Lecture Notes in Electrical Engineering 1049

Udai Pratap Rao
Mamoun Alazab
Bhavesh N. Gohil
Pethuru Raj Chelliah *Editors*

Security, Privacy and Data Analytics

Select Proceedings of the 2nd International Conference, ISPDA 2022

Springer

Lecture Notes in Electrical Engineering

Volume 1049

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

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

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

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

China

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

India, Japan, Rest of Asia

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

Southeast Asia, Australia, New Zealand

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

USA, Canada

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

All other Countries

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

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

Udai Pratap Rao · Mamoun Alazab ·
Bhavesh N. Gohil · Pethuru Raj Chelliah
Editors

Security, Privacy and Data Analytics

Select Proceedings of the 2nd International Conference, ISPDA 2022

Springer

Editors

Udai Pratap Rao
Department of Computer Science
and Engineering
National Institute of Technology Patna
Bihar, India

Bhavesh N. Gohil
Department of Computer Science
and Engineering
Sardar Vallabhbhai National Institute
of Technology
Surat, India

Mamoun Alazab
Faculty of Science and Technology
Charles Darwin University
Northern Territory, NT, Australia

Pethuru Raj Chelliah
Reliance Jio Platforms
Bangalore, India

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-99-3568-0 ISBN 978-981-99-3569-7 (eBook)
https://doi.org/10.1007/978-981-99-3569-7

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

Preface

With the advancements in computing and networking facilities, the day-to-day activities are transforming online. The information generated in these online facilities demand strong Security and Privacy guarantees. The research problems pertaining to security, privacy and data analytics are covered in this book. This book constitutes the referred proceedings of Second International Conference on Security, Privacy and Data Analytics, ISPDA 2022 organised by Computer Science and Engineering Department, Sardar Vallabhbhai National Institute of Technology, Surat, India in December 2022. The proceedings cover recent contributions and novel developments from researchers across industry and academia who are working in the field of security, privacy and data analytics from technological and social perspectives. The salient features of this book are:

- Coverage of the novel technological advancements in Security and Privacy domain.
- Coverage of the novel technological advancements in Data Analytics domain.
- Discussion of Theoretical and Empirical studies on Security, Privacy and Data Analytics.

This book will emerge as a valuable reference for researchers, instructors, students, scientists, engineers, managers and industry practitioners. The ISPDA series proceedings' content will be useful for different researchers and practitioners at a variety of learning levels.

Bihar, India	Udai Pratap Rao
Northern Territory, Australia	Mamoun Alazab
Surat, India	Bhavesh N. Gohil
Bangalore, India	Pethuru Raj Chelliah

Contents

About the Editors

Udai Pratap Rao (Senior Member, IEEE) received a Ph.D. in Computer Engineering from Sardar Vallabhbhai National Institute of Technology, Surat (Gujarat), India, in 2014. He has been working as an Associate Professor in the Department of Computer Science and Engineering at the National Institute of Technology Patna since Nov 2022. Before joining the National Institute of Technology Patna, he was associated as an Assistant Professor with Sardar Vallabhbhai National Institute of Technology, Surat, in the Department of Computer Science and Engineering for more than 15 years. His research interests include information security & privacy, privacy in location-based services, big data privacy, security, and trust management in online social networks (OSNs), security and privacy in IoT and cyber-physical systems, and distributed computing. He has published about 90 papers extensively in journals, book chapters, and refereed conference proceedings. He has supervised 05 Ph.D. theses in the fields of data privacy, IoT security, and Security and Trust Management in OSN. He has also supervised a Post Doctoral Fellow (PDF) under ISEA Project Phase II in the Department of Computer Engineering at SVNIT Surat. He is currently the PI of the research project entitled “Design and Implementation of Secure Service and Attribute-based Authorization Model in Dynamic and Constrained-specific IoT Environment” funded by IHUB NTIHAC Foundation, IITK, under the aegis of the National Mission on Interdisciplinary Cyber-Physical System (NM-ICPS), Department of Science and Technology, Government of India. He was the Principal Investigator (PI) of the Micro Research project entitled “Investigating Light-Weight Cryptography Algorithms and Its Application to Various IoT Devices” funded by TEQIP-III from July 2019 to Jan 2021. He also acted as the Chief Investigator of the “Information Security Education and Awareness Project Phase II” project from July 2018 to Dec 2019, funded by the Ministry of Electronics and Information Technology (MeitY) Govt. of India. He has also edited two books published by reputed international publishers.

Mamoun Alazab is a full Professor at Faculty of Science and Technology, Charles Darwin University, Northern Territory, Australia. He received his Ph.D. degree in Computer Science from the Federation University of Australia, School of Science, Information Technology and Engineering. He is the recipient of the prestigious award: NT Young Tall Poppy (2021) of the year from the Australian Institute of Policy and Science (AIPS), and the Japan Society for the Promotion of Science (JSPS) fellowship through the Australian Academy of Science. He published more than 300 research papers in many international journals and conferences, more than 100 in IEEE/ACM Transactions, 11 authored and edited books, and 3 patents. He is a Senior Member of the IEEE and the founding chair of the IEEE Northern Territory (NT) Subsection. He serves as the Associate Editor of IEEE Transactions on Computational Social Systems, IEEE Transactions on Network and Service Management (TNSM), IEEE Internet of Things Journal, ACM Digital Threats: Research and Practice, and Complex & Intelligent Systems.

