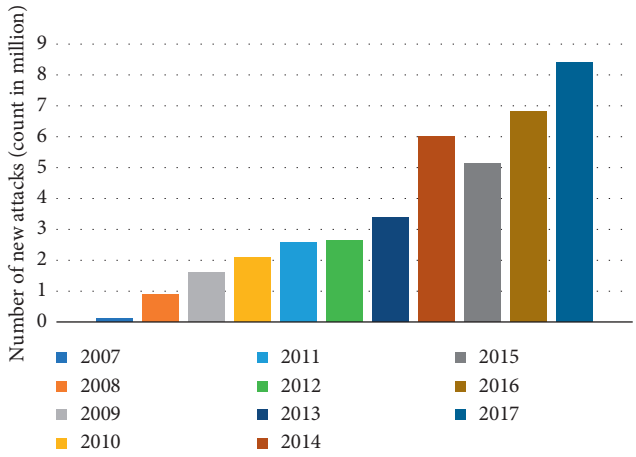


Figure 1: New attacks evolution.

The rest of this paper is organized as follows: in Section 2, we discuss some related work, Sections 3 and 4 are devoted to the introduction of machine learning and honeypot technologies for cyber security, and in Section 5, we discuss the proposed predictive design.

2. Related Work

Anomalies characterization receives a lot of attention; all seek to protect themselves against fraudulent use of their data or against malicious intrusions into computer systems. A lot of security solutions were proposed in the last decade, but results still present some limitations [2], and the most recent works are based on machine learning algorithms to model anomalies activities using data collected by information technologies such as honeypots.

The authors in [6] propose an intelligent honeypot which improves IoTs devices' security, based on machine learning. In order to store each device response, an IoT-scanner was proposed to probe accessible IoT devices on the Internet and scan the Internet for each malicious interaction, and a model called IoTLearner was trained to be used by the intelligent honeypot that can optimize a model to reply attackers.

The authors in [2] propose an autonomous method for attacks characterization, based on unsupervised anomalies learning, using the collected information by honeypots. This approach is based on clustering techniques such as density-based clustering, subspace clustering, and evidence accumulation for classifying flow ensembles in traffic classes. The advantage of this method is that it does not require a training phase.

The authors in [7] propose an automatic classification of social spam-based machine learning (e.g., SVM), for network communities such as Facebook and MySpace using a social honeypot to gather information about malicious profiles.

The authors in [8] propose a linkage defense system [9] based honeypot to overcome the limitations of the traditional tools. The linkage technique will ensure management and communication between the honeypot and components of the defense system, constructing a linkage management module based on SNMP protocol for network management. In order to overcome the problem of new attacks, the system is centroid honeypot for treating suspicious flows arrived from the traditional defense system, and the decision to block or not will depend on the state of the honeypot. If the honeypot is damaged, then the correspondent intruder will be blocked by the firewall [8].

3. Machine Learning

A computer is not smart; it performs tasks described in a program form, as orders of what to do and how to do it, and this is called traditional programming. While writing a traditional program, the decision is made directly into the program. Machine learning is a subarea of artificial intelligence [10] that aims to give computers the opportunity to learn; its techniques allow understanding the structure of the data and integrating them into models that can be understood and used to solve complex problems in real-life situations, and its techniques represent an efficient tool to address the significant challenges posted by the big data [11].

Machine learning has turned the concept of traditional programming around (Figure 2), by training models and making them able to learn and make decisions without being explicitly programmed. The job of a machine learning model is to output predictions based on mathematical hypothesis functions, and while passing data, this hypothesis maps from input variables to outputs or to find structures or clusters in them.

A machine learning model is designed in two phases: the first is to estimate the model from the available data, by executing practical tasks such as animal recognition in pictures, speech translation, or participating in autonomous vehicles driving, this is called the training phase, and it is generally performed before the practical use of the model. The second is the production phase, meaning the phase of passing new data to obtain the result corresponding to the desired task. According to the information available during the learning phase, learning is qualified in different ways; if the data is labelled, then the learning would be supervised, in a more general case, and without labels, the learning is unsupervised.

Depending on the nature of the problem, there are different approaches that vary depending on the type and volume of data. Machine learning categories are divided into supervised, unsupervised, and reinforcement learning [12]. In this section, we discuss briefly each category.

The supervised learning [11, 12] consists of learning prediction functions from a database of pairs input-output or from labelled examples. The supervised learning problems can be divided into two categories, namely, regression and classification problems [13]. In regression problems, results are output within continuous values by mapping input variables to some continuous functions. In contrast, classification problems allow to output results within discrete values. There are several supervised algorithms widely used such as linear regression (LR), support vector machine (SVM), and decision trees (DT). For the unsupervised learning [11, 12], the data have no labels, and we only receive

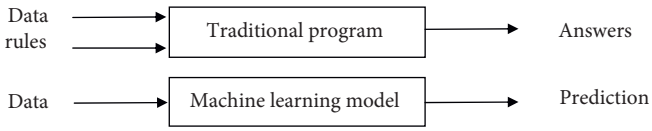


Figure 2: Comparison between machine learning and traditional programming.

raw observations of random variables. Therefore, the training algorithm applies in this case, to find similarities and distinctions within the data and group together all those which have common characteristics [12]. For example, an epidemiologist would like to try to bring out, in a large set of liver cancer victims, an explanatory hypothesis to this illness. The computer could differentiate some groups, which the epidemiologist would, then, associate with various explanatory factors, geographical origins, genetics, consumption habits or practices, and exposure to various potentially or actually toxic agents (heavy metals and toxins such as aflatoxin). Another approach of semisupervised learning [12] was proposed; it consists of a combination between a set of labelled and not labelled data; it is a good compromise between the two "supervised" and "unsupervised" types of learning, it allows to analyze a large number of data without the need to label them all, and takes advantage of both types mentioned. The last type of learning is the reinforcement learning [3, 11, 12], in which an agent interacts iteratively with the environment to learn and to improve its comportment simultaneously. Reinforcement learning differs from the first type of learning in the way that, in supervised learning, the training data are labelled, so the model is trained with the correct answer itself, whereas in reinforcement learning, there are no labels; instead, the agent learns from its experience to performs the given task [3].

The next part is devoted to discuss some machine learning examples.

3.1. Linear Regression. LR (Linear regression) [14] is used to estimate a linear hypothesis function between the output and the input variables, as regression or a classification tool [15], and it is written as follows:

$$h_\theta(X) = \theta_0 + \theta_1 \cdot X_1 + \ldots + \theta_n \cdot X_n, \tag{1}$$

where h_θ is the hypothesis function, X_i represents input variables, and θ_{ii} represents weights of the hypothesis function.

Weights represent the parameters of the hypothesis function. For estimating weight values, the first step is to calculate the error between the estimated result $(\widehat{y})$ and the expected result (y) using the cost function. The mean squared error (MSE) is the most widely used cost function, and it is written as follows:

$$J = \frac{1}{N} \sum_{i=1}^{N} \left(\widehat{y}\left(x^{(i)}\right) - y^{(i)}\right)^2. \tag{2}$$

The second step is to apply the gradient descent algorithm [16], which represents the most important part in linear regression, is the simplest algorithm to apply, and it allows to estimate the optimal weight values by minimizing the cost function. The gradient descent is an iterative algorithm which updates the weights at each iteration to minimize the cost function by setting a threshold value.

3.2. K-Means Algorithm. *K*-means [17] is one of the most widely used clustering algorithms; it is an iterative algorithm, where the first step is to randomly initialize cluster centroids [17] and start moving the centroids until every point is allocated to the nearest cluster, keeping the centroids as small as possible. The most widely used clustering criterion is the sum of the squared Euclidean distances between each data point $(x_{1,}x_2, \ldots, x_N)$ and the centroid m_k, and this criterion is called clustering error *E*.

$$E\left(m_{1,}m_2, \ldots, m_M\right) = \sum_{1}^{N} \sum_{1}^{M} \left\|x_i - m_k\right\|^2. \tag{3}$$

The K-means algorithm finds locally optimal solutions with respect to the clustering error. It is a fast-iterative algorithm that has been used in many clustering applications [18], but it is still having a major disadvantage which backs to its sensitivity to initial positions of the cluster centers and the number of clusters [17, 19]. Therefore, in order to obtain near optimal solutions using the k-means algorithm, several runs must be scheduled differing in the initial positions of the cluster centers. To overcome this problem, different approaches have been proposed such as the Global K-means approach [20].

3.3. Decision Trees. A decision tree is a very simple model representing a set of choices in a graphic form of a tree [21], and it is a hierarchical representation of the data structure in the form of decision sequences (tests) for classes or results prediction. Given several characteristics, the decision begins with one of these characteristics, and if that is not enough, another one is used and so on. Each individual (or observation), must be assigned to a class which is described by a set of variables tested in the tree's nodes. Tests are performed in internal nodes, and decisions are made in leaf nodes [22]. DT (decision trees) can be used for both classification and regression problems. There are several automatic algorithms for building decision trees: ID3, C4.5, and CART.

ID3 starts with placing all the learning examples in the root node. Then, each node is cut to one of the remaining attributes (which has not been tested yet). The choice of this attribute is made through a measure of homogeneity; it uses the entropy function and the gain of information according to an attribute to decide which is the best characteristic [21].

ID3 shows some limitations when it is about continuous characteristics and a large number of feature values [22]. C4.5 is an extension of ID3; it was proposed in 1993 by Ross Quinlan as an amelioration of ID3, to support continuous attributes and missing values, and it is based on the pruning technique to reduce the prediction error rate [22]. The authors in [22] proposed a comparative study of decision trees ID3 and C4.5, they compared the execution time and

the accuracy in function of the dataset size, and the results show that C4.5 is similar to ID3 in terms of accuracy, but it is more effective than ID3 in the execution time.

Machine learning is used for a wide spectrum of applications, and it represents an efficient analysis method for smart cities, especially in intelligent transportation systems (ITS) and smart grids, due to the large amount of generated data by control systems, information and communication technologies, and advanced sensors. In ITS, different machine learning methods such as deep neural network (DNN) and deep reinforcement learning (DRL) were proposed for monitoring and estimating real-time traffic flow data, estimating the possibility of accidents, trajectory design, and cyber physical security [12]. Smart grid networks use computer technologies to optimize the production, distribution, consumption, and possibly, the storage of energy in order to better coordinate all the meshes of the electrical network, from producers to the final consumer. A lot of proposed works are based on machine learning algorithms to analyze heterogeneous data arrived from different sources, and power grid control, a Deep Long Short-Term Memory (DLSTM) model, was proposed to forecast the price and demand for electricity for a day and week ahead [12].

4. Honeypot

Information security policy generally aims to set up mechanisms to guarantee services in terms of integrity, confidentiality, authentication, identification, availability, and access control. Attacks are based on tools that scan the entire networks looking for vulnerable systems [23]; hence, the originality of the honeypot lies in the fact that the system is voluntarily presented as a weak source able to hold the attention of attackers [5]. The general purpose of honeypots is to make the intruder believe that he can take control of a real production machine, which will allow the administrator to observe the means of compromising the attackers, to guard against new attacks, and give him more time to react.

Honeypots are very flexible and exist in different forms. Most works classify honeypots within two ways: the first classification consists in classifying honeypots according to the interactions they allow, and a second classification categorizes them according to their usefulness [5, 24]. In our classification, we will be interested in presenting the advantages and disadvantages of each class of Honeypots and giving some examples.

Low-interaction Honeypots [25] are limited to the degree of emulation offered by the honeypot; hence, the interaction between the attacker and honeypot system is low. These honeypots offer few privileges to the intruder who will have a limited scope. For example, the Honeypot can emulate an FTP (File Transfer Protocol) service on port 21, but emulate only the login command or one other command. The advantage of low-interaction Honeypots is their simplicity; they are easier to implement and manage and pose little risk since the attacker is limited. But, this type records limited information and can only monitor known activities, and emulated services cannot do a lot. It is, therefore, easier for an attacker to detect a low-interaction honeypot. Tools such as KFSENSOR (refer to the next section) are examples of low-interaction honeypots. Medium-interaction honeypots [26] are also discussed in some works; this type of honeypots allows the intruder to get little more access than low-interaction honeypots, offers better simulation services, and enables logging of more advanced attacks, but it requires more time to implement and certain level of expertise. High-interaction honeypots [27] involve the use of real operating systems and real applications, and this is not about emulation; we provide the attacker with something real. The risks are numerous since our machine is intended to be compromised [24]. One of the advantages of this solution is that it is possible to obtain a lot of information because attackers have access to real systems. Hence, high-interaction honeypots allow examining all behaviors and methods, as well as the tools used by the attacker, and take knowledge if the attack is new or not. An example of high-interaction honeypots is the honeynet. On the other hand, honeypots classification is performed according to the specification of utilization; one type is production oriented, the other is research oriented. Honeypots are used in production to protect a company (prevention and detection) and help to find solutions against attacks. Low-interacting honeypots are often used for production [24]. They can be used for research, to collect information about attackers, tools, and motivations, in order to predict future attacks or to provide judicial elements.

4.1. Honeypot Deployment. Deployment of honeypots depends on whether the decoy system is intended to monitor external or internal attacks to the organization's network; hence, it can be installed in front of the firewall, in a demilitarized zone (DMZ), or behind the firewall [5].

The main advantage of choosing the first position (Figure 3) is that there is no change to be made to the firewall filtering rules which protects the internal network, and the location does not introduce new risks for machines on the internal network, but it does not detect attacks carried out from inside the network since, generally, the outgoing flows are blocked by the firewall. The second position is to place the honeypot in a demilitarized zone, and the advantage of the DMZ is that it provides public servers on the Internet isolated from the internal network. This DMZ can be used for production servers (Figure 4) or dedicated to honeypot, while the firewall only allows incoming flows to pass through to the DMZ for available services. This makes it possible to analyze only attempted attacks for the services in question. For the last position (Figure 5) and in case the decoy system is used to detect external attacks, it can induce greater risks of vulnerabilities since once compromised, the decoy system can be used by the attacker to launch other attacks on the internal network. But, this position allows detecting attacks from internal users of the organization to internal services or detecting a bad firewall configuration [5].

4.2. KFSensor. This tool is very simple to set up and does not require a lot of resources systems, and it is based on intrusion detection systems [28]. KFSensor is a server listening

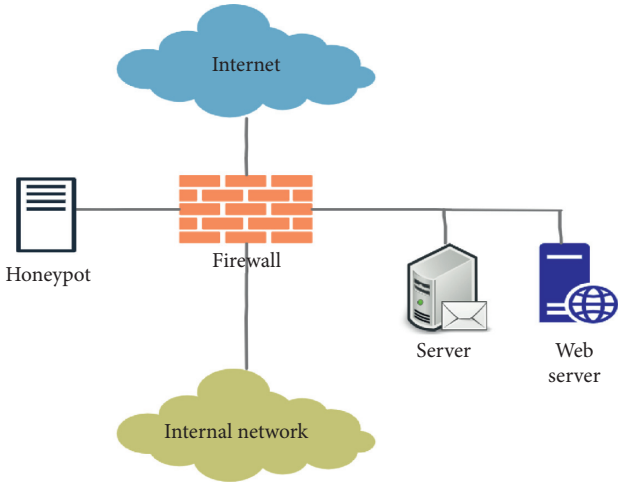


Figure 3: Honeypot deployed independently.

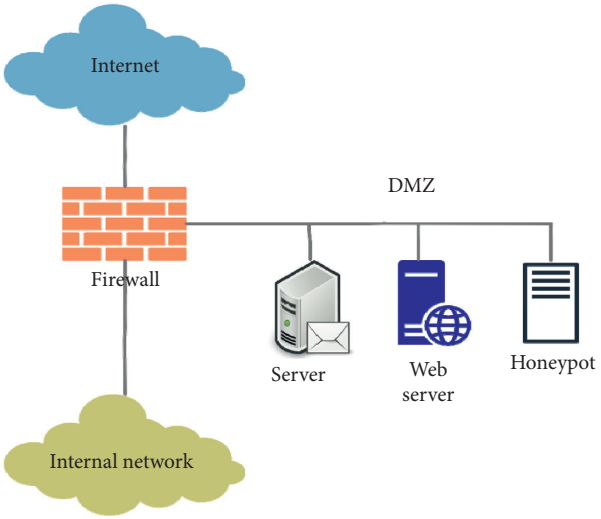


Figure 4: Honeypot deployed in a DMZ.

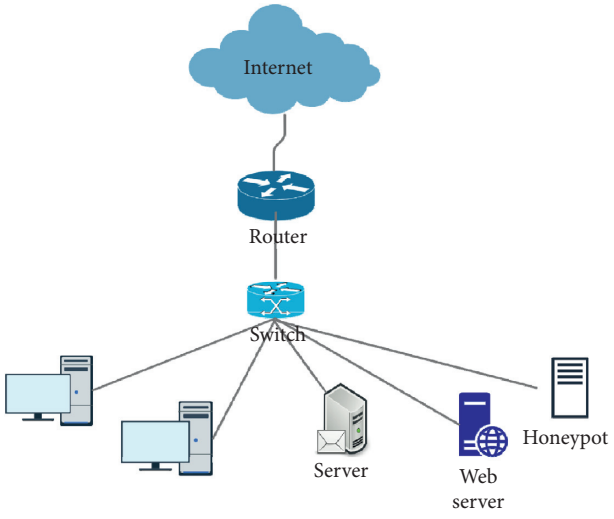


Figure 5: Honeypot deployed in the LAN.

for connections to the network input and monitoring the different port numbers [28], which has a standby user interface (monitoring user interface).

KFSensor professional monitor attacks on every TCP and UDP port, as well as detects ICMP [29] or ping messages. When KFSensor passes new connection information and the data received from visitors to the signature engine, for comparing its data with each signature rule stored in its signature base, and if a match is found, the signature ID is stored with the event in the event log.

4.3. Netfacade. Netfacade allows to simulate a network of vulnerable hosts (honeynet) using the redirect unused IP addresses on a range of addresses to vulnerable services [28].

Netfacade provides a medium level of interaction, it is able to emulate services such as FTP, SendMail, IMAP, HTTP, NFS, and SSH and various operating systems such as Linux, Solaris, and Windows NT.

4.4. Specter. Specter can emulate some network services, as well as some systems of the most common operations. It is classified between as a medium-interaction honeypot and emulates various services such as FTP, POP3, and HTTP [30]. Specter is usable on operating systems based on Windows NT, 2000 or XP [28].

4.5. CurrPorts. CurrPort displays the list of all opened TCP/IP and UDP ports on the local network, with the process that opened the port, and allows to close unwanted TCP connections, kill the process that opened the ports, save the TCP/UDP ports information to HTML file, XML file, or to tab-delimited text file.

5. Smart Agent for Cyber Security Attacks Prevention and a Prediction-Based Machine Learning and Honeypot System

Companies have invested a great deal on time and money in manual networks reconfiguration, in order to protect information systems from infiltration. It is well known that the locks break and the keys can be copied; therefore, it is an illusion to think that a lock and a key represent perfect security. So, the real challenge in terms of cyber security is to accept the probability of an imminent attack and to understand what is really going on within complex information systems. Traditional security tools such as IDS, Firewalls, and IPS can protect systems against simple attacks that use the same tools and tactics repeatedly. They are implemented independently; hence, there is no contact between them to block intrusion detected in an IDS by the firewall [8], for example, they represent a passive solution when it is about 0-day attacks.

The proposed solution is based on honeypot systems and ML techniques combination, as tools for gathering information, analysis, and threat predictions, in order to ensure the security of companies' networks. The implementation of honeypots depends on the services offered to customers by the production company's servers. In Figure 6,

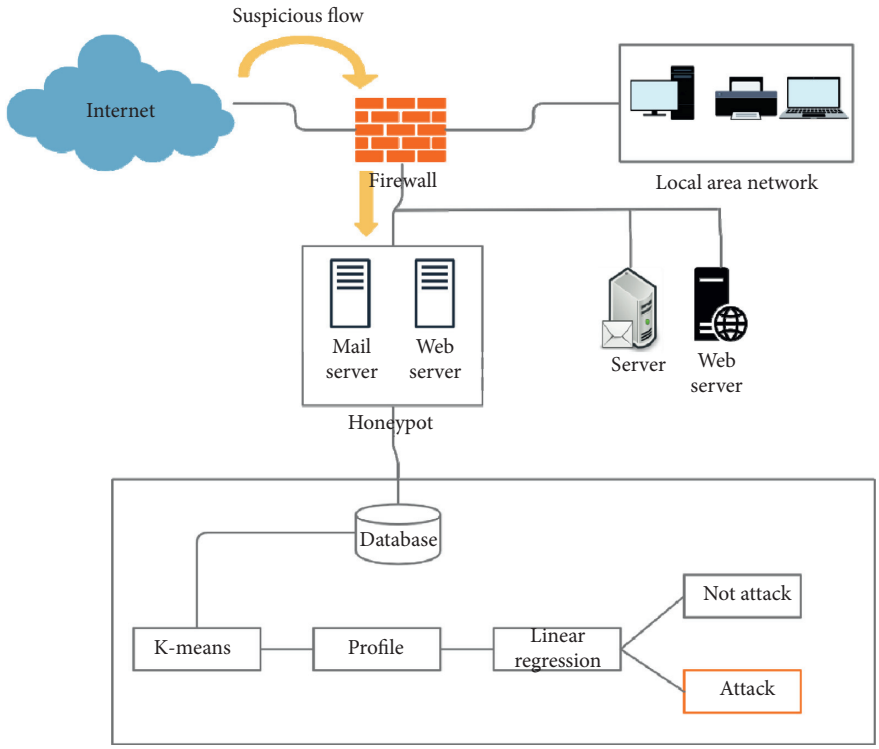


Figure 6: Suspicious flow path.

it is shown that a company deploys tree servers, a web server for its site web hosting, an FTP server for files transfer, and a mail server. In order to monitor suspicious profiles through the services provided by this company, in the same honeypot server, three virtual machines can be implemented, and each one is configured to emulate one of the previous services with the secure shell (SSH) module to allow remote access to any of the virtual machines.

This implementation allows detecting suspicious profile patterns on services, for predicting attacker profiles based on machine learning analysis. Decisions will allow reconfiguring the security policy (e.g., Firewall) in order to block the attackers. Hence, the firewall should be configured in a way to redirect suspicious flow to the honeypots in order to gather information about the application and the transport layers. The collected data will be submitted to a combination of algorithms. A clustering technique will be used to cluster the data into homogenous classes and create the user profile. The profile will, then, be classified into an attacker or nonattacker profile based on another classification algorithm.

In each interaction with a suspicious flow, the honeypot stores the collected information vector V_i^{user}(IP, logging, and packet length) in a database, in order to construct a user profile. Based on K-means algorithm, the data will be clustered into homogenous classes for creating a profile. Linear regression will be used for modeling each class, to give more significant and homogenous presentation to the data (Figures 7 and 8).

Let there be a vector V_i^{user} with n elements of information collected by the honeypot system; the V_i^{user} data will be adapted into qualitative N_{cd} and quantitative data N_{cd}. The qualitative data such as the IP address will be stocked directly in the user profile, while the quantitative quantities are clustered into homogenous groups using K-means $C_j[K]$. Each produced class will be represented by a linear model $CL_j[f]$ using linear regression. The qualitative data and the proposed quantitative modeling form the suspicious profile, the decision (decision stage) is made by projecting the suspicious profile on attacker profiles using distance metric between the trained models (learning stage) and the suspicious models.

In the learning phase (Algorithm 1), three parameters are required: the number of clusters, the initialization of centroids, and parameters of the linear function. The higher the precision of initialization and calculation of these parameters, the higher the precision of the fitting model.

The algorithm returns the following information (Algorithm 2): profile creation and classification based on the attackers model. In the Euclidean sense, $HCL_j[f]$ is the closest vector to $HCL_j[f]$, the decision is made based on projections of the new profile on the hackers profiles.

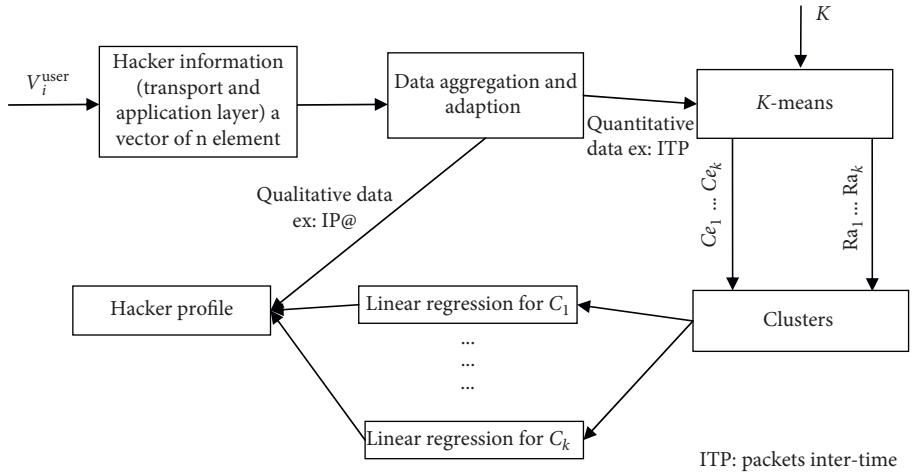


Figure 7: Learning stage.

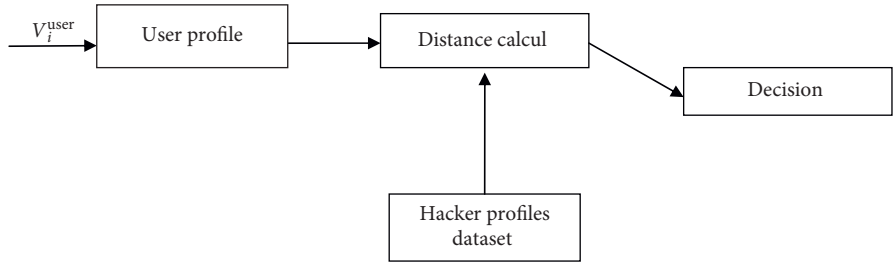


Figure 8: Decision stage.

INPUT
K//number of clusters
Vpn = ($\mathbf{V}_1^{\mathbf{user}}$, $\mathbf{V}_2^{\mathbf{user}}$,...,$\mathbf{V}_n^{\mathbf{user}}$)//Hacker information array (vector of vectors)
START
$\mathbf{M}_{\mathbf{ab}}$ = **Q1A**(**Vpn**)//qualitative adaptation
$\mathbf{N}_{\mathrm{cd}}$ = **Q2A**(**Vpn**)//quantitative adaptation
f = **c**−1//linear regression order = space dimension−1
for $j = 1$; $i < c$; j++//For each row of N_{cd}
(Cj [K], Rj [K]) = K-means ($\mathbf{N}_{id}$, K)//creation of clusters center et radius
for $j = 1$; $i < c$; j++
CLj[f] = Linear Regression (Cj [K], Rj [K], $\mathbf{N}_{id}$)//Linear Regression of every clusters
OUTPUT//Hacker profile
$\mathbf{M}_{\mathbf{ab}}$//qualitative adaptation
CLc[f]//linear regression coefficients

Algorithm 1: Learning.

6. Conclusions

In this paper, we have presented an introduction of machine learning and honeypot as solutions for cyber security. We also presented an efficient algorithm which returns two important information, one for profile creation and the other for classifying this profile. In fact, the specific solution based on honeypot and the combination of machine learning algorithms forms a solid modeling and predictive system for suspicious profile recognition and classification. Hence, it represents an integrated efficient system for cyber security to deal with future and 0-day attacks. Our next work

INPUT
K//number of clusters
$\mathbf{Vpn} = (\mathbf{V_1}^{\mathbf{user}}, \mathbf{V_2}^{\mathbf{user}}....,\mathbf{V_n}^{\mathbf{user}})$//user information array (vector of vectors)
$(\mathbf{HM_{ab}}, \mathbf{HCLc[f]})$ Hackers profiles
START
$\mathbf{UM_{ab}} = \mathbf{Q1A}(\mathbf{Vpn})$//qualitative adaptation
$\mathbf{N}_{cd} = \mathbf{Q2A}(\mathbf{Vpn})$//quantitative adaptation
$\mathbf{f} = \mathbf{c}-1$//linear regression order = space dimension−1
for $j = 1$; $i < c$; j++//For each row of N_{cd}
$(Cj\ [K], Rj\ [K]) = K$-means $(\mathbf{N}_{id}, K)$//creation of clusters center et radius
for $j = 1$; $i < c$; j++
$\mathbf{UCLj[f]}$ = Linear Regression $(Cj\ [K], Rj\ [K], \mathbf{N}_{id})$//Linear Regression of every clusters
OUTPUT
Distance($HCLc[f]$, $UCLj[f]$)
Isequal(HM_{ab}, UM_{ab})

Algorithm 2: Decision.

will be devoted to implement the smart agent in a real environment in order to evaluate and test its performances.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] GData, *Malware Numbers*, http://www.gdatasoftware.com, 2017.

[2] P. Owezarski, "Unsupervised classification and characterization of honeypot attacks," in *Proceedings of 10th International Conference on Network and Service Management (CNSM) and Workshop*, pp. 10–18, Rio de Janeiro, Brazil, November 2014.

[3] S. Dowling, M. Schukat, and E. Barrett, "Improving adaptive honeypot functionality with efficient reinforcement learning parameters for automated malware," *Journal of Cyber Security Technology*, vol. 2, no. 2, pp. 75–91, 2018.

[4] I. M. M. Matin and B. Rahardjo, "Malware detection using honeypot and machine learning," in *Proceedings of 2019 7th International Conference on Cyber and IT Service Management (CITSM)*, Bandung Institute of Technology, Bandung, Indonesia, pp. 1–4, November 2019.

[5] L. Spitzner, *Honeypots: Tracking Hackers*, Addison-Wesley, Clemson, SC, USA, 2003.

[6] T. Luo, Z. Xu, X. Jin, Y. Jia, and X. Ouyang, "Iotcandyjar: towards an intelligent-interaction honeypot for iot devices," in *Proceedings of the Black Hat*, Las Vegas, NV, USA, 2017.

[7] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots+ machine learning," in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR'10*, pp. 435–442, The ACM Digital Library, New York; NY, USA, July 2010.

[8] G. Feng, C. Zhang, and Q. Zhang, *A Design of Linkage Security Defense System Based on Honeypot: Trustworthy Computing and Services*, Springer, Berlin, Heidelberg, Germany, 2014.

[9] L.-j. Li and H. Peng, "A defense model study based on IDS and firewall linkage," in *Proceedings of 2010 International Conference of Information Science and Management Engineering*, pp. 91–94, IEEE, Xi'an, China, August 2010.

[10] Y. LeCun, "L'apprentissage profond, une révolution en intelligence artificielle," *La lettre du Collège de France*, vol. 41, p. 13, 2016.

[11] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, p. 67, 2016.

[12] Z. Ullah, F. Al-Turjman, L. Mostarda, and R. Gagliardi, "Applications of artificial intelligence and machine learning in smart cities," *Computer Communications*, vol. 154, pp. 313–323, 2020.

[13] J. H. Lee, J. Shin, and M. J. Realff, "Machine learning: overview of the recent progresses and implications for the process systems engineering field," *Computers & Chemical Engineering*, vol. 114, pp. 111–121, 2018.

[14] G. A. Seber and A. J. Lee, *Linear Regression Analysis*, John Wiley & Sons, Hoboken, NJ, USA, 2012.

[15] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.

[16] M. Schleich, D. Olteanu, and R. Ciucanu, "Learning linear regression models over factorized joins," in *Proceedings of SIGMOD '16: Proceedings of the 2016 International Conference on Management of Data*, ACM, San Francisco, CA, USA, pp. 3–18, July 2016.

[17] T. M. Kodinariya and P. R. Makwana, "Review on determining number of cluster in K-means clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[18] M. Eddabbah, M. Moussaoui, and Y. Laaziz, "A smart architecture design for health remote monitoring systems and heterogeneous wireless sensor network technologies: a machine learning breathlessness prediction prototype," *International Journal of Intelligent Enterprise*, vol. 6, no. 2–4, pp. 293–310, 2019.

[19] S. Ray and R. H. Turi, "Determination of number of clusters in k-means clustering and application in colour image

segmentation," in *Proceedings of 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99)*, Narosa Publishing House, New Delhi India, pp. 137–143, December 1999.

[20] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[21] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[22] B. Hssina, A. Merbouha, H. Ezzikouri, and M. Erritali, "A comparative study of decision tree ID3 and C4. 5," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 2, pp. 13–19, 2014.

[23] H. J. Wang, C. Guo, D. R. Simon, and A. Zugenmaier, "Shield: vulnerability-driven network filters for preventing known vulnerability exploits," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols*, pp. 193–204, The ACM Digital Library, New York; NY, USA, August 2004.

[24] K. Sadasivam, B. Samudrala, and T. A. Yang, "Design of network security projects using honeypots," *Journal of Computing Sciences in Colleges*, vol. 20, no. 4, pp. 282–293, 2005.

[25] P. S. Negi, A. Garg, and R. Lal, "Intrusion detection and prevention using honeypot network for cloud security," in *Proceedings of 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 129–132, IEEE, Noida, India, Janauary 2020.

[26] D. Fraunholz, F. Pohl, and H. D. Schotten, "Towards basic design principles for high-and medium-interaction honeypots," in *Proceedings of 16th European Conference on Cyber Warfare and Security*, p. 120, Dublin, Ireland, June 2017.

[27] H. Wang and B. Wu, "SDN-based hybrid honeypot for attack capture," in *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1602–1606, Chengdu, China, March 2019.

[28] F. Pouget and M. Dacier, "White paper: honeypot, honeynet: a comparative survey," Rep. RR-03-082, Eurecom, Biot, France, 2003.

[29] N. Naik and P. Jenkins, "A fuzzy approach for detecting and defending against spoofing attacks on low interaction honeypots," in *Proceedings of 2018 21st International Conference on Information Fusion (FUSION)*, pp. 904–910, IEEE, Cambridge, CA, UK, July 2018.

[30] B. Nagpal, N. Singh, N. Chauhan, and P. Sharma, "Catch: comparison and analysis of tools covering honeypots," in *Proceedings of 2015 International Conference on Advances in Computer Engineering and Applicationspp*, pp. 783–786, IEEE, Ghaziabad, India, March 2015.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8824659, 13 pages
https://doi.org/10.1155/2020/8824659

Research Article

Preprocessing Method for Encrypted Traffic Based on Semisupervised Clustering

Rongfeng Zheng,[1] Jiayong Liu,[2] Weina Niu,[3] Liang Liu,[2] Kai Li,[2] and Shan Liao[2]

[1]*College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China*
[2]*College of Cybersecurity, Sichuan University, Chengdu 610065, China*
[3]*School of Computer Science and Engineering, Institute for Cyber Security, University of Electronic Science and Technology of China (UESTC), Chengdu 611731, China*

Correspondence should be addressed to Jiayong Liu; ljy@scu.edu.cn

Received 25 March 2020; Revised 25 May 2020; Accepted 8 July 2020; Published 27 July 2020

Academic Editor: Sajjad Shaukat

The explosive growth in network traffic in recent times has resulted in increased processing pressure on network intrusion detection systems. In addition, there is a lack of reliable methods for preprocessing network traffic generated by benign applications that do not steal users' data from their devices. To alleviate these problems, this study analyzed the differences between benign and malicious traffic produced by benign applications and malware, respectively. To fully express these differences, this study proposed a new set of statistical features for training a clustering model. Furthermore, to mine the communication channels generated by benign applications in batches, a semisupervised clustering method was adopted. Using a small number of labeled samples, our method aggregated historical network traffic into two types of clusters. The cluster that did not contain labeled malicious samples was regarded as a benign traffic cluster. The experimental results were compared using four types of clustering algorithms. The density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm was selected to mine benign communication channels. We also compared our method with two other methods, and the results demonstrated that the benign channels mined through our method were more reliable. Finally, using our method, 1,811 benign transport layer security (TLS) channels were mined from 18,357 TLS communication channels. The number of flows carried by these benign channels comprised 65.37% of the entire network flows, and no malicious flow was included in our results, which proves the effectiveness of our method.

1. Introduction

Most of the communications making up internet traffic are generated by benign applications. If these communications are directly imported into the network intrusion detection system (NIDS) without any preprocessing, they invariably impose huge computational pressure on the NIDS. Therefore, the exclusion of benign network traffic in advance is a widely adopted strategy in the industry.

Before encryption technology was popularized on the internet, antimalware manufacturers could recognize network traffic using the deep packet inspection (DPI) method. However, with the popularization of the transport layer security (TLS) protocol [1], the malware also gradually adopted this protocol to complete command and control (C&C) communication. This led to a gradual failure in identifying network traffic using the DPI method. In response to this situation, some security vendors use the server name field or the domain name field in the certificate to preprocess the TLS traffic generated by benign applications. However, server names and certificates are easily forged by malware, making this preprocessing strategy unreliable.

Currently, the most efficient preprocessing method uses IP whitelisting technology. However, there is a lack of reliable IP whitelist sources on the internet. Threat intelligence communities such as AlienVault [2], IBM X-Force Exchange [3], and Recorded Future [4] usually provide available IP blacklist resources that are used by malware, but seldom

present whitelist resources. Furthermore, the IP whitelist usually needs to be updated occasionally to ensure its validity. Therefore, it is necessary to provide a fast and reliable method for collecting IP whitelist.