Bhavesh N. Gohil is an Assistant Professor at the Department of Computer Science and Engineering, Sardar Vallabhbhai National Institute of Technology (SVNIT), Surat. He received a Ph.D. degree from the same institute and his research interests include Security and performance issues in Distributed/Cloud/Fog/Edge/Mobile systems/computing.

Pethuru Raj Chelliah is a chief architect and vice president in the Edge AI division of Reliance Jio Platforms Ltd. Bangalore. He has over 23 years of IT industry experience and 08 years of research experience. He completed his Ph.D. degree at Anna University, Chennai, and continued with the UGC-sponsored postdoctoral research in the Department of Computer Science and Automation, Indian Institute of Science (IISc), Bangalore. He was granted a couple of international research fellowships (JSPS and JST) to work as a research scientist for 3.5 years in two leading Japanese universities. He has published over 45 research papers in peer-reviewed journals, authored/edited 43 books, and contributed 20 book chapters for various technology books edited by highly acclaimed and accomplished professors and professionals. His research interests focus on emerging technologies such as IoT, artificial intelligence, big and fast data analytics, blockchain, digital twins, cloud-native computing, edge/fog clouds, reliability engineering, microservices architecture, event-driven architecture (EDA), etc.

Recent Advancements on Energy-Saving LEACH Protocol in Wireless Sensor Network—Review

Bhupesh B. Lonkar and Swapnili Karmore

Abstract Energy-efficient routing is the biggest challenge for the researcher to implement in the various wireless networks. Nowaday, Wireless Sensor Network (WSN) has a popular choice for researchers to implement energy-efficient routing techniques. WSN is using energy-efficient routing protocols to decrease sensor node energy and improve network lifetime. Clustering is the best approach to consume less power and enhances speed. Researchers worked on an energy-efficient technique called Low Energy Adaptive Clustering Hierarchy (LEACH). This technique obtains few defects on the initial state for random selection of Cluster Head (CH). LEACH has several disadvantages like cluster head selection while each round consumes more energy of node, non-consideration surplus energy. LEACH is most frequently used in clustering routing algorithms with certain limitations, so the researchers are focusing on improving the LEACH algorithm and creating more efficiency for practical use. The main issues of LEACH inequality of the amount of CHs and forgetting to unused energy of nodes. That's why researchers find the solution to propose a new energy-efficient routing protocol, or enhance the existing routing protocol to increase network/battery lifetime. Therefore, we studied a literature survey on best routing techniques like LEACH and energy-efficient LEACH clustering protocols. Also, we give a comparative analysis of the Enhancement of Energy Efficient Routing Protocol (EERP) of LEACH in WSN and elaborate it.

Keywords Energy-saving · Energy-efficient · LEACH · EERP · EERP-LEACH · WSN

B. B. Lonkar (✉)
Department of Computer Science and Engineering, G. H. Raisoni University, Saikheda, India
e-mail: bhupesh.lonkar@gmail.com

S. Karmore
Department of Data Science, G. H. Raisoni Institute of Engineering Technology, Nagpur, India

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_1

1 Introduction

Wireless sensor network has larger sensor nodes, limited power, enumerate and communication capabilities. These sensor nodes are placed in a large space with one or more base stations [1]. LEACH is a remarkable directing convention in WSN. It is a Grouping-based show, which helps in improving the lifetime of a far-off sensor organization. This task is studying different directing rules and presuming that the cooperative timetable is used to improve the grouping in steering conventions in WSN [2]. The routing protocols are responsible to select the optimum path for forwarding data among the networks. A recent advancement done in new routing techniques depends on network structure and applications. It proposed new energy-efficient routing techniques for improving network performance [3].

WSNs are divided into two categories: proactive and reactive sensor networks. In proactive sensor networks, the sensors control the nodes at continuous intervals in which the transmitter detects the atmosphere and sends data to sensors [4]. The applications used in proactive sensor networks require surveillance data in a fixed interval. In reactive sensor networks, the node is responsible for some event that occurs at the end of communication [5].

WSN term is derived from devices like mobile phones, laptops, PDAs, etc. Today, most wireless sensor devices, control, modest technologies, computing power, memory, competencies, etc. Most of the researchers' studies focused on designing energy-efficient and computationally intensive reliable algorithms or protocols [6].

In WSN, nodes may be static and dynamic. In Static Mode, the position of fixed nodes accomplished the network lifetime. In Dynamic Mode, free nodes of the network change their location from one end to another. The sensor nodes are transmitting data specifically or in between the network and data collected to the BS [7]. Mobilization is performing a difficult job due to uncertain activities to nodes in the network. The networks identify themselves of synchronization make communication, accuracy and consonant [8].

WSN is one of the most imminent, favorable, and prime technology enabled with various applications like location monitoring, traffic surveillance and controlling, target tracing, climate monitoring, disaster relief, etc. [9]. The quality of communication is the prime aspect of self-organized WSN which can calculate energy efficiency, security, robustness, and network lifetime. Among these, most of the researchers worked in energy consumption and network lifetime [10]. In WSN, data will consume more energy when it does for transmission and reception compared with sensing and processing data. Therefore, optimization techniques are used to increase the durability of the network, and minimized energy consumption [11].

In WSN, sensor nodes have limited energy resources, storage capacity, and weak computing power. In most of the cases, it is not possible to recover or recharge batteries due to higher node quantity and the problem of the environment [12]. The challenging task for researchers is to optimize battery energy through an actual energy-aware protocol. The researchers have suggested several protocols and algorithms for the direction of energy-aware clustering in WSN [13].

The rest of the paper has been organized as follows: In Sect. 2, we discuss different literature surveys on energy-efficient routing techniques like LEACH. In Sect. 3, we discuss different literature surveys on recently defined routing algorithms. Section 4 gives a comparative study on defining routing algorithms and routing protocols on the main two parameters of power consumption and a lifetime of the network. Finally, in Sect. 5, we define the result archived of all defined routing algorithms.

2 Discussion on Literature Survey Based on Routing Technique

Very few routing algorithms can work to improve the energy and speed of WSN. Here, we studied some energy-efficient routing algorithms like LEACH.

2.1 Low-Energy Adaptive Clustering Hierarchy (LEACH)

Heinzelman et al. [14] and Qubbaj et al. [15] proposed the LEACH protocol utilizes grouping topology to collect a large number of data and these transfer it to BS. It is adjustable and also well-organized. The sensor nodes are divided into groups including fixed-size CHs selected from each cluster. In the TDMA Schedule, the nodes collect actual data and send it to specific cluster heads. If a node has carried on being a cluster head, then it spends more energy until the node expires shortly. Thus, to avoid such a situation, the cluster head changed the operation after every round. SNs transmit information to their respective CH with the same collected through CH from every node and compute the quantity of the collected data and the same data is sent to the sink node.

LEACH is the first clustering protocol used for the selection of CH. It used random-based techniques for cluster CH election, cluster generation, random distribution, and self-organization. The main objective of LEACH is power utilization and increases network lifetime. The main disadvantage of the LEACH protocol is the random election of CH. This is an uneven election of CH to effectively utilized high power for clusters with huge and low amounts of nodes in clusters. It is hierarchical protocol cluster-based routing in which every mode sends more information to the CHs; the CHs are grouped together and minimize data size and are forwarded to the BS.

In the setup phase, the member nodes will automatically establish the cluster according to the rules of protocol. Each cluster selects a CH, and the remaining nodes are treated like member nodes. The work of the member nodes is to collect all information and transmit cluster information to the CH. CH is responsible to assign the time slab in the cluster, accepting and initially processing information in the cluster, setting up the route between clusters, and circulating the true information to BS.

3 Discussion on Literature Survey Based on Routing Algorithm

Thi Quynh and Viet [16] suggested two algorithms: K-means algorithm used for clustering/grouping and the bat algorithm (BA) for selecting CHs. In the LEACH protocol, sensor nodes classify in the cluster, and it elects any one SN to act as CH randomly. The Clusters work as a mediator for collecting data from SNs and transmit it to BS through CH, so LEACH is helpful to enhance network lifespan along with decreasing the power consumption of every node. The main drawback of the LEACH is it does not observe the power of the present node, and randomly selects cluster heads giving unequal power consumption in the nodes of the network. To overcome such a problem, it's improving LEACH protocol. It is called the proposed BA-LEACH routing algorithm. In this algorithm, BA assigns the sensor node as CH when it finds the minimum distance and power consumption. The K-means algorithm divides sensor nodes within clusters that are each sensor node be part of a cluster with the cluster centers. Therefore, combined BA and K-means introduced the BA-LEACH algorithm. This algorithm is better than PSO-LEACH in terms of on speed, stability, and perfection. BA-LEACH is used Particle Swarm Optimization (PSO) to improve the LEACH protocol. In this paper, simulation analysis of BA-LEACH can be decreasing the power consumption of networks and increase the lifespan in wireless sensor nodes. BA-LEACH is extending the life of WSNs as compared with LEACH and PSO-LEACH.

The study by Saleh et al. [17] aimed to intensify LEACH searching a cluster head at a short stage of energy consumption. LEACH protocol has some drawbacks at the initial phase of a random selection of CH. To overcome this drawback of LEACH, it improved the CH selection technique. The proposed IE^2-LEACH initiates the new approach to determine CH rather than random CH selection. Also, the CH selection in IE^2-LEACH is more accurate than LEACH based on energy saving. To discover CH in various parameters like the number of transmitting packets, transmission media properties, physical distance, the number of nodes in the transmission route, etc., these parameters are enough for power consumption that's not enough for the selection of CHs. To enhance the life span of the CHs, IE^2-LEACH excludes the unnecessary selection of CHs and states the badness of CHs. Therefore, the geographic location of each header is considered a factor of IE^2-LEACH. Afterward, the usages of the sleeping-waking scheduling technique reduce the decease probability of CHs and enhance power utilization by reducing unnecessary data sent to BS. Therefore, the simulation work shows IE^2-LEACH performs better than E-LEACH and LEACH in reducing power utilization and increasing network lifetime.