The reverse domain name lookup method can be used to obtain benign IP addresses based on the collection of benign domain names. However, under the primary domain name, there are usually many subdomains that bind to different IP addresses. As a result, it is difficult to enumerate all of the subdomains. Nevertheless, because the domain name whitelists are usually on the order of millions, using the reverse domain lookup method to collect the IP whitelist is significantly inefficient. In addition, the reliability of the IP whitelist is based on the reliability of the domain name whitelist, which makes it difficult to guarantee that all IP addresses in the whitelist are benign.

In recent years, classification and clustering techniques based on machine learning have been widely used in the identification of encrypted communication traffic. Additionally, there have been numerous studies on the application of coarse classification models [5–7] and clustering models [8–11] for preprocessing network traffic. Classification models usually require a large number of labeled samples for training, which results in the improved ability of the classification model to identify the trained samples. In other words, in the classification results of the coarse classification model for identifying malicious traffic, the nonmalicious class cannot be regarded as the benign class, because it may contain untrained malicious network traffic. For the clustering model, it is difficult to ensure the purity of the samples in the clusters based only on the single flow-based features [12]. In particular, if the benign cluster contains malicious samples, it will produce many false negatives in the NIDS.

Our study demonstrates that there are many differences between benign and malicious samples in TLS communication channels, such as the amount of inbound and outbound traffic, the connected devices, and the communication frequency. Based on these differences, the features of TLS communication channels can be extracted and a clustering model for benign applications can be established. Our network traffic preprocessing method can be realized by excluding the benign traffic contained in a cluster. Therefore, the contributions of this study are as follows:

(1) This study proposes a new network traffic preprocessing method based on a semisupervised model. To distinguish it from the traditional single flow-based features, this study presents a new set of statistical features for building a clustering model based on the TLS communication channels.

(2) This study proposes a new feature selection method. In the proposed method, the spectral clustering feature selection algorithm is used to select the top-200 features based on unlabeled samples. Further, by redesigning the evaluation algorithm, the wrapper method based on a semiunsupervised model is used to further select the best performing feature subset.

(3) Experimental results show that the preprocessing method proposed in this study can identify 1,811 benign TLS communication channels from 18,357 TLS communication channels. These channels carry 65.37% of the entire TLS flows. Furthermore, through contrast experiments, the proposed preprocessing method was verified to perform better than two other machine learning-based methods.

2. Related Work

The geometric growth in network traffic in recent times has resulted in increased processing pressure in the detection of malicious communication. Although the traditional preprocessing method based on DPI technology can accurately identify unencrypted communication traffic, it cannot cope with the currently increasing encrypted communication traffic. Machine learning technology has been used extensively in the identification of encrypted traffic and is mainly divided into classification and clustering algorithms. By applying a supervised learning algorithm, the coarse classification model [5–7] is used to preprocess network traffic, whereas the fine classification model [13–15] is used to identify the type of network traffic accurately. The clustering model [8–11] based on unsupervised learning algorithms is mainly used to identify unknown network applications and can also be used as a preprocessing method.

2.1. Supervised Learning Model. The coarse classification model is usually used for preprocessing before identifying the type of network traffic. Zhao [5] proposed a three-layer classifier to detect known and unknown network traffic. The first layer consists of a coarse classification model (a binary classifier), which is mainly used to quickly identify the unknown network traffic and thus reduce the processing pressure of the fine classification model. A similar process can be seen whereby the coarse classification model of the first layer is mainly used to exclude the benign traffic, the second layer is used to classify the different types of malicious traffic, and the third layer is used to identify different malware families [6]. However, in this process, the preprocessing method of the first layer is not described in detail, and its impacts on the detection results are rarely evaluated. To deal with the problem of accurately identifying abnormal network traffic, a two-stage deep learning detection model [7] has been proposed. This method introduced a detailed design scheme of the first stage's binary model and used the probability score value calculated from the binary model as the second stage's input. Experiments show that this method has an accuracy rate of 99.996% for the KDD99 dataset and 89.134% for the UNSW-NB15 dataset, which are higher than those of other current methods. However, because such methods do not exclude normal samples in the first stage, they reduce the efficiency of the entire detection model. Additionally, all the methods that are based on the supervised model process network flow individually and cannot process network flows in batches, which leaves room for the further improvement of preprocessing efficiency.

2.2. Unsupervised Learning Model. Unsupervised learning algorithms are also widely adopted in the preprocessing of network traffic. Zhang [9] used an unsupervised model to preprocess the network traffic to find zero-day network traffic clusters. They then used the zero-day traffic and labeled samples to train a binary classifier that was used to identify zero-day traffic more effectively. Experiments show that their method can more accurately recognize known network applications and also identify zero-day network applications. Similar methods can also be found in the research of Zhao [10]. They used a clustering model to achieve two preprocessing goals, namely, screening out unknown network traffic and expanding more labeled samples based on a few known samples. Experiments show that their method can improve classification accuracy after the preprocessing step. The research of Sacramento [11] is based on the assumption that most network traffic is benign, and only a small part of it is malicious traffic. During their preprocessing step, the largest cluster was considered to be a benign cluster, while the smaller clusters needed to be further analyzed. Through experiments, a variety of network attack behaviors were detected. However, their impact on the detection effect was not evaluated after carrying out the preprocessing step. Liya [8] used a hierarchical clustering model to preprocess a set of samples. First, they divided the set of samples into multiple clusters, after which they selected several representative samples from each cluster. Then, they used the Bayesian algorithm to classify these flows. The classification results of the selected flows represented the classification results of the entire cluster. Thus, network traffic could be quickly processed. Although the clustering model can be used to preprocess network traffic in batches, most of the current research is based on the single flow-based feature, which cannot guarantee the purity of the clusters. The sample purity in some clusters can only reach 35% [12]. Therefore, the current preprocessing method based on the clustering model can be further improved.

2.3. Single Flow-Based and Multiple Flow-Based Feature. Presently, in the field of network traffic classification, most studies focus on single flow-based features. A single flow is composed of packets with the same five-tuple information. Gezer [16] mainly extracted the single flow-based features from multiple dimensions (including the duration of the flow, the maximum, minimum, and average packet length), the interarrival time of the flow, and the number of inbound and outbound packets. Korczyński and Duda [17] used the sequence of packet length to build a Markov model. Yang [14] proposed the packet length and interpackets arrival time's distribution features in a flow. These features can be regarded as single flow-based features and are commonly used in most network traffic classification experiments.

Multiple flow-based features usually represent those that are extracted from multiple flows produced in a sliding time window. In a study on the identification of proxy application traffic based on the characteristic of flow bursts in a short time window [18], these features were designed to include the number of flow bursts, the maximum flow burst lengths, and the sum of all flow burst lengths. To detect network intrusion behaviors, Patil [19] not only applied single flow-based features but also added some multiple flow-based features, such as the number of flows with the same source IP address, and the number of flows with the same destination IP address. The application of these multiple flow-based features in different traffic classification scenarios allows for the improved performance of traffic classification.

Numerous studies [9, 20, 21] have utilized the concept of a bag of flow (BoF). A BoF is a set of flows with the same destination IP address, destination port, and transport protocol, which represents the network traffic generated by the same server application on the same port over time. As long as the type of a certain flow can be determined, the BoF type can also be determined. These studies regard a BoF as the total of flows generated on an application's communication channel. As mentioned earlier, most of the current traffic classification studies are based on single flow-based features. To the best of our knowledge, no feature design is based purely on applications' communication channels. This study sought to find the difference between the benign application and the malware on the TLS communication channel and propose a preprocessing method to exclude the benign TLS traffic. Applying this method to the NIDS can significantly reduce the processing pressure on the detection system.

3. Benign Traffic Characteristics

Before introducing feature design, it is necessary to analyze the behavioral differences between benign applications and malware on the TLS communication channel. We considered flows with the same destination IP address or server IP address and destination port, namely, port 443, as flows on the same communication channel. There are many differences between the communication behaviors of benign applications and those of malware. Firstly, in the transmission direction, benign applications usually initiate a request to the server and obtain resources, such as text, picture, audio, and video data from the server. The transmission payload is concentrated from the server to the client, which is the inbound direction. For example, by analyzing its historical traffic records, github.com has a total of 19 TLS communication channels (19 independent destination IP addresses), which carry a total of 4,080 network flows. Further analysis shows that there are a total of 3,933 flows and that the inbound payload is greater than the outbound payload, which accounts for 96.40% of the total flows. However, malware that focuses on stealing information from the users' host usually produces more outbound traffic. By collecting large amounts of traffic samples as described in Section 4, we compared the inbound and outbound traffic differences between benign and malicious application samples on their communication channels. Figure 1 shows the distribution of the outbound and inbound payload sizes using the same number of samples.

As shown in Figure 1, the abscissa represents the inbound payload sizes, and the ordinate represents the outbound payload sizes. Red dots represent malicious samples

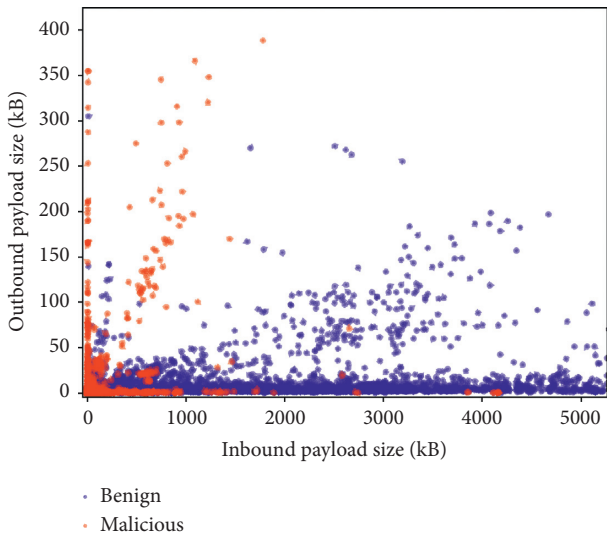


Figure 1: Distribution of inbound and outbound payload sizes.

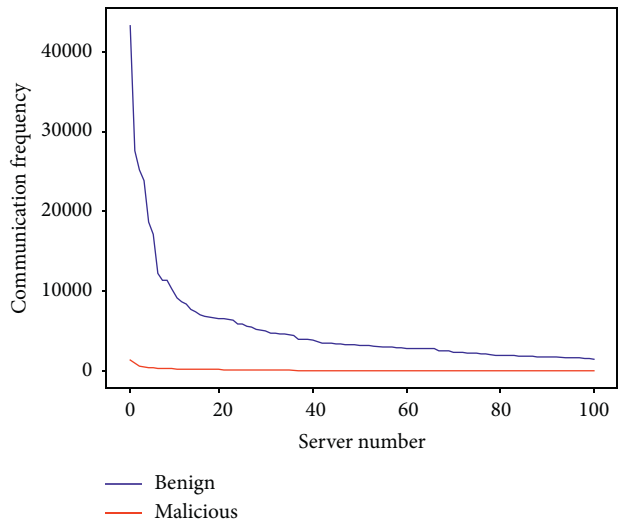


Figure 2: Top 100 servers in terms of communication frequency.

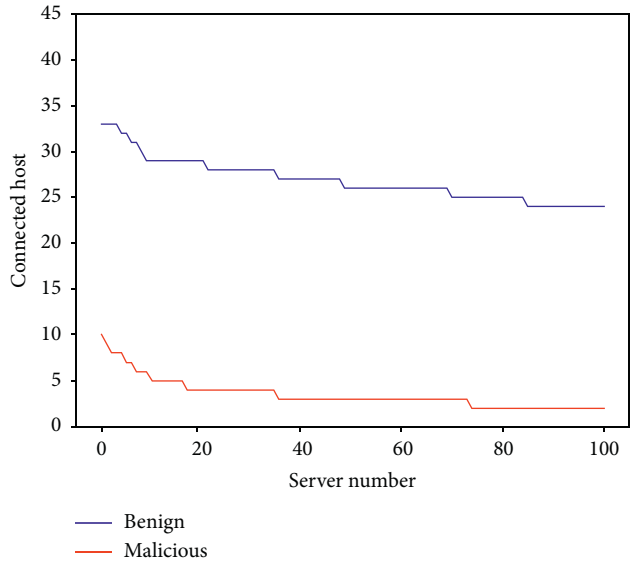


Figure 3: Top 100 servers in terms of connected devices.

and blue dots represent benign samples. It can be seen that malicious samples are more distributed near the vertical axis, whereas benign samples are mostly distributed near the horizontal axis. The proportion of the malicious flows is insignificant when the inbound payload size is higher than 1,000 kB.

Additionally, as a result of the users' online habits, some benign applications are frequently used to obtain resources from the server. This means that the communication frequency to benign application servers is higher than that of malware C&C servers. Because concealment is put first by malware, the frequent connection to the C&C server should be avoided. Figure 2 shows the top 100 servers in terms of communication frequency for both benign and malicious samples.

In addition to some niche applications, other applications have a specific user base, which results in more hosts accessing certain servers. On the contrary, malware generally chooses high-value targets for infection, so that there are relatively fewer devices that connect to the C&C servers in the local area network. Figure 3 shows the top 100 servers in terms of connected devices.

Based on the above analysis of the differences between benign and malicious traffic and their characteristics, we can categorize the network traffic into two types of clusters. The first type is characterized by a larger inbound payload, a higher communication frequency, and more connected devices. This type of cluster can be regarded as the network traffic generated by benign applications. However, theoretically, the other type of cluster contains not only malicious traffic but also traffic generated by some benign niche software.

4. Feature Representation

Based on the analysis of the characteristics of benign traffic mentioned in the previous section, this section mainly introduces the statistical features from five aspects, namely, inbound payload size, outbound payload size, inbound and outbound payload ratio, communication frequency, and connected devices. However, in terms of statistical features, the statistical feature values among benign applications vary significantly. For example, the inbound payload size of some benign applications, such as youtube.com, can reach the gigabyte (GB) scale, while the inbound payload size of other benign applications, such as baidu.com are only at the kilobyte (kB) scale. At the same time, other statistical features show the same problem. Therefore, quantization is needed before extracting features.

The quantitative scheme adopted in this study is divided into three steps. The first step is to set granularity. Based on the historical network traffic records, different statistical intervals can be divided according to different granularities, and these statistical intervals can represent the candidate features.

The second step is to calculate and normalize the feature values on each interval. For a server, there may be many instances of communication behavior, and the statistical value, such as payload size, is different each time. Therefore, we need to map these values into different statistical intervals and use the number of mapping times to calculate the feature value. Figure 4 shows a mapping process for calculating the feature value.

The third step is to compare the differences between benign and malicious samples in each interval and select the training features. To select more appropriate features, we used 5 kB, 10 kB, 50 kB, and 500 kB as the granularities for dividing intervals and compared the proportion of the inbound payload size between the benign and malicious samples in each interval using the same sample size.

As shown in Figure 5, benign and malicious samples show great differences in these intervals. It can be seen that the inbound payload size of malware is mainly concentrated in the 0–5 kB interval, accounting for more than 80% of the total traffic. When the interval is greater than 5 kB, the proportion of the malware is always smaller than that of benign applications. When the interval is greater than 500 kB, the proportion of malicious flow accounts for only 1.30%, which is almost negligible. Other features like outbound payload size, inbound/outbound payload ratio (in- and out-payload ratio), communication frequency, and connected devices show the same trend as shown in Figure 6.

Therefore, for these statistical features, it is feasible to divide the statistics according to different granularities. We designed our feature set, as shown in Table 1, which contains a total of 500 features.

The feature set in Table 1 mainly describes the network behavior in a TLS communication channel. This feature set is completely different from the traditional single flow-based feature set described in other studies [14, 16, 17]. The feature design used in this study can express the accessing behavior of a certain server from a higher level and bring together TLS communication channels with similar network behaviors.

4.1. Spectral Feature Selection Algorithm. This study adopted the spectral feature selection algorithm (SPEC) [22] to select relevant features for an unlabeled sample set. SPEC is a feature selection algorithm based on spectral graph theory [23]. The theory of SPEC is not complex and its performance is superior to other algorithms such as Laplacian Score. It can be used for both supervised and unsupervised feature selection. SPEC calculates the relevance of a feature by evaluating the feature consistency of the spectral matrix derived from the similarity matrix S. The similarity between the two samples x_i and x_j is evaluated using a radial basis function (RBF):

$$S_{ij} = e^{-\left(\left\|x_i - x_j\right\|^2 / 2\sigma^2\right)}. \tag{1}$$

By calculating S_{ij}, the similarity matrix $\mathbb{S}$ can be constructed to represent the relationships among samples. Given $\mathbb{S}$, the undirected graph G and adjacency matrix W can be constructed, where $W(i, j) = w_{ij}$ and the weight w_{ij} is determined by $\mathbb{S}$. The defining vector $\mathbf{d} = \{d_1, d_2, d_3, \ldots, d_n\}$, where $d_i = \sum_{k=1}^{n} w_{ik}$. The degree matrix D is defined as follows: $D(i, j) = d_i$, if $i = j$, and zero otherwise. Given the adjacency with W and the degree matrix D, the Laplacian matrix L and the normalized Laplacian matrix $\mathscr{L}$ are defined as follows:

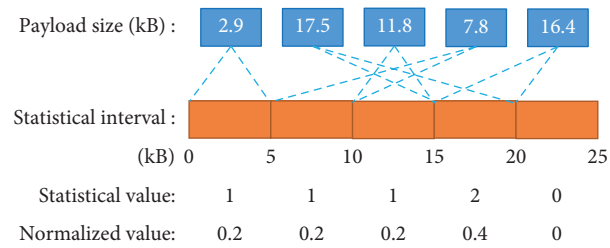


Figure 4: Mapping payload sizes to statistical intervals.

$$\begin{aligned} L &= D - W, \\ \mathscr{L} &= D^{-(1/2)} L D^{-(1/2)}. \end{aligned} \tag{2}$$

The weight of each feature vector f_i can be obtained using three ranking functions, namely, $\varphi_1, \varphi_2, \varphi_3$. Considering that the φ_2 function performs better on the test set used in Zheng and Huan's research study [22], this study selects the φ_2 function as the ranking function:

$$\varphi_2(F_i) = \frac{f_i'^T \mathscr{L} f_i'}{1 - f_i'^T \xi_0}, \tag{3}$$

in equation (3), where $f_i' = (D^{1/2} f_i)$, F_i represents the i-th feature. Given the normalized Laplacian matrix L, its spectral decomposition (λ_i, ξ_i) can be calculated, where λ_i is the eigenvalue and ξ_i is the eigenvector. According to spectral graph theory [23], we have the following: $\lambda_0 = 0$ and $\xi_0 = D^{1/2} e$. Accordingly, using the ranking function φ_2, the weight of a feature can be readily calculated. The entire calculation process is shown in Algorithm 1 which can be used to obtain the top k relevant features. There are three steps in the feature selection process: (1) building similarity set $\mathbb{S}$ and constructing its graph representation according to equations (1) and (3) (lines 1–3); (2) calculating $\varphi_2(F_i)$ according to equation (3) (lines 4–6); (3) ranking features in ascending order for $\varphi_2(F_i)$ (lines 7-8). In fact, a smaller $\varphi_2(F_i)$ represents the improved separability among samples. Hence, the smaller the value of $\varphi_2(F_i)$, the more important the feature f_i is.

To evaluate the selected feature set, we prepared 6,978 samples and used Algorithm 1 to rank the importance of the features. Table 2 shows an example of the top 20 features. It can be seen that the features of the downstream payload size have the highest proportion, meaning that these types of features are the most important.

We selected the top 200-feature subsets as our candidate feature subsets by using Algorithm 1 to quickly exclude ineffective features due to the sparsity of the designed feature set. To further select the relevant features, we used the wrapper method to evaluate whether the feature subset could meet the requirements of clustering. This part can be seen in the next section.

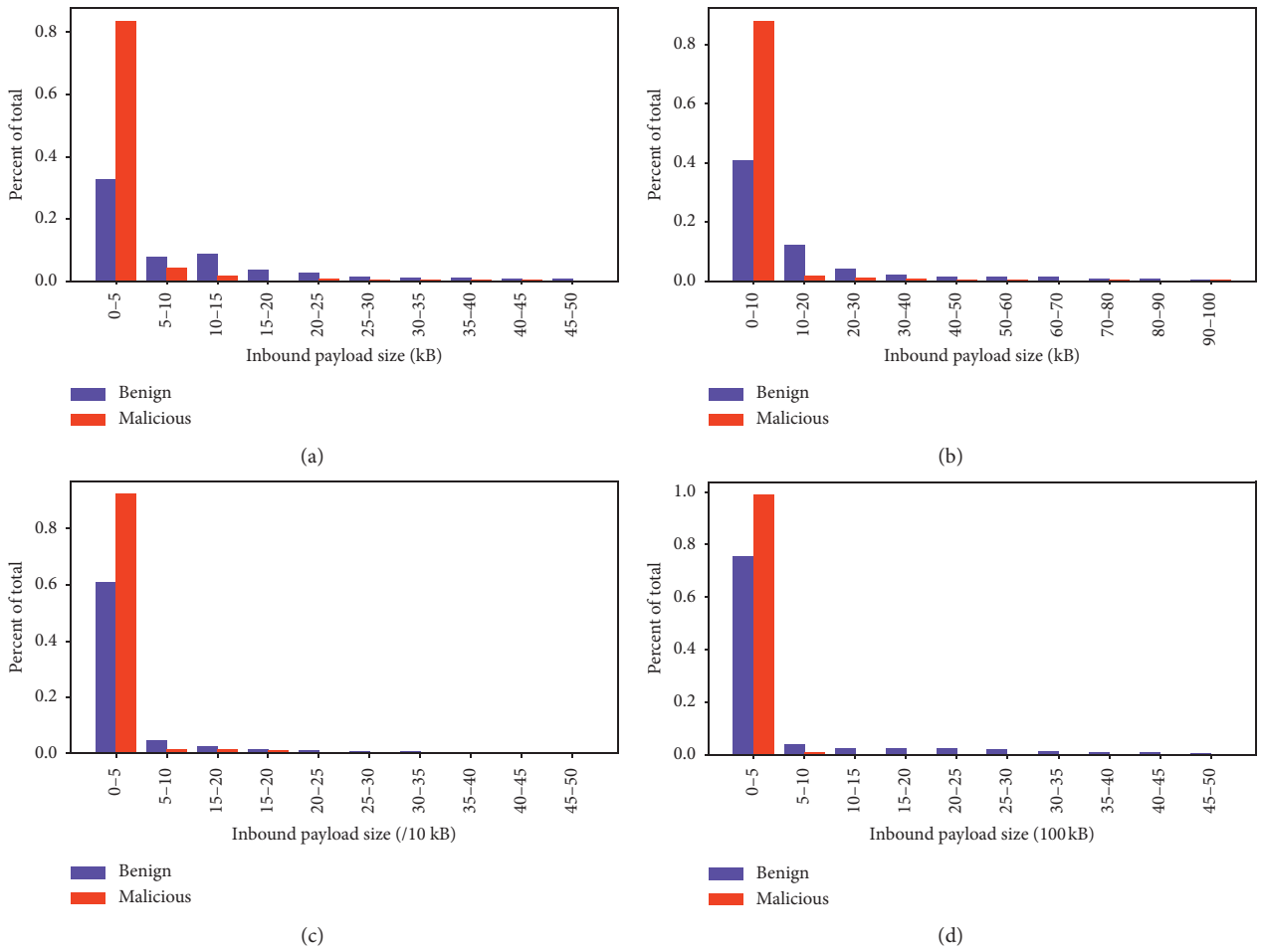


Figure 5: Proportions in different intervals.

5. Model Training

The goal of preprocessing is to mine the TLS communication channels used by benign applications as much as possible. However, the feature subset selected by the SPEC algorithm did not necessarily meet the requirements of the clustering model; i.e., the cluster of benign TLS channels should contain as few malicious TLS channels as possible. Figure 7 shows an example of the description of different clustering effects that originated from different feature subsets. Labeled samples are represented by + and – where + indicates benign TLS samples and – indicates malicious TLS samples. Unlabeled samples are represented by △. F_i and F_i' are used to denote different feature subsets. In the clustering process, we labeled the attribute of a cluster by calculating its proportion of positive and negative samples. If the proportion of positive samples in the cluster is higher, the cluster is considered to be benign; otherwise, it is marked as malicious.

We did not require distinguishing the two types of samples. We only needed to ensure that the benign cluster contained as few malicious samples as possible. In Figure 7, although the clustering results based on F_i' were better, we preferred the clustering result based on F_i to ensure the purity of the benign cluster. Hence, we needed to redesign the evaluation algorithm used in the process of training the model. In this process, two goals can be achieved simultaneously: (1) selecting the best performance feature subset; (2) selecting a more appropriate clustering algorithm.

Semisupervised clustering was adopted to evaluate the clustering effect, and the wrapper method was used to select a subset of the best performing features based on the ranked top 200-feature sets obtained in the previous section. Two rules were used to evaluate the performance of each subset of features: (1) whether the subset of features reduced the proportion of malicious channels in the benign cluster; (2) whether the subset of features could improve the recognition rate of benign channels when the proportion of malicious channels in the benign cluster did not change. We evaluated the performance of the feature subsets by calculating two indicators of the labeled samples: the false positive rate (FPR) and the true positive rate (TPR). The evaluation algorithm is outlined in Algorithm 2.

Algorithm 2 is divided into three main steps: (1) selecting a clustering algorithm and calculating the confusion matrix of the X_{labeled} (labeled samples) to obtain the initial FPR and TPR (lines 1-2); (2) constructing feature subsets using backward selection and judging whether a feature should be excluded by comparing FPR and TPR with the last result (lines 3–13); (3) obtaining the final feature subset and clustering result of the labeled and unlabeled samples (lines

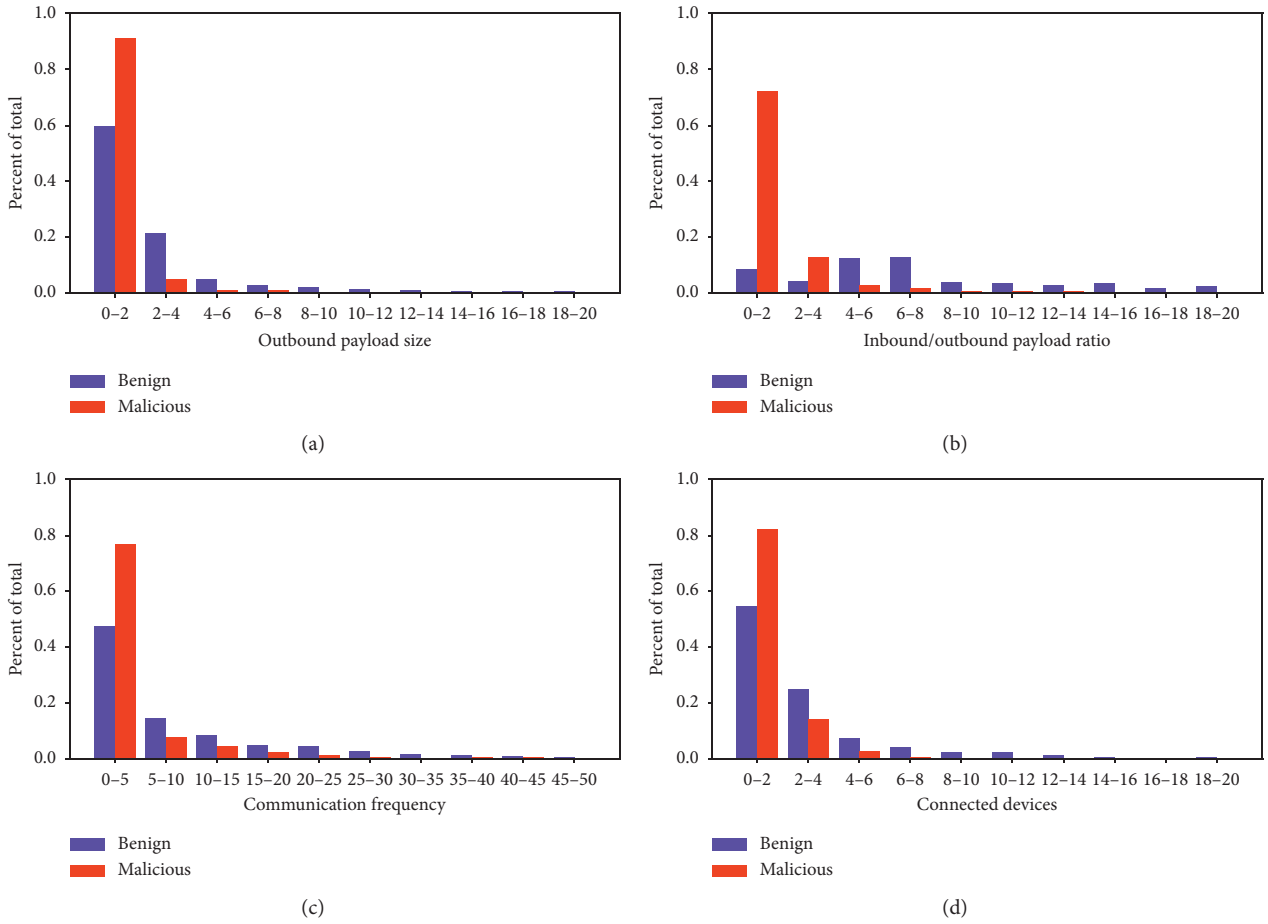


Figure 6: Comparison of other feature expressions.

Table 1: Designed feature set.

Feature name	Description	Number
Inbound payload size	Interval granularities with 2, 3, 5, 10, 20, 50, 100, 200, 500, 5000 kB	100
Outbound payload size	Interval granularities with 2, 3, 5, 10, 20, 50,100, 200, 500, 5000 kB	100
In-and-out payload ratio	Interval granularities with 2, 3, 5, 10, 20, 50, 100, 200, 500, 5000	100
Communication frequency	Interval granularities with 2, 3, 5, 10, 20, 50, 100, 200, 500, 5000	100
Connected devices	Interval granularities with 2, 3, 5, 10, ,20, 50, 100, 200, 500, 5000	100

Input: X, $\gamma(\cdot)$, k, F_i
Output: $\mathrm{SF}_{\mathrm{SPEC}}$–the ranked feature list
(1) construct $\mathbb{S}$, the similarity set from X (and Y);
(2) construct graph G from $\mathbb{S}$;
(3) build W, D and L from G ;
(4) **for** each feature vector f_i **do**
(5) $f_i' \leftarrow (D^{1/2} f_i / \|D^{1/2} f_i\|)$; $\mathrm{SF}_{\mathrm{SPEC}}(i) \leftarrow \varphi_2(F_i)$
(6) **end**
(7) ranking $\mathrm{SF}_{\mathrm{SPEC}}$ in ascending order for $\varphi_2(F_i)$
(8) return $\mathrm{SF}_{\mathrm{SPEC}}$.

Algorithm 1: SPEC.

14-15). Additionally, by comparing the clustering effects of different clustering algorithms, the best performing clustering algorithm can also be obtained.

6. Experiment and Evaluation

The experiment was composed of three steps: (1) data collection and pretreatment; (2) algorithm selection, whereby the best performing algorithm could be selected from the four clustering algorithms; (3) method comparison. The method proposed in this study was compared with other existing methods and their effectiveness evaluated.

6.1. Data Collection. The test data of TLS traffic used in this method were collected from the gateway of our laboratory; we mirrored all network traffic including TLS flows to our experimental platform. In this study, we only used TLS traffic to mine TLS benign communication channels.

The network flow was collected using a tool developed by us [24]. The basic information of the flow included the

Table 2: Top 20 features.

Feature description	$\varphi_2(F_i)$
Proportion of payload ratio in [200, 400) to the total	0.5381
Proportion of payload ratio in [8, 10) to the total	0.5421
Proportion of in-payload size in [0, 2) kB to the total	0.5433
Proportion of payload ratio in [30, 40) to the total	0.5436
Proportion of out-payload size in [20, 40) kB to the total	0.5451
Proportion of in-payload size in [40000, 45000) kB to the total	0.5473
Proportion of in-payload size in [25000, 30000) kB to the total	0.5531
Proportion of payload ratio in [150, 200) to the total	0.5553
Proportion of in-payload size in [450, 500) kB to the total	0.5558
Proportion of in-payload size in [1500, 2000) kB to the total	0.5572
Proportion of payload ratio in [500, 1000) to the total	0.5575
If the communication frequency in [20, 40)	0.5642
Proportion of payload ratio in [6, 8) to the total	0.5656
Proportion of in-payload size in [45000, 50000) kB to the total	0.5663
Proportion of out-payload size in [16, 18) kB to the total	0.5664
If the communication frequency in [200, 400)	0.5670
Proportion of out-payload size in [90, 100) kB to the total	0.5704
If the communication frequency in [70, 80)	0.5726
Proportion of payload ratio in [18, 20) to the total	0.5735
Proportion of in-payload size in [50, 60) kB to the total	0.5738

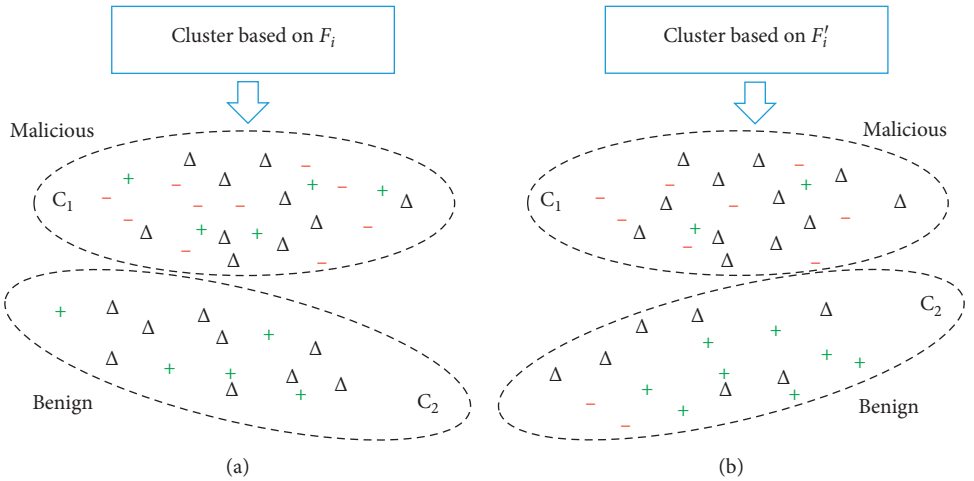


Figure 7: Comparison of clustering results using different feature subsets.

Input: $F_{\text{top-200}}$, X_{labeled}, $X_{\text{unlabeled}}$
Output: Best feature subset (BFS), Clustering result (CR)
(1) select a cluster algorithm
(2) calculate initial confusion matrix for X_{labeled}
obtain FPR_{ini}, TPR_{ini}
(3) **for** backward selection $F_{\text{top-200}}$ and exclude f_i **do**
(4) calculate confusion matrix for X_{labeled}
(5) **if** $\text{FPR} < \text{FPR}_{\text{last}}$:
(6) exclude (f_i)
(7) $\text{FPR}_{\text{last}} = \text{FPR}$, $\text{TPR}_{\text{last}} = TPR$
(8) **if** $\text{FPR} = \text{FPR}_{\text{last}}$ and $\text{TPR} \geq TPR_{\text{last}}$:
(9) exclude (f_i)
(10) $\text{FPR}_{\text{last}} = \text{FPR}$, $\text{TPR}_{\text{last}} = \text{TPR}$
(11) **if** $\text{FPR} > \text{FPR}_{\text{last}}$:
(12) retain (f_i)
(13) **end**
(14) obtain feature subset as BFS, CR for $X_{\text{unlabeled}}$ and X_{labeled}
(15) return BFS, CR.

Algorithm 2: SMFS.

source IP address, destination IP address, inbound and outbound payload size, number of inbound and outbound packets, and the inbound/outbound payload ratio. This information on network flow can also be acquired using tools such as NetFlow [25], which was developed by Cisco or Moloch [26].

To verify this method, we collected all the network traffic based on the TLS protocol generated from July 1, 2019, to July 15, 2019. We collected a total of 1,655,498 TLS flows containing 18,357 TLS communication channels (18,357 unique destination IP addresses). In the experiment, we mined the benign TLS channels from the abovementioned 18,357 communication channels. In addition, this study also used the malicious traffic samples provided by Stratosphere Lab [27]. We downloaded a total of over 300 GB in traffic samples and extracted 14,544 TLS flows generated by malware, which makes up a total of 970 TLS communication channels. These malware samples were mainly used to verify the reliability of benign TLS channels mined using this method.

Before the experiments, it was necessary to carry out some pretreatment work on the samples to improve the efficiency of our method and reduce noise samples. Some channels with low accessing behaviors and low payload sizes were excluded in advance. The pretreatment rules were as follows: (1) The total number of access behaviors must not exceed 20 times; (2) The inbound payload size must not exceed 40 kB. Samples that satisfied these two rules were filtered out in advance. After pretreatment, we had 6,700 test samples and 278 labeled malicious samples remaining.