The work of El Idrissi et al. [18] introduces the latest clustering algorithm to the increased lifetime of the network and intensifies power efficiency in WSN, called Optimal Selection of Cluster Head-Grid6 (OSCH-G6). The SNs are arranged randomly in the coverage space. The nodes received data moving through cluster head to BS. The proposed OSCH-G6 protocol split the network area within the Cluster grid. Every Cluster grid selects a cluster head based on unused power and finds each node's distance to the BS. The main difference between the proposed and existing LEACH protocol is the formation of clusters. The proposed protocol constructs clusters in the form of a grid and the existing one constructs a cluster in each round. If some nodes are dead, then it becomes large. Due to the long distance between nodes of cluster heads, it consumes more energy. Therefore, results derived by the OSCH-G6 protocol are more successful than LEACH to improve network lifetime and decrease energy consumption.

The work of El Khediri et al. [19] introduced the novel technique of Minimum Weight Low Energy Adaptive Clustering Hierarchy (MW-LEACH). The MW-LEACH protocol is an identical clustering technique to decrease power consumption and improved the lifetime of the network. Consequently reducing energy and distance between nodes, it sends data through uniform clustering and multi-hop techniques in WSN. It calculates distances together with nodes, and selects cluster head nodes and the rest of the power. Also, it calculates the number of nodes for each cluster head. The prime process constructs with an initial place of cluster head successor. The member nodes transfer the data to the successors and send the same to BS in different directions. Therefore, the specified approach can have fewer complexes in terms of message and time.

El Aalaoui and Hajraoui [20] proposed a modern methodology to enhance the performance of the LEACH protocol and introduced the algorithm Organized LEACH (O-LEACH). It determines each CH and connected nodes, then finds which of them is nearer to the base station. If someone else node has smallest distance to BS becomes new CH plus broadcast advertisement message send to the earliest CH to switch the position of the regular node. It informed the rest of the group nodes to ask to join the current CH. If the selected CH is the closest, it remains constant. It balanced the energy distribution for all nodes in the cluster and reduced energy loss at the time of communication of the network. The proposed technique focused on the correct option of CH. The proposed approach has compared LEACH with specified parameters like network lifetime, power utilization, and total packets transferred to BS.

The work of Akleek et al. [21] proposed a technique of AI to enhance LEACH using a super-cluster head (SCH) and combine improvement with the Consume Power Fairly (CPF) protocol with a proposed enhancement for obtaining the best route from source to destination. As per the enhancement in LEACH, at the same time, it utilized both super-cluster head (SCH) and CPF. SCH played a key in transferring data from regular CHs to BS. If the distance of CHs is higher than SCH, then it consumes more energy. Therefore, the CPF protocol obtained the best route from a source to a destination node like SCH. It used a multi-hop routing technique between CH with SCH has to do with CPF to obtain the best path that depends on the equal energy distribution WSN nodes. That is enough approach to accomplished retaining energy for the entire network. The initiative work proved enhancement in the network lifetime.

The study by Kumar et al. [22] suggested changes in the well-known energy-efficient clustering protocol like LEACH and Modified-LEACH (M-LEACH), defined as ACHs-LEACH. Using multi-hop techniques, the cluster head gathers information from the nodes and transmits it via Assistant Cluster Head (ACH) to the sink node. The cluster head observes information through its correlated nodes and transfers the same to ACH. The main objective of ACH interchanges the information between cluster head and neighbor nodes. The proposed work focused on various factors like energy utilization, bit rate, data loss, and data transmission rate. Therefore, the proposed technique worked for decreasing power consumption for the selection of cluster head and assistant cluster head so that it can improve network lifetime.

The work of Us Sama et al. [23] proposed a modification in the LEACH protocol, named Energy-efficient Least Edge Computation routing protocol (ELEC). ELEC-LEACH worked in two states: the first is Setup and the second is Steady. In the setup state, both ELEC and LEACH are the same. The CHs are selected by the same process as in LEACH. In the steady state, the ELEC-LEACH goes along with the route process technique of the ELEC routing protocol. CHs send collected data through multi-hop to the base station and choose neighbors similar to that used in the ELEC routing protocol. With the help of route processing, the ELEC generates a specific route table conforming to essentially obtain the events. The sensor nodes accept a particular route table that recognized adjacent events and sends them to the CH by using a single or multi-hop technique based on distance. If data are far away from the CH, then sensor nodes will transfer data through the multi-hop techniques or directly send it to the cluster head. After collecting all information, CH sends the information to the BS through multi-hop. Therefore, the source CH elects the neighboring hop CH with the smallest value edge count, energy level, and link weight. The proposed work shows the ELEC-LEACH protocol is more efficient than the MR-LEACH focused on the parameters like improved network lifetime, unused energy, the reduction ratio of failed nodes, and a packet drop. Also, the simulation result of the proposed protocol proved that it increased network lifetime by double that of the MR-LEACH protocol.