6.2. Algorithm Selection. The preprocessing method proposed in this study adopted a semisupervised clustering method to mine the benign TLS channels. The requirements for clustering were clear. We only needed to train two clusters. The first cluster was the cluster that conformed to the benign communication characteristics described in Section 2. This cluster was mainly composed of benign TLS channels. The other cluster was composed of malicious TLS channels and also contained some benign TLS channels generated by niche applications. Additionally, it should be noted that the method used in this study focused on clustering part of the benign TLS channels and not all the benign TLS channels. The reason is that some benign applications also have similar communication behaviors as malware.

To evaluate the effectiveness of the clustering algorithm, we selected 297 representative benign samples and 278 malicious samples and mixed these labeled samples with unlabeled samples for training the model. We used labeled samples to determine the cluster in which the benign samples were located and to verify whether the method could effectively mine benign TLS channels.

Because different sample sets were adapted to different types of clustering algorithms, we needed to determine the clustering algorithm that was more suitable for training the samples. Four clustering algorithms were selected for comparative experiments, namely, K-means [28], based on the central point of samples; DBSCAN [29], based on the density of samples; Gaussian Mixture Models (GMM) [30], based on the distribution of samples; and Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [31], which is a hierarchical clustering algorithm. We used Algorithm 2 to test these four clustering algorithms and compared the clustering results among them. Table 3 shows the clustering results of the four clustering algorithms.

It can be seen that the UPGMA and GMM clustering algorithms cannot effectively distinguish the labeled samples, indicating that the characteristics of the sample set are not suitable for these two clustering algorithms. The K-means clustering algorithm can recognize all the labeled malicious samples. However, a majority of the labeled benign samples are also identified as malicious. Only 95 of the 297 benign samples are correctly distinguished. The DBSCAN algorithm has the best clustering effect, whereby 204 benign samples are correctly recognized, and the recognition rate of labeled malicious samples reaches 100%. This means that all the labeled malicious samples are in the same cluster, which guarantees the purity of the benign samples contained in other clusters. Additionally, the testing set contained 6,700 TLS channels, with 4,889 in the malicious cluster, and 1,811 in the benign cluster. In other words, we mined 1,811 benign TLS channels that can be excluded through preprocessing. In the DBSCAN clustering algorithm, the selected best feature subset is shown in Table 4, with a total of 25 features.

6.3. Method Comparison and Evaluation. To further verify the effectiveness of our method, we compared it with a classification method proposed by Anderson et al. [13] and a clustering method proposed by Su [8], respectively. These methods are used to recognize benign traffic and mine benign TLS channels.

In the process of reproducing the method proposed by Anderson [13], we first selected 11,500 malicious samples and 11,500 benign samples. We then completed the feature extraction according to the feature set given in the method. The logistic regression algorithm (the random forest algorithm works best in actual tests) was used to train the sample set and evaluate the model using 10-fold cross-validation. The average accuracy of the classification model was 94.44%, and the recall rate was 89%. The test results achieved the effect described in the literature [13].

In the method proposed by Su [8], they built a hierarchical clustering process. In their method, the 7-dimensional structural features were first used for coarse-grained traffic clustering, after which the 7-dimensional temporal features were used for fine-grained traffic clustering. Finally, using Naïve Bayes classifiers, a small number of samples were selected from each cluster, and the classification results of these samples represented the classification results of the entire cluster. Therefore, their methods can also be used to identify benign network traffic.

In the first comparative experiment, we also used 297 labeled benign samples and 278 malicious samples as our experimental dataset. Our method used a feature set derived from TLS communication channels, whereas Anderson et al.

Table 3: Comparison of the clustering results of the four clustering algorithms.

Clustering algorithm	Testing samples		Labeled benign samples		Labeled malicious samples		Features in BFS
	Benign	Malicious	Benign	Malicious	Benign	Malicious	
K-means	733	5967	95	202	0	278	82
DBSCAN	1811	4889	204	93	0	278	25
GMM	4115	2385	229	68	129	149	93
UPGMA	2	6698	0	297	0	278	39

Table 4: Subset of features in DBSCAN.

Feature description	Category
Proportion of in-payload size in [10, 15) kB to the total	Numerical
Proportion of in-payload size in [15, 20) kB to the total	Numerical
Proportion of in-payload size in [20, 30) kB to the total	Numerical
Proportion of in-payload size in [0, 500) kB to the total	Numerical
Proportion of in-payload size in [500, 1500) kB to the total	Numerical
Proportion of in-payload size in [1500, 2000) kB to the total	Numerical
Proportion of out-payload size in [5, 10) kB to the total	Numerical
Proportion of out-payload size in [20, 30) kB to the total	Numerical
Proportion of out-payload size in [50, 100) kB to the total	Numerical
Proportion of out-payload size in [300, 400) kB to the total	Numerical
Proportion of payload ratio in [5, 10) to the total	Numerical
Proportion of payload ratio in [10, 20)to the total	Numerical
Proportion of payload ratio in [30, 40) to the total	Numerical
Proportion of payload ratio in [150, 200) to the total	Numerical
Proportion of payload ratio in [200, 400) to the total	Numerical
Proportion of payload ratio in [500, 1000) to the total	Numerical
If the communication frequency in [20, 40)	Boolean
If the communication frequency in [40, 60)	Boolean
If the communication frequency in [70, 80)	Boolean
If the communication frequency in [100, 150)	Boolean
If the communication frequency in [200, 300)	Boolean
If the communication frequency in [400, 500)	Boolean
If the connected hosts in [3, 6)	Boolean
If the connected hosts in [9, 12)	Boolean
If the connected hosts in [15, 20)	Boolean

and Su et al. used feature sets based on a single flow in their proposed methods. Therefore, in the methods of Anderson [13] and Su [8], we stipulate that in a communication channel, as long as any of the flows is determined to be malicious, the entire communication channel is untrustworthy. By reproducing these two methods, we obtained the results of the comparative experiments as shown below.

As shown in Figure 8, the A-Method represents the method proposed by Anderson et al. [13], and the S-Method represents the method proposed by Su et al. [8]. It can be seen that the A-Method has the highest recognition rate for benign samples, reaching 97.97%. However, it cannot identify all malicious samples, which means that the A-Method is not reliable. Both the S-Method and our method can identify all malicious samples, but the S-Method has a lower recognition rate for benign samples. Our method can not only cluster all the malicious samples but also have the highest recognition rate for benign samples.

In the next experiment, we compared the ability of these three methods to mine the benign TLS channels based on the test set. At the same time, we used open-source threat intelligence and manual inspection methods to evaluate the candidate benign TLS channels by checking whether the server IP address contained in the TLS channels was used by malware.

We selected the AlienVault [2] threat intelligence community, which owns a significantly comprehensive threat intelligence library that can be used to determine whether a server IP address is malicious. In AlienVault, some of the benign IP addresses, such as 8.8.8.8 (Google Public DNS), are also considered to be malicious (This is related to AlienVault's strategy for collecting the indicator of compromise (IoC). Further discussion is not required here). Hence, manual inspection is also needed to evaluate whether the malicious results provided by AlienVault are correct. It is worth mentioning that we cannot directly mine the benign TLS channels using AlienVault because it is unable to collect all malicious IP addresses worldwide. Another reason is that some benign IP addresses are also marked as malicious. However, AlienVault can reflect the reliability of the mined benign TLS channels to some extent. The test results are shown in Table 5.

It can be seen that the number of candidate benign channels mined using the A-Method is the largest, reaching

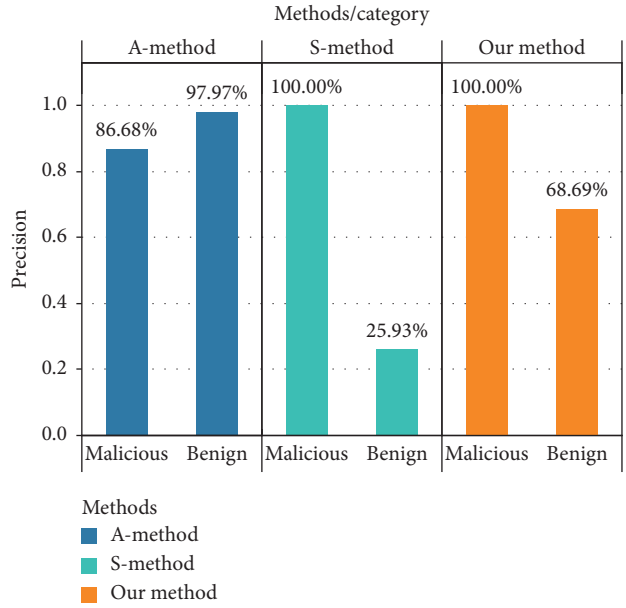


Figure 8: Comparison of different methods used on the labeled dataset.

Table 5: Evaluation of benign channels mined using different methods.

Benign channels	A-method	S-method	Our method
Candidate benign channels	11,286	1,722	1,811
Checking by AlienVault	11,166	1,699	1,779
Manual inspection	11,261	1,716	1,811

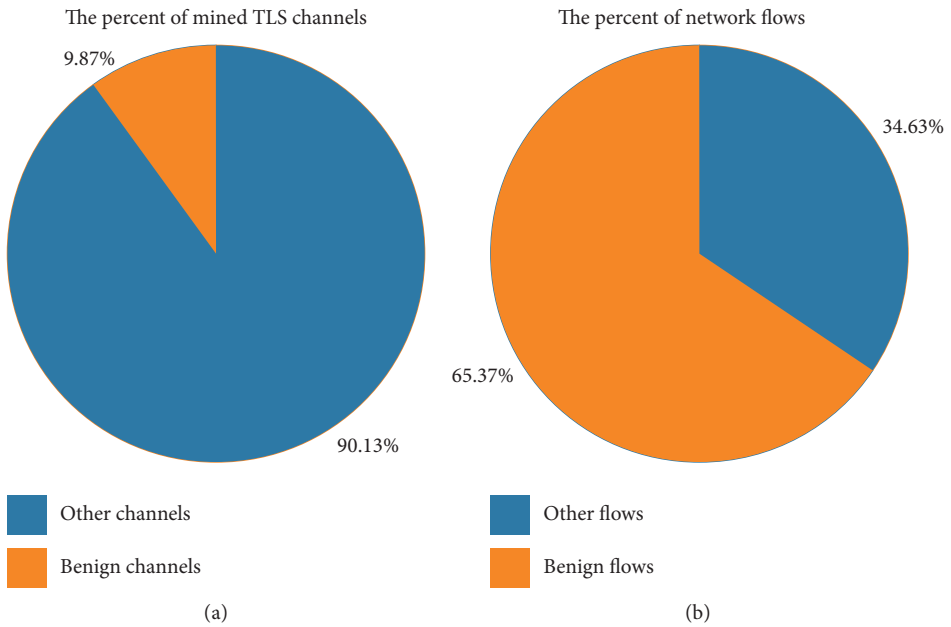


Figure 9: Statistics related to the benign TLS channels.

11,286, and that of benign channels mined using the S-Method is the least. Our method's results are slightly higher than those of the S-Method. However, through the step of checking using AlienVault, the candidate benign channels mined using the A-Method fall the fastest from 11,286 to 11,166. Manual inspection demonstrates that these IP addresses are mostly benign. However, there are still 25 malicious IP addresses in the candidate benign channels mined using the A-Method. We also found six malicious IP addresses mined using the S-Method. The candidate benign

channels mined using our method did not contain any malicious IP addresses. Therefore, the results show that our method is more reliable than the other two methods.

Using our method, we mined 1,811 benign channels from 18,357 channels. These channels account for 9.87% of the total mined IP addresses, as shown in Figure 9. The number of network flows carried by these 1,811 channels accounts for 65.37% of the total number of network flows. In other words, applying our preprocessing method in the NIDS can reduce the processing pressure by at least 65.37%. Other TLS channels are not all malicious because the characteristics of some niche benign applications are inconsistent with our assumptions. Therefore, our method is only suitable for mining benign TLS channels and cannot be used to identify malicious TLS channels.

Finally, it is worth mentioning that the benign channels can be used to form IP whitelist rules. The preprocessing method we proposed can also be used for mining whitelist IP addresses.

7. Conclusions

Because of the increasing TLS traffic, importing them into the NIDS indiscriminately will undoubtedly result in substantial processing pressure. Hence, it is a consensus to preprocess network traffic before completing detection. However, current studies seldom evaluate the impact of the results brought about by the preprocessing methods. Moreover, the classification and clustering models based on the single flow-based features are not significantly reliable in the preprocessing of TLS traffic. This study proposed a semiunsupervised model for quickly mining benign TLS channels to cope with such problems. We analyzed the differences between benign applications and malware on the communication behaviors of TLS channels in detail and proposed a set of new features. By adopting a spectral clustering algorithm and a wrapper method, a set of relevant features were selected, and a preprocessing model was established by applying a semiunsupervised algorithm. Through a set of experiments, the DBSCAN algorithm was selected from three other clustering algorithms to build a preprocessing model. Additionally, by comparing our method with two other methods, our experiments demonstrate that it not only performs better in terms of processing efficiency but is also significantly reliable for mined benign TLS channels.

The preprocessing model based on TLS channel features proved that it can mine the benign TLS channels used in this study. Indeed, TLS channel features have the potential to be further mined, and subsequently used to recognize malicious channels based on supervised models. For high camouflaged TLS flows produced by malware, it is difficult to detect them only based on a single flow, but we can observe and evaluate their behaviors, such as data-stealing behavior, from the perspective of their communication channels.

Therefore, in future studies, two main points need to be further explored for recognizing malicious TLS channels. One is to explore new TLS channel features that are solely effective in detecting malware traffic; the other is to ascertain whether the performance of classifiers can be improved by combining TLS channel-based features with traditional single flow-based features.

Data Availability

The malware traffic data used to support the findings of this study have been deposited in the Stratosphere IPS project repository (Available: https://www.stratosphereips.org/datasets-malware).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Defense Innovation Special Zone Program of Science and Technology (Grant no. JG2019055), in part by the Frontier Science and Technology Innovation Projects of National Key Research and Development Program (Grant no. 2019QY1405), and in part by the Key Research and Development Program of Sichuan Province (Grant no. 2020YFG0076).

References

[1] T. Dierks and C. Allen, *The TLS Protocol Version 1.0*, pp. 1–75, Internet Engineering Task Force (IETF), Fremont, CA, USA, 1999.

[2] Alienvault, *Alienvault, Inc.*,Alienvault, San Mateo, CA, USA, 2020, https://otx.alienvault.com.

[3] IBM X-Force Exchange, *IBM Security*, IBM X-Force Exchange, Atlanta, GA, USA, 2020, https://exchange.xforce.ibmcloud.com.

[4] Recorded Future, *Recorded Future, Inc*, Recorded Future, Somerville, MA, USA, 2020, https://support.recordedfuture.com.

[5] S. Zhao, S. Chen, Y. Sun, Z. Cai, and J. Su, "Identifying known and unknown mobile application traffic using a multilevel classifier," *Security and Communication Networks*, vol. 2019, Article ID 9595081, 11 pages, 2019.

[6] Y. C. Chen, Y. J. Li, A. Tseng, and T. Lin, "Deep learning for malicious flow detection," in *Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, Montreal, QC, Canada, pp. 1–7, October 2017.

[7] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: a two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[8] L. Su, Y. Yao, N. Li, J. Liu, Z. Lu, and B. Liu, "Hierarchical clustering based network traffic data reduction for improving suspicious flow detection," in *Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, pp. 744–753, IEEE, New York, NY, USA, August 2018.

[9] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust network traffic classification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1257–1270, 2015.

[10] S. Zhao, Y. Zhang, and P. Chang, "Network traffic classification using tri-training based on statistical flow characteristics," in *Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS*, pp. 323–330, IEEE, Sydney, NSW, Australia, August 2017.

[11] L. Sacramento, I. Medeiros, J. Bota, and M. Correia, "Flowhacker: detecting unknown network attacks in big traffic data using network flows," in *Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, pp. 567–572, IEEE, New York, NY, USA, August 2018.

[12] T. Glennan, C. Leckie, and S. M. Erfan, "Improved classification of known and unknown network traffic flows using semi-supervised machine learning," in *Proceedings of the 21st Australasian Conference on Information Security and Privacy*, Springer, Melbourne, VIC, Australia, pp. 493–501, June 2016.

[13] B. Anderson, S. Paul, and D. McGrew, "Deciphering malware's use of TLS (without decryption)," *Journal of Computer Virology and Hacking Techniques*, vol. 14, no. 3, pp. 195–211, 2018.

[14] Y. Yang, C. Kang, G. Gou, Z. Li, and G. Xiong, "TLS/SSL encrypted traffic classification with autoencoder and convolutional neural network," in *Proceedings of the 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 362–369, IEEE, Exeter, United Kingdom, June 2018.

[15] B. A. AlAhmadi and I. Martinovic, "Malclassifier: Malware family classification using network flow sequence behaviour," in *Proceedings of the 2018 APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–13, IEEE, San Diego, CA, USA, May 2018.

[16] A. Gezer, G. Warner, C. Wilson, and P. Shrestha, "A flow-based approach for trickbot banking trojan detection," *Computers & Security*, vol. 84, pp. 179–192, 2019.

[17] M. Korczyński and A. Duda, "Markov chain fingerprinting to classify encrypted traffic," in *Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 781–789, IEEE, Toronto, ON, Canada, April 2014.

[18] X. Zeng, X. Chen, G. Shao et al., "Flow context and host behavior based shadowsocks's traffic identification," *IEEE Access*, vol. 7, pp. 41017–41032, 2019.

[19] R. Patil, H. Dudeja, and C. Modi, "Designing an efficient security framework for detecting intrusions in virtual network of cloud computing," *Computers & Security*, vol. 85, pp. 402–422, 2019.

[20] M. Baldi, A. Baldini, N. Cascarano, and F. Risso, "Service-based traffic classification: principles and validation," in *2009 IEEE Sarnoff Symposium*, pp. 1–6, IEEE, Princeton, NJ, USA, April 2009.

[21] J. Ma, K. Levchenko, C. Kreibich, S. Savage, and G. M. Voelker, "Unexpected means of protocol inference," in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, pp. 313–326, Rio de Janeiro, Brazil, October 2006.

[22] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th International Conference on Machine Learning*, pp. 1151–1157, Corvallis, OR, USA, June 2007.

[23] F. Chung, "Spectral graph theory," in *Proceedings of the of CBMS Regional Conference Series in Mathematics*, AMS, Providence, RI, USA, December 1997.

[24] scu_igroup: Streamdump, 2020, https://github.com/scu-igroup/StreamDump.

[25] Netflow, 2020, https://netflow.us/.

[26] Moloch, 2020, https://molo.ch/.

[27] S. Lab: Malware Capture Facility Project, 2020, https://www.stratosphereips.org/datasets-malware.

[28] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, pp. 281–297, Berkeley, CA, USA, 1967.

[29] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Kdd-96*, vol. 96, no. 34, pp. 226–231, Portland, OR, USA, August 1996.

[30] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication*, vol. 17, no. 1-2, pp. 91–108, 1995.

[31] R. R. Sokal and C. D. Michener, "A statistical method for evaluating systematic relationships," *The University of Kansas Science Bulletin*, vol. 38, pp. 1409–1438, 1958.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8830683, 19 pages
https://doi.org/10.1155/2020/8830683

Review Article

A Systematic Literature Review on Using Machine Learning Algorithms for Software Requirements Identification on Stack Overflow

Arshad Ahmad,[1,2] **Chong Feng**,[1] **Muzammil Khan**,[3] **Asif Khan**,[1] **Ayaz Ullah**,[2] **Shah Nazir**,[2] **and Adnan Tahir**[4]

[1]*School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China*
[2]*Department of Computer Science, University of Swabi, Anbar, Pakistan*
[3]*Department of Computer Science, University of Swat, Mingora, Pakistan*
[4]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China*

Correspondence should be addressed to Chong Feng; fengchong@bit.edu.cn

Received 13 May 2020; Revised 1 June 2020; Accepted 25 June 2020; Published 15 July 2020

Academic Editor: Iqtadar Hussain

Context. The improvements made in the last couple of decades in the requirements engineering (RE) processes and methods have witnessed a rapid rise in effectively using diverse machine learning (ML) techniques to resolve several multifaceted RE issues. One such challenging issue is the effective identification and classification of the software requirements on Stack Overflow (SO) for building quality systems. The appropriateness of ML-based techniques to tackle this issue has revealed quite substantial results, much effective than those produced by the usual available natural language processing (NLP) techniques. Nonetheless, a complete, systematic, and detailed comprehension of these ML based techniques is considerably scarce. *Objective*. To identify or recognize and classify the kinds of ML algorithms used for software requirements identification primarily on SO. *Method*. This paper reports a systematic literature review (SLR) collecting empirical evidence published up to May 2020. *Results*. This SLR study found 2,484 published papers related to RE and SO. The data extraction process of the SLR showed that (1) Latent Dirichlet Allocation (LDA) topic modeling is among the widely used ML algorithm in the selected studies and (2) precision and recall are amongst the most commonly utilized evaluation methods for measuring the performance of these ML algorithms. *Conclusion*. Our SLR study revealed that while ML algorithms have phenomenal capabilities of identifying the software requirements on SO, they still are confronted with various open problems/issues that will eventually limit their practical applications and performances. Our SLR study calls for the need of close collaboration venture between the RE and ML communities/researchers to handle the open issues confronted in the development of some real world machine learning-based quality systems.

1. Introduction

The RE activity is steered in the very first phase of software development lifecycle and plays a very pivotal role in ensuring the development of quality and secure software systems [1, 2]. There are various activities (i.e., elicitation, specification, validation, and management) associated with it that need to be effectively performed to somehow guarantee developing a quality software [1–8]. There has been a rapid surge in the RE community of effectively using the diverse online user feedback offered on various social media/online platforms, for instance, the Stack Overflow Q&A site [9], Twitter, bug reporting systems, and mobile app stores (i.e., Google's Play Store and Apple's Play Store) [10] as amongst the latent and rich sources of diverse user requirements [11, 12]. SO Q&A online programming community is commonly used by diverse programmers for learning, problem solving, and sharing knowledge on various issues of software development. For instance, the posts of SO can be about pursuing assistance on programming issues/problems, stating a bug in a software development tool, and requesting or recommending new software

requirements (both functional requirements (FRs) and nonfunctional requirements (NFRs)) omitted in the software development tool, which were totally overlooked previously [13].

Normally software requirements are of two types, namely, FRs and NFRs. The research work on the difference between FRs and NFRs is defined and well known; however, the automatic identification and classification of the software requirements stated in different natural language is still a huge challenge [14–21]. The problem situation becomes more complicated and worst when identifying software requirements from the Q&A on SO (unstructured text) [13, 22, 23]. In addition, some other reasons which make this task problematic are the diversity of stakeholders, difference in the terminologies used, structures of the sentences, and the language used to specify the same kind of requirements [14, 24]. These high levels of inconsistencies and the unstructuredness of the Q&A posts text data on SO make the automatic identification and classification more intricate and challenging [13].

To overcome this complex challenge, several RE researchers have suggested software requirements extraction methods or techniques through using and integrating various ML algorithms applicable in several domains: for instance, effectively eliciting software requirements from the Q&A posts text on SO [22, 23], software requirements elicitation from the documents [12], and automatically classifying user requests in diverse crowd sourced requirements [11]. The results obtained from using these different ML algorithms/techniques for software requirements identification/extraction are significant and more specifically for the SO platform. Nonetheless, a complete, systematic, and detailed comprehension of these emerging ML based techniques for identification or recognition of software requirements on SO is currently unavailable in the existing literature [13]. This SLR study is aimed at filling this gap for both researchers and practitioners' community.

For this SLR study, we have thoroughly followed the systematic literature review (SLR) method as our primary research method [25, 26], with the aim of identifying and classifying the available empirical evidence about the use of emerging ML methods or algorithms for diverse software requirements identification on the SO platform for developing quality systems. The SLR method has been used successfully on diverse topics within the area of requirements engineering [27–30].

The research method will assist in getting a more in-depth understanding of different emerging ML algorithms or methods for identifying/recognizing and comprehending software requirements on the SO, facilitating the following:

Identifying research gaps (research opportunities)

Better decision-making (practitioners) when selecting an ML algorithm

Effectively planning the software requirements identification and avoiding pitfalls

This work presented a detailed SLR work of 12 primary studies related to the emerging ML based approaches or techniques for software requirements recognition or identification on the SO platform. The main research goal of this SLR study is to recognize or identify and categorize or classify the type of machine learning algorithms or techniques used for identifying software requirements on the Stack Overflow platform. Our SLR work is aimed at identifying various types of machine learning algorithms or techniques that have been properly utilized to identify the diverse software requirements on the SO, their working, and evaluation mechanisms. These outcomes will ultimately help and enable us to recognize the main complex issues and challenges that need to be properly tackled to enhance the working capabilities of the different machine learning based techniques. The specific contributions of our SLR study worth mentioning are as follows.

Contribution 1: detecting and categorizing the type of software requirements (RQ1) and ML algorithms or techniques used for identifying diverse software requirements on the SO (RQ2)

Contribution 2: identifying the processes (RQ3) and the performance evaluations of the used ML algorithms (RQ4)

Contribution 3: identifying and collecting basic demographic data, for instance, active researchers (DQ1), types of organizations (DQ2) and countries involved (DQ3), and the most frequently reported publication venues (DQ4)

Contribution 4: collecting diverse evidence giving a centralized source to facilitate the research

The goal of our SLR work is broad enough so we divided it into the set of different research questions (RQs) specified as follows.

RQ1: what types of software requirements identified or reported in the selected studies?

RQ2: what are the types of ML algorithms that have been used for identifying software requirements on SO in the selected studies? Do the ML based approaches outperform the non-ML based approaches? Are there any ML based techniques that considerably outperform the other ML based techniques?

RQ3: what are the types of procedures the reported machine learning algorithms use to identify software requirements on SO?

RQ4: what are the methods utilized to assess the performance of the machine learning algorithms applied in the selected studies? What are the performance outcomes of the reported ML algorithms?

The remainder of the research paper is organized as follows: Section 2 basically describes all the related work. The research methodology followed for this study is explained comprehensively in Section 3. In Section 4, we briefly presented the main results and discussion of the SLR study. Section 5 briefly summarizes the key findings, limitations, and open challenges identified in our SLR study. The different types of the validity threats of the SLR study are

discussed in detail in Section 6. Finally, Section 7 concludes the paper and discusses briefly how the key findings of our SLR study can be further effectively used by the researchers and practitioners in their future research endeavors.

2. Related Work

As per our knowledge till now, there is no single SLR study performed on utilizing machine learning algorithms or techniques for identifying software requirements on the SO online Q&A website. There are some surveys available on the SO [13, 31], but as of now none of them are related in any way to ML algorithms or methods for detecting/identifying software requirements on the SO online Q&A website.

In 2018, Ahmad et al. [13] performed an ad hoc literature review, by selecting 166 research papers on the SO that were mainly classified about software development life cycle from the start of the SO website till the year 2016 positively. They categorized the studies on SO platform into two: "SO design and usage" and "SO content applications." This categorization not only gives detailed comprehensions to the SO Q&A forum or platform providers about the possible limitations in the design and usage of such platforms but also offers ways to address them in the days to come. It also empowers software programmers to utilize such platforms for the recognized underutilized different tasks of software development lifecycle, e.g., software requirements specification, design, analysis, and validation.

Similarly, the work of Baltadzhieva and Chrupała [31] thoroughly reviewed and analyzed various questions quality posted on diverse community question answering (CQA) websites like SO. Besides, they pointed out the different metrics through which the quality of the posted questions can be identified, which in large ultimately lead to affecting the question quality.

There are various other surveys available [32–34] that are performed for identifying ML algorithms used for recognizing, collecting, and categorizing/classifying the nonfunctional requirements in software documents. In 2013, Meth et al. [32] conducted an SLR on investigating the works on automated requirements elicitation. They selected 36 primary papers published between January 1992 and March 2012. They categorized the identified works through an analysis framework, including tool categorization, technological concepts, and assessment approaches.

Later on, Binkhonain and Zaho [33] conducted an SLR on ML algorithms for identifying and classifying NFRs. They have selected 24 primary studies published. The key findings of their work revealed that ML approaches could identify and classify NFRs, but they still have many challenges that need more attention. Recently, Iqbal et al. [34] presented a survey on ML algorithms and requirements engineering. They provided a bird's-eye view of how ML algorithms are aiding different requirements engineering activities. The results revealed that the impact of ML algorithms could be found in different phases of RE lifecycle, e.g., software requirements elicitation, analysis, specification, validation, and management.

Besides, there are some surveys, SLR's, and systematic mapping studies done in other areas on sentiment analysis of scientific citations [35], data preprocessing methods for class imbalance problem [36], ML algorithms or techniques based software development effort estimation models [37], usability in agile software development [38, 39], and requirements prioritization [40], among others.

Unlike the different related works cited above, our study is focused on identifying and classifying ML algorithms or techniques used for identifying diverse software requirements on the SO only. We have selected 12 primary studies for our SLR work published until May 2020. Thus, we are the first to perform a comprehensive SLR study aimed at identifying, reviewing, summarizing, assessing, and thoroughly reporting the diverse works of ML algorithms or techniques for identifying diverse software requirements on the SO.

3. SLR Research Methodology

There has been a rapid surge in the popularity of using the Evidence-Based Software Engineering (EBSE) among researchers due to the applicability of systematic literature review (SLR) in various domains [41–43]. The key goal of the SLR study is to systematically identify, classify, and thoroughly synthesize any new evidence based on the data extracted from the selected research publications.

We conducted a comprehensive SLR study, thoroughly following the systematic guidelines defined and stated in [25, 26, 44–46], with the aim of identifying and classifying the types of machine learning algorithms or techniques used for identifying the software requirements on the Stack Overflow. The steps of the SLR research method encompassed the following activities:

(1) Planning for the SLR study (the SLR protocol):
   (a) Proper definition of the research goals and the set of different research questions
   (b) Description of the strategies for search, selection, and data extraction processes
   (c) Identification and description of potential validity threats
   (d) Tasks allocation and assignment (roles and responsibilities of every researcher involved)

(2) Conducting the SLR study (executing the SLR protocol):
   (a) Thoroughly searching for the primary studies
   (b) Selecting cautiously the relevant primary studies
   (c) Data extraction and thorough analysis of the selected primary studies to produce a classification schema (the map)

(3) Reporting the SLR results:

The initial findings of our SLR study have been published previously in a conference [47]; since then, the research questions have been added (demographic research questions) and modified a little bit (specifically RQ2 and RQ4). In addition, we

have also added snowballing search strategy [48] as a complementary strategy in addition to the automated electronic data sources with the aim of not overlooking any relevant paper. Basically, the conference paper just reported the initial findings with no detailed analysis of the results. In this work we have deeply assessed all the findings of the research questions, reported the key findings and limitations, and discussed the open challenges for future researchers. The subsequent subsections give comprehensive information regarding the main activities of the SLR protocol used.

3.1. Goal and Research Questions. The key research goal of our SLR study is to recognize or identify and classify or categorize the different types of machine learning algorithms or techniques used for identifying the software requirements on the Stack Overflow.

The goal of our SLR work is broad enough, so we divided it into the set of different research questions (RQs) specified in Table 1.

We also collected evidence to answer some interesting demographic questions (DQs) as suggested in [44–46] associated with the identification of the most actively participating researchers, organizations affiliations (academia or industry), and countries, as well as the top publication venues. Table 2 presents briefly the description of every DQ.

3.2. Search Strategy (EDS and Snowballing Method). The EBSE technique is totally dependent on the approach of identifying, collecting, and summarizing all the existing empirical evidence. Nonetheless, it is quite hard to fully ensure that all the existing empirical evidence was recognized; we ultimately need to abate the validity threat of not relying on single search strategy [49]. Therefore, two diverse search approaches, i.e., EDS and snowballing, were considered, with the aim of complementing each other, retrieving the highest number (recall) of potentially relevant studies (precision). The relevant experts on the RE and Stack Overflow areas validated the diverse search strategies.

The search was thoroughly performed in four diverse electronic data sources (EDS), namely, the ACM Digital Library, IEEE Xplore, Scopus, and Web of Science (WoS), respectively [50, 51]. These different EDS ensure including the diverse main venues (i.e., conference proceedings, workshops, and journals) specifically in the field of Software Engineering [52, 53]. The automated search strings were developed from the combination of the key terms extracted from our defined research questions, keywords from the different research publications retrieved by a pilot search, and the list of synonyms.

The initial set of search terms were integrated and adopted to the particular EDS with the help of using the "AND" (if allowed) and "OR" (if allowed) logical operators wherever possible. We conducted several search rounds in the diverse EDS until we accomplished the best balance between precision and recall measures. Table 3 presents the set of final selected search strings, adapted to each of the four electronic data sources, respectively. The automated search strategy was carried out on 23/08/2019 and successfully retrieved 1,073 results only.

We also performed both backward and forward snowballing search methodology as outlined in [48, 54] in addition to the automated electronic databases search strategy on 27/05/2020. To start the snowballing process, the 12 primary studies were used as initial seeds that were selected from the automated search strategy. In the snowballing process, it is vital to consider only the suitable or relevant research studies, so to ensure this, we adopted a top-down sequential process to include only the relevant research papers for each stage of the new iteration. To guarantee the relevance, we also performed the data extraction with the aim that the preselected papers were suitable for answering the defined research questions. All those papers were selected as the new seeds for the next stage or iteration of the snowballing process which passed the data extraction criterion.

Finally, the snowballing process retrieved 1,411 papers and ended the process at the third iteration with no new primary papers found. To minimize the possible biasness, two of the authors individually conducted the selection of the papers, and a third author carefully reviewed the data generated from every iteration, integrated the individual outcomes, and thoroughly assessed them for any disagreements. Tables 4 and 5, respectively, present the outputs of the two adopted search approaches for the SLR study.

3.3. Research Paper Selection Criteria (Inclusion and Exclusion Criteria). The selection process primarily consists of two tasks: a principally perfect definition of both inclusion and exclusion criteria and truly applying these definite benchmarks to select the pertinent primary research studies [55, 56]. As inclusion and exclusion are principally two conflicting activities, we chose to categorically focus our efforts on the exclusion criterion, by outlining a clear set of criteria, both objectively and subjectively appropriate. The former one does not cause any sort of threat to the validity, and, henceforth, its application is much easier and simpler. While applying the very first exclusion criterion, specifically, those related to the language and duplicity assisted us to remove irrelevant data quite rapidly. For this SLR study, the following objective exclusion criteria were applied to all the retrieved papers:

(a) Exclusion criterion 1: research papers not written in English language

(b) Exclusion criterion 2: short research papers (less than four pages in length)

(c) Exclusion criterion 3: research papers not published in peer-reviewed publication venues

(d) Exclusion criterion 4: research papers that are not a primary research study (secondary and tertiary research studies)

Table 1: Description of the research questions.

Research question	Description
RQ1: what are the types of software requirements that are identified or reported in the selected studies?	The different types of software requirements are functional requirements (FRs) and nonfunctional requirements (NFRs), or others.
RQ2: what are the types of ML algorithms that have been used for identifying software requirements on SO in the selected studies? Do the ML based approaches outperform the non-ML based approaches? Are there any ML based techniques that considerably outperform the other ML based techniques?	To identify all the techniques or methods (algorithms) used in the selected primary studies.
RQ3: what are the types of procedures the reported machine learning algorithms use to identify software requirements on SO?	To know all the processes (natural language processing); those are used in the selected primary studies.
RQ4: what are the methods utilized to assess the performance of the machine learning algorithms applied in the selected studies? What are the performance outcomes of the reported ML algorithms?	To know the different performance evaluation criteria applied in the selected primary studies, their results, strengths, and weaknesses, respectively.

Table 2: Descriptions of the demographic questions.

Demographic questions	Description
DQ1: who are the most actively participating researchers?	All authors, ordered by the number of papers
DQ2: which organizations are the most active?	Based on the affiliations of all the authors
DQ3: which countries are the most active based on authors affiliations?	Based on the affiliations of all the authors
DQ4: which are the top venues for the publications?	Type (conference, journal, or workshop) and the name of the publishing venue

Table 3: Set of search strings, adapted to each of the EDS.

EDS	Search strings
IEEE Xplore	(("Document Title" : "requirements*") OR ("document Title" : "NFR") OR ("document Title" : "functional requirements") OR ("document title" : "non-functional requirements") OR ("document title" : "quality requirements")) AND (("Abstract" : Stackoverflow) OR ("Abstract" : "stack overflow") OR ("Abstract" : techniques) OR ("Abstract" : "machine learning")) AND (("author Keywords" : "requirements*") OR ("author Keywords" : "stack overflow") OR ("author Keywords" : stackoverflow) OR ("author Keywords" : "NFR") OR ("author keywords" : "non-functional") OR ("author keywords" : "functional"))
ACM Digital Library	(acmdlTitle : (+requirements* + NFR + quality + functional + non-functional) OR acmdlTitle : (Stackoverflow stack overflow)) and recordAbstract : (algorithm techniques "machine learning") OR keywords.author.keyword : (+requirements* + stackoverflow + functional + non-functional + FR + NFR))
Web of Science (WoS)	TI = ("requirements engineering" OR "NFR" OR "non-functional requirements" OR "quality requirements" OR "functional requirements") AND TS = (Stack overflow OR Stackoverflow OR "machine learning" OR algorithms OR techniques) and language: (english) refined by: web of science categories: (computer science software engineering)
Scopus	TITLE ("requirements engineering" OR NFR OR non-functional OR functional OR quality) and TITLE-ABS-KEY (stack overflow OR stackoverflow OR "machine learning" OR techniques OR algorithms) AND (LIMIT-TO (SUBJAREA, "COMP")) AND (LIMIT-TO (LANGUAGE, "English"))

(e) Exclusion criterion 5: any kind of grey literature (books, presentations, poster sessions, forewords, talks, editorials, tutorials, panels, keynotes, etc.)