The work of Panchal et al. [24] proposed a Residual Energy-based Cluster Head Selection in LEACH (RCH-LEACH) algorithm that balanced cluster formation in the network. This algorithm used extra parameters like threshold energy, node surplus energy, and prime number of clusters in the cluster head selection process. The selected parameters update their information after each round. In RCH-LEACH, the sensor nodes will send their information (location and surplus energy) to BS in each round. Then, BS immediately begins the CH selection process. Every sensor node is defined a random value in terms of 0 or 1. In the beginning, the base station identifies the threshold value of each live node, then compares both random and threshold values. If the threshold value is greater than the random value of the node, it is called G-node. Thus, it counts the number of G-nodes and compares the value of clusters. When the G-node value exceeds the number of cluster values, then it finds the optimum number of G-nodes depending on the higher value of surplus energy. So, it considers it as CH. After obtaining CH, BS sends a broadcast message to the network, and nodes will receive the id of the newly selected CH. Then all non-CH nodes will be selecting the nearest CH depending on the received signal strength from CH. Therefore, the proposed technique will save energy and enhance the performance of the network.

The work of Sahib et al. [25] suggested hierarchical energy-efficient routing protocols like LEACH, HEEP, TEEN, and PEGASIS. The proposed work specifies the LEACH protocol which CH election procedure is called "round" (r). The structure of specific round and sensor networks is defined in a balance state. Now, the selection of CH utilized a hierarchy low endurance change. The sensor hubs select their lead unexpected, such factor likes a channel best sign's. The new hub is defined as CHs to reduce the intensity used of CH finally. Typical CHs obtain for evaluating standard utilization intensity with sensor hubs. In the proposed work, the researcher concentrated Energy Model (EM) in the LEACH protocol. Therefore, The LEACH protocol includes EM in CH and non-CH nodes to balance network energy.

The directive provides stability of the energy utilization in the wireless sensor network study of Umbreen et al. [26] introduced the Energy-Efficient Mobility-based Cluster head Selection (EEMCS) protocol. CH selection is the main factor for sensor energy consumption. Each node weightage has computed the nodes strength, surplus energy, calculating base station distance, and neighbor node weights. In EEACH, cluster heads are elected by considering remarkable parameters like the remaining node's energy, mobility, BS, distance, and neighbor counts. After determining all factors of selecting the CH, it evaluates each node against a threshold value. The energy threshold is criteria for the selection of cluster heads. When all nodes are satisfied with the above criteria, then it selects a cluster head. The researcher finds the results in design protocol EEMCS worked exceptionally superior to current routing protocols like CRPD, LEACH, and MODLEACH. The proposed protocol helps the network adjust the load, network balancing, energy consumption, and throughput. EEMCS utilized less energy and increased network lifetime as compared to current routing protocols.

The work of Takele et al. [27] proposed an algorithm for improving LEACH. In a study of modified LEACH, the first round will form the clusters, and nodes will be coordinated to support being CH on each round shift by shift continuously. The organized nodes will consequently turn as a CH up to the total energy of its two-thirds energy consumed. After two-thirds energy consumed by the cluster, its reforms and removes low energy with expired nodes from the organized time frame. If the coordinated node is dead ahead of it serving as CH, the substitute will help the CH. It introduced the new work on upgrading the LEACH routing protocol focused on decreasing energy dissipation and redundant overload of cluster heads.

To balance energy in WSNs, Zhang et al. [28] introduced a clustered-based cross-layer optimization technique. In this technique, the physical layer provides the corresponding information to the network layer and data link layer to send packets efficiently on the network. The cross-layer optimization technique proposed an energy-efficient cross-layer optimization routing algorithm called LEACH Cross-layer Optimization (CLO). The LEACH-CLO is similar to the classic LEACH algorithm. The proposed LEACH-CLO performs six stages in each round. The first stage is initialization, second CH selection, third clustering, fourth inter-cluster communication, fifth intra-cluster communication, and sixth energy control. In the first stage, the BS broadcasts the message to whole nodes of the network. Each node stored the calculated distance from BS and the remaining energy of the node. In the second stage, the node along with high surplus energy gets a chance of cluster head. In the third stage, the cluster will be established when it broadcasts information sent by the cluster head and information received by member nodes, and then joins CH and member node finalized cluster. In the fourth stage, member nodes want to send observed data and transfer data to CH via an agent, and the CH node collects data. In the fifth stage, Inter-cluster used the multi-hop technique because the distance between CH and BS was far away with data sent to BS, and it used more transmission power to consume more energy. In the sixth stage, the forwarding distance defines the optimal transmit energy after routing. Therefore, the simulation results of the proposed algorithm could efficiently reduce and balance energy from the WSN compared to the existing algorithm.