(f) Exclusion criterion 6: all sorts of research thesis whether Ph.D. or master or bachelor theses

It is obvious that subjective criteria are very complex to address adequately in any SLR study including this one. They are prone to create biasness into the SLR study, and, thus, a predefined protocol principally needs to be applied with the aim of minimizing this threat. On the contrary, applying these criteria might also leads to a substantial reduction in the number of research papers to consider as being relevant. For this SLR study, the authors applied the two exclusion criteria described as follows:

(a) Not focus: research studies not related to any of the RE activities on the Stack Overflow

(b) Out of scope: research studies not related to any of the RE phases of software development lifecycle

Any research paper not excluded by the aforementioned criteria was deemed relevant and included in the set of final selected primary research studies. The authors primarily adopted a top-down method to the application of these criteria on the research papers. In the first stage, some metadata information such as the title, abstract, and keywords of the research paper was taken into consideration. If these data were not sufficient to exclude any research paper at hand, then the authors reviewed the full text of the research publication, more specifically the introduction

Table 4: Automated searching in EDS results.

Source	Works retrieved
ACM Digital Library	269
IEEE Xplore	481
Scopus	104
Web of Science	219
Total	1,073

Table 5: Snowballing method search results.

Snow balling iterations	Number of seeds	Number of citations
First iteration	12	783
Second iteration	82	624
Third iteration	3	4
Total	97	1,411

(problems and contributions of the research study), the results, and conclusions sections of the research study.

To handle appropriately with any disagreements, the authors primarily followed the inclusive criteria as systematically suggested in [44] and described in detail in Table 6. The authors excluded a research publication at hand only when both of the reviewers agreed (category "F") or marked the research publication as the borderline (category "E").

The complete diagrammatic flow of both the searches performed (EDS and Snowballing), detailed systematic selection processes and the outcome of every task of the SLR study are reflected in Figure 1. A final set of 12 research papers was selected for this SLR study (the detailed list with full bibliographic references is provided in Table 7). Besides, the details of the quality assessment criterion adopted for the SLR study are described in the next section.

3.4. Quality Assessment Criteria. For any research publication to pass the defined selection phase, a comprehensive quality assessment criterion was defined. The authors defined principally "4" quality assessment criteria questions to assess the rigorousness, reliability, and significance of the relevant studies as suggested by [6, 57–59] for the research paper as shown in Table 8. Thus, a research study which accomplished a quality score of "4" was thus considered in the final selection.

3.5. Data Extraction Process. To minimize any sort of biasness in the data extraction process, one of the authors' developed a comprehensive data extraction form (DEF) in the spread sheet format. The DEF (see Table 9) was mainly used to extract and store the data for each of the selected research studies. The rest of the authors thoroughly reviewed, improved, and agreed upon the DEF before properly starting the data extraction process. The proper use of the DEF facilitates a detailed, systematic, explicit, and consistent approach to conduct the data extraction process of an SLR study [60, 61].

Table 6: Dealing with disagreements.

		Reviewer X		
		Include	Uncertain	Exclude
Reviewer Y	Include	A	B	D
	Uncertain	B	C	E
	Exclude	D	E	F

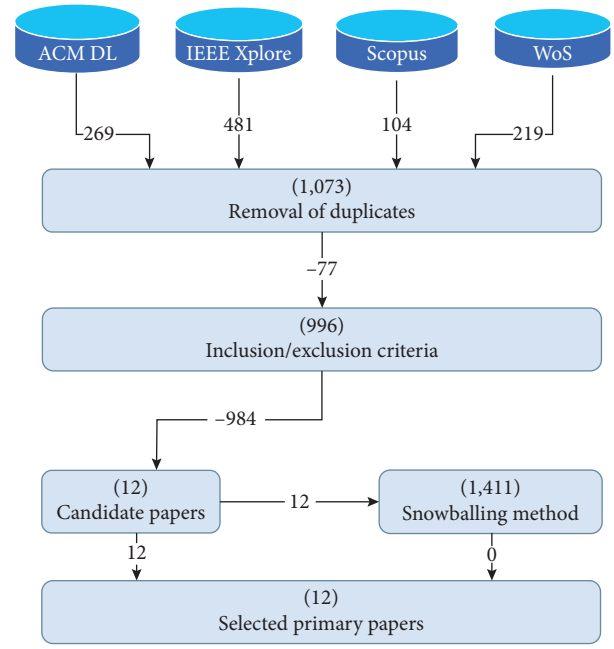


Figure 1: Summary of the searches and selection processes of the SLR study.

To conduct the data extraction process properly, the authors distributed the set of selected primary research publications into two halves, as stated below.

The first half of the research papers was blindly allocated to the reviewers R1 (the first author) and R2 (the second author), respectively. Both of the reviewers R1 and R2 thoroughly gauged the work independently and, after finishing their respective tasks, fixed all the discrepancies (if any) to yield an agreed dataset. The second half of the research papers, too, was blindly allocated to the reviewers R1 (the first author) and R3 (third author), respectively. Each pair of the reviewers populated the DEF individually. When any sort of disagreements arose, a consensus meeting was arranged, with the participation of a third reviewer, until all (if any) discrepancies were fixed [62].

4. Results and Discussion

In this section, the authors present and discuss briefly the results of the SLR study. For each of the RQs, the authors instigate them with a summary of the most noteworthy results, a discussion about the most relevant facets, and, based on them, a suggestion of some explanatory hypotheses.

Table 7: List of selected primary studies.

Paper ID	Full bibliographic reference
S01	Yin, H., and Pfahl, D. (2017, November). A preliminary study on the suitability of stack overflow for open innovation in requirements engineering. In Proceedings of the 3rd International Conference on Communication and Information Processing (pp. 45–49). ACM.
S02	Zou, J., Xu, L., Guo, W., Yan, M., Yang, D., and Zhang, X. (2015, May). Which non-functional requirements do developers focus on? An empirical study on stack overflow using topic analysis. In 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories (pp. 446–449). IEEE.
S03	Pinto, G., castor, F., and liu, Y. D. (2014, May). Mining questions about software energy consumption. In Proceedings of the 11th Working Conference on Mining Software Repositories (pp. 22–31). ACM.
S04	Ahmad, A., li, K., Feng, C., and sun, T. (2018, October) "An empirical study on how iOS developers report quality Aspects on stack overflow," international journal of machine learning and computing vol. 8, no. 5, pp. 501–506, 2018. IJMLC.
S05	Treude, C., Barzilay, O., and storey, M. A. (2011, May). How do programmers ask and answer questions on the web?: Nier track. In 2011 33rd International Conference on Software Engineering (ICSE) (pp. 804–807). IEEE.
S06	Zou, J., Xu, L., Yang, M., Zhang, X., and Yang, D. (2017). Towards comprehending the non-functional requirements through developers' eyes: An exploration of stack overflow using topic analysis. Information and Software Technology, 84, 19–32.
S07	Bajaj, K., pattabiraman, K., and mesbah, A. (2014, May). Mining questions asked by web developers. In Proceedings of the 11th Working Conference on Mining Software Repositories (pp. 112–121). ACM.
S08	Ahmad, A., Feng, C., li, K., Asim, S. M., and sun, T. (2019). Toward empirically investigating non-Functional requirements of iOS developers on stack overflow. IEEE Access, 7, 61145–61169.
S09	Xiao, M., Yin, G., wang, T., Yang, C., and chen, M. (2015). Requirement acquisition from social q&a sites. In Requirements Engineering in the Big Data Era (pp. 64–74). Springer, Berlin, Heidelberg.
S10	Rosen, C., and shihab, E. (2016). What are mobile developers asking about? A large scale study using stack overflow. Empirical Software Engineering, 21(3), 1192–1223.
S11	Abad, Z. S. H., shymka, A., pant, S., currie, A., and ruhe, G. (2016, September). What are practitioners asking about requirements engineering? An exploratory analysis of social q&a sites. In 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW) (pp. 334–343). IEEE.
S12	Pinto, G. H., and Kamei, F. (2013, October). What programmers say about refactoring tools?: An empirical investigation of stack overflow. In Proceedings of the 2013 ACM workshop on Workshop on refactoring tools (pp. 33–36). ACM.

Table 8: Quality assessments.

Quality criteria	Score
Are the findings and results explicitly mentioned in the study?	Yes = 1 No = 0
Is there any empirical evidence on the findings in the study?	Yes = 1 No = 0
Are the arguments well-presented and justified in the study?	Yes = 1 No = 0
Is the study well referenced?	Yes = 1 No = 0

4.1. RQ1: What Are the Types of Software Requirements Identified or Reported in the Selected Studies? For the data extraction process, the authors started with a preestablished classification scheme of software requirements to group the types of reported software requirements. The classification scheme includes primarily two types of software requirements: that is, functional requirements (FRs) and nonfunctional requirements (NFRs). The authors found direct evidence about the presence of both types of software requirements, FRs and NFRs, on the SO. The authors identified seven primary papers (S01, S03, S07, S09, S10, S11, and S12) related to functional requirements and five studies (S02, S04, S05, S06, and S08) related to nonfunctional requirements identified on the SO, as depicted in Table 10. Table 7 shows the full list of selected papers.

The majority (approx. 58.33 %) of the selected primary studies specifically focused on extracting functional requirements (FRs) related to general software development tools. Some examples of the identified and extracted FRs questions were about extracting developers' needs of Eclipse IDE tools (e.g., S01), suggesting features (e.g., S03), reporting issues (e.g., S07), and requesting new features (e.g., S09, S10, S11, and S12) among others. The remaining approximately 41.66% of the selected primary studies, identified the nonfunctional requirements (NFRs) on the SO. Some examples of the identified/recognized NFRs were primarily related to the mobile application development tools (e.g., S02, S04, S06, and S08) and the web (e.g., S05).

4.2. RQ2: What Are the Types of ML Algorithms That Have Been Used for Identifying Software Requirements on SO in the Selected Studies? Do the ML-Based Approaches Outperform the Non-ML-Based Approaches? Are There Any ML-Based Techniques That Considerably Outperform the Other ML-Based Techniques? There is a plethora of diverse machine learning algorithms and techniques available for text processing; however, they can be classified primarily into two main categories named as the supervised learning (SL) and the unsupervised learning (USL) algorithms/techniques. The aim of the supervised learning algorithm is to deduce a function from the labelled training comprised of data instances and the anticipated outcome value for each of the instances. Then, the inferred function could be utilized to predict the label for all the hidden data instances [63]. On the contrary, the unsupervised learning algorithm deduces a function to specify the unseen structure of the unlabelled data [63]. Hence, it explores the data for similar patterns,

Table 9: Description of the data extraction form.

Data item	Description
Bibliographic information	Authors, title of the paper, name and type of the organization, country, publication type and venue, and year of publication.
Study goal	The main intent of the selected primary research studies.
ML algorithm/technique	The different types of the ML algorithms/techniques employed to identify and classify the software requirements.
Types of the requirements	The different types of software requirements FRs/NFRs.
Process/approach	The process/approach used to identify the FRs/NFRs from posts.
Performance evaluation	How were the performance and efficiency of the ML algorithms measured in the selected primary research studies and what were their actual outcomes?

Table 10: Papers reporting different types of software requirements.

Primary papers	Type of software requirements	Support level
S01, S03, S07, S09, S10, S11, S12	FRs	7
S02, S04, S05, S06, S08	NFRs	5

and then grouping (clustering) of the similar instances is produced. Besides, there is another type called the semi-supervised learning (SSL) that lies between the supervised and the unsupervised learning algorithms/techniques. The aim in the semisupervised learning is to deduce a function grounded on lesser amount of labelled training dataset and huge amount of unlabelled dataset [63].

The Latent Dirichlet Allocation (LDA) is one of the most widely adopted and naïve statistical topic modeling algorithms. The basic idea behind LDA is that each document in the text corpus may be related to several different topics. LDA is clearly defined by its generative process, the process by which it is expected that each text document of the corpus is produced. In this generative model, it is supposed that each text document of the corpus is created mainly in two stages: in the first stage, the set of topics are selected for the text document based on the distribution of the topics. In the second stage, to generate each word in the text document, a topic is selected based on the distribution in the first stage, and then a word is arbitrarily chosen conferring to the distribution of words for each of the topics. In this manner, it generates a set of words and places it together as a text document. In this generative model, each of the text documents is characterized as a bag-of-words, that is, a naive characterization of a text as a multiset of words, ignoring grammar and the ordering of the words in the text document [64, 65].

The Support Vector Machines (SVM) are a group of SL ML algorithms, primarily utilized for classifications or/and regressions. The SVM method is basically grounded upon the statistical learning theory and the certain dimensions mentioned in [66, 67]. The purpose of the SVM technique is to create the optimal separating line aimed at categorizing all of the input data into various classes.

The thematic analysis is not an ML algorithm but a process/technique (manual or automatic) that involves exploring, identifying, extracting, and storing the patterns (or themes) that emerge within data at hand. Besides, the identified themes are basically the patterns that emerge across the datasets significant to the description of an occurrence and are related to a specific research question (RQ). Nevertheless, the themes are recognized by combining the modules or pieces of concepts or experiences, which are usually worthless when examined in separation. Finally, these themes ultimately serve as the categories for the analysis [68–70].

The outcome of this RQ revealed that there were mainly two different machine learning algorithms identified in the selected primary research studies. These machine learning algorithms basically fall into two types, comprising one SL, and eight USL ML algorithms. Besides, we also found three primary studies that used thematic analysis or qualitative coding techniques, as comprehensively summarized in Table 11. Table 11 also depicts that USL algorithms, specifically LDA, are the most popular/common type of machine learning algorithm, as they were utilized in eight primary research studies (averaging approximately 67%).

Besides, the FRs were identified in the four studies (e.g., S01, S07, S10, and S11) and the NFRs were also identified in the four selected studies (e.g., S02, S04, S06, and S08), respectively, out of the eight primary studies that used the LDA algorithm. SVM was the second most popular ML algorithm category (approximately 8.3%). It was quite interesting to observe that three of the selected primary studies (25%) did not use any algorithm and have used thematic coding or qualitative coding technique for identifying and classifying the different types of software requirements on the SO.

The detailed performance evaluations of all the ML algorithms and techniques of the selected studies of our SLR study are discussed in the results and discussion section of RQ4 (see Section 4.4).

4.3. RQ3: What Are the Types of Procedures the Reported Machine Learning Algorithms Use to Identify Software Requirements on SO? The outcomes of the detailed analysis of the 12 selected primary research studies of the SLR study have depicted a general process pattern for utilizing or

Table 11: Different ML algorithms.

Type	ML algorithms/ technique	Purpose	Type of software requirements	Support level
USL	LDA	Finding the hidden or latent topic(s) that the documents (posts) contain based on their probabilities	FRs (S01, S07, S10, and S11), NFRs (S02, S04, S06, and S08)	8
	Thematic coding	It involves exploring, identifying, extracting, and storing the patterns or themes	FRs (S03 and S12), NFRs (S05)	3
SL	SVM	It is primarily used for classification and is grounded upon the statistical learning theory	FRs (S09)	1

applying the machine learning algorithms approach to recognize or identify and categorize or classify the different types of software requirements on the SO. This whole process could possibly be classified into three (3) main stages or phases: the text preprocessing phase is the phase in which the irrelevant text from the data is removed (RQ3); the learning phase basically involves applying the actual different ML algorithms (addressed in RQ2); and finally the evaluation phase (addressed in RQ4) involves assessing or evaluating an ML algorithm's approach to identify or recognize and categorize or classify the different types of software requirements on the SO as depicted in Figure 2.

The text preparation phase is the text preprocessing phase, where it takes textual data of SO Q&A as input and applies some NLP techniques to clean the textual data for further processing. A total of six different NLP preprocessing techniques were identified in the selected research papers as depicted in Table 12. The identified different preprocessing techniques are explained briefly as follows:

Stop words removal: it is the process of removing certain auxiliary verbs (e.g., be, do, and have), conjunctions (e.g., and, or), and different articles (e.g., the, a, and an) from the text [71]

Case unification: it is the process of converting the text into a uniform style, i.e., lowercase or uppercase

Tokenization: it is the process of splitting the sentence into words [71]

Stemming: it is the process of stemming or reducing the words to their origin or roots. For instance, words like 'goes,'' gone,' and "going" will be reduced to 'go' [72]

Punctuations removal: it is the process of removing different punctuations (e.g., commas, semicolons, question marks, and exclamation marks) from the text

The outcome of this research question (RQ) revealed that approximately 66.67% of the selected studies used different preprocessing techniques. The remaining studies averaging approximately (33.33%) lack reporting about using any of the preprocessing steps. The different preprocessing approaches or techniques used in the selected research studies are the stop words removal (approx. 58.33%), case unification (approx. 50%), tokenization (approx. 41.66%), stemming (approx. 25%), punctuations removal (approx. 16.67%), and filtration (approx. 8.33%) as depicted in Table 12. These findings reveal that there is lack of using different appropriate and uniform NLP techniques for preprocessing the SO Q&A text.

As depicted in Figure 2, predefined data is a prerequisite for building or constructing FRs/NFRs classifiers using the various ML algorithms. Specifically, the SL machine learning algorithms need a prelabelled dataset, whereas the USL machine learning algorithms need a set of predefined groups or keywords. Besides, the SSL machine learning approaches, alike the SL machine learning approaches, also need a prelabelled dataset. This process is not been explicitly stated in the surveyed techniques/approaches of this SLR research study. In addition, most of the selected primary research studies treat the machine learning based methods as 'black boxes' and offer no explicit explanation on how actually all these machine learning methods work.

In the evaluation phase, the assessment of both the FRs/NFRs classifier is performed using several methods to find the efficacy of the classifiers. Besides, the performance evaluation of the techniques and results is discussed in the results and discussion section of RQ4 (see Section 4.4).

4.4. RQ4: What Are the Methods Utilized to Assess the Performance of the Machine Learning Algorithms Applied in the Selected Studies? What Are the Performance Outcomes of the Reported ML Algorithms? Out of the 12 selected research studies of this SLR study, merely 6 of them (averaging approx. 50%) have stated their performance evaluation criterion. Table 13 briefly sums up the four performance evaluation criteria recognized from the selected research studies and their respective outcomes.

It was quite significant to observe (as depicted in Table 13) that though the LDA and SVM were used in various selected research papers, their performance results considerably vary from one another. For instance, LDA performed well in S08; however it did not perform that good in the S11. In the S04, LDA got a fairly well recall outcome compared to the precision outcome. Nevertheless, all of the selected studies achieved considerably good recall results compared to the precision results irrespective of the ML algorithm applied, that is, LDA and SVM.

Amongst the four different performance evaluation criteria, the precision and the recall are deemed amongst the famous ones, trailed by the F-measure and the word and topic intrusion. In the following, we briefly describe how effectively the four measures are used to enumerate and

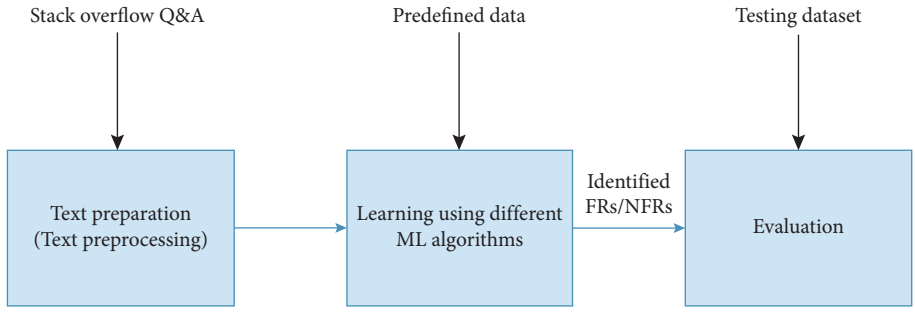


Figure 2: A generic process of using MLl algorithms to identify software requirements on the SO.

Table 12: NLP techniques used for text processing.

NLP process	Paper ID	Support level
Stop words removal	S02, S04, S06, S07, S08, S09, and S11	7
Case unification	S02, S04, S06, S08, S09, and S11	6
Tokenization	S02, S04, S06, S08, and S09	5
Stemming	S07, S09, and S11	3
Punctuations removal	S09 and S11	2
Filtration	S10	1

Table 13: Performance evaluations used.

ML algorithm	Study ID	Precision (%)	Recall (%)	F-measure (%)	Word and topic intrusion (%)
LDA	S02	68.7	75.9		
	S04	70.33	77		
	S06	68.7	75.9		
	S08	70.33	77		
	S11				58
SVM	S09	72	77	74.42	

evaluate the performances of the described ML techniques/algorithms in the 12 selected research studies of the SLR study.

4.4.1. Precision and Recall Measures. Out of the 12 selected research papers of this SLR study that state their diverse performance measures, merely 4 of them (averaging approx. 41.66%) have properly employed these two matrices with the aim of effectively assessing the performance of the utilized machine learning techniques and algorithms. The precision measure could be defined as the aggregate number of correctly classified FRs/NFRs in ratio to the number of FRs/NFRs retrieved. It is mathematically represented as P = true positive/(true positive + false positive). Recall measures can be defined as the ratio of FRs/NFRs that have been correctly categorized/distributed and mathematically can be represented as R = true positives/(true positives + false negatives) [73]. True positive basically characterizes the number of correctly classified software requirements (FRs/NFRs), the false positive characterizes the amount of incorrectly categorized/distributed diverse software requirements (FRs/NFRs), and true negative signifies the amount of diverse software requirements (FRs/NFRs) correctly not categorized/distributed, whereas false negative is basically the amount of diverse software requirements (FRs/NFRs) incorrectly not classified [74].

The precision and recall evaluation measures are mostly used together [75], and there always exists a trade-off between them. Besides, the goal of the precision evaluation measure is to ensure that all of the retrieved diverse software requirements are correctly relevant, while the recall aims at retrieving all of the relevant diverse software requirements. Nevertheless, which value (Precision or Recall) is highly significant to quantify the identification or classification results can be considerably argued. The research studies S02, S04, S06, S08, and S09 emphasized on accomplishing a high recall value as it was considered that the accuracy of automatic software requirements labeling (FRs/NFRs) in the topics/posts is important. It is worth mentioning that the selected studies S02, S04, S06, S08, and S09 have achieved high recall rate which indicates the acceptability of accurately labeling or classifying the requirements (FRs/NFRs) identified. Among the selected 12 studies, S09 achieved high recall and precision rates followed by S04 and S08, which signifies the performance of the ML algorithms used.

Nevertheless, S09 viewed that precision was also equally significant if the recall was satisfactory for the automatic categorization/classification to ignore huge number of irrelevant diverse software requirements (FRs/NFRs) from

being wrongly classified as pertinent (false positives). It is also evident from the values of both the precision and the recall as depicted in Table 13 that there are no studies that achieved higher recall and considerable lower precision results, or there were no such research papers that have achieved considerably higher precision and considerably lower recall results.

4.4.2. F-Measures. The F-measure takes into consideration both the precision and the recall measures and is basically the weighted average of the precision and recall measures. F-measure can be defined as F-measure = 2 × (precision × recall)/(precision + recall). We can also define the F-measure as the harmonic mean of precision and recall measures [76]. Only one (8.33%) of the selected primary research studies applied the F-measure. However, they did not give explicitly the reasons and importance of employing this measure in their study. They only claimed that this measure is commonly used for information extraction tasks. Nevertheless, they utilized this measure to integrate the precision and recall measures as the F-measure corresponds to the weighted average value of precision and recall measures [77] and then used the output value as a gauge for the performance evaluation.

4.4.3. Word and Topic Intrusion. Only one (averaging approx. 8.33%) of the selected studies used the word intrusion and topic intrusion method for evaluating the performance of the applied ML algorithm. The word intrusion method basically assesses the quality of the inferred topic(s) by computing their level of "cohesiveness." The topic intrusion method assesses whether the content of a document matches with the topic(s) it has been allotted to, conferring to the human judgment [78]. The authors did not give any reasons for employing this measure in their study and just claimed to assess the performance of the used ML algorithm (topic modeling) following the stepwise guidelines of [78].

4.5. DQs: Who Are the Most Actively Participating Researchers? A total of 45 researchers appear as authors in the 12 primary papers selected by our study. We only report the top 10 active researchers, based on their number of published works. It is worth stating that the data allowed us to identify groups of researchers who often published collaborative works (e.g., Zou, Xu, Zhang and Yang and Ahmad, Li, Feng and Sun). Besides, all of the active researchers are affiliated with academic organizations. It is also worth mentioning that all of the active researchers, those with two publications, are based in China except one author (Pinto) who is based in Brazil. Moreover, all of the identified researchers are currently active. The identified top ten active researchers and their affiliations are depicted in Table 14.

4.6. DQs: Which Organizations Are the Most Active? We have identified a total of 13 different organizations based on the number of mentions in the selected studies. All of the identified organizations are academic institutions, and the majority is based in China. Table 15 shows a list of the ten most active organizations. It is noteworthy that there is no single study reported from the industry; thus, there is a huge imbalance and gap between the amount of research from academia and industry (100% academic, 0% industry), which highlights the need for research in industrial contexts, or in academia-industry collaborations.

4.7. DQs: Which Countries Are the Most Active Based on the Authors' Affiliations? Based on the affiliations of all the authors, we have been able to identify seven different countries located in the four continents. The top three countries with the greatest representation are China, Canada, and Brazil. The rest of the countries do not reach a representation higher than 50% of the previous three. Figure 3 shows the number of authors whose affiliations belong to the seven countries with the number of mentions.

4.8. DQs: Which Are the Top Venues for the Publications? The outcome of this DQ revealed that the favorite venues for publication of the selected studies were mainly conferences with 50% of the primary works as depicted in Figure 4. The ACM/IEEE Working Conference on Mining Software Repositories, with the most significant primary papers published, topped the list of the most cited, as shown in Table 16. Combining with the workshops (16.7%), the conferences cover approximately 66.7% of the selected studies. Nonetheless, 33.3% of the papers, published in journals, are a worthy fact and provide a solid empirical basis, considering the quality and the standard prestige of publication venues (e.g., Information and Software Technology, Empirical Software Engineering, and IEEE Access). Table 16 shows the complete list of mentioned conferences, journals, and workshops.

Moreover, Figure 5 shows that the interest in identifying software requirements on SO has remained at a moderate level (approximately two papers per year) since 2014, with higher peaks in 2014, 2015, and 2017, respectively.

5. Findings, Limitations, and Open Issues

In this section, we will primarily discuss in brief some of the key research findings, the limitations of the reviewed ML approaches, and finally the open issues from our SLR results.

5.1. Key Findings of the SLR. Some of the significant findings that can be drawn from the SLR study are mentioned as follows:

The identified ML based techniques/approaches have considerably performed well by accomplishing an accuracy of approximately more than 70% in detecting/identifying and categorizing/classifying the software requirements (FRs and NFRs) on SO.

Overall, the SVM (SL algorithm) performed better than LDA (USL algorithm) though having the same recall

Table 14: Active researchers based on number of papers published.

Author's name	Organization	Support level
Pinto	Federal University of Pernambuco, Brazil	2
Zou	The School of Software Engineering, Chongqing University, China	2
Zhang	The School of Software Engineering, Chongqing University, China	2
Yang	The School of Software Engineering, Chongqing University, China	2
Xu	The School of Software Engineering, Chongqing University, China	2
Ahmad	Beijing Institute of Technology, China	2
Feng	Beijing Institute of Technology, China	2
Li	Beijing Institute of Technology, China	2
Sun	Beijing Institute of Technology, China	2
Yin	Institute of Computer Science, University of Tartu, Estonia	1

Table 15: Top 10 active organizations.

Organization's name and country	No. of mentions
The School of Software Engineering, Chongqing University, China	11
Beijing Institute of Technology, Beijing, China	8
College of Computer Science, National University of Defense Technology, China	5
Department of Computer Science, University of Calgary, Canada	5
Federal University of Pernambuco, Brazil	4
Electrical and Computer Engineering, University of British Columbia, Canada	3
Institute of Computer Science, University of Tartu, Estonia	2
Department of Computer Science, University of Victoria, Canada	2
Department of Statistics, University of Peshawar, Pakistan	1
SUNY Binghamton, USA	1

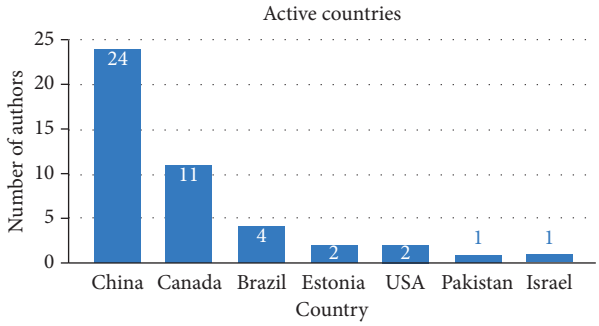


Figure 3: Most active countries (by authors' affiliation).

but better precision/F-measures and was used in only one of the selected primary studies. Besides, among the reviewed ML algorithms, SVM proved to have the excellent performance, with LDA (USL algorithm) being the most recurrently utilized ML algorithm.

The machine learning algorithms seem to perform well when separate words are employed as features requests, rather than the different phrases. Likewise, the ML techniques/algorithms used in the selected studies tend to yield better results when the words used are in the original form, instead of performing some preprocessing steps (stemming and lemmatization, among others) on the words.

The key findings discussed above depict that despite of the fact of being still at the infancy stage of the research, the machine learning based methods have produced considerably good results for identifying and classifying software requirements (FRs/NFRs). This is deemed to be an encouraging prospect.

5.2. SLR Limitations. During this SLR study, we have noticed some limitations specifically related to reporting and evaluating the ML techniques used in the selected research studies discussed as follows.

5.2.1. Deficient Reporting Standards. Predefined data is a needed both for SL and USL algorithms: the SL algorithms need a labelled data, whereas the USL algorithms require predefined classifications and related terms to accomplish better performance. Nonetheless, before employing any machine learning algorithm (the SL or the USL algorithm), the machine learning methods must follow some phases to preprocess the input data (in our case the SO posts) and find the pertinent features for the machine learning algorithm. But, this process is not explicitly stated in the surveyed techniques in the selected research studies of our SLR study. Besides, the majority of the chosen primary research studies treat the machine learning approaches as 'black boxes' and offer no explicit explanation on how actually all these machine learning approaches work. Consequently, this makes the SLR study fairly challenging. To guarantee the consistency of our SLR study, we have made a general process (see Figure 2) as a common framework to evaluate each of the selected studies individually. Moreover, we also thoroughly assessed the reported results of the selected studies to infer how the employed approaches worked.

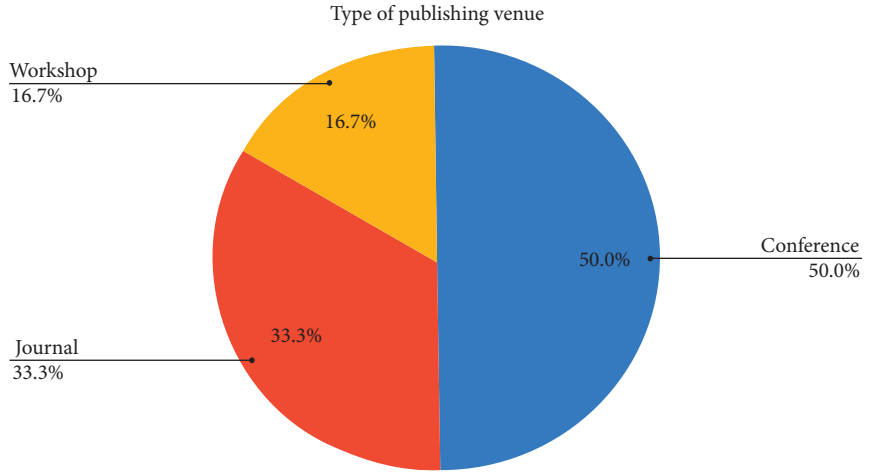


Figure 4: Distribution of publishing venues.

Table 16: Most cited conferences, journals, and workshops.

Publication name	Type of publication	Support level
ACM/IEEE Working Conference on Mining Software Repositories	Conference	3
IEEE International Conference on Software Engineering	Conference	1
ACM International Conference on Communication and Information Processing	Conference	1
Requirements Engineering in the Big Data Era	Conference	1
Information and Software Technology	Journal	1
Empirical Software Engineering	Journal	1
IEEE Access	Journal	1
International Journal of Machine Learning and Computing (IJMLC)	Journal	1
IEEE International Requirements Engineering Conference Workshops	Workshops	1
ACM workshop on Workshop on Refactoring Tools (WRT)	Workshops	1

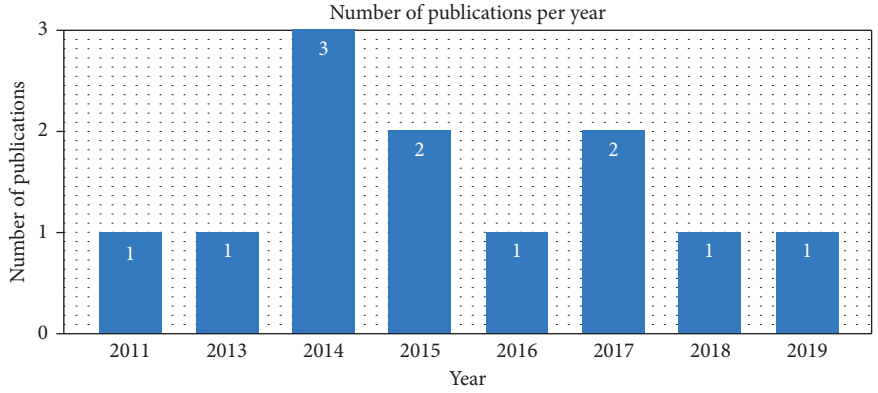


Figure 5: Evolution of the number of papers published.

5.2.2. Deficient Performance Evaluation Standards. In our SLR study, half of the selected research studies gave the evaluation results (6 out of 12 studies); the majority of them did not give the reasons behind using a certain performance evaluation technique. For instance, considering the precision and recall measure, the selected research studies did not entirely clarify which one was more significant: low recall and high precision measures or low precision and high recall measures. Besides, they did not state the reasons behind providing F-measure, word intrusion, and topic intrusion measure and not providing accuracy measure, or vice versa. Thus, we can conclude that the majority of the selected studies of our SLR study did not determine why they have used a certain technique that varies from others or how to describe their outcomes.

5.3. Open Issues. Through thorough assessment and thinking on this SLR study, we finally managed to identify five open issues normally confronted by practitioners or researchers, briefly explained as follows.

5.3.1. Need for Shared Standard Prelabelled Datasets. The first critical issue identified in the existing research is the deficiency of a standard prelabelled dataset. A preclassified standard dataset is mandatory to employ or utilize the SL algorithms more effectively. It considerably needs a lot of efforts and time to develop such a standard dataset. Besides, it needs the pool of real-life FRs/NFRs, the standard classification of these requirements FRs/NFRs, and the validation of the classifications from researchers and practitioners (domain experts). As per our knowledge, unfortunately there are no shared prelabelled datasets available; developing a shared prelabelled dataset can assist the researchers/practitioners to perform more experimentation and offer benchmarks for future performance. Nonetheless, the abundance of such prelabelled dataset will not only solve the problem of automated learning, but also aid in improving the systematic authentication of such applications.

5.3.2. Need for Standardized Nonfunctional Requirements. RE plays a key role in any software project. Specifically, the role of NFRs is considered significant and critical to the ultimate success of any type of software projects [79]; nonetheless, till now the RE community (both researchers and practitioner) could not develop a consensus on what actually nonfunctional requirements (NFRs) are and how effectively can they be classified and represented [79]. Subsequently, there exists a plethora of varied terms (e.g., qualities, attributes, properties, characteristics, constraints, and performance) used to define the nonfunctional requirements, ultimately causing not only terminologies, but also vast conceptual differences [79].

The variety in the definitions of the NFRs ultimately causes different classification of the NFRs. Our SLR results show that only few of the selected studies (e.g., S04 and S08) used the same NFRs categorization while S05 used a different category. Thus, there is a need for standard definition and framework of NFRs which will ultimately assist future researchers to do more consistent experimentation.

Along with the problems of the standardized definition and categorization or classification of the NFRs, the third challenge is the proper NFRs representation: till now there is no standard agreement on how uniformly the NFRs could possibly be described and particularly what precise level of detail is desired [79]. One possible effect is that an NFR might turn into a functional requirement, dependent upon how you describe or express it. For instance, the software requirements example stated by Glinz [79], "the probability for successful access to the customer data by an unauthorized person shall be smaller than 10–5," is actually an NFR. Nevertheless, if we further polish this software requirement to "the database shall grant access to the customer data only to those users that have been authorized by their user name and password," it eventually converts to a functional requirement.

These three aforementioned issues make the different machine learning algorithms quite problematic, as they could possibly be merely trained and utilized for particular NFRs with particular terminologies and categorization. They specifically could not be generalized or scaled up. These issues ultimately make the automated categorization/classification of the software requirements quite difficult and easily disposed to errors. Moreover, these sorts of issues create difficulties for comparing and evaluating the performance of similar methods or techniques and to set up some performance milestones.

5.3.3. Finding and Selecting Useful Features. Mining and extracting precisely the valuable features from the SO posts is a challenging task. The ML approaches utilized by the selected research studies do not necessarily offer significant features. In the USL ML approaches (LDA), the features were either too abstract or worthless. For instance, the abstract features adapted by S06 are, for example, 'failed' and 'fails' for reliability and the features adopted by S08 are, for example, 'user' and 'friend' for usability. Besides, in S06 the "node" and "code" can be deemed as worthless features for the category of the usability. On the other hand, the SL approaches too revealed considerable performance; however, no clear examples of insufficient or inappropriate features were reported which can be discussed here.

It is obvious that worthless features could considerably cause the surge in the number of the false positives and will ultimately lead to the poor performance of the classifiers. Nevertheless, this also hints that the classifiers can be limited in effectively recognizing and extracting NFRs from the domain for which they are trained leading to creating the overfitting problem. We also think that the SL approaches might not be enough capable of bing effectively useful with a diverse style of writing in the similar domains.

Besides, the text of SO posts is considerably in various lengths (shorter or longer) and composition (code written inside texts) compared to the sentences utilized in routine language. These likely cause the lower cooccurrences of the words and the sparse features. Thus, we think that identifying and extracting precisely the meaningful features for NFRs is still an open challenging issue which needs more investigations in the future.

5.3.4. Need for Sentiment Analysis. The selected studies of our SLR study revealed that all the ML approaches (both the SL and USL) and other techniques did not perform any sentiment analysis. For instance, they just found the list of requirements (FRs/NFRs) from the SO posts and did not further analyze the extracted requirements. More specifically, they just treated all the extracted requirements as "black box" and none of the selected studies further classified whether the extracted requirements are feature requests, bug reports, praise, and positive or negative, among others. Thus, one possible way of addressing and performing the sentiment analysis would be adopting the work of [80, 81] for Q&A posts. Nevertheless, another challenging issue that needs to be addressed is investigating the suitability of adopting different sentiment analysis techniques as reported in [82–85] for our domain.

5.3.5. Need for Applying Advanced ML Algorithms. In our SLR study, there were mainly two ML algorithms used LDA (USL) and SVM (SL) for identifying requirements in SO posts. There is a lack of applying not only other SL and SSL algorithms but also advanced ML algorithms. The results and performance of both of these ML algorithms were good enough; however, keeping in view the importance and structure of the text and volume of the data and applying some advanced algorithms (i.e., deep learning) are worth investigating in the future.

6. Validity Threats

In this study, we have considered the five most frequently reported validity threats of software engineering research [60, 86]. These validity threats include descriptive validity, interpretive validity, theoretical validity, generalizability, and reliability. In the subsequent sections, each of these validity threats is discussed in detail.

6.1. Descriptive Validity. This type of validity is associated with the threats to a person's ability to apprehend and document the observations in an objective and accurate way. In SLR, the most significant activity related to this kind of threat is the precise extraction of data since the researchers' biasness may affect the type and quantity of the extracted data. Besides, there might be variations in the interpretation, meaning, and description of the extracted data among researchers. To minimize this threat, the authors arranged work sessions using examples to realize a uniform methodology for determining how and what data need to be extracted. To accomplish this aim, we designed a form called DEF, which was unanimously agreed upon by all the authors. To ensure uniformity and traceability and ultimately reduce the researcher's bias, every entry in the DEF was commented that linked the value assigned by the researcher to a specific text in the original source. Finally, two of the researchers independently reviewed every piece of data extracted, and agreement sessions were arranged whenever there is a discrepancy in the outputs or situations when some descriptive problems arose.

6.2. Interpretive Validity. This kind of validity threats is also called conclusions validity, as they might occur while figuring out any conclusions. Nevertheless, again, the main threat is the researchers' bias when comprehending the extracted data.

To minimize this threat, we applied two different mechanisms. First, regular sessions were arranged after the data extraction, to make sure that all the authors agreed on the correct interpretation of the results, a set of coding rules, and their effects. Secondly, four of the authors in two independent teams acquired the conclusions from the results. Besides, the first author steered, matched, and combined the conclusions, harmonizing the writing style. Finally, all of the authors double-checked the conclusions to ensure that they could be traceable to all of the previous results stored in the DEF.

6.3. Theoretical Validity. This is one of the crucial types of validity threats that contain the highest risk. There are many activities that are mostly affected which are searching and selecting primary papers; nonetheless individual researchers' bias during data extraction (descriptive validity) is also a crucial factor. The main problem related to the search process is the inability to discover all the related available evidence. During the selection process, three challenging situations may occur: selection of the irrelevant papers, or exclusion of the relevant papers, or both. Some other threats are publication bias towards positive results and the quality of the selected works, which ultimately have impact on the data extracted and the acquired conclusions.

To reduce the threat to the search process, we developed the automated search strings from the diverse key terms generated from our defined set of the research questions (RQs), keywords from the research publications retrieved by a pilot search, and the list of various synonyms. Besides, we performed an automated search in four separate electronic database sources. The protocol to perform the search process is detailed in Search Strategy (Section 3.2). The details concerning the selection process and the protocol followed to minimize the theoretical validity threats can be found in Research Paper Selection Criteria (Inclusion and Exclusion Criteria) (Section 3.3).

6.4. Generalizability. There are two types of validity threats that need to be considered under this validity category, namely, internal validity threat and external validity threat. Both of these types are related to the possibility of generalizing the results, within the groups known as internal validity or between the groups known as external validity.

Internal generalizability is related to the validity of our results within other groups identifying software requirements on SO. This validity is reliant on the selected primary paper because of the diversity in reported contexts (type of Q&A, number of Q&A, techniques or algorithms used, domain, assessed metrics, and so on) and the low number of primary studies. It is also dependent upon the fact that the quality of the reported information is also low, due to the scarcity of enough details).

External generalizability is primarily related to other external groups or communities. We did not investigate the identification of software requirements in other settings different from the SO. All of the selected studies deal with identification of software requirements on SO. Henceforth, it is impossible to make judgments regarding other contexts.

To lessen the possible internal generalizability threat, we depend upon the objectivity of the data extraction process and form (DEF) and the SLR protocol to assess the results and acquire significant conclusions. Nonetheless, due to the sample size (12 primary studies), the generalization of the results cannot be guaranteed by our study and needs more investigation.

6.5. Reliability. This kind of the validity threat is concerned about the capability of other researchers to replicate the work and to get similar outcomes. It is challenging to

replicate the experiments in the software engineering domain [41]; nevertheless SLR is an objective mechanism that assists the critical facets of this replication. By providing adequate detailed data on the different information sources (e.g., research databases, search engines, and search strings), well defined exclusion and inclusion criteria, an objective DEF, and granting access to all of the data, comprising those obtained in intermediary processes, the probabilities of other researchers to replicate the study increase.

To improve the reliability of our work, we conducted a comprehensive report of the entire process pursued, from the start of the protocol to the final conduction phase. Lastly, to minimize the validity threats during the conduction phase of this work, we described the rubrics used for the self-evaluation, pursuing the detailed procedures defined by Kitchenham et al. [25, 26].

7. Conclusions

This research study reports comprehensively the planning, conducting, and execution phases of the SLR work on using ML techniques for recognizing diverse software requirements on the SO online Q&A platform. The SLR study was primarily based on the identification, analysis, and classification of 12 selected primary studies that were published until May 2020. The authors have thoroughly investigated several RQs pertaining to what machine learning algorithms/techniques have been effectively utilized for diverse software requirements identification/recognition on the SO, assessed the actual working of these approaches/techniques, and eventually identified the performance measures that have been utilized to assess these techniques or approaches. The authors have thoroughly investigated addressing these RQs and provided a systematic comprehension of the identified machine learning based techniques.

One of the conclusive points that come from the SLR outcomes is that properly using machine learning algorithms to extract or identify software requirements on the SO platform is a very challenging job, as it needs suitable techniques to elicit features and to identify and categorize/classify the text. For the SL machine learning approaches, a prelabelled dataset is compulsory as well. On the other side, comparing with the old-fashioned manual classification methods, using machine learning approaches can possibly support in improving the identification and extraction process of the NFRs, as it decreases the human labours and mistakes. This SLR revealed that machine learning algorithms, specifically SVM (SL algorithm), possess great potential in the RE domain, as they proved to have considerably better performance, LDA (USL algorithm) is the most widely used machine learning algorithm, and the precision and recall are amongst the commonly utilized evaluation methods to measure the performance of these machine learning algorithms in the selected research studies.

Our SLR study, being of an empirical research nature, pointed out several avenues for future researchers, in terms of not only tackling the gaps identified, but also developing new and improved machine learning methods. We have thoroughly discussed the suggestions derived from this SLR study in Key Findings (see Section 5.1), Limitations (see Section 5.2), and some of the avenues for future researchers in the open issues (see Section 5.3).

In conclusion, our SLR findings advocate that this area of research is neglected before and believe that our SLR study is the first step in improving the knowledge base, as it briefly reported the outcomes of all the recent developments related to the RE research on SO. Another interesting finding from our SLR revealed that it appears like this domain is passing through a prescientific phase: all the selected studies analyzed in our SLR seldom compared themselves with the existing works. Finally, we conclude that this SLR study calls for the effective collaboration venture between the RE and machine learning researchers or practitioners to tackle the open challenges confronted in the development of the real world machine learning applications to the enormous field of RE.

Data Availability

The data used to support the findings of the study are included within the manuscript.

Conflicts of Interest

All authors declare that they have no conflicts of interest relevant to this article.

Acknowledgments

This work was funded by the National Key R&D Program of China (no. 2017YFB1002101), National Natural Science Foundation of China (no. U1636203), and the Joint Advanced Research Foundation of China Electronics Technology Group Corporation (CETC) (no. 6141B08010102).

References

[1] H. Khan, A. Ahmad, and M. A. Alnuem, "Knowledge management: a solution to requirements understanding in global software engineering," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, pp. 2087–2099, 2012.

[2] H. Khan, A. Ahmad, C. Johansson, and M. A. Alnuem, "Requirements understanding in global software engineering industrial surveys," in *Proceedings of the 2011 International Conference on Computer and Software Modeling (IPCSIT)*, pp. 167–173, Izmir, Turkey, July 2011.

[3] A. Ahmad and H. Khan, "The importance of knowledge management practices in overcoming the global software engineering challenges in requirements understanding," Master thesis research, Blekinge Institute of Technology, Karlskrona, Sweden, 2008.

[4] M. A. Alnuem, A. Ahmad, and H. Khan, "Requirements understanding: a challenge in global software development, industrial surveys in Kingdom of Saudi Arabia," in *Proceedings of the 2012 IEEE 36th Annual Computer Software and Applications Conference (COMPSAC)*, pp. 297–306, Izmir, Turkey, July 2012.

[5] A. Ahmad, C. Feng, M. Tao, A. Yousif, and S. Ge, "Challenges of mobile applications development: initial results," in *Proceedings of the 8th IEEE International Conference on Software*

Engineering and Service Science (ICSESS 2017), pp. 464–469, Beijing, China, November 2017.

[6] A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, and S. Ge, "An empirical study of investigating mobile applications development challenges," *IEEE Access*, vol. 6, pp. 17711–17728, 2018.

[7] J. Dick, E. Hull, and K. Jackson, *Requirements Engineering*, Springer, Berlin, Germany, 2017.

[8] A. Ahmad, "Research on comprehending software requirements on social media," *PhD ComputerScience [amp] Technology Research*, School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China, 2018.

[9] M. Linares-Vásquez, B. Dit, and D. Poshyvanyk, "An exploratory analysis of mobile development issues using stack overflow," in *Proceedings of the 10th Working Conference on Mining Software Repositories*, pp. 93–96, San Francisco, CA, USA, May 2013.

[10] E. C. Groen, S. Kopczýnska, M. P. Hauer, T. D. Krafft, and J. Doerr, "Users – the hidden software product quality experts? A study on how app users report quality Aspects in online reviews," in *Proceedings of the IEEE 25th International Requirements Engineering Conference*, pp. 80–89, Lisbon, Portugal, September 2017.

[11] C. Li, L. Huang, J. Ge, B. Luo, and V. Ng, "Automatically classifying user requests in crowdsourcing requirements engineering," *Journal of Systems and Software*, vol. 138, pp. 108–123, 2018.

[12] Z. Kurtanovic, W. Maalej, and Ieee, "Automatically classifying functional and non-functional requirements using supervised machine learning," in *Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference*, pp. 490–495, Lisbon, Portugal, September 2017.

[13] A. Ahmad, C. Feng, S. Ge, and A. Yousif, "A survey on mining stack overflow: question and answering (Q&A) community," *Data Technologies and Applications*, vol. 52, no. 2, pp. 190–247, 2018.

[14] Z. S. H. Abad, O. Karras, P. Ghazi, M. Glinz, G. Ruhe, and K. Schneider, "What works better? a study of classifying requirements," in *Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 496–501, Lisbon, Portugal, September 2017.

[15] N. A. Ernst and J. Mylopoulos, "On the perception of software quality requirements during the project lifecycle," in *Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp. 143–157, Essen, Germany, June 2010.

[16] E. Parra, C. Dimou, J. Llorens, V. Moreno, and A. Fraga, "A methodology for the classification of quality of requirements using machine learning techniques," *Information and Software Technology*, vol. 67, pp. 180–195, 2015.

[17] F. Petcuşin, L. Stănescu, and C. Bădică, "An experiment on automated requirements mapping using deep learning methods," in *Proceedings of the International Symposium on Intelligent and Distributed Computing*, pp. 86–95, Saint-Petersburg, Russia, October 2019.

[18] J. Winkler and A. Vogelsang, "Automatic classification of requirements based on convolutional neural networks," in *Proceedings of the 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, pp. 39–45, Beijing, China, September 2016.

[19] F. Dalpiaz, D. Dell'Anna, F. B. Aydemir, and S. Çevikol, "Requirements classification with interpretable machine learning and dependency parsing," in *Proceedings of the 2019 IEEE 27th International Requirements Engineering Conference (RE)*, pp. 142–152, Jeju Island, Korea, September 2019.

[20] Y. Ko, S. Park, J. Seo, and S. Choi, "Using classification techniques for informal requirements in the requirements analysis-supporting system," *Information and Software Technology*, vol. 49, no. 11-12, pp. 1128–1140, 2007.

[21] M. Galster, F. Gilson, and F. Georis, "What quality attributes can we find in product backlogs? A machine learning perspective," in *Proceedings of the European Conference on Software Architecture*, pp. 88–96, Paris, France, September 2019.

[22] A. Ahmad, C. Feng, K. Li, S. M. Asim, and T. Sun, "Toward empirically investigating non-functional requirements of iOS developers on stack overflow," *IEEE Access*, vol. 7, pp. 61145–61169, 2019.

[23] A. Ahmad, L. Kan, C. Feng, and T. Sun, "An empirical study on how iOS developers report quality Aspects on stack overflow," *International Journal of Machine Learning and Computing (IJMLC)*, vol. 8, pp. 501–506, 2018.

[24] Z. S. H. Abad and G. Ruhe, "Using real options to manage technical debt in requirements engineering," in *Proceedings of the 2015 IEEE 23rd International Requirements Engineering Conference (RE)*, pp. 230–235, Ottawa, Canada, August 2015.

[25] B. Kitchenham, *Procedures for Performing Systematic Reviews*, vol. 33, pp. 1–26, Keele University, Keele, UK, 2004.

[26] B. Kitchenham and S. Charters, *Guidelines for Performing Systematic Literature Reviews in Software Engineering*, Keele University and Durham University, Keele, UK, 2007.

[27] C. Pacheco, I. García, and M. Reyes, "Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques," *IET Software*, vol. 12, no. 4, pp. 365–378, 2018.

[28] V. Anu, W. Hu, J. C. Carver, G. S. Walia, and G. Bradshaw, "Development of a human error taxonomy for software requirements: a systematic literature review," *Information and Software Technology*, vol. 103, pp. 112–124, 2018.

[29] D. Dermeval, J. Vilela, I. I. Bittencourt et al., "Applications of ontologies in requirements engineering: a systematic review of the literature," *Requirements Engineering*, vol. 21, pp. 405–437, 2016.

[30] S. Jayatilleke and R. Lai, "A systematic review of requirements change management," *Information and Software Technology*, vol. 93, pp. 163–185, 2018.

[31] A. Baltadzhieva and G. Chrupała, "Question quality in community question answering forums," *ACM SIGKDD Explorations Newsletter*, vol. 17, no. 1, pp. 8–13, 2015.

[32] H. Meth, M. Brhel, and A. Maedche, "The state of the art in automated requirements elicitation," *Information and Software Technology*, vol. 55, no. 10, pp. 1695–1709, 2013.

[33] M. Binkhonain and L. Zhao, "A review of machine learning algorithms for identification and classification of non-functional requirements," *Expert Systems with Applications*, vol. 1, 2019.

[34] T. Iqbal, P. Elahidoost, and L. Lúcio, "A bird's eye view on requirements engineering and machine learning," in *Proceedings of the 2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 11–20, Nara, Japan, December 2018.

[35] A. Yousif, Z. Niu, J. K. Tarus, and A. Ahmad, "A survey on sentiment analysis of scientific citations," *Artificial Intelligence Review*, vol. 52, no. 3, pp. 1805–1838, 2019.

[36] H. Ali, M. N. M. Salleh, K. Hussain et al., "A review on data preprocessing methods for class imbalance problem,"

International Journal of Engineering & Technology, vol. 8, pp. 390–397, 2019.

[37] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Information and Software Technology*, vol. 54, no. 1, pp. 41–59, 2012.

[38] D. A. Magües, J. W. Castro, and S. T. Acuna, "HCI usability techniques in agile development," in *Proceedings of the 2016 IEEE International Conference on Automatica (ICA-ACCA)*, pp. 1–7, Curico, Chile, October 2016.

[39] D. A. Magües, J. W. Castro, and S. T. Acuña, "Usability in agile development: a systematic mapping study," in *Proceedings of the 2016 XLII Latin American Computing Conference (CLEI)*, pp. 1–8, Valparaiso, Chile, October 2016.

[40] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. r. Mahrin, "A systematic literature review of software requirements prioritization research," *Information and Software Technology*, vol. 56, no. 6, pp. 568–585, 2014.

[41] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-based software engineering and systematic reviews*, Vol. 4, CRC Press, Boca Raton, FL, USA, 2015.

[42] S. Beecham, D. Bowes, and K.-J. Stol, *Introduction to the EASE 2016 Special Section: Evidence-Based Software Engineering: Past, Present, and Future*, Elsevier, Amsterdam, Netherlands, 2017.

[43] T. Dyba, B. A. Kitchenham, and M. Jorgensen, "Evidence-based software engineering for practitioners," *IEEE Software*, vol. 22, no. 1, pp. 58–65, 2005.

[44] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: an update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[45] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, vol. 12, pp. 1–10, Swindon, UK, June 2008.

[46] J. L. Barros-Justo, F. B. V. Benitti, and S. Tiwari, "The impact of Use Cases in real-world software development projects: a systematic mapping study," *Computer Standards & Interfaces*, vol. 66, p. 103362, 2019.

[47] A. Ahmad, C. Feng, A. Tahir et al., "An empirical evaluation of machine learning algorithms for identifying software requirements on Stack Overflow: initial Results," in *Proceedings of the 10th IEEE International Conference on Software Engineering and Service Science (ICSESS 2019)*, Beijing, China, October 2019.

[48] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, p. 38, London, UK, May 2014.

[49] B. A. Kitchenham, Z. Li, and A. J. Burn, "Validating search processes in systematic literature reviews," in *Proceedings of the 1st International Workshop on Evidential Assessment of Software Technologies (EAST-2011)*, pp. 3–9, Beijing, China, 2011.

[50] J. Bailey, C. Zhang, D. Budgen, M. Turner, and S. Charters, "Search engine overlaps: do they agree or disagree?" in *Proceedings of the Second International Workshop on Realising Evidence-Based Software Engineering*, p. 2, Minneapolis, MN, USA, May 2007.

[51] N. B. Ali and M. Usman, "Reliability of search in systematic reviews: towards a quality assessment framework for the automated-search strategy," *Information and Software Technology*, vol. 99, pp. 133–147, 2018.

[52] L. Chen, M. Ali Babar, and H. Zhang, "Towards an evidence-based understanding of electronic data sources," in *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering*, London, UK, April 2010.

[53] M. Turner, *Digital Libraries and Search Engines for Software Engineering Research: An Overview*, Keele University, Keele, UK, 2010.

[54] D. Badampudi, C. Wohlin, and K. Petersen, "Experiences from using snowballing and database searches in systematic literature studies," in *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, p. 17, Nanjing, China, April 2015.

[55] N. B. Ali and K. Petersen, "Evaluating strategies for study selection in systematic literature studies," in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 45, Torino, Italy, September 2014.

[56] K. Petersen and N. B. Ali, "Identifying strategies for study selection in systematic reviews and maps," in *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement*, pp. 351–354, Banff, Canada, September 2011.

[57] M. Niazi, S. Mahmood, M. Alshayeb et al., "Challenges of project management in global software development: a client-vendor analysis," *Information and Software Technology*, vol. 80, pp. 1–19, 2016.

[58] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: a systematic review," *Information and Software Technology*, vol. 50, no. 9-10, pp. 833–859, 2008.

[59] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard et al., "Preliminary guidelines for empirical research in software engineering," *IEEE Transactions on Software Engineering*, vol. 28, no. 8, pp. 721–734, 2002.

[60] A. Ampatzoglou, S. Bibi, P. Avgeriou, M. Ver-beek, and A. Chatzigeorgiou, "Identifying, categorizing and mitigating threats to validity in software engineering secondary studies," *Information and Software Technology*, vol. 106, pp. 201–230, 2018.

[61] D. Budgen, P. Brereton, S. Drummond, and N. Williams, "Reporting systematic reviews: some lessons from a tertiary study," *Information and Software Technology*, vol. 95, pp. 62–74, 2018.

[62] M. Kuhrmann, D. M. Fernández, and M. Daneva, "On the pragmatic design of literature studies in software engineering: an experience-based guideline," *Empirical Software Engineering*, vol. 22, no. 6, pp. 2852–2891, 2017.

[63] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised Learning*, MIT Press, Cambridge, MA, USA, 2006.

[64] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[65] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[66] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," in *Measures of Complexity*, pp. 11–30, Springer, Berlin, Germany, 2015.

[67] R. Afridi, Z. Iqbal, M. Khan, A. Ahmad, and R. Naseem, "Fetal heart rate classification and comparative analysis using cardiotocography data and KNOWN classifiers," *International Journal of Grid and Distributed Computing (IJGDC)*, vol. 12, pp. 31–42, 2019.

[68] J. Fereday and E. Muir-Cochrane, "Demonstrating rigor using thematic analysis: a hybrid approach of inductive and deductive coding and theme development," *International Journal of Qualitative Methods*, vol. 5, no. 1, pp. 80–92, 2006.

[69] G. A. Bowen, "Document analysis as a qualitative research method," *Qualitative Research Journal*, vol. 9, no. 2, pp. 27–40, 2009.

[70] V. Braun and V. Clarke, *Thematic Analysis*, Springer, Berlin, Germany, 2012.

[71] A. Khan, B. Baharudin, L. H. Lee, and K. Khan, "A review of machine learning algorithms for text-documents classification," *Journal of Advances in Information Technology*, vol. 1, pp. 4–20, 2010.

[72] A. G. Jivani, "A comparative study of stemming algorithms," *International Journal of Computer Applications in Technology*, vol. 2, pp. 1930–1938, 2011.

[73] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "Automated classification of non-functional requirements," *Requirements Engineering*, vol. 12, no. 2, pp. 103–120, 2007.

[74] A. Casamayor, D. Godoy, and M. Campo, "Identification of non-functional requirements in textual specifications: a semi-supervised learning approach," *Information and Software Technology*, vol. 52, no. 4, pp. 436–445, 2010.

[75] W. Zhang, Y. Yang, Q. Wang, and F. Shu, "An empirical study on classification of non-functional requirements," in *Proceedings of the Twenty-Third International Conference on Software Engineering and Knowledge Engineering (SEKE 2011)*, pp. 190–195, Miami Beach, FL, USA, July 2011.

[76] M. Riaz, J. King, J. Slankas, and L. Williams, "Hidden in plain sight: automatically identifying security requirements from natural language artifacts," in *Proceedings of the 2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pp. 183–192, Karlskrona, Sweden, August 2014.

[77] D. Mladenić, J. Brank, M. Grobelnik, and N. Milic-Frayling, "Feature selection using linear classifier weights: interaction with classification models," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 234–241, Sheffield, UK, July 2004.

[78] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei, "Reading tea leaves: how humans interpret topic models," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 288–296, Vancouver, BC, Canada, December 2009.

[79] M. Glinz, "On non-functional requirements," in *Proceedings of the 15th IEEE International Requirements Engineering Conference (RE 2007)*, pp. 21–26, Delhi, India, October 2007.

[80] W. Maalej, Z. Kurtanović, H. Nabil, and C. Stanik, "On the automatic classification of app reviews," *Requirements Engineering*, vol. 21, no. 3, pp. 311–331, 2016.

[81] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? on automatically classifying app reviews," in *Proceedings of the 2015 IEEE 23rd international requirements engineering conference (RE)*, pp. 116–125, Ottawa, Canada, August 2015.

[82] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto, "Sentiment analysis for software engineering: how far can we go?" in *Proceedings of the 2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pp. 94–104, Gothenburg, Sweden, May 2018.

[83] D. M. E.-D. M. Hussein, "A survey on sentiment analysis challenges," *Journal of King Saud University - Engineering Sciences*, vol. 30, no. 4, pp. 330–338, 2018.

[84] N. Novielli, D. Girardi, and F. Lanubile, "A benchmark study on sentiment analysis for software engineering research," in *Proceedings of the 2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*, pp. 364–375, Gothenburg, Sweden, May 2018.

[85] H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10760–10773, 2009.

[86] K. Petersen and C. Gencel, "Worldviews, research methods, and their relationship to validity in empirical software engineering research," in *Proceedings of the 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement*, pp. 81–89, Ankara, Turkey, October 2013.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 3701067, 11 pages
https://doi.org/10.1155/2020/3701067

Research Article

Deep Learning-Based Cryptanalysis of Lightweight Block Ciphers

Jaewoo So

Department of Electronic Engineering, Sogang University, Seoul 04107, Republic of Korea

Correspondence should be addressed to Jaewoo So; jwso@sogang.ac.kr

Received 5 February 2020; Revised 21 June 2020; Accepted 26 June 2020; Published 13 July 2020

Academic Editor: Umar M. Khokhar

Most of the traditional cryptanalytic technologies often require a great amount of time, known plaintexts, and memory. This paper proposes a generic cryptanalysis model based on deep learning (DL), where the model tries to find the key of block ciphers from known plaintext-ciphertext pairs. We show the feasibility of the DL-based cryptanalysis by attacking on lightweight block ciphers such as simplified DES, Simon, and Speck. The results show that the DL-based cryptanalysis can successfully recover the key bits when the keyspace is restricted to 64 ASCII characters. The traditional cryptanalysis is generally performed without the keyspace restriction, but only reduced-round variants of Simon and Speck are successfully attacked. Although a text-based key is applied, the proposed DL-based cryptanalysis can successfully break the full rounds of Simon32/64 and Speck32/64. The results indicate that the DL technology can be a useful tool for the cryptanalysis of block ciphers when the keyspace is restricted.

1. Introduction

Cryptanalysis of block ciphers has persistently received great attention. In particular, recently, many cryptanalytic techniques have emerged. The cryptanalysis based on the algorithm of algebraic structures can be categorized as follows: a differential cryptanalysis, a linear cryptanalysis, a differential-linear cryptanalysis, a meet-in-the-middle (MITM) attack, and a related-key attack [1, 2]. Differential cryptanalysis, which is the first general cryptanalytic technique, analyses how differences evolve during encryption and how differences of plaintext pairs evolve to differences of the resultant ciphertext pairs [3]. The differential cryptanalysis has evolved to various types of differential cryptanalysis such as an integral cryptanalysis, which is sometimes known as a multiset attack, a boomerang attack, an impossible differential cryptanalysis, and an improbable differential cryptanalysis [1, 2]. Linear cryptanalysis is also a general cryptanalytic technique, where it analyses linear approximations between plaintexts bits, ciphertexts bits, and key bits. It is a known plaintext attack. The work in [4] showed that the efficiency of the linear cryptanalysis can be improved by use of chosen plaintexts. The authors in [5] proposed a zero-correlation linear cryptanalysis, which is a key recovery technique. The MITM attack, which employs a space-time tradeoff, is a generic attack which weakens the security benefits of using multiple encryptions [6]. The biclique attack, which is a variant of the MITM attack, utilizes a biclique structure to extend the number of possibly attacked rounds by the MITM attack [6]. In a related-key attack, an attacker can observe the operation of a cipher under several different keys whose values are initially unknown, but where some mathematical relationship connecting the keys is known to the attacker [7].

However, the conventional cryptanalysis might be impractical or have limitations to be generalized. First, most of conventional cryptanalytic technologies often require a great amount of time, known plaintexts, and memory. Second, although the traditional cryptanalysis is generally performed without the keyspace restriction, only reduced-round variants are successfully attacked on recent block ciphers. For example, no successful attack on the full-round Simon or the full-round Speck, which is a family of lightweight block ciphers, is known [8–10]. Third, we need an automated and generalized test tool for checking the safety of various lightweight block ciphers for Internet of Things [11]. There are various automated techniques that can be used to build distinguishers against block ciphers [12–14]. Because resistance against differential cryptanalysis is an important design criterion for modern block ciphers, most designs rely

on finding some upper bound on probability of differential characteristics [12]. The authors in [13] proposed a truncated searching algorithm which identifies differential characteristics as well as high probability differential paths. The authors in [14] applied a mixed integer linear programming (MILP) to search for differential characteristics and linear approximations in ARX ciphers. However, most automated techniques have endeavoured to search for differential characteristics and linear approximations. Hence, the machine learning- (ML-) based cryptanalysis can be a candidate to solve the above problems.

This paper proposes a generic deep learning- (DL-) based cryptanalysis model that finds the key from known plaintext-ciphertext pairs and shows the feasibility of the DL-based cryptanalysis by applying it to lightweight block ciphers. Specifically, we try to utilize deep neural networks (DNNs) to find the key from known plaintexts. The contribution of this paper is two-fold: first, we develop a generic and automated cryptanalysis model based on the DL. The proposed DL-based cryptanalysis is a promising step towards a more efficient and automated test for checking the safety of emerging lightweight block ciphers. Second, we perform the DL-based attacks on lightweight block ciphers, such as S-DES, Simon, and Speck. In our knowledge, this is the first attempt to successfully break the full rounds of Simon32/64 and Speck32/64 although we apply the text-based key for the block ciphers.

The remainder of this paper is organized as follows: Section 2 presents the related work; Section 3 describes the attack model for cryptanalysis; Section 4 introduces the DL-based approach for the cryptanalysis of lightweight block ciphers and presents the structure of the DNN model; Section 5 describes how to learn and evaluate the model; in Section 6, we apply the DL-based cryptanalysis to lightweight block ciphers and evaluate the performance of the DL-based cryptanalysis; finally, Section 7 concludes this paper.

Notations: we give some notations, which will be used in the rest of this paper. A plaintext and ciphertext are, respectively, denoted by $p = (p_0, p_1, \ldots, p_{n-1})$ and $c = (c_0, c_1, \ldots, c_{n-1})$, where n is the block size, p_i is the ith bit of the plaintext, c_i is the ith bit of the ciphertext, and $\mathbf{p}_i$, $\mathbf{c}_i \in \{0, 1\}$. A key is denoted by $k = (k_0, k_1, \ldots, k_{m-1})$, where m is the key length and k_i is the ith bit of the key, $k_i \in \{0, 1\}$. Let $k|_i^j$ denote the key bits from the ith bit to the jth bit of the key, that is, $k|_i^j \triangleq (k_i, k_{i+1}, \ldots, k_j)$. A block cipher is specified by an encryption function, $E(p, k)$, that is, $c = E(p, k)$.

2. Related Work

ML has been successfully applied in a wide range of areas with significant performance improvement, including computer vision, natural language processing, speech, and game [15]. The development of ML technologies provides a new development direction for cryptanalysis [16]. The idea of the relationship between the fields of cryptography and ML is introduced in [17] at 1991. After that, many researchers have endeavoured to apply the ML technologies for the cryptanalysis of block ciphers.

The studies on the ML-based cryptanalysis can be classified as follows: first, some studies focused on finding the characteristics of block ciphers by using ML technologies. The authors in [18] used a recurrent neural network to find the differential characteristics of block ciphers, where the recurrent neural network represents the substitution functions of a block cipher. The author in [19] applied an artificial neural network to automate attacks on the classical ciphers of a Caesar cipher, a Vigenère cipher, and a substitution cipher, by exploiting known statistical weakness. They trained a neural network to recover the key by providing the relative frequencies of ciphertext letters. Recent work [20] experimentally showed that a CipherGAN, which is a tool based on a generative adversarial network (GAN), can crack language data enciphered using shift and Vigenère ciphers.

Second, some studies used ML technologies to classify encrypted traffic or to identify the cryptographic algorithm from ciphertexts. In [21], an ML-based traffic classification was introduced to identify SSH and Skype encrypted traffic. The authors in [22] constructed three ML-based classification protocols to classify encrypted data. They showed the three protocols, hyperplane decision, Naïve Bayes, and decision trees, efficiently perform a classification when running on real medication data sets. The authors in [23] used a support vector machine (SVM) technique to identify five block cryptographic algorithms, AES, Blowfish, 3DES, RC5, and DES, from ciphertexts. The authors in [24] proposed an unsupervised learning cost function for a sequence classifier without labelled data, and they showed how it can be applied in order to break the Caesar cipher.

Third, other researchers have endeavoured to find out the mapping relationship between plaintexts, ciphertexts, and the key, but there are few scientific publications. The work in [25] reported astonishing results for attacking the DES and the Triple DES, where a neural network was used to find the plaintexts from the ciphertexts. The authors in [26] used a neural network to find out the mapping relationship between plaintexts, ciphertexts, and the key in simplified DES (S-DES). The author in [27] developed a feedforward neural network that discovers the plaintext from the ciphertext without the key in the AES cipher. The authors in [28] attacked on the round-reduced Speck32/64 by using deep residual neural networks, where they trained the neural networks to distinguish the output of Speck with a given input difference based on the chosen plaintext attack. The attack in [28] is similar to the classical differential cryptanalysis. However, the previous work failed to attack the full rounds of lightweight block ciphers, and moreover, they failed to develop a generic deep learning- (DL-) based cryptanalysis model.

3. System Model

We consider (n, m) lightweight block ciphers such as S-DES, Simon, and Speck, where n is the block size and m is the key length. Our objective is to find the key, $\mathbf{k}$, in which the attacker has access to M pairs, $[\mathbf{p}^{(i)}, \mathbf{c}^{(j)}]$, of known plaintexts, and their resultant ciphertexts encrypted with the

same key, that is, $\mathbf{c}^{(j)} = E(\mathbf{p}^{(j)}, \mathbf{k})$, $j = 1, 2, \ldots, M$. Hence, the cryptanalytic model is a known plaintext attack model. Because the algorithms of block ciphers have been publicly released, we assume that the algorithms of block ciphers are known.

4. Deep Learning-Based Approach

4.1. DNN Learning Framework. The modern term "DL" is considered as a better principle of learning multiple levels of composition, which uses multiple layers to progressively extract higher level features from the raw input [29]. In the DL area, a DNN is considered as one of the most popular generative models. As a multilayer processor, the DNN is capable of dealing with many nonconvex and nonlinear problems. The feedforward neural network forms a chain, and thus, the feedforward neural network can be expressed as

$$f(\mathbf{x}; \boldsymbol{\theta}) = f^{(L+1)}\left(f^{(L)}\left(\cdots f^{(1)}(\mathbf{x})\right)\right), \tag{1}$$

where $\mathbf{x}$ is the input, the parameter $\boldsymbol{\theta}$ consists of the weights $\mathbf{W}$ and the biases $\mathbf{b}$, $f^{(l)}$ is called the lth layer of the network, and L is the number of hidden layers. Each layer of the network consists of multiple neurons, each of which has an output that is a nonlinear function of a weighted sum of neurons of its preceding layer. The output of the jth neuron at the lth layer can be expressed as

$$j^{(l)} = f^{(l)}\left(\sum_i w_{ij}^{(l)} u_i^{(l-1)} + b_j^{(l)}\right), \tag{2}$$

where $w_{ij}^{(l)}$ is the weight corresponding to the output of the ith neuron at the preceding layer and $b_j^{(l)}$ is the bias. We apply a DNN to find the key of lightweight block ciphers. The multilayer perception mechanism and special training policy promote the DNN to be a commendable tool to find affine approximations to the action of a cipher algorithm. We train the DNN by using N_r pairs of $(\mathbf{p}, \mathbf{c})$ randomly generated with different keys in order that the system f finds affine approximations to the action of a cipher, as shown in Figure 1. In Figure 1, the loss function can be the mean square error (MSE) between the encryption key, $\mathbf{k}$, and the output of the DNN, $\widehat{\mathbf{k}}$. The performance of the trained DNN is evaluated by using N_t pairs randomly generated with different keys. Finally, given M known plaintexts, we find the key by using the trained DNN and the majority decision.