The work of Dhami et al. [29] proposed an approach that improves throughout the execution in WSN, and introduced an energy-saving genetic algorithm from Virtual Grid-based Dynamic Routes Adjustment (VGDRA). In the existing technique, the medial node is selected as CH, and the node restricts data transmission in the network. This node absorbs more power instead of another node shown to the expired node. Whenever it observes a straight path route is overloaded and consumed more energy to reduce the network lifetime. To overcome such a problem, the researchers used the VGDRA algorithm. The proposed genetic algorithm works on four principles such as selection, assessment, crossover, and mutation. In the first step, the virtual grid represents an area divided into equal and constant size blocks. Also, it split the same number of nodes into all blocks. The grid algorithm assigned input to create a population and announced the number of iterations and population size. This process can assess the strength operation for iteration and find the best path. In the second step of creating a population, each cluster selected the nodes and defined the route. With the help of total energy and distance, the researchers are analyzing strength operation, nodes energy, and location from source to destination with high strength. In the third step, it used crossover to examine crossover output. If crossover output is higher than the initial strength, then save or else implement the mutation. In the fourth step, it calculates mutation strength and distance and finds output is maximized. The proposed genetic algorithm is the best energy-saving protocol compared to the LEACH protocol.

Ghosh and Misra [30] introduced three clustering protocols LEACH, E-LEACH, and modified Enhance Energy-Efficient Adaptive Clustering (m-EEEAC). An upgrade type of LEACH is the m-EEEAC protocol. All nodes transferred residual energy value to BS. Every node used residual energy over adaptive clustering, and it produces constant CHs over the network. BS selects CHs and broadcasts messages to all clusters. The base station selects the top three values of residual energy nodes and broadcasts their node id. These types of processes decrease the rate problem, volume of sensor nodes, and non-operating Global Positioning System in the indoor surroundings. A proposed m-EEEAC protocol is well-defined for an improvised selection of cluster heads, decreases network overhead, and enhances the network lifetime. This expands the strength for increasing live nodes in the network.

Taj and Kbir [31] aimed to enhance the multipath LEACH protocol named Intermediate Cluster Head (ICH-LEACH). The main drawback of LEACH is CH sends data directly to the base station. It uses high energy to transmit data. If BS is out of transmission range, then it's possible to lose the data. Therefore, the researchers introduced ICH-LEACH. In ICH-LEACH, if CH is far away from BS, it makes sure to consume less energy in each round and is then transmitted. So the intermediate CH selects the actual position of the CH. Therefore, each CH has a neighbor intermediate CH. This technique worked in three phases such as initialization, cluster setup, and steady. So in the first phase, BS broadcasts the location to all member nodes of the network. Then these location coordinates are received and stored in nodes. So that the location coordinate of a base station used by the selected CH and determine either send data directly to BS or not. In the second phase, each CH recommended itself as a member node, sending its coordinate and distance information from BS to other CH. Immediately after, the CH received the location coordinates, and signal strength and stored them with BS information. Therefore, each CH will store the information to the existing CH in the routing table and create once in every round and update it when received from the neighbor CH. Now in the last phase, if BS is closest to the member node, then data is sent directly to the BS. Rather, it will select the distance from intermediate CH to BS, not more than the distance from BS. A proposed ICH-LEACH protocol expands network lifespan and transfer new data compared with existing LEACH. Therefore, simulation results show this proposed ICH-LEACH protocol increases the lifetime of networks.

4 Comparative Study

In this section, Tables 1 and 2 show the classification of a summary of the literature review, comparative analysis of EERP, and enhancement in Energy-Efficient Routing Protocol (EERP)-LEACH on various parameters.

5 Result Achieved

As per Table 1, Fig. 1 shows the year-wise comparative analysis of EERP-LEACH protocols and techniques to enhance network lifetime. The researcher has improved the above LEACH protocols to enhance the network lifetime. The *x*-axis shows Enhanced LEACH protocols and the *y*-axis shows network lifetime. These kinds of protocols not only save energy but also enhance the lifetime of the network. This is a comparative analysis showing the year-wise growth in the LEACH protocol to improve the network lifetime.

As per Table 1, Fig. 2 shows the year-wise comparative analysis of EERP-LEACH protocols and techniques for energy consumption. The researcher has improved the above LEACH protocols for energy consumption. The *x*-axis shows Enhanced

Table 1 Summary of literature review

Year	Name of author	Literature survey based on	Methodology	No. of nodes utilized	Maximum number of round taken	Total count of dead nodes	Energy consumption (J) (%)	Network lifetime extended span (%)
2021	Trang Pham Thi Quynh	Algorithm	BA-LEACH	100	3000	100	50	51.23
2021	Safa'a S. Saleh	Protocol	IE^2-LEACH	100	1200	100	50	75
2020	Nezha El Idrissi	Protocol	OSCH-G6	100	4000	100	50	50
2020	Salim EL Khediri	Protocol	MW-LEACH	300	5000	300	60	60
2020	Abderrahmane El Aalaoui	Protocol	O-LEACH	500	25	500	20	44
2020	Fatima Abu Akleek	Technique	SCH and CPF	100	4500	100	20	40
2020	Naveen Kumar	Protocol	ACHs-LEACH	82	82	82	41	66
2020	Najm Us Sama	Protocol	ELEC-LEACH	500	500	500	60	58
2020	Panchal	Algorithm	RCH-LEACH	150	3000	150	30	20
2020	Haneen Ali Sahib	Protocol	Block-based routing	100	1548	100	50	29

(continued)

Table 1 (continued)

Year	Name of author	Literature survey based on	Methodology	No. of nodes utilized	Maximum number of round taken	Total count of dead nodes	Energy consumption (J) (%)	Network lifetime extended span (%)
2020	Sehar Umbreen	Protocol	EEMCS	250	1335	250	88	18.72
2019	Atallo Kassaw Takele	Protocol	M-LEACH	100	38	100	30	15.27
2018	Wenbo Zhang	Technique	LEACH-CLO	100	200	100	20	10
2018	Mandeep Dhami	Algorithm	VGDRA	180	3500	180	60	3.08
2017	Sreya Ghosh	Protocol	m-EEEAC	100	100	100	60	3
2016	M. Bennani Mohamed Taj	Protocol	ICH-LEACH	100	40	100	51	1.27

Table 2 Comparative analysis of enhancement in energy-efficient routing protocols (EERP)–LEACH

Routing protocol	Type of protocol	Network lifetime	Power consumption	Data delivery model	Specific path
BA-LEACH	Hierarchical	High	Low	Cluster	Yes
IE^2-LEACH	Hierarchical	Very high	Low	Cluster	Yes
OSCH (G6)	Hierarchical	High	Low	Cluster	Yes
MW-LEACH	Hierarchical	High	Very high	Cluster	Yes
O-LEACH	Hierarchical	Low	Very low	Cluster	Yes
SCH and CPF	Hierarchical	High	Very low	Cluster	Yes
ACHs-LEACH	Hierarchical	High	Very high	Cluster	Yes
ELEC-LEACH	Hierarchical	Very high	Very high	Cluster	Yes
RCH-LEACH	Hierarchical	High	Very low	Cluster	Yes
Block-based routing	Hierarchical	High	Low	Cluster	Yes
EEMCS	Hierarchical	High	Very high	Cluster	Yes
M-LEACH	Hierarchical	High	Very low	Cluster	Yes
LEACH-CLO	Hierarchical	High	Very low	Cluster	Yes
VGDRA	Hierarchical	High	Very high	Cluster	Yes
m-EEEAC	Hierarchical	High	Very high	Cluster	Yes
ICH-LEACH	Hierarchical	High	Low	Cluster	Yes

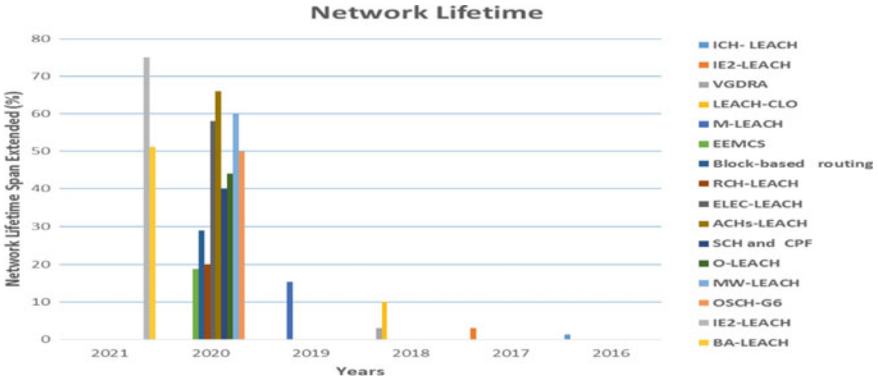


Fig. 1 Comparative results of EERP-LEACH

LEACH protocols and the *y*-axis shows energy consumption. This is the comparative analysis showing the year-wise growth in the LEACH protocol to reduce energy for enhancing the lifetime of the network.

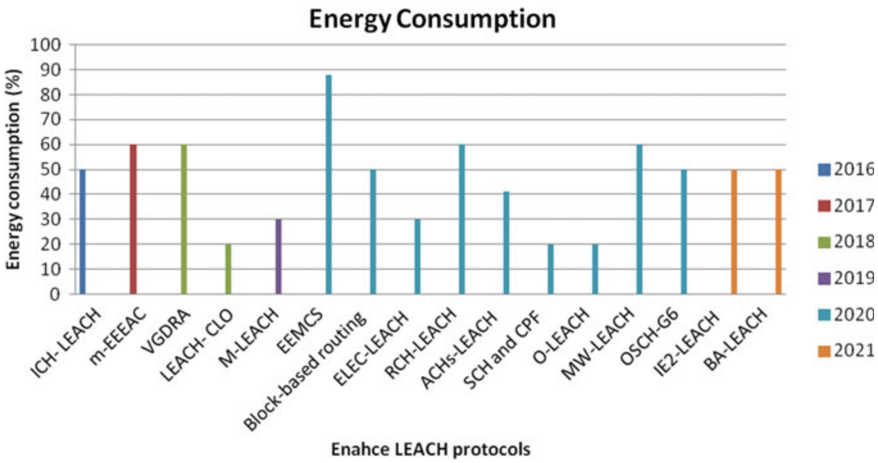


Fig. 2 Comparative results in enhanced LEACH protocols for energy consumption of the network

As per Table 1, Fig. 3 shows comparative results of the number of nodes utilized with maximum rounds taken to improve the network lifetime. The *x*-axis shows node utilization and the *y*-axis shows rounds taken. This is the comparative analysis showing the total nodes used against rounds taken to save energy and increase network lifetime.