4.2. DNN Structure for the Cryptanalysis. The structure of a DNN model for the cryptanalysis is shown in Figure 2. We consider a ReLU function, $f_{\mathrm{ReLU}}(x) = \max(0, x)$, as the nonlinear function. The DNN has η_l neurons at the lth hidden layer, where $l = 1, \ldots, L$. Each neuron at the input layer associates each bit of the plaintext and ciphertext; that is, the ith neuron represents $\mathbf{p}_i$, and the $(j + n - 1)$th neuron represents $\mathbf{c}_j$, where $i, j = 0, 1, \ldots, n - 1$. The number of neurons at the input layer is $2n$. Each neuron at the output layer associates each bit of the key; that is, the output of the ith neuron corresponds to $\mathbf{k}_i$, where $i = 0, 1, \ldots, m - 1$. Hence, the number of neurons at the output layer is m. The output of the DNN, $\widehat{\mathbf{k}}$, is a cascade of nonlinear transformation of the input data, $[\mathbf{p}, \mathbf{c}]$, mathematically expressed as

$$\widehat{\mathbf{k}} = f([\mathbf{p}, \mathbf{c}]; \boldsymbol{\theta}) = f^{(L+1)}\left(f^{(L)}\left(\cdots f^{(1)}([\mathbf{p}, \mathbf{c}])\right)\right), \tag{3}$$

where L is the number of hidden layers and $\boldsymbol{\theta}$ is the weights of the DNN.

5. Model Training and Testing

5.1. Data Generation. The ML algorithm learns from data. Hence, we need to generate data set for training and testing the DNN. Because the algorithms of modern block ciphers are publicly released, we can generate N plaintext-ciphertext pairs with different keys, where $N = N_r + N_s$, N_r is used for training the DNN, and N_s is used for testing the DNN. Let the jth sample represent $[\mathbf{p}^{(j)}, \mathbf{c}^{(j)}; \mathbf{k}^{(j)}]$, $j = 1, 2, \ldots, N$, as shown in Figure 3, where $\mathbf{c}^{(j)} = E(\mathbf{p}^{(j)}, \mathbf{k}^{(j)})$ for $i \neq j$, $\mathbf{p}^{(i)} \neq \mathbf{p}^{(j)}$, and $\mathbf{k}^{(i)} \neq \mathbf{k}^{(j)}$.

5.2. Training Phase. The goal of our model is to minimize the difference between the output of the DNN and the key. Let $\mathbf{X}$ represent the training plaintext-ciphertext pairs $[\mathbf{p}^{(j)}, \mathbf{c}^{(j)}]$, and let $\mathbf{K}$ represent the training keys $\mathbf{k}^{(j)}$ corresponding to the jth pair $[\mathbf{p}^{(j)}, \mathbf{c}^{(j)}]$, where $1 \leq j \leq N_r$.

The DNN learns the value of the parameter $\boldsymbol{\theta}$ that minimizes the loss function, from the training samples, as follows:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} L(f(\mathbf{X}; \boldsymbol{\theta}), \mathbf{K}), \tag{4}$$

where because the samples are i.i.d., the MSE loss function can be expressed as follows:

$$\mathrm{MSE} = \frac{1}{N_r \cdot m} \sum_{j=1}^{N_r} \sum_{i=0}^{m-1} \left(\mathbf{k}_i^{(j)} - \widehat{\mathbf{k}}_i^{(j)}\right)^2, \tag{5}$$

where N_r is noted as the number of training samples, $\mathbf{k}_i^{(j)}$ is the ith bit of the key corresponding to the jth sample, and $\widehat{\mathbf{k}}_i^{(j)}$ is the ith output of the DNN corresponding to the jth sample.

5.3. Test Phase. After training, the performance of the DNN is evaluated in terms of the bit accuracy probability (BAP) of each key bit. Here, the BAP of the ith key bit is the number of the DNN finding the correct ith key bit, divided by the total number of test samples.

Because the output of the DNN is a real number, $\widehat{\mathbf{k}}_i \in \mathbb{R}$, we quantize the output of the DNN into $\{0, 1\}$. The quantized output of the DNN can then be expressed as

$$\widetilde{\mathbf{k}}_i = \begin{cases} 0, & \text{if } \widehat{\mathbf{k}}_i < 0.5, \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

Then, the BAP of the ith key bit is given as

$$\rho_i = \frac{1}{N_s} \sum_{j=1}^{N_s} \mathrm{XNOR}\left(\mathbf{k}_i^{(j)}, \widetilde{\mathbf{k}}_i^{(j)}\right), \tag{7}$$

where N_s is the number of test samples. $\mathrm{XNOR}(a, b)$ has one if two input values, a and b, are identical, and otherwise, it

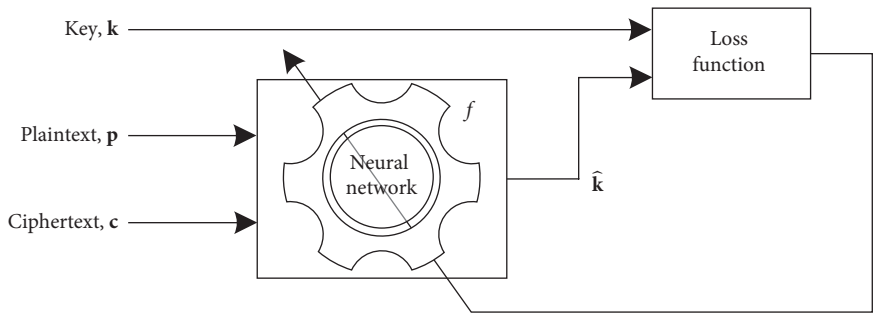


Figure 1: A schematic diagram of the DL-based cryptanalysis.

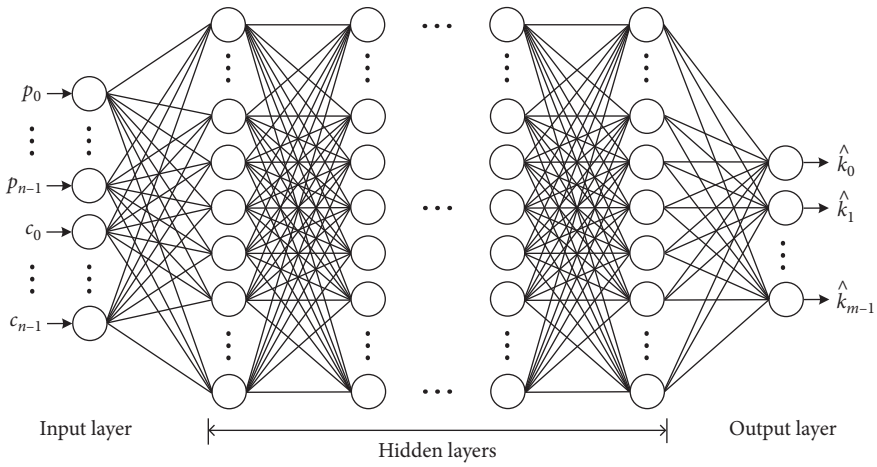


Figure 2: A DNN model.

has zero. $\mathbf{k}_i^{(j)}$ is the ith key bit corresponding to the jth test sample, and $\widetilde{\mathbf{k}}_i^{(j)}$ is the quantized output of the DNN with the input of the jth test sample.

5.4. Majority Decision When M Plaintexts Are Known. Assume that we have M plaintext-ciphertext pairs encrypted with the same key. If we have a probability of finding the ith key bit, ρ_i, then the attack success probability of finding the ith key bit, which is the probability of a correct majority decision, is given as

$$\begin{aligned}\alpha_i(M) &= \Pr\left(X \geq \frac{M}{2} + 1\right) = 1 - \Pr\left(X \leq \frac{M}{2}\right) \\ &= 1 - \sum_{j=0}^{M/2} \binom{M}{j} \rho_i^j (1-\rho_i)^{M-j}.\end{aligned} \tag{8}$$

By using the de Moivre–Laplace theorem, as M grows large, the normal distribution can be used as an approximation to the binomial distribution, as follows:

$$\alpha_i(M) = 1 - \Phi\left(\frac{(M/2) - M\rho_i}{\sqrt{M\rho_i(1-\rho_i)}}\right), \tag{9}$$

where $\Phi(z) = \int_{-\infty}^{z} 1/\sqrt{2\pi} e^{-x^2/2} \mathrm{d}z$. Hence, in order to find the ith key bit with a success probability greater than or equal to τ, the number of required known plaintexts is

$$M_i^* = \min\left\{M \,\middle|\, \alpha_i(M) \geq \tau\right\}. \tag{10}$$

6. Performance Evaluation

6.1. Data Set and Performance Metric. For the data set, we generate the plaintext as any combination of a random binary digit, that is, $\mathbf{p}_i \in \mathrm{rand}\{0, 1\}$. However, for the encryption key, we consider two methods. The first method is a "*random key*," where the key has any combination of a random binary digit, that is, $\mathbf{k}_i \in \mathrm{rand}\{0, 1\}$, $i = 0, 1, \ldots, m-1$. Hence, the probability that the ith key bit is one is 0.5 for all i. The other method is a "*text key*," where the key has any combination of characters. For the simplicity, as shown in Figure 4, the character is one out of 64 ASCII characters, which consists of lowercase and uppercase alphabet characters, 10 digits, and two special characters: $\mathscr{T} = \{a, b, \ldots, z, A, B, \ldots, Z, 0, 1, \ldots, 9, ?, @\}$ and $|\mathscr{T}| = 64$. Hence, in the text key generation, each eight bits belongs to the set of $\mathscr{T}$, that is, $\mathbf{k}|_{8\cdot i}^{8\cdot i+7} \in \mathrm{rand}(\mathscr{T})$, where $i = 0, 1, \ldots, [m/8] - 1$. For example, for a 64-bit key, the key consists of 8 characters. In the text key, the probability that the ith key bit is one depending on the order in each character. Let the occurrence probability denote $\mu_i = \max(\Pr(\mathbf{k}_i = 1), \Pr(\mathbf{k}_i = 0))$, where $\Pr(\mathbf{k}_i = x)$ is the probability that the ith key bit is x. Figure 5 shows the occurrence probability of the ith key bit μ_i. For example, the

$$\begin{bmatrix} p_0^{(1)} & p_1^{(1)} & \cdots & p_{n-1}^{(1)}, & c_0^{(1)} & c_1^{(1)} & \cdots & c_{n-1}^{(1)}; & k_0^{(1)} & k_1^{(1)} & \cdots & k_{m-1}^{(1)} \\ p_0^{(2)} & p_1^{(2)} & \cdots & p_{n-1}^{(2)}, & c_0^{(2)} & c_1^{(2)} & \cdots & c_{n-1}^{(2)}; & k_0^{(2)} & k_1^{(2)} & \cdots & k_{m-1}^{(2)} \\ & & \vdots & & & & \vdots & & & & \vdots & \\ p_0^{(j)} & p_1^{(j)} & \cdots & p_{n-1}^{(j)}, & c_0^{(j)} & c_1^{(j)} & \cdots & c_{n-1}^{(j)}; & k_0^{(j)} & k_1^{(j)} & \cdots & k_{m-1}^{(j)} \\ & & \vdots & & & & \vdots & & & & \vdots & \\ p_0^{(N)} & p_1^{(N)} & \cdots & p_{n-1}^{(N)}, & c_0^{(N)} & c_1^{(N)} & \cdots & c_{n-1}^{(N)}; & k_0^{(N)} & k_1^{(N)} & \cdots & k_{m-1}^{(N)} \end{bmatrix}$$

Figure 3: Data set.

BIN	HEX	Char.
00000000	0	NUL
00000001	1	SOH
00000010	2	STX
00000011	3	ETX
00000100	4	EOT
00000101	5	ENQ
00000110	6	ACK
00000111	7	BEL
00001000	8	BS
00001001	9	HT
00001010	0A	LF
00001011	0B	VT
00001100	0C	FF
00001101	0D	CR
00001110	0E	SO
00001111	0F	SI
00010000	10	DLE
00010001	11	DC1
00010010	12	DC2
00010011	3	DC3
00010100	14	DC4
00010101	15	NAK
00010110	16	SYN
00010111	17	ETB
00011000	18	CAN
00011001	19	EM
00011010	1A	SUB
00011011	1B	ESC
00011100	1C	FS
00011101	1D	GS
00011110	1E	RS
00011111	1F	US
00100000	20	SPACE
00100001	21	!
00100010	22	"
00100011	23	#
00100100	24	$
00100101	25	%
00100110	26	&
00100111	27	'
00101000	28	(
00101001	29	)
00001010	2A	*
00101011	2B	+
00101100	2C	,
00101101	2D	-
00101110	2E	.
00101111	2F	/
00110000	30	0
00000110	31	1
00110010	23	2
00110011	33	3
00110100	34	4
00110101	35	5
00110110	36	6
00001100	37	7
00111000	38	8
00111000	39	9
00111010	3A	:
00111011	3B	;
00111100	3C	<
00111101	3D	=
00111110	3E	>
00111111	3F	?
01000000	40	@
01000001	41	A
01000010	42	B
01000011	43	C
01000100	44	D
01000101	45	E
01000110	46	F
01000111	47C	G
01001000	48	H
01001001	49	I
01001010	4A	J
01001011	4B	K
01001100	4C	L
01001101	4D	M
01001110	4E	N
01001111	4F	O
01010000	50	P
01010001	51	Q
01010010	52	R
01010011	53	S
01010100	54	)
01010101	55	T
01010110	56	V
01010111	57	W
01011000	58	X
01011001	59	Y
01011010	5A	Z
01011011	5B	[
01011100	5C	₩
01011101	5D	]
01011110	5E	^
01011111	5F	–
01100000	60	`
01100001	61	a
01100010	62	b
01100011	63	c
01100100	64	d
01100101	65	e
01100110	66	f
01100111	67	g
01101000	68	h
01101001	69	i
01101010	6A	j
01101011	6B	k
01101100	6C	l
01101101	6D	m
01101110	6E	n
01101111	6F	o
01110000	70	p
01110001	71	q
01110010	72	r
01110011	73	s
01110100	74	t
01110101	75	u
01110110	76	v
01110111	77	w
01111000	78	*x*
00100011	79	y
01111010	7A	z
01111011	7B	{
01111100	7C	|
01111101	7D	}
01111110	7E	~
01111111	7F	DEL

Figure 4: Characters used in the text key generation.

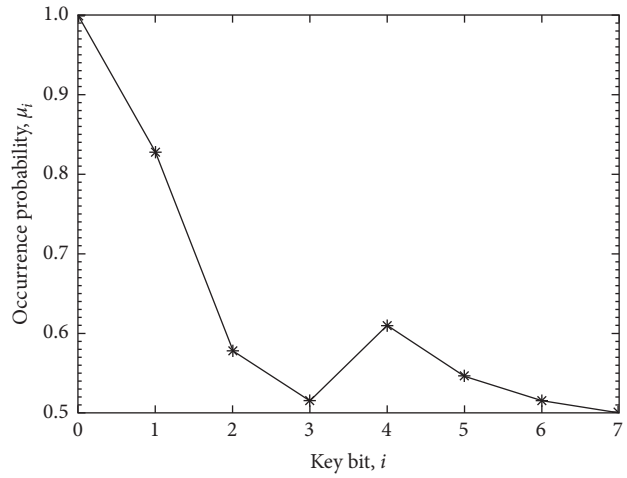


Figure 5: Occurrence probability in the text key generation.

first bit of the key character is always 0, and the second bit is one with the probability of 0.828.

Taking the occurrence probability of each key bit into consideration, the performance of finding the ith key bit can be expressed as the deviation as follows:

$$\varepsilon_i = \rho_i - \mu_i, \tag{11}$$

where ρ_i is the BAP and μ_i is the occurrence probability of the ith key bit. If M known plaintexts is given, the performance of finding the ith key bit is given by $\varepsilon_i(M) = \alpha_i(M) - \mu_i$, where $\alpha_i(M)$, which is the probability of a correct majority decision, is obtained from equation (9).

6.2. Simulation Environment. The performance of the DL-based cryptanalysis is evaluated for the lightweight block ciphers: S-DES, Simon32/64, and Speck32/64, as shown in Table 1.

In order to train the DNN with an acceptable loss rate, it is necessary to expand the network size. Hyperparameters, such as the number of hidden layers, the number of neurons per hidden layer, and the number of epochs, should be tuned in order to minimize a predefined loss function. The traditional way of performing hyperparameter optimization has been grid search or random search. Other hyperparameter optimizations are Bayesian optimization, gradient-based optimization, evolutionary optimization, and population-based training [30, 31]. Moreover, automated ML (AutoML) has been proposed to design and train neural networks automatically [30]. In our simulation, by using the data set of Simon32/64 and Speck32/64 ciphers, we simply perform an exhaustive searching to set the number of hidden layers, L, and the number of neurons per hidden layer, η_l, through a manually specified subset of the hyperparameter space, $L \in \{3, 5, 7\}$ and $\eta_l \in \{128, 256, 512\}$. Additionally, to reduce the complexity, we choose a smaller number of hidden layers if the performance difference is not greater than 10^{-5}. If the number of epochs is greater than 3000, the error becomes small, and when it reaches 5000, it is sufficiently minimized, so we set the number of epochs is fixed to 5000. Consequently, the parameters used for training the DNN models are as follows: the number of hidden layers is 5, the number of neurons at each hidden layer is 512, and the number of epochs is 5000. We use the adaptive moment (Adam) algorithm for the learning rate optimization of the DNN.

Table 1: Block ciphers used in case studies.

Item	S-DES	Simon	Speck
Block size (bits), n	8	32	32
Key size (bits), m	10	64	64
Round, R	2	32	22

The powerful "*Tensorflow*" is introduced to design and process the DNN. Also, we deploy a GPU-based server, which is equipped with Nvidia GeForce RTX 2080 Ti and its CPU is Intel Core i9-9900K. The implemented DL-based cryptanalysis tool is shown in Figure 6. The GUI was implemented by using PyQt over Python 3.7. The implemented tool provides various combinations of ML architectures, hyperparameters, and training/test samples.

6.3. Simplified DES

6.3.1. Overview of S-DES. S-DES, designed for education purposes at 1996, has similar properties and structure as DES but has been simplified to make it easier to perform encryption and decryption [32]. The S-DES has an 8-bit block size and a 10-bit key size. The encryption algorithm involves five functions: an initial permutation (IP); a complex function labelled f_K, which involves both permutation and substitution operations and depends on a key input; a simple permutation function that switches the two halves of the data; the function f_K again; and finally a permutation function that is the inverse of the initial permutation (IP^{-1}). S-DES may be said to have two rounds of the function f_K.

Because the length of the key is limited, the brute-force attack, which is known as an exhaustive key search, is available. Some previous work presented an approach for breaking the key using genetic algorithm and particle swarm optimization [33, 34], which is concluded that the genetic algorithm is a better approach than the brute force for analysing S-DES.

6.3.2. Test Results. For training and testing the DNN, we generate N plaintext-ciphertext pairs with different keys, as follows:

$$\mathbf{c}^{(j)} = S\text{-}DES\left(\mathbf{p}^{(j)}, \mathbf{k}^{(j)}\right), \quad j = 0, 1, \ldots, N, \tag{12}$$

where $\mathbf{k}^{(i)} \neq \mathbf{k}^{(j)}$ for $i \neq j$ and $N = N_r + N_s$. Here, N_r is the number of samples for training and N_s is the number of samples for testing. In the simulation, we use $N_r = 50000$ and $N_s = 10000$. The plaintext is any combination of a random binary digit, that is, $p_i \in \mathrm{rand}\{0, 1\}$. We generate the encryption key by using two methods: a random key and a text

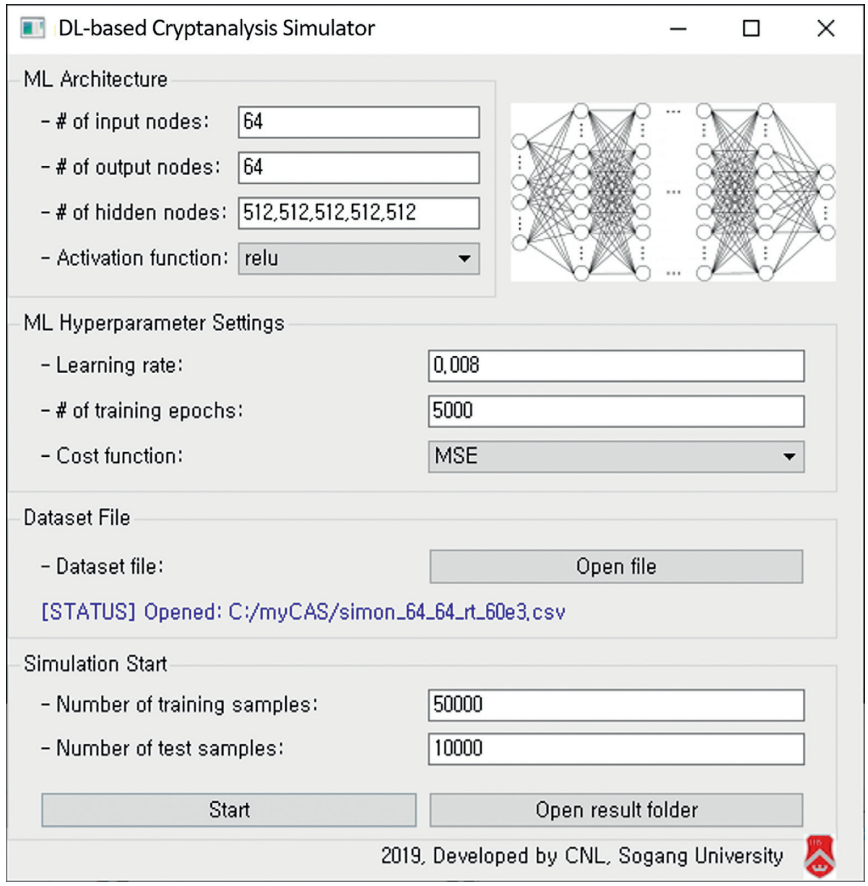


Figure 6: Implemented DL-based cryptanalysis simulator.

key. In the S-DES with a 10-bit key, the text key has any combination of one character and two random binary bits.

Figure 7 shows the BAP of the DNN when we apply a random key and a text key. The results show the DL-based cryptanalysis can break the S-DES cipher. When we apply a random key, the key bits, $\mathbf{k}_1$, $\mathbf{k}_5$, and $\mathbf{k}_8$, are quite vulnerable to the attack and the key bit of $\mathbf{k}_6$ is the safest. Because the minimum value of the BAP is $\rho_{\min} = 0.5389$ at the 6th key bit, from equation (10), we need $M = 271$ known plaintexts to find all the key bits with a probability of 0.9 and we need $M = 891$ known plaintexts to find all the key bits with a probability of 0.99. When we apply a text key, the BAP becomes high, thanks to the bias of the occurrence probability of each key bit, μ_i, as shown in Figure 5. Because the minimum value of the BAP is $\rho_{\min} = 0.6484$ at the 6th key bit, from equation (10), we need $M = 19$ known plaintexts to find all the key bits with a probability of 0.9 and we need $M = 59$ known plaintexts to find all the key bits with a probability of 0.99.

Figure 8 shows the deviation between the BAP and the occurrence probability of each key bit. Because of the bias of the occurrence probability of each key bit in the text key, we need to eliminate the bias characteristics of each key bit. The DNN shows that the key bits, which are quite vulnerable to the attack, are ($\mathbf{k}_2$, $\mathbf{k}_5$, $\mathbf{k}_8$) in the text key and ($\mathbf{k}_1$, $\mathbf{k}_5$, $\mathbf{k}_8$) in the random key. The key bit of $\mathbf{k}_6$ is the safest both in the text key and in the random key.

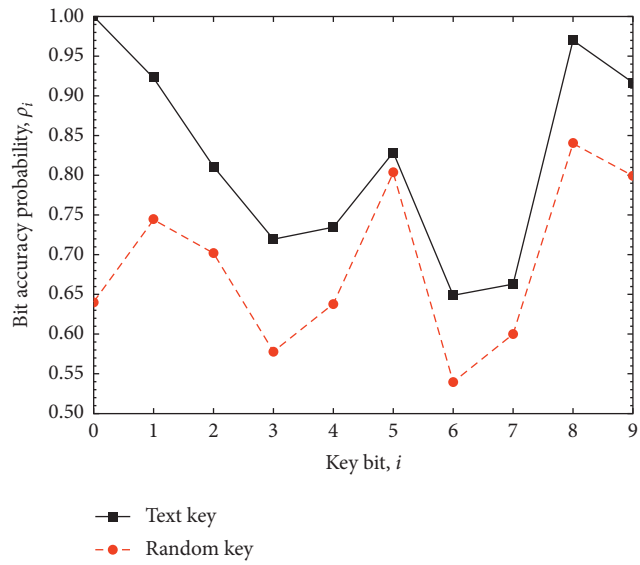


Figure 7: Bit accuracy in the S-DES with a random key.

6.4. Lightweight Block Ciphers

6.4.1. Overview of Simon and Speck. Lightweight cryptography is a rapidly evolving and active area, which is driven by the need to provide security or cryptographic measures to resource-constrained devices such as mobile phones, smart

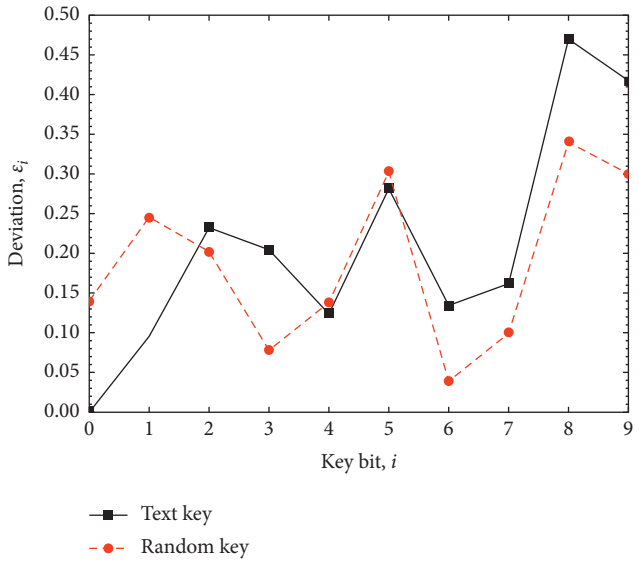


Figure 8: Deviation in the S-DES.

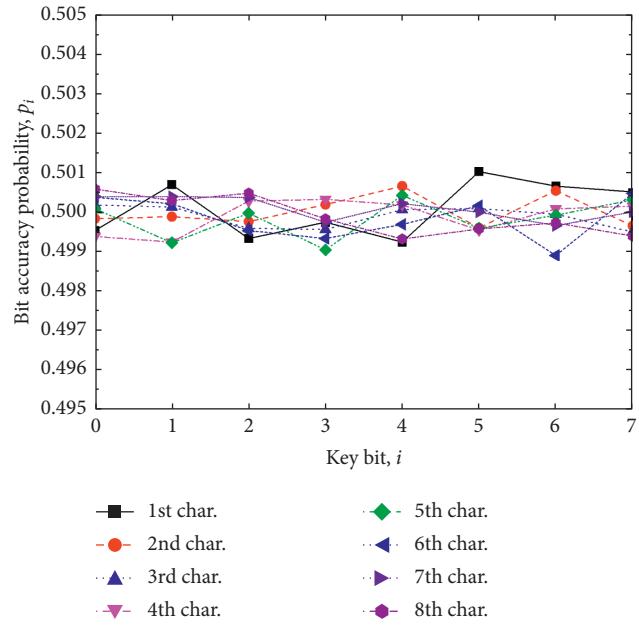


Figure 9: Bit accuracy probability of the Simon32/64 with a random key.

cards, RFID tags, and sensor networks. Simon and Speck is a family of lightweight block ciphers publicly released in 2013 [35, 36]. Simon has been optimized for performance in hardware implementations, while Speck has been optimized for software implementations. The Simon block cipher is a balanced Feistel cipher with a u-bit word, and therefore, the block length is $n = 2u$. The key length, m, is a multiple of u by 2, 3, or 4. Simon supports various combinations of block sizes, key sizes, and number of rounds [35]. In this paper, we consider a Simon32/64 which refers to the cipher operating on a 32-bit plaintext block that uses a 64-bit key. The Speck is an add-rotate-xor (ARX) cipher. The block of the Speck is always two words, but the words may be 16, 24, 32, 48, or 64 bits in size. The corresponding key is 2, 3, or 4 words. Speck also supports various combinations of block sizes, key sizes, and number of rounds [35].

As of 2018, no successful attack on full-round Simon or full-round Speck of any variant is known. The authors in [37] showed differential attacks of up to slightly more than half of the number of rounds for Simon and Speck families of block ciphers. The authors in [38] showed an integral attack on 24-round Simon32/64 with time complexity of 2^{63} and the data complexity of 2^{32}. The work in [39] showed an improved differential attack on 14-round Speck32/64 with time complexity of 2^{63} and the data complexity of 2^{31}.

6.4.2. Data Generation. For training and testing the DNN, we generate N plaintext-ciphertext pairs with different keys, as follows:

$$\begin{aligned} \mathbf{c}^{(j)} &= \text{Simon}\frac{32}{64}\left(\mathbf{p}^{(j)}, \mathbf{k}^{(j)}\right), \\ \mathbf{c}^{(j)} &= \text{Speck}\frac{32}{64}\left(\mathbf{p}^{(j)}, \mathbf{k}^{(j)}\right), \end{aligned} \tag{13}$$

where $j = 0, 1, \ldots, N$ and $N = N_r + N_s$. Here, N_r is the number of samples for training and N_s is the number of samples for testing. The plaintext is any combination of a random binary digit, that is, $\mathbf{p}_i \in \text{rand}\{0, 1\}$. We generated the encryption key by using two methods: a random key and a text key. In the text key, the 64-bit key consists of 8 characters, where each character is one of 64-character set, $\mathcal{T}$. Hence, although the total keyspace is 2^{64}, the actual keyspace is reduced to 2^{48}. For training, we use $N_r = 5 \times 10^5$ samples, and for the test, we use $N_s = 10^6$ samples.

6.4.3. Test Results. Figure 9 shows the BAP of the Simon32/64 with a random key in unit of character. The DNN shows that the BAP of each key bit varies randomly with an average of almost 0.5. Moreover, the results vary with each simulation with different hyperparameters. That is, the DNN failed to attack the Simon32/64 with a random key.

Figure 10 shows the BAP and the deviation of the Simon32/64 with a text key in unit of character. The BAP of each key bit is almost identical to the occurrence probability of the text key because the DNN learns the characteristics of the training data. However, when we eliminate the bias characteristics of the text key, the DNN shows the positive deviations, which means the DNN can break a Simon32/64 with a text key. For example, from equation (10), we need just $M = 215$ known plaintexts in order to find the key bit of $\mathbf{k}_2$ with a probability of 0.99. The minimum value of BAPs is 0.51603 at $\mathbf{k}_3$, which is greater than μ_3 by about $\varepsilon_3 = 0.00040$, except the last bits of each character. Hence, we can find the encryption key with a probability of 0.9 given $M \approx 2^{10.58}$ known plaintexts, and we can find the encryption key with a probability of 0.99 given $M \approx 2^{12.34}$ known plaintexts.

Figure 11 shows the BAP of the Speck32/64 with a random key in unit of character. The BAP of each key bit varies randomly with an average of almost 0.5, similar to the results of the Simon32/64. Moreover, the results vary with

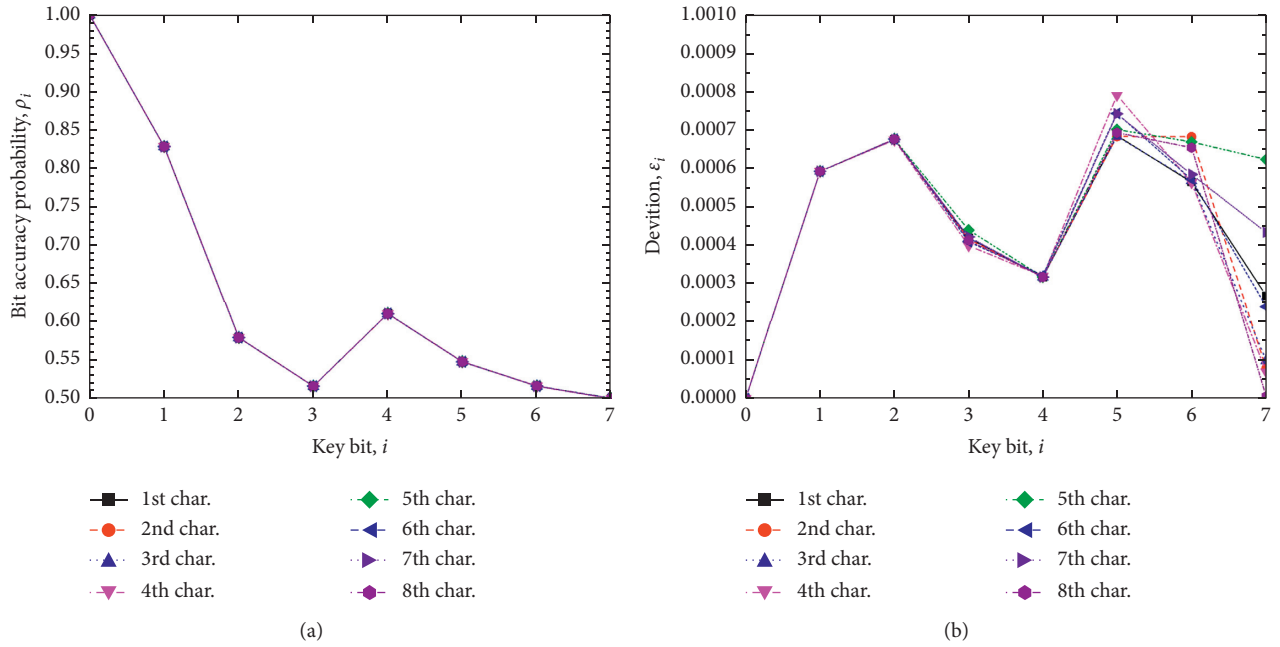


Figure 10: Bit accuracy probability and deviation of the Simon32/64 with a text key.

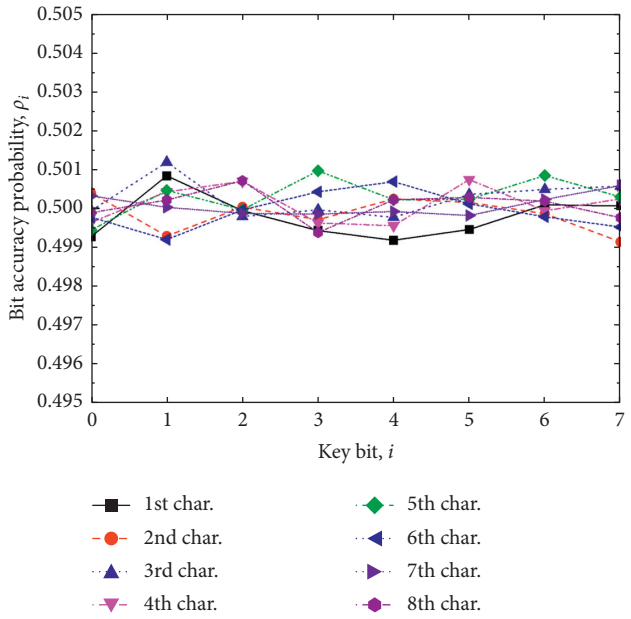


Figure 11: Bit accuracy probability of the Speck32/64 with a random key.

different hyperparameters. That is, the DL-based attacks against the Speck32/64 with a random key have been failed.

Figure 12 shows the BAP and the deviation of the Speck32/64 with a text key in unit of character. The DNN shows the positive deviations. That is, the DNN shows the possibility of breaking a Speck32/64 with a text key. The minimum value of BAPs is 0.51607 at $\mathbf{k}_3$, which is greater than μ_3 by about $\varepsilon_3 = 0.00044$, except the last bits of each character. Hence, we can find the encryption key with a probability of 0.9 given $M \approx 2^{10.57}$ known plaintexts, and we can find the encryption key with a probability of 0.99 given $M \approx 2^{12.33}$ known plaintexts.