As per Table 1, Fig. 4 shows comparative results of the number of nodes utilized with dead nodes to improve network lifetime. The *x*-axis shows node utilization and the *y*-axis shows dead nodes. This is the comparative analysis showing the number of nodes utilized against dead nodes to save energy and enhance network lifetime.

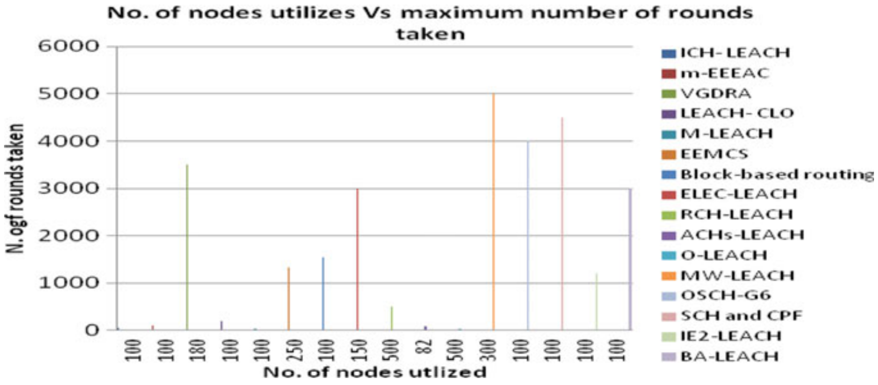


Fig. 3 Comparative results of number of nodes utilized versus maximum rounds taken

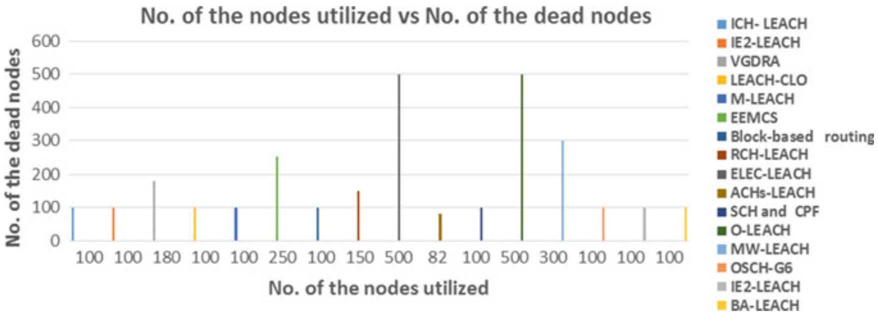


Fig. 4 Comparative results of number of nodes utilized versus dead nodes

6 Conclusion

In this paper, we finally conclude that all existing LEACH protocols worked efficiently for energy saving and improved the lifetime of the network. The researchers apply new techniques to improve the LEACH protocols and their experimental result analysis based on some parameters like network performance, node utilization, data transmission rate, dead nodes, delay, etc. These parameters actually calculate the energy of the sensor node and its consumptions. Also, we did an analysis of these EERP-LEACH algorithms to find the best option for researchers to evaluate LEACH in their further results in WSN.

References

1. Nigam GK, Dabas C (2021) ESO-LEACH: PSO based energy efficient clustering in LEACH. J King Saud Univ - Comput Inf Sci 33(8):947–954. https://doi.org/10.1016/j.jksuci.2018.08.002
2. Chithaluru PK, Khan MS, Kumar M, Stephan T (2021) ETH-LEACH: an energy enhanced threshold routing protocol for WSNs. Int J Commun Syst 34(12). https://doi.org/10.1002/dac.4881
3. Sajedi SN, Maadani M, Nesari Moghadam M (2022) F-LEACH: a fuzzy-based data aggregation scheme for healthcare IoT systems. J Supercomput 78(1). https://doi.org/10.1007/s11227-021-03890-6
4. Lin C, Jiang F (2021) Research of multidimensional optimization of LEACH protocol based on reducing network energy consumption. J Electr Comput Eng 2021:1–9. https://doi.org/10.1155/2021/6658454
5. Nasr S, Quwaider M (2020) LEACH protocol enhancement for increasing WSN lifetime. In: 2020 11th international conference on information and communication systems (ICICS). https://doi.org/10.1109/icics49469.2020.239542
6. Fu C, Zhou L, Hu Z, Jin Y, Bai K, Wang C (2021) LEACH-MTC: a network energy optimization algorithm constraint as moving target prediction. Appl Sci 11(19):9064. https://doi.org/10.3390/app11199064
7. Devika G, Ramesh D, Karegowda AG (2021) Energy optimized hybrid PSO and wolf search based LEACH. Int J Inf Technol 13(2):721–732. https://doi.org/10.1007/s41870-020-00597-4

8. Nasrollahzadeh S, Maadani M, Pourmina MA (2021) Optimal motion sensor placement in smart homes and intelligent environments using a hybrid WOA-PSO algorithm. J Reliab