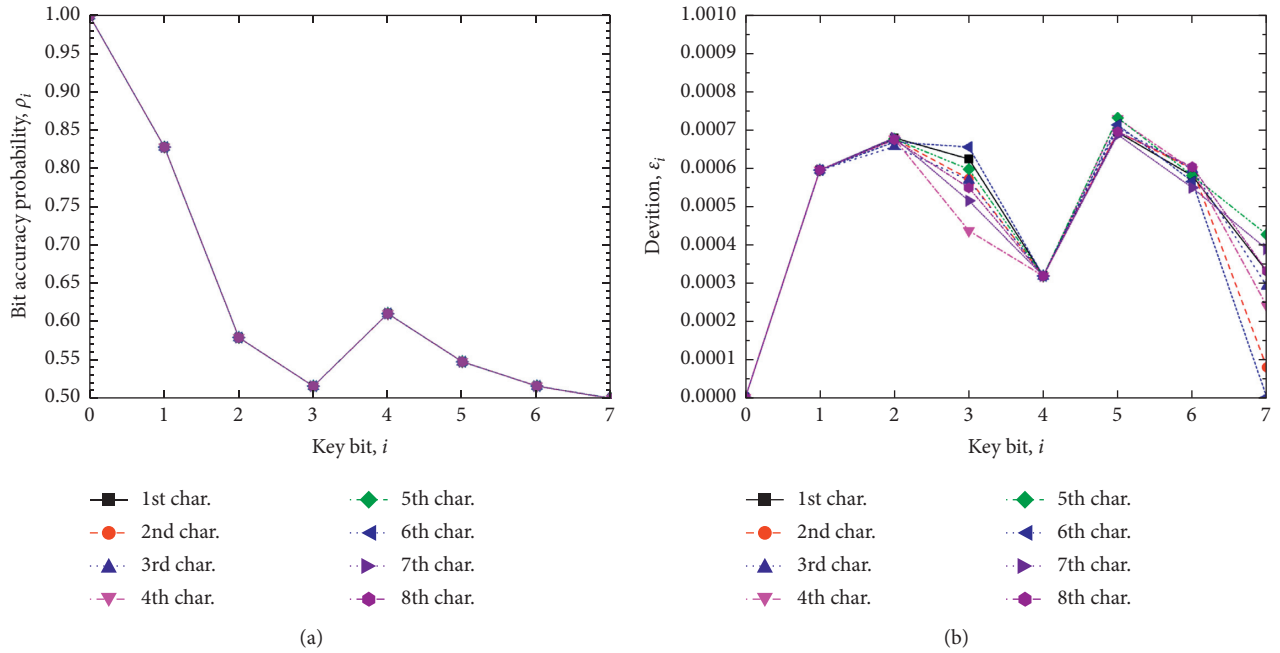


Figure 12: Bit accuracy probability and deviation of the Speck32/64 with a text key.

7. Conclusions

We developed a DL-based cryptanalysis model and evaluated the performance of the DL-based attack on the S-DES, Simon32/64, and Speck32/64 ciphers. The DL-based cryptanalysis may successfully find the text-based encryption key of the block ciphers. When a text key is applied, the DL-based attack broke the S-DES cipher with a success probability of 0.9 given $2^{8.08}$ known plaintexts. That is, the DL-based cryptanalysis reduces the search space nearly by a factor of 8. Moreover, when a text key is applied to the block ciphers, the DL-based cryptanalysis finds the linear approximations between the plaintext-ciphertext pairs and the key, and therefore, it successfully broke the full rounds of Simon32/64 and Speck32/64. When a text key is applied, with a success probability of 0.99, the DL-based cryptanalysis finds 56 bits of Simon32/64 with $2^{12.34}$ known plaintexts and 56 bits of Speck32/64 with $2^{12.33}$ known plaintexts, respectively. Because the developed DL-based cryptanalysis framework is generic, it can be applied to attacks on other block ciphers without change.

The drawback of our proposed DL-based cryptanalysis is that the keyspace is restricted to the text-based key. However, although uncommon, a text-based key can be used to encrypt. For example, the login password entered with the keyboard can be text based if the input data are not hashed. Modern cryptographic functions are designed to be very random looking and to be very complex, and therefore, ML can be difficult to find meaningful relationships between the inputs and the outputs if the keyspace is not restricted. Hence, our approach limited the keyspace to only text-based keys, and the proposed DL-based cryptanalysis could successfully break the 32 bit variants of Simon and Speck ciphers. If the keyspace is not limited, the DL-based cryptanalysis failed to attack the block ciphers. In the future, the accuracy of ML will be improved, and the accuracy becomes more precise, thanks to the development of algorithms and hardware. Moreover, advanced data transformation that efficiently maps cryptographic data onto ML data will help the DL-based cryptanalysis to be performed without the keyspace restriction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (nos. 2019R1F1A1058716 and 2020R1F1A1065109).

References

[1] R. Avanzi, *A Salad of Block Ciphers-The State of the Art in Block Ciphers and Their Analysis*, IACR, Lyon, France, 2017.

[2] X. Dewu and C. Wei, "A survey on cryptanalysis of block ciphers," in *Proceedings of the International Conference on Computer Application and System Modeling (ICCASM)*, pp. 1–6, Taiyuan, China, October 2010.

[3] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer, Berlin, Germany, 1993.

[4] L. R. Knudsen and J. E. Mathiassen, "A chosen-plaintext linear attack on DES," in *Proceedings of the International Workshop*

on Fast Software Encryption (FSE), pp. 262–272, New York, NY, USA, April 2000.

[5] A. Bogdanov and V. Rijmen, "Zero-correlation linear cryptanalysis of block ciphers," Report 2011/123, IACR, Lyon, France, 2011.

[6] S. Zhao, X. Duan, Y. Deng, Z. Peng, and J. Zhu, "Improved meet-in-the middle attacks on generic Feistel constructions," *IEEE Access*, vol. 7, pp. 34416–34424, 2019.

[7] C. Guo, "Understanding the related-key security of Feistel ciphers from a provable perspective," *IEEE Transactions on Information Theory*, vol. 65, no. 8, pp. 5260–5280, 2019.

[8] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, *The SIMON and SPECK Lightweight Block Ciphers*, pp. 1–23, IACR, Lyon, France, 2018.

[9] K. Fu, L. Sun, and M. Wang, "New integral attacks on SIMON," *IET Information Security*, vol. 11, no. 5, pp. 277–286, 2017.

[10] A. D. Dwivedt, P. Morawiecki, and G. Spivastava, "Differential cryptanalysis of round-reduced SPECK suitable for Internet of Things devices," *IEEE Access*, vol. 7, pp. 16476–16486, 2019.

[11] X. Guo, J. Hua, Y. Zhang, and D. Wang, "A complexity-reduced block encryption algorithm suitable for Internet of Things," *IEEE Access*, vol. 7, pp. 54760–54769, 2019.

[12] R. Ankele and S. Kölbl, "Mind the gap-a closer look at the security of block ciphers against differential Cryptanalysis," in *Proceedings of the Selected Areas in Cryptography (SAC)*, pp. 163–190, Alberta, Canada, August 2018.

[13] J. Chen, J. Teh, Z. Liu, C. Su, A. Samsudin, and Y. Xiang, "Towards accurate statistical analysis of security margins: new searching strategies for differential attacks," *IEEE Transactions on Computers*, vol. 66, no. 10, pp. 1763–1777, 2017.

[14] K. Fu, M. Wang, Y. Guo, S. Sun, and L. Hu, "MILP-based automatic search algorithms for differential and linear trails for Speck," Report 2016/407, pp. 1–20, IACR, Lyon, France, 2016.

[15] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[16] J. Blackledge, S. Bezobrazov, and P. Tobin, "Cryptography using artificial intelligence," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, Killarney, Ireland, July 2015.

[17] R. L. Rivest, "Cryptography and machine learning," in *Proceedings of the Advances in Cryptology (ASIACRYPT)*, pp. 427–439, Fujiyoshida, Japan, November 1991.

[18] A. G. Bafghi, R. Safabakhsh, and B. Sadeghiyan, "Finding the differential characteristics of block ciphers with neural networks," *Information Sciences*, vol. 178, no. 15, pp. 3118–3132, 2008.

[19] R. Focardi and F. L. Luccio, "Neural cryptanalysis of classical ciphers," in *Proceedings of the Italian Conference on Theoretical Computer Science (ICTCS)*, pp. 104–115, Urbino, Italy, September 2018.

[20] A. N. Gomez, S. Huang, I. Zhang, B. M. Li, M. Osama, and L. Kaiser, "Unsupervised cipher cracking using discrete GANs," in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–6, Vancouver, Canada, January 2018.

[21] R. Alshammari and A. N. Zincir-Heywood, "Machine learning based encrypted traffic classification: identifying SSH and Skype," in *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–8, Ottawa, Canada, July 2009.

[22] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, pp. 1–34, San Diego, CA, USA, February 2015.

[23] C. Tan and Q. Ji, "An approach to identifying cryptographic algorithm from ciphertext," in *Proceedingsof the IEEE International Conference on Communication Software and Networks (ICCSN)*, pp. 19–23, Beijing, China, Junuary 2016.

[24] Y. Liu, J. Chen, and L. Deng, "Unsupervised sequence classification using sequential output statistics," in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pp. 1–10, Long Beach, CA, USA, December 2017.

[25] M. M. Alani, "Neuro-cryptanalysis of DES and triple-DES," in *Proceedingsof the International Conference on Neural Information Processing (ICONIP)*, pp. 637–646, Doha, Qatar, November 2012.

[26] M. Danziger and M. A. A. Henriques, "Improved cryptanalysis combining differential and artificial neural network schemes," in *Proceedings of the International Telecommunications Symposium (ITS)*, pp. 1–5, Vienna, Austria, August 2014.

[27] X. Hu and Y. Zhao, "Research on plaintext restoration of AES based on neural network," *Security and Communication Networks*, vol. 2018, Article ID 6868506, 9 pages, 2018.

[28] A. Gohr, "Improving attacks on round-reduced speck32/64 using deep learning," *Advances in Cryptology-CRYPTO 2019*, Springer, Berlin, Germany, pp. 150–179, 2019.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.

[30] K. Pešková and R. Neruda, "Hyperparameters search methods for machine learning linear workflows," in *Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1205–1210, Boca Raton, Florida, USA, December 2019.

[31] T. Yu and H. Zhu, "Hyper-parameter optimization: a review of algorithms and applications," pp. 1–56, 2020, http://arxiv.org/abs/2003.05689.

[32] E. F. Schaefer, "A simplified data encryption standard algorithm," *Cryptologia*, vol. 20, no. 1, pp. 77–84, 1996.

[33] R. Vimalathithan and M. L. Valarmathi, "Cryptanalysis of simplified-DES using computational intelligence," *WSEAS Transactions on Computers*, vol. 10, pp. 77–84, 2011.

[34] L. Sharma, B. K. Pathak, and N. Sharma, "Breaking of simplified data encryption standard using binary particle swarm optimization," *International Journal of Computer Science Issues*, vol. 9, no. 3, pp. 307–313, 2012.

[35] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," Report 2013/404, IACR, Lyon, France, 2013.

[36] S. Dehnavi, "Further observations on SIMON and SPECK block cipher families," *Cryptography*, vol. 3, no. 1, pp. 1–12, 2018.

[37] F. Abed, E. List, S. Lucks, and J. Wenzel, "Differential cryptanalysis of round-reduced Simon and speck," in *Proceedings of the International Conference on Fast Software Encryption (FSE)*, pp. 525–545, London, UK, March 2014.

[38] Z. Chu, H. Chen, X. Wang, X. Dong, and L. Li, "Improved integral attacks on SIMON32 and SIMON48 with dynamic key-guessing techniques," *Security and Communication Networks*, vol. 2018, Article ID 5160237, 12 pages, 2018.

[39] I. Dinur, "Improved differential cryptanalysis of round-reduced Speck," in *Proceedings of the International Workshop on Selected Areas in Cryptography*, pp. 147–164, Montreal, Canada, August 2014.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8873639, 6 pages
https://doi.org/10.1155/2020/8873639

Research Article

Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms

Luo GuangJun,[1] **Shah Nazir**,[2] **Habib Ullah Khan**,[3] **and Amin Ul Haq**[4]

[1]*Education Science Department, Xianyang Normal University, Xianyang, Shaanxi, China*
[2]*Department of Computer Science, University of Swabi, Swabi, Pakistan*
[3]*Department of Accounting & Information Systems, College of Business & Economics, Qatar University, Doha, Qatar*
[4]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 61173, China*

Correspondence should be addressed to Luo GuangJun; 1653446869@qq.com

Received 14 May 2020; Revised 4 June 2020; Accepted 6 June 2020; Published 9 July 2020

Academic Editor: Amir Anees

The spam detection is a big issue in mobile message communication due to which mobile message communication is insecure. In order to tackle this problem, an accurate and precise method is needed to detect the spam in mobile message communication. We proposed the applications of the machine learning-based spam detection method for accurate detection. In this technique, machine learning classifiers such as Logistic regression (LR), K-nearest neighbor (K-NN), and decision tree (DT) are used for classification of ham and spam messages in mobile device communication. The SMS spam collection data set is used for testing the method. The dataset is split into two categories for training and testing the research. The results of the experiments demonstrated that the classification performance of LR is high as compared with K-NN and DT, and the LR achieved a high accuracy of 99%. Additionally, the proposed method performance is good as compared with the existing state-of-the-art methods.

1. Introduction

Mobile message is a way of communication among the people, and billions of mobile device users exchange numerous messages. However, such type of communication is insecure due to lack of proper message filtering mechanisms. One cause of such insecurity is spam, and it makes the mobile message communication insecure. Spam is considered to be one of the serious problems in e-mail and instance message services. Spam is a junk mail or message. Spam e-mails and messages are unwanted for receivers which are sent to the users without their prior permission. It contains different forms such as adult content, selling item or services, and so on [1]. The spam increased in these days due more mobile devices deployed in environment for e-mail and message communication. Currently, 85% of mails and messages received by mobile users are spam [2]. The cost of mails and messages are very low for senders but high for receipts of these messages. The cost paid some time by service providers and the cost of spam can be measured in the loss of human time and loss of important messages or mails [3]. Due to these spam mails and messages, the values able e-mails and messages are affected because each user have limited Internet services, short time, and memory [4].

To handle these problems caused by the spam, researchers proposed different techniques to detect the spam e-mails and messages and secure the communication. Details of some of the techniques are presented in this article. Sharaff [5] proposed a method based on machine learning classifiers to classify ham and spam. In the proposed methods, they used four classifiers including iterative dichotomiser, decision tree, simple cart and active directory tree. The weka tool was used for experimental simulations. The proposed method achieved high performance in terms of accuracy. In [6], the e-mail classification method was proposed for the detection of spam. In the system, four predictive machine learning classifiers were used with various data partitions for training and testing of the models. Additionally different hyper parameters values were used in the models. The system obtained good results. Bhat [7]

designed ensemble methods based on techniques such as bagging, boosting, and stacking for classification of spam and ham. The data set used in the study was collected from Facebook. The experimental results demonstrated that the bagging ensemble learning approach, using J48 (decision tree) base classifier, performs well than its individual model, and the method achieved high performance in terms of detection accuracy. In [8], a method is proposed for ham and spam detection and principle components analysis and support vector machine were used in the designing of the system. Additionally, the performance evaluation and cross validation methods were used in the system. The proposed technique achieved high performance, and the method effectively detected the spam. Kumar [9] used various classifiers for ham and spam detection. They used different feature selection algorithms for selection of suitable features. The experimental results show that the classifier random tree with fisher algorithm achieved high results. The proposed method achieved 99% accuracy. In [10], the spam detection method was proposed using machine learning classifiers and 92% accuracy was achieved. Yang at al. [11] proposed spam detection approach based on multimodal fusion (SDAMF). They used the deep neural networks model for detection of spam and achieved 98.48% accuracy. In [12], a spam detection method was proposed based on the artificial immune system (ISAIS) and 98.05% accuracy was achieved. In [13], the Phishing e-mail detection system framework was proposed based on supervised and unsupervised methods. Ruano-Ordás et al. [14] proposed the spam detection method. They used evolutionary computation for discovering spam patterns from e-mail samples.

In this research study, we proposed a spam detection method using machine learning algorithms such as LR, k-nearest neighbor, and decision tree for classification of ham and spam messages. The SMS spam collection dataset was considered for testing of the current research. The dataset was divided into two categories: 30% for testing and 70% for training purpose for the predictive models. The evaluation metrics for performance such as specificity, accuracy, and sensitivity were considered evaluating the proposed study. The results obtained from experiments confirmed that the proposed research achieved high accuracy.

The remaining paper is organized as follows: Section 2 is about the related work to the methodology. In Section 3, experimental work is analyzed and presented in detail. The paper concludes in Section 4.

2. Methods and Materials

This section shows the research methods and materials of the paper.

2.1. Dataset. The dataset considered in the current research is available on kaggle, a machine learning repository [15]. The dataset "SMS spam collection dataset" contains 5572 instances and two attributes v1 and v2. The v2 is the input messages which are either spam or nonspam. The predicted label v1 has two classes: 0 = nonspam and 1 spam. In the data, 4900 are nonspam samples and 672 are spam samples. The dataset is given in Table 1.

2.2. Classification Algorithms. The following machine learning algorithms were considered for classifications of ham and spam.

2.2.1. Logistic Regression. LR is a classifier [16, 17]. The problem in binary classification is computing the value of predictive y while $y \in [0, 1]$; 0 and 1 are for class negative and positive. The LR predicts the variable value of multiclassification such as $y \in [0, 1, 2, 3]$.

2.2.2. Decision Tree. A DT is a supervised machine learning algorithm [18, 19]. Its shape is like a tree in which each node is a decision node or leaf. This technique of DT is easily understandable and simple for making the decisions. A DT contains external and internal nodes interlinked with each other. Decision can be made based on the internal nodes and the child node to access the preceding node. There is no child of the leaf node and is linked with a label.

2.2.3. K-Nearest Neighbor. K-NN is a classification supervised learning algorithm [18]. It predicts the label of class as a fresh input and utilizes the same to its inputs in the training set. The performance of K-NN is not enough good. Let (x, y) be the the training observation and the learning function h: $X \longrightarrow Y$, so that an observation x, h (x) can establish y value.

2.2.4. Division of Dataset. The set data were split into 30% and 70% for validation and training of the predictive model.

2.2.5. Measure for Evaluation of Performance. To validate the classifier performance, we used metrics such as specificity, accuracy, sensitivity, and execution time which are expressed in equations (1), (2), and (3) which are computed from confusion matrix as given in Table 2.

The formulation of measures is as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%, \quad (1)$$

$$\text{sensitivity } (Sn) = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \quad (2)$$

$$\text{specificity } (Sp) = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\%. \quad (3)$$

3. Experiments and Result Analysis

Diverse approaches are used for spam detection. Abayomi-Alli et al. [22] presented a comprehensive review of the soft techniques in spam classifications. Acceptability users of

Table 1: Description of SMS Spam collection dataset.

Sample no number	V1	V2
1	Ham	Go until jurong point, crazy.. Available only in bugis *n* great world la e buffet... Cine there got amore wat...
2	Ham	Ok lar... Joking wif u oni...
3	Spam	Free entry in 2 a wkly comp to win FA cup final tkts 21st May 2005. Text FA to 87121 to receive entry question (std txt rate) T&C's apply 08452810075over18's.
.	.	.
.	.	.
.	.	.
.	.	
.	.	
.	.	
.	.	Rofl. Its true to its name
.		
5572	Ham	

Table 2: Confusion matrix [20, 21].

	Predicted spam (1)	Predicted ham (0)
Actual spam (1)	TP	FN
Actual ham (0)	FP	TN

SMS spam application on the store of Android App were assessed. Roy et al. [23] proposed a technique to identify short-text spam messages. The proposed model is helpful for different strategies of business. Kaur et al. [24] presented a detailed report on techniques of detection-cum-analysis of compromised accounts and spam detection. Jeong et al. [25] presented a spam detection approach. Cheah et al. [26] proposed an approach for security testing of automotive interface of Bluetooth. Halabi and Bellaiche [27] presented an approach to quantify the performance and service evaluation of cloud security. Tsui et al. used diverse composition for the consequences of development of components on the properties of security [28]. Zhang et al. [29] presented a novel method for evaluating the crowd security of OSN trustworthiness. Mao et al. [30] made a security network of dependency from the access behavior to measure the significance of object security from with broad perspectives.

We performed experiments to classify the ham and spam using the SMS spam collection dataset. Classifiers LR, decision tree, and k-nearest neighbor were used for the classification in this study. The dataset is divided as follows: 30% for validation and 70% for training. The results obtained from experiments are shown in tables and presented in figures graphically. The python on an Intel (R) Core™ i5 -2400CPU and Windows 10 were used for the experiments and setup to obtain the computation results of the experimental work.

3.1. Visualization of SVM Spam Collection Dataset. In the data, 4900 are ham samples and 672 are spam samples which are shown in Figure 1.

Figure 2 shows the ratio of spam and ham messages.

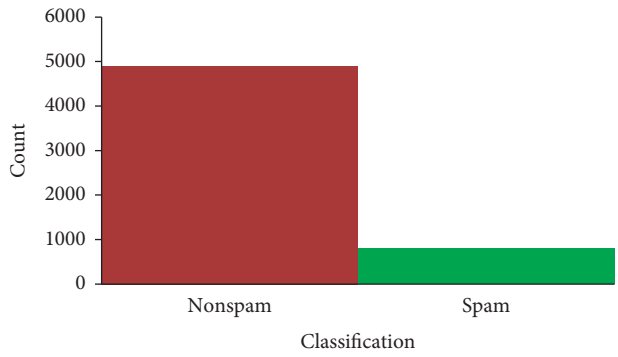


Figure 1: Classification of spam and ham messages.

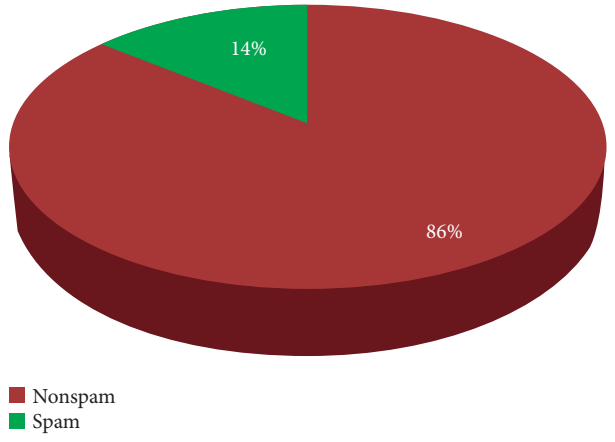


Figure 2: Ratio of ham and spam messages.

Table 3: Classification performance of classifiers.

Predictive model	Evaluation performance measures				
	Accuracy (%)	Specificity (%)	Sensitivity (%)	MCC (%)	Processing Time (s)
Logistic regression ($C=1$)	99	93	86	93	0.494
K-nearest neighbor (K-NN, $K=1$)	95	80	60	80	0.630
DT	98	95	86	95	46.032

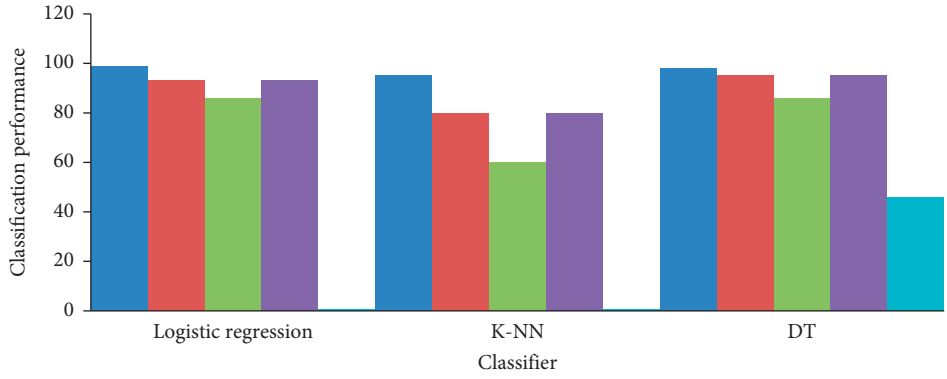


Figure 3: Performance of classifications for classifiers.

3.2. Classification Results of Classifiers. To perform the classification of the ham and spam messages, in this paper, we used the classification algorithms such as LR, decision tree, and K-nearest neighbor with essential basic hyperparameters. The dataset was divided into two parts for training and testing. The classifiers were trained with 70% of the samples and validated with 30% samples of the data set. All the experiential results are reported in Table 3. According to Table 3, the LR at hyperparameter $C=1$ achieved 99% accuracy, 100% specificity, sensitivity 86%, and MCC, 93% and the processing time is 0.494 seconds. The classifier decision tree obtained 98% accuracy, 94% specificity, sensitivity 86%, and MCC 95%, and the processing time is 46.032 seconds. Similarly, the k-nearest neighbor classifier achieved 95% accuracy, 100% specificity, sensitivity 60%, and MCC 80%, and the processing time is 0.630 seconds. The experimental results (according to Table 3), the classification performance of LR is high as compared with the decision and k-nearest neighbor in terms of accuracy. The classification accuracy of classifiers is shown in Figure 3. Similarly, the computation time of LR is low as compared with k-NN and DT. Figure 4 shows the processing time graphically for better understanding. From these experiential results analysis, we concluded that the LR effectively classifies the ham and spam because the achieved accuracy is high. The 100% specificity of the LR model correctly detected the ham messages. Similarly, 86% sensitivity shows that LR spam message capability is good. Thus, the experimental results suggest that LR is a the best classifier for the classifications of ham and spam successfully.

Figure 3 shows the performance of classifications for classifiers including LR, K-NN, and DT.

The classifier processing time for K-NN, LR, and DT is shown in Figure 4.

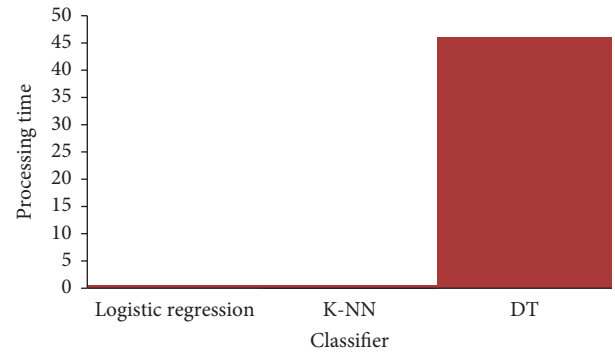


Figure 4: Processing time of classifiers.

Table 4: Comparison of the current study with existing approaches in terms of accuracy.

Reference	Method	Accuracy (%)
[11]	SDAMF	98.48
[12]	ISAIS	98.48
Our study 2019	LR	99

3.3. Comparison of Performance with Existing Methods. The comparison performance of classifications of the current approach is done with the existing approaches in term of accuracy. The current approach achieved an accuracy of 99% which is high as compared with the available approaches. Table 4 shows the accuracy obtained from the current approach along with other approaches available.

The performance comparison is graphically shown in Figure 5.

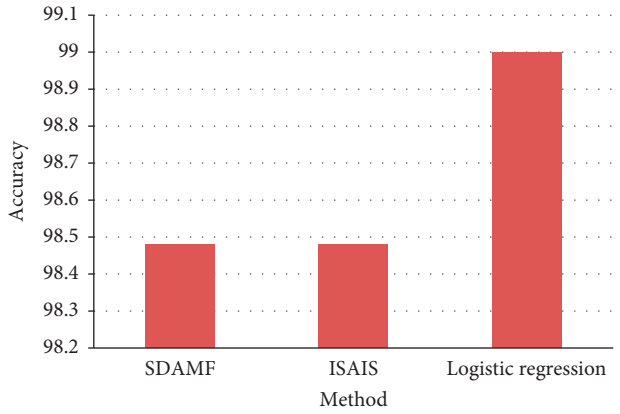


Figure 5: Performance comparison.

4. Conclusion

Detection of spam is important for securing message and e-mail communication. The accurate detection of spam is a big issue, and many detection methods have been proposed by various researchers. However, these methods have a lack of capability to detect the spam accurately and efficiently. To solve this issue, we have proposed a method for spam detection using machine learning predictive models. The method is applied for the purpose of detection of spam. The experimental results obtained show that the proposed method has a high capability to detect spam. The proposed method achieved 99% accuracy which is high as compared with the other existing methods. Thus, the results suggest that the proposed method is more reliable for accurate and on-time detection of spam, and it will secure the communication systems of messages and e-mails.

Data Availability

No data were used to support the study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the Key scientific research of the Ministry of Education, China (Grant no. DCA190332).

References

[1] L. Zhang, J. Zhu, and T. Yao, "An evaluation of statistical spam filtering techniques," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 3, no. 4, pp. 243–269, 2004.

[2] M. Bassiouni, M. Ali, and E. A. El-Dahshan, "Ham and spam E-mails classification using machine learning techniques," *Journal of Applied Security Research*, vol. 13, no. 3, pp. 315–331, 2018.

[3] I. Alsmadi and I. Alhami, "Clustering and classification of email contents," *Journal of King Saud University—Computer and Information Sciences*, vol. 27, no. 1, pp. 46–57, 2015.

[4] B. Yu and Z.-B. Xu, "A comparative study for content-based dynamic spam classification using four machine learning algorithms," *Knowledge-Based Systems*, vol. 21, no. 4, pp. 355–362, 2008.

[5] A. Sharaff, "Comparative study of classification algorithms for spam email detection," in *Emerging Research in Computing, Information, Communication and Applications*, pp. 237–244, Springer, Berlin, Germany, 2016.

[6] S. Youn and D. McLeod, "A comparative study for email classification," in *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, pp. 387–391, Springer, Berlin, Germany, 2007.

[7] S. Y. Bhat, "Spammer classification using ensemble methods over structural social network features," in *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pp. 454–458, Warsaw, Poland, August 2014.

[8] J. C. Gomez and M.-F. Moens, "PCA document reconstruction for email classification," *Computational Statistics & Data Analysis*, vol. 56, no. 3, pp. 741–751, 2012.

[9] R. K. Kumar, "Comparative study on email spam classifier using data mining techniques," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, pp. 14–16, Hong Kong, March 2012.

[10] S. K. Trivedi and S. Dey, "Interplay between probabilistic classifiers and boosting algorithms for detecting complex unsolicited emails," *Journal of Advances in Computer Networks*, vol. 1, pp. 132–136, 2013.

[11] H. Yang, Q. Liu, S. Zhou, and Y. Luo, "A spam filtering method based on multi-modal fusion," *Applied Sciences*, vol. 9, no. 6, p. 1152, 2019.

[12] A. J. Saleh, A. Karim, B. Shanmugam et al., "An intelligent spam detection model based on artificial immune system," *Information*, vol. 10, no. 6, p. 209, 2019.

[13] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decision Support Systems*, vol. 107, pp. 88–102, 2018.

[14] D. Ruano-Ordás, F. Fdez-Riverola, and J. R. Méndez, "Using evolutionary computation for discovering spam patterns from e-mail samples," *Information Processing & Management*, vol. 54, no. 2, pp. 303–317, 2018.

[15] SMS, "Spam collection dataset," 2019, https://www.kaggle.com/datasets.

[16] K. Larsen, J. H. Petersen, E. Budtz-Jørgensen, and L. Endahl, "Interpreting parameters in the logistic regression model with random effects," *Biometrics*, vol. 56, no. 3, pp. 909–914, 2000.

[17] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Science & Business Media, Berlin, Germany, 2013.

[18] X. Wu, V. Kumar, J. Ross Quinlan et al., "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.

[19] A. U. Haq, "A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms," *Mobile Information Systems*, vol. 2018, Article ID 3860146, 21 pages, 2018.

[20] A. U. Haq, J. P. Li, M. H. Memon et al., "Feature selection based on L1-norm support vector machine and effective recognition system for Parkinson's disease using voice recordings," *IEEE Access*, vol. 7, pp. 37718–37734, 2019.

[21] A. U. Haq, "Comparative analysis of the classification performance of machine learning classifiers and deep neural network classifier for prediction of Parkinson disease," in *Proceedings of the 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 101–106, Chengdu, China, December 2018.

[22] O. Abayomi-Alli, S. Misra, A. Abayomi-Alli, and M. Odusami, "A review of soft techniques for SMS spam classification: methods, approaches and applications," *Engineering Applications of Artificial Intelligence*, vol. 86, pp. 197–212, 2019.

[23] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter SMS Spam," *Future Generation Computer Systems*, vol. 102, pp. 524–533, 2020.

[24] R. Kaur, S. Singh, and H. Kumar, "Rise of spam and compromised accounts in online social networks: a state-of-the-art review of different combating approaches," *Journal of Network and Computer Applications*, vol. 112, pp. 53–88, 2018.

[25] S. Jeong, G. Noh, H. Oh, and C.-K. Kim, "Follow spam detection based on cascaded social information," *Information Sciences*, vol. 369, pp. 481–499, 2016.

[26] M. Cheah, S. A. Shaikh, O. Haas, and A. Ruddle, "Towards a systematic security evaluation of the automotive Bluetooth interface," *Vehicular Communications*, vol. 9, pp. 8–18, 2017.

[27] T. Halabi and M. Bellaiche, "Towards quantification and evaluation of security of cloud service providers," *Journal of Information Security and Applications*, vol. 33, pp. 55–65, 2017.

[28] F. Tsui, E. Jung, and S. Duggins, "Software composition of different security level components," *Computer Technology and Application*, vol. 2, no. 11, pp. 835–842, 2011.

[29] Z. Zhang, J. Wen, X. Wang, and C. Zhao, "A novel crowd evaluation method for security and trustworthiness of online social networks platforms based on signaling theory," *Journal of Computational Science*, vol. 26, pp. 468–477, 2017.

[30] W. Mao, Z. Cai, D. Towsley, Q. Feng, and X. Guan, "Security importance assessment for system objects and malware detection," *Computers & Security*, vol. 68, pp. 47–68, 2017.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8852124, 9 pages
https://doi.org/10.1155/2020/8852124

Research Article

Modelling Features-Based Birthmarks for Security of End-to-End Communication System

Meilian Li,[1] Shah Nazir,[2] Habib Ullah Khan,[3] Sara Shahzad,[4] and Rohul Amin[5]

[1]*School of Electronic and Electrical Engineering, Anhui Sanlian Univeristy, Hefei, Anhui 230601, China*
[2]*Department of Computer Science, University of Swabi, Ambar, Khyber Pakhtunkhwa 23430, Pakistan*
[3]*Department of Accounting and Information System, College of Business and Economics, Qatar University, Doha 2713, Qatar*
[4]*Department of Computer Science, University of Peshawar, Peshawar, Khyber Pakhtunkhwa 25120, Pakistan*
[5]*Department of Mathematics, University of Peshawar, Peshawar, Khyber Pakhtunkhwa 25120, Pakistan*

Correspondence should be addressed to Habib Ullah Khan; habib.khan@qu.edu.qa

Received 12 May 2020; Revised 30 May 2020; Accepted 4 June 2020; Published 29 June 2020

Academic Editor: Iqtadar Hussain

Feature-based software birthmark is an essential property of software that can be used for the detection of software theft and many other purposes like to assess the security in end-to-end communication systems. Research on feature-based software birthmark shows that using the feature-based software birthmark joint with the practice of software birthmark estimation together can deliver a right and influential method for detecting software piracy and the amount of piracy done by a software. This can also guide developers in improving security of end-to-end communication system. Modern day software industry and systems are in demand to have an unbiased method for comparing the features-based birthmark of software competently, and more concretely for the detecting software piracy and assessing the security of end-to-end communication systems. In this paper, we proposed a mathematical model, which is based on a differential system, to present feature-based software birthmark. The model presented in this paper provides an exclusive way for the features-based birthmark of software and then can be used for comparing birthmark and assessing security of end-to-end communication systems. The results of this method show that the proposed model is efficient in terms of effectiveness and correctness for the features-based software birthmark comparison and security assessment purposes.

1. Introduction

Software piracy is considered to be a foremost anxiety for the industry of software. Software piracy is done due to the large growth of Internet and software industry. Wide-ranging research [1] into the way to do piracy of software detection has encouraged the progress of techniques such as water-marking in software, fingerprints, and recently the birth-mark of software. Birthmark of software is inherent characteristic or property of software to be effectively used for theft of software and detection of software piracy. Software watermark and fingerprint have been used for a long time with the realization but these techniques have some limitations. Some of the researchers and practitioners of industry are using forward-looking versions of software watermark [1–12], fingerprints [13, 14], software clone [15, 16], and software birthmark [17–29]. Detection of plagiarism is relevant area to these mentioned software detection methods which are used for source code theft and discovery of similarities among the original and duplicated source codes [30–35]. Watermark of software is used to express the proprietorship of a software. The watermarks add some supplementary code or detail information to the existing software to show the ownership. Software finger-print is used to find the intellectual property. Cloning of software is done by copy-past of source code of copyrighted software that may be in parts or full in another version of the software. The methods of clone detection of software are used to sense the piracy in such cases. Software birthmark is considered to be the recently used technique for the software piracy detection. Birthmarks of software use the inherent characteristics or software properties to identify the

originality of software. Birthmark similarities of two software programs show the extent of piracy done among the software.

The concept of birthmark of software is offered for the similar determination of theft identification of software and detection of piracy. Birthmark of software is till now recognized to be resilient to any obliteration or obfuscation technique(s). Several researches have been accomplished to recognize diverse types of birthmark of software [20, 24–27, 29, 36–40]. Nazir et al. [36] offered the strategy of feature-based software birthmark and a proper estimation process for birthmark of software [37]. Though birthmark of software has been extensively deliberated in research from several viewpoints of the area of software piracy and detection of theft, yet there is no objective measure to compare birthmarks of software efficiently for the detection of piracy and to assess the security of end-to-end communication systems. The aim of the proposed work is to deliver a mathematical model for the purpose of comparison of feature-based birthmark of software and to assess the security of end-to-end communication systems. The proposed model is based on differential equations system and uses the features of birthmark, presented by Nazir et al. [36] and can be assessed for the comparison purpose of features-based birthmark of another (duplicate) software and assessment purpose of the security of end-to-end communication systems. These comparisons will ultimately endorse or reject the piracy performed in software and security changes that occurred in the applications.

The organization of the paper is as follows: Section 2 of the paper presents related work done for software birthmark and detection of piracy. Section 3 gives the details of the mathematical model used for the proposed research, with logic of using mathematical model for birthmark comparisons. This section further provides explanation for the use of differential equations as system model. The results and discussion of the proposed research are discussed in Section 4. This section further discusses the case study of the method. The paper concludes in Section 5.

2. Related Work

Software industry and productions are facing with a dreadful problem of piracy and changes of security in software. On the other hand, the pirates of software make vast sums of money from the trade performed in piracy and changes in security of software. According to the report of Business Software Alliance (BSA) [41] of year 2013, about 43% of software programs that are configured on personnel computer systems in the globe were pirated and not appropriately licensed. The marketable value of these unlicensed software programs was about 62.7 billion dollars. Taking this point further, Myles and Collberg [29] outlined the three foremost threats to industry of software. These threats include the illegal re-selling of the legitimate software, malicious reverse engineering, and software tampering. The industries of software adopt diverse practices to trace the theft of software. Among these practices, the software birthmark is one of the techniques which are used for the detection of pirated software and by the assistance of which the pirated or duplicated version of the software would be traced. The software birthmark types and history could be taken at length. Tamada et al. [42] designed the very first birthmark method which is based on four types of birthmark; these birthmarks are constant values in field variables (CVFV), inheritance structure (IS), sequence of method calls (SMC), and used classes (UC). This technique of birthmark was effectively used by the software industry for the purpose of detection theft of software. Myles and Collberg [29] suggested a method of "Whole Program Path Birthmark." This method is based on the whole control flow of the program. The properties of resilience and credibility were used to assess the effectiveness of the method. The method further reveals that the WPPB is more resilient than the existing methods of birthmark. Zeng et al. [43] proposed a framework of semantic-based abstract interpretation for software birthmark. Mahmood et al. [44] proposed a method-based similarity level for software birthmark. By help of the proposed method, the elements of code and their properties can be found. This method traces the modification occurring in the program. Wang et al. [45] suggested the operand stack dependence-based static software birthmark for the difficulty of semantic lost when mining birthmark with the help of k-gram algorithm.

Moreover, through offering different types of birthmark, several researchers have provided some case studies for the work of their analysis and evaluation they performed. Choi et al. [23] analyzed the static API-based birthmark of software for binary executable of Windows and compared 49 executables. They described that the birthmark used by them can easily distinguish and identify the program copies. The birthmark is checked with the Windows dynamic birthmark and presented to likely suitable for the applications with Graphical User Interfaces. Kakimoto et al. [28] did analysis of the birthmark similarities in Argo UML and then visualized them using multidimensional scale. Park et al. [24] proposed a static API trace birthmark for detection of theft of Java-based programs. This technique assesses the birthmark for the properties of resilience and credibility. Results obtained from their experiment of the proposed method show that the static API birthmark can identify related components of two packages while the other techniques of birthmarks fail to do so. Xie et al. [46] suggested a static birthmark for k-gram and their weights. The weight is computed by analysis rate of change in the k-gram frequency of the actual and modified version of the program. Myles and Collberg [47] accomplished an empirical analysis of the k-gram-based software birthmark by analysis of 111 programs in the Java programming language. Several studies [20, 24–27, 29, 35, 36, 42, 43, 48–51] were explored for the types of birthmark, their analysis, and assessment, but the work of [48, 49] analyzed the birthmark in depth used for different purposes. From most of the studies, it is derived that in majority of the cases only the results of case study and empirical suggestions are provided to support the given studies.

The current research work is endeavouring to propose a mathematical model for the purpose of comparisons of

feature-based software birthmark and to evaluate the security of end-to-end communication systems. The model is based on differential equations system and uses the features of birthmark presented in the literature.

3. Methodology

The methodology is described in the following subsections which present the proposed research methodology for the features-based birthmark of software.

3.1. Need for a Mathematical Model. Diverse methods based on mathematics are used by the researchers and practitioners for modelling the real life occurrences. A number of these techniques include exact equations, linear equations, separable variable methods, substitution solution, and numerical method. These techniques are used to solve the first-order differential equations [52].

Software industry is endeavouring to have a policy and strategic independent description for birthmarks of software, which can then be used as proper estimating and comparisons of birthmark of software. This definition and description will ease the industry of software to detect software theft and piracy with further changes in security of end-to-end communication systems. The recommended feature-based software birthmark [36] is currently mathematically modelled to enable the birthmark comparison based on the defined features. This feature-based birthmark comparison will identify the similarities among software programs for the purpose of piracy detection and changes in security of end-to-end communication systems.

In this research work, the essential model is planned in the form of homogeneous linear differential system. For the design of this type of system, generally three methods are used. These methods are repeated Eigen values, distinct real Eigen values, and Complex Eigen values. In the situation of the proposed research, the Eigen values are complex.

Mathematically, if $\lambda_1 = \alpha + i\beta$ and $\lambda_2 = \alpha - i\beta$, where $i = \sqrt{-1}$ are Complex Eigen values of the matrix "A," then the corresponding Eigen vector also contains complex values [52]. This study proposed a mathematical model for the features-based software birthmark to enable the comparisons among the birthmark based on the predefined features.

3.2. Terminologies Used for Modelling Software Piracy Detection. The following subsections briefly discuss the method and terminologies used in this research for modelling features-based birthmark of software.

3.2.1. Differential Model for Software Birthmark. The differential equations have the derivatives of one or more dependent variable(s), with respect to one or more independent variable(s) [52]. Let there be an equation with unknown variables, without any information available about its construction. Such type of an equation (function) can be represented as, for example, $y' = \phi(x)$?

3.2.2. Eigen Values and Eigen Vector. The characteristic polynomial of a square matrix "A" is defined by [53]

$$p(\lambda) = \det(A - \lambda I). \quad (1)$$

If p is the characteristic polynomial of matrix "A", then the roots of p are the Eigen values of matrix "A." If λ is Eigen value of "A" and $x \neq 0$ satisfies $(A - \lambda I)x = 0$, then x is Eigen vector corresponding to the Eigen value λ. In the context of this research, there are three main features (categories), from which a differential system is obtained. This differential system is also called linear differential system. To solve this differential system, we need the Eigen vector for the corresponding Eigen values.

3.3. Model for Comparison of Birthmark for Detection of Software Piracy and Assessment of Security in End-to-End Communication Systems. Diverse approaches have been used in literature in the area of development of healthcare mobile applications. The proposed technique for comparisons of suggested features-based software birthmark is mathematically modelled to enable and facilitate the comparisons of birthmarks and assessment of security of end-to-end communication systems based on the identified features. The features followed by the proposed study are the features that are already identified in the previous research work [36, 48, 49]. This features-based comparison advises the similarity among different modules of the software which can further investigate the changes occurring in the security of end-to-end communication systems. Here, in this study, we considered the four main features that were previously identified [36]. These features include preconditional features, input features, nonfunctional features, and functional features. These categories are further divided into subcategories of features. The preconditional features have three subfeatures categories that are program availability, runnable, and identification of components. These features are significant which can be patterned even for all kinds of programs for detecting the similarities. Figure 1 shows the detail of the feature-based birthmark of software as already defined [36].

After performing the early analysis, the rest of the three features categories are used as the base of mathematical model for the proposed study, while the category of preconditional features is excluded, as this features category can be examined for all types of software while detecting the piracy and changes in security of software. The input feature category is further divided into 17 features that are program context, program contents, internal data structure, program flow, configurable terminologies, program responses, control flow, size of program, interface description, number of statements in program, naming, functions, restriction, limitation and constraints, comprehensive documentation, global data structure, user interface, and internal quality. The nonfunctional feature category is further divided into 12 subfeatures that are automation, ease of use, friendly, scalability, applicability, interface connections, robustness, dependency, portability, scope, standard, and

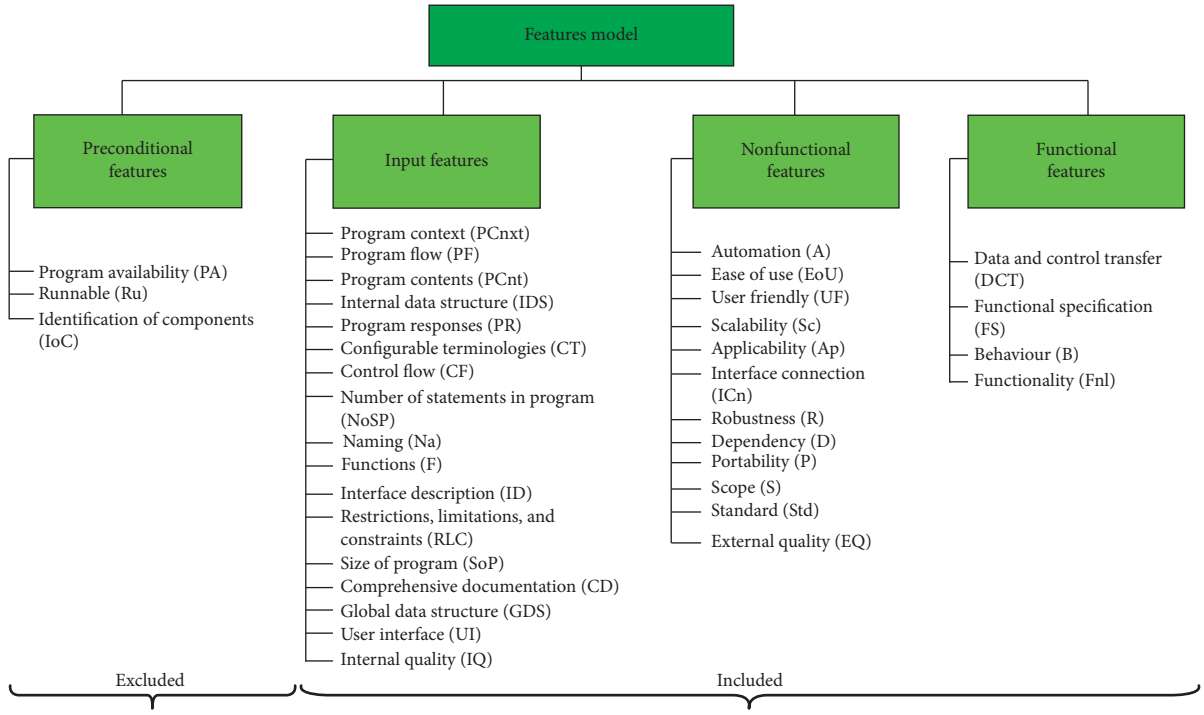


Figure 1: Software features and their subfeatures.

external quality. The functional feature category is divided into further four subfeatures that are data and control process, functional specification, behaviour, and functionality. All the categories of these features are combined and then plotted in the form of differential system mathematically as

$$\begin{aligned} x'(f) &= 17x + 12y + 4z, \\ y'(f) &= 4x + 17y + 12z, \\ z'(f) &= 12x + 4y + 17z, \end{aligned} \tag{2}$$

where x, y, and z are the three features. Then, from equation (2), we have

$$X'(f) = AX(f). \tag{3}$$

To find these three features x, y, and z, we need to find the solution of equation (2). For this purpose of finding the exact solution, we have to find the Eigen values and Eigen vectors of the matrix A. The proposed process has been carried out in the following steps.

Step 1. To find Eigen value,

$$\text{Since } A = \begin{bmatrix} 17 & 12 & 4 \\ 4 & 17 & 12 \\ 12 & 4 & 17 \end{bmatrix}. \tag{4}$$

According to Section 3.2.2, by using equation (1), the characteristic polynomial of the matrix "A" is given by

$\det(A - \lambda I) = 0$. That is,

$$\begin{vmatrix} 17-\lambda & 12-0 & 4-0 \\ 4-0 & 17-\lambda & 12-0 \\ 12-0 & 4-0 & 17-\lambda \end{vmatrix} = 0. \tag{5}$$

After simplification, we have

$$\lambda^3 - 51\lambda^2 + 723\lambda - 4257 = 0. \tag{6}$$

By using syntactic division, we have

$$\begin{aligned} \lambda_1 &= 33, \\ \lambda_2 &= 9 + 6.9282i, \\ \lambda_3 &= 9 - 6.9282i. \end{aligned} \tag{7}$$

Thus, the Eigen values of the matrix "A" are 33, $9 + 6.9282i$, and $9 - 6.9282i$, where λ_1 is real, λ_2 is complex, and λ_3 is complex conjugate of λ_2.

Step 2. To find Eigen vector of corresponding Eigen values,

If $\lambda = 33$, then the corresponding Eigen vector is given by $AX = \lambda X$.

$$\begin{bmatrix} 17 & 12 & 4 \\ 4 & 17 & 12 \\ 12 & 4 & 17 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 33 \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \tag{8}$$

By solving this, we have

$$\begin{aligned} -16a + 12b + 4c &= 0, \\ 4a - 16b + 12c &= 0, \\ 12a + 4b - 16c &= 0. \end{aligned} \tag{9}$$

By solving this, we have

$$\begin{aligned} a &= 1, \\ b &= 1, \\ c &= 1. \end{aligned} \tag{10}$$

Thus, the corresponding Eigen vector for Eigen value $\lambda = 33$ is $V1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

Similarly, the corresponding Eigen vectors for $9 + 6.9282i$ and $9 - 6.9282i$ are given by

$$\begin{aligned} V_2 &= \begin{pmatrix} \frac{1}{2}i(i + \sqrt{3}) \\ -\frac{1}{2}i(-i + \sqrt{3}) \\ 1 \end{pmatrix}, \\ V_3 &= \begin{pmatrix} -\frac{1}{2}i(-i + \sqrt{3}) \\ \frac{1}{2}i(i + \sqrt{3}) \\ 1 \end{pmatrix}. \end{aligned} \tag{11}$$

Step 3. Thus, the solution of equation (2) is given by

$$\begin{aligned} X = {} & c_1 V_1 e^{\lambda_1 f} + c_2 (B_1 \cos \beta f - B_2 \sin \beta f) e^{\alpha f} \\ & + c_3 (B_2 \cos \beta f + B_1 \sin \beta f) e^{\alpha f}, \end{aligned} \tag{12}$$

where $\lambda = \alpha + i\beta$, $B_1 =$ real part (Eigen vector) and $B_2 =$ imaginary part (Eigen vector). Putting the values in the above equation, we get

$$\begin{aligned} X = {} & c_1 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} e^{33f} + c_2 \left(\begin{pmatrix} -\frac{1}{2} \\ -\frac{1}{2} \\ 1 \end{pmatrix} \cos 6.9282 f - \begin{pmatrix} \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} \\ 0 \end{pmatrix} \sin 6.9282 f \right) e^{9f} \\ & + c_3 \left(\begin{pmatrix} \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} \\ 0 \end{pmatrix} \cos 6.9282 f + \begin{pmatrix} -\frac{1}{2} \\ -\frac{1}{2} \\ 1 \end{pmatrix} \sin 6.9282 f \right) e^{9f}, \\ x(f) = {} & c_1 e^{33f} + c_2 (\cos(6.9282 f)) e^{9f} + c_3 (\sin(6.9282 f)) e^{4f}. \end{aligned} \tag{13}$$

Similarly, we have

$$\begin{aligned} y(f) &= c_1 e^{33f} + c_2 \left[-\frac{1}{2}\cos(6.9282 f) + \frac{\sqrt{3}}{2}\sin(6.9282 f) \right] e^{9f} + c_3 \left[-\frac{\sqrt{3}}{2}\cos(6.9282 f) - \frac{1}{2}\sin(6.9282 f) \right] e^{9f}, \\ z(f) &= c_1 e^{33f} + c_2 \left[-\frac{1}{2}\cos(6.9282 f) - \frac{\sqrt{3}}{2}\sin(6.9282 f) \right] e^{9f} + c_3 \left[\frac{\sqrt{3}}{2}\cos(6.9282 f) - \frac{1}{2}\sin(6.9282 f) \right] e^{9f}. \end{aligned} \tag{14}$$

Putting the value of $f = 0$ in the above equations and using the initial conditions, we have

$$\begin{aligned} c_1 - \frac{1}{2}c_2 + \frac{\sqrt{3}}{2}c_3 &= 17, \\ c_1 - \frac{1}{2}c_2 - \frac{\sqrt{3}}{2}c_3 &= 4, \\ c_1 + c_2 &= 12. \end{aligned} \tag{15}$$

By solving these equations, we get

$$\begin{aligned} c_1 &= 11, \\ c_2 &= 1, \\ c_3 &= -7.5056. \end{aligned} \tag{16}$$

Thus, the required solution of (2) is given by

$$\begin{aligned} x(f) &= 11e^{33f} + \cos(6.9282 f) e^{9f} - 7.5056 \sin(6.9282 f) e^{4f}, \\ y(f) &= 11e^{33f} + \left[-\frac{1}{2}\cos(6.9282 f) + \frac{\sqrt{3}}{2}\sin(6.9282 f) \right] e^{9f} - 7.5056 \left[-\frac{\sqrt{3}}{2}\cos(6.9282 f) - \frac{1}{2}\sin(6.9282 f) \right] e^{9f}, \\ z(f) &= 11e^{33f} + \left[-\frac{1}{2}\cos(6.9282 f) - \frac{\sqrt{3}}{2}\sin(6.9282 f) \right] e^{9f} - 7.5056 \left[\frac{\sqrt{3}}{2}\cos(6.9282 f) - \frac{1}{2}\sin(6.9282 f) \right] e^{9f}, \end{aligned} \tag{17}$$

where $x(f)$, $y(f)$, and $z(f)$ represent the required solution of the differential system (2) for the available features of software birthmark.

For the process of comparisons of birthmark of software for the detection purpose of software piracy and assessment of security of end-to-end communication systems, birthmark(s) of various occurrences of (the same) software application defined over the same features based birthmark [36] can be modelled using the given differential system. If the solutions of both of the resulting differential systems are found the same or nearly the same, then the software is copy of the original software; hence, it is proved to be pirated and changes have occurred in the security of end-to-end communication systems.

4. Results and Discussion

The following subsections briefly discuss the results and discussion section of the paper.

4.1. Experimentation with a Case Study and the Results. The proposed research work based on mathematical model for features-based software birthmark has been validated by performing a case study. The case study was intended to test an Android mobile application for features-based software birthmark. Multiple versions of the application were generated to bear and validate the process of comparison. Copies instances of the Android mobile applications were modified (in parts) to enhance and eliminate a portion of functionality. These modifications were made through third-party developers. This process was done to mimic pirated copies of the test cases application and to assess the security of end-to-end communication systems.

After getting modified copies of the mobile (Android) application, the features-based birthmark of software was individually derived from all of the copies of the application as shown in equation (2). This was performed by extracting each individual copy of the features-based application. The features along with their details were taken into consideration for checking the piracy among the applications and to assess the security of end-to-end communication systems. The features of each copy were then extracted and the birthmarks of pirated copies of the applications were then compared with the features-based software birthmark of the actual application to show that piracy and changes in security were done/or not to further show the similarities and security view among the actual and pirated copies of the application. Figure 2 shows case study performed for features extraction from the actual and pirated version of the software and their comparison process.

A case study of the equations below was taken as an example to show the validity:

$$\frac{\partial u(x,y,z)}{\partial x} = 17x + 15xy, \tag{18}$$

$$\frac{\partial u(x,y,z)}{\partial y} = 17y + 15xz, \tag{19}$$

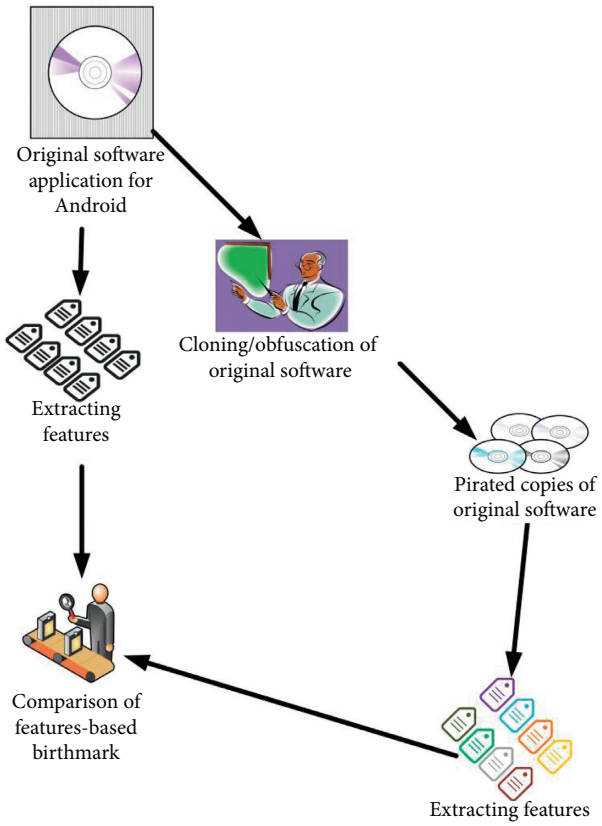


Figure 2: Case study for feature extractions and comparisons process of actual (original) and pirated software.

$$\frac{\partial u(x,y,z)}{\partial z} = 17z + 15xy. \tag{20}$$

And the exact equation of the above system of partial differential equation is

$$u(x,y,z) = \frac{17}{2}\left(x^2 + y^2 + z^2\right) + 15xyz. \tag{21}$$

Equation (21) satisfies equations (18)–(20) and hence shows that the proposed model works well. If equation (21) is put in equations (18)–(20), then the left-hand side is equal to the right-hand side. Equation (21) is the exact solution of equations (18)–(20). So, it will satisfy for all values of the variables x, y, and z. It can be any real number. The threshold can be any real number for the variables x, y, and z.

The proposed features-based model accepts inputs of software for comparison of features of original and pirated software that is fully or partially pirated. This comparison can ultimately show the extent of piracy and changes done in security of end-to-end communication systems. In the current scenario of the case study, features of original software were extracted as shown in the top of Figure 2. Then features from pirated copies of software as shown in the right side of Figure 2 were extracted. A comparison of these features was done which is mathematically shown as equations (18)–(20) and their solution in equation (7). From the above description, it is clear that the proposed model

works very well; hence, piracy and changes can be found up to optimal level.

Furthermore, some other examples were tried to show the validity of the proposed method. These examples follow:

$$\frac{\partial x(x,y,z)}{\partial x} = 34xyz + 16y^2z + 2xz^2, \quad (22)$$

$$\frac{\partial x(x,y,z)}{\partial y} = 17x^2z + 32xyz + 2xz^2, \quad (23)$$

$$\frac{\partial x(x,y,z)}{\partial z} = 17x^2y + 16xy^2 + 4xyz. \quad (24)$$

We can find the numerical solution which always contains some error. The proposed mathematical model accepts features as input(s) shown in equation (2) to check the piracy and changes in security of features-based software birthmark. In the context of the current case study, features were extracted from multiple copies of the Android mobile application to show the piracy and security changes among multiple copies of the application.

5. Discussion

Industry of software development and end-to-end communication systems is using diverse approaches and methods to detect and identify the software piracy and assessment of security in end-to-end communication systems. Different techniques like watermarks, fingerprints, and digital signatures were used for showing the originality of the software, but these techniques have some limitations such as with the use of code obfuscation and semantic preserving transformation the watermarks and digital signature can be removed. Due to these limitations, the concept of software birthmark came into existence. The software birthmarks are considered to be of the utmost value and resilient to obliteration, and uniquely identify specific software. Software features are categorised into several categories. A program of software is a combination of several types of features of software. The investigation of code of a program, based on the defined features, resultantly supports the detection of similarity among more than one instance of seemingly the identical application of software. Such detection of similarity will eventually facilitate identifying and detecting the theft and piracy of software. The features-based birthmark of software provides further wide-ranging birthmark and hence representation of a software. The proposed differential-system-based mathematical model in this study using the idea of Eigen values and Eigen vector provides an exclusive solution for the features-based birthmark of software. This exclusive solution provides an unbiased measure for comparisons of features-based software birthmark that can be checked to piracy and assessment of security in end-to-end communication systems.

5.1. Threat to Validity. Software birthmark is the inherent characteristic of software used for the detection of theft in the software and can also be used for other purposes like to show the ownership of the software and detect the level of piracy in the software. So far, the existing literature was searched to analyze the existing efforts made in the area of software birthmark but maybe some work is missed due to the open access and availability of the research work. Validation of the work is also mandatory which is not mostly covered by this research and the work was validated through experts' opinion.

6. Conclusion

The proposed research work has presented a mathematical model based on differential system for comparisons of features-based birthmark of software and assessment of security in end-to-end communication systems. These comparisons of feature-based software birthmark will eventually find piracy and changes in security performed among the end-to-end communication systems. The main objective of the proposed study is to do comparisons of the feature-based software birthmark that was addressed by Nazir et al. [36]. The birthmark of software in terms of feature-based birthmark is categorised into different types. These categories include input features, functional features, and nonfunctional features. These features-based software birthmark categories are jointly known as software birthmark. This paper contributes to present a mathematical model based on differential system for the features-based software birthmark to support the comparisons of software birthmark to be checked for piracy and security assessment of end-to-end communication systems. The solutions of the differential equation as defined by using the idea of Eigen values designed for the feature categories of the birthmarks provide an unbiased measure and an effective means to compare birthmarks of software for the purpose of detecting piracy. Therefore, this comparison of model can make the process of software piracy and theft detection smooth and assesses the security of end-to-end communication systems.

Data Availability

No primary data were collected.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by Qatar National Library, Qatar.

References

[1] R. Thabit and B. E. Khoo, "Robust reversible watermarking scheme using Slantlet transform matrix," *Journal of Systems and Software*, vol. 88, pp. 74–86, 2014.

[2] Y. Zeng, F. Liu, X. Luo, and C. Yang, "Software watermarking through obfuscated interpretation: implementation and analysis," *Journal of Multimedia*, vol. 6, no. 4, 2011.

[3] F. Liu, B. Lu, and X. Luo, "A chaos-based robust software watermarking," in *Information Security Practice and Experience*, vol. 3903, pp. 355–366, Springer, Berlin, Germany, 2006.

[4] C. Collberg and T. R. Sahoo, "Software watermarking in the frequency domain: implementation, analysis, and attacks," *Journal of Computer Security*, vol. 13, no. 5, pp. 721–755, 2005.

[5] G. Myles and C. Collberg, "Software watermarking through register allocation: implementation, analysis, and attacks," in *Information Security and Cryptology–ICISC 2003*, vol. 2971, pp. 274–293, Springer, Berlin, Germany, 2004.

[6] C. Collberg, E. Carter, S. Debray, A. Huntwork, C. Linn, and M. Stepp, "Dynamic path-based software watermarking," in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 04)*, pp. 1–10, Washington, DC, USA, June 2004.

[7] G. e. Arboit, "A method for watermarking java programs via opaque predicates," in *Proceedings of the Fifth International Conference on Electronic Commerce Research (ICECR-5)*, pp. 1–8, Montreal, Canada, October 2002.

[8] R. Venkatesan, V. Vazirani, and S. Sinha, "A graph theoretic approach to software watermarking," in *Proceedings of the 4th International Information Hiding Workshop*, pp. 157–168, Pittsburgh, PA, USA, April 2001.

[9] J. P. Stern, G. e. Hachez, F. c. Koeune, and J.-J. Quisquater, "Robust object watermarking: application to code," in *Information Hiding*, vol. 1768, pp. 368–378, Springer, Berlin, Heidelberg, 2000.

[10] A. Monden, H. Iida, K.-i. Matsumoto, K. Inoue, and K. Torii, "A practical method for watermarking java programs," in *Proceedings of the 24th Computer Software and Applications Conference compsac2000*, pp. 191–197, Taipei, Taiwan, October 2000.

[11] C. Collberg and C. Thomborson, "Software watermarking: models and dynamic embeddings," in *Proceedings of the Conference Record of POPL '99: The 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 311–324, San Antonio, Texas, USA, January 1999.

[12] G. Qu and M. Potkonjak, "Analysis of watermarking techniques for graph coloring problem," in *Proceedings of the 1998 in Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design–ICCAD'98*, pp. 190–193, New York, NY, USA, November 1998.

[13] J. Pieprzyk, "Fingerprints for copyright software protection," in *Information Security*, vol. 1729, pp. 178–190, Springer, Berlin, Heidelberg, 1999.

[14] C. S. Collberg, C. Thomborson, and G. M. Townsend, "Dynamic graph-based software fingerprinting," *ACM Transactions on Programming Languages and Systems*, vol. 29, no. 6, p. 35, 2007.

[15] I. D. Baxter, A. Yahin, L. Moura, M. Sant'Anna, and L. Bier, "Clone detection using abstract syntax trees," in *Proceedings of the International Conference on Software Maintenance*, Bethesda, MD, USA, November 1998.

[16] D. Rattan, R. Bhatia, and M. Singh, "Software clone detection: a systematic review," *Information and Software Technology*, vol. 55, no. 7, pp. 1165–1199, 2013.

[17] H. T. e. al, "Detecting the theft programs using birthmarks," in *Information Science Technical Report*, Nara Institute of Science and Technology, Ikoma, Japan, 2003.

[18] H.-i. Lim, "Customizing k-gram based birthmark through partial matching in detecting software thefts," in *Proceedings of the IEEE 37th Annual Computer Software and Applications Conference Workshops (COMPSACW)*, pp. 1–4, IEEE, Japan, July 2013.

[19] Z. Xin, H. Chen, X. Wang et al., "attacks: automatically evading behavior-based software birthmark," *International Journal of Information Security*, vol. 11, no. 5, pp. 293–304, 2012.

[20] H. Chen, H.-i. Lim, S. Choi, and T. Han, "Detecting common modules in Java packages based on static object trace birthmark," *The Computer Journal*, vol. 54, no. 1, pp. 108–124, 2011.

[21] P. P. F. Chan, L. C. K. Hui, and S. M. Yiu, "Dynamic software birthmark for java based on heap memory analysis," in *Communications and Multimedia Security*, vol. 7025, pp. 94–107, Springer, Berlin, Heidelberg, 2011.

[22] Y. Mahmood, S. Sarwar, Z. Pervez, and H. F. Ahmed, "Method based static software birthmarks: a new approach to derogate software piracy," in *Proceedings of the 2nd International Conference on Computer, Control and Communication*, pp. 1–6, IEEE, Karachi, Pakistan, February 2009.

[23] S. Choi, H. Park, H.-i. Lim, and T. Han, "A static API birthmark for Windows binary executables," *Journal of Systems and Software*, vol. 82, no. 5, pp. 862–873, 2009.

[24] H. Park, S. Choi, H.-i. Lim, and T. Han, "Detecting java theft based on static API trace birthmark," in *Advances in Information and Computer Security*, vol. 5312, pp. 121–135, Springer, Berlin, Heidelberg, 2008.

[25] H. Park, S. Choi, H.-i. Lim, and T. Han, "Detecting code theft via a static instruction trace birthmark for Java methods," in *Proceedings of the 6th IEEE International Conference on Industrial Informatics*, pp. 551–556, Daejeon, South Korea, July 2008.

[26] H.-i. Lim, H. Park, S. Choi, and T. Han, "Detecting theft of java applications via a static birthmark based on weighted stack patterns," *IEICE Transactions on Information and Systems*, vol. E91-D, no. 9, pp. 2323–2332, 2008.

[27] J. Yang, J. Wang, and D. Li, "Detecting the theft of natural language text using birthmark," in *Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 1–4, Pasadena, CA, USA, December 2006.

[28] T. Kakimoto, A. Monden, Y. Kamei, H. Tamada, M. Tsunoda, and K.-i. Matsumoto, "Using software birthmarks to identify similar classes and major functionalities," in *Proceedings of the 2006 International Workshop on Mining Software Repositories*, Shanghai, China, 2006.

[29] G. Myles and C. Collberg, "Detecting software theft via whole program path birthmarks," in *Information Security*, vol. 3225, pp. 404–415, Springer, Berlin, Heidelberg, 2004.

[30] A. Aiken, *Moss: A System for Detecting Software Plagiarism*, University of California, Berkeley, CA, USA, 1994.

[31] G. Whale, "Identification of program similarity in large populations," *The Computer Journal*, vol. 33, no. 2, pp. 140–146, 1990.

[32] M. J. Wise, "Detection of similarities in student programs: yap'ing may be preferable to plague'ing," in *Proceedings of the 23rd SIGCSE Technical Symposium*, St. Louis, Missouri, USA, March 1992.

[33] S. Schleimer, D. Wilkerson, and A. Aiken, "Winnowing: local algorithms for document fingerprinting," in *Proceedings of 2003 SIGMOD Conference*, San Diego, CA, USA, June 2003.

[34] Z. Tian, Q. Zheng, T. Liu, M. Fan, X. Zhang, and Z. Yang, "Plagiarism detection for multithreaded software based on thread-aware software birthmarks," in *Proceedings of the 22nd International Conference on Program Comprehension*, Hyderabad, India, May 2014.

[35] Z. Tian, Q. Zheng, T. Liu, and M. Fan, "DKISB: dynamic key instruction sequence birthmark for software plagiarism

detection," in *Proceedings of the IEEE International Conference on High Performance Computing and Communications & IEEE International Conference on Embedded and Ubiquitous Computing*, pp. 619–627, Zhangjiajie, China, November 2013.

[36] S. Nazir, S. Shahzad, Q. U. A. Nizamani, R. Amin, M. A. Shah, and A. Keerio, "Identifying software features as birthmark," *Sindh University Research Journal*, vol. 47, no. 3, pp. 535–540, 2015.

[37] S. Nazir, S. Shahzad, S. A. Khan, N. Binti Alias, and S. Anwar, "A novel rules based approach for estimating software birthmark," *The Scientific World Journal*, vol. 2015, Article ID 579390, 8 pages, 2015.

[38] D. Lee, Y. Choi, Y. Choi, J. Jung, J. Kim, and D. Won, "efficient categorization of the instructions based on binary excutables for dynamic software birthmark," *International Journal of Information and Education Technology*, vol. 5, no. 8, pp. 571–576, 2015.

[39] S. Won, S. Shahzad, and S. B. S. Abid, "Selecting software design based on birthmark," *Life Science Journal*, vol. 11, no. 12s, pp. 89–93, 2014.

[40] J. Choi, Y. Han, S.-j. Cho, HaeYoungYoo, and J. Woo, "A static birthmark for MS Windows applications using import address table," in *Proceedings of the Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 129–134, Taichung, Taiwan, July 2013.

[41] BSA, *The Compliance Gap BSA Global Software Survey*, Business Software Alliance, Washington, DC, USA, 2014.

[42] H. Tamada, M. Nakamura, and A. Monden, "Design and evaluation of birthmarks for detecting theft of Java programs," in *Proceedings of IASTED International Conference on Software Engineering*, pp. 569–575, Innsbruck, Austria, February 2004.

[43] Y. Zeng, F. Liu, X. Luo, and S. Lian, "Abstract interpretation-based semantic framework for software birthmark," *Computers & Security*, vol. 31, no. 4, pp. 377–390, 2012.

[44] Y. Mahmood, Z. Pervez, S. Sarwar, and H. F. Ahmed, "Similarity level method based static software birthmarks," in *Proceedings of the High Capacity Optical Networks and Enabling Technologies*, pp. 205–210, Penang, Malaysia, November 2008.

[45] Y. Wang, F. Liu, Z. Zhao, B. Lu, and X. Xie, "Operand stack dependence based java static software birthmark," in *Proceedings of the 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Shenyang, China, July 2013.

[46] X. Xie, F. Liu, B. Lu, and L. Chen, "A software birthmark based on weighted k-gram," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent System (ICIS)*, pp. 400–405, Xiamen, China, October 2010.

[47] G. Myles and C. Collberg, "K-gram based software birthmarks," in *Proceedings of the 2005 ACM Symposium on Applied Computing*, Santa Fe, New Mexico, 2005.

[48] S. Nazir, S. Shahzad, R. Wirza et al., "Birthmark based identification of software piracy using Haar wavelet," *Mathematics and Computers in Simulation*, vol. 166, pp. 144–154, 2019.

[49] S. Shahzad, S. Shahzad, and N. Mukhtar, "Software birthmark design and estimation- A systematic literature review," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3905–3927, 2019.

[50] K. Fukuda and H. Tamada, "A dynamic birthmark from analyzing operand stack runtime behavior to detect copied software," in *Proceedings of the 2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 505–510, Honolulu, HI, USA, July 2013.

[51] Y. Bai, X. Sun, G. Sun, X. Deng, and X. Zhou, "Dynamic k-gram based software birthmark," in *Proceedings of the IEEE ASWEC 2008 19th Australian Conference*, Perth, WA, Australia, March 2008.

[52] D. G. Zill and M. R. Cullen, *Differential Equation S with Boundary Value Problem*, Brooks/Cole Cengage Learning, Boston, MA, USA, 7th edition, 2009.

[53] R. L. Burden and J. D. Faires, *Numerical Analysis*, Brooks/Cole, Cengage Learning, Boston, MA, USA, 9th edition, 2011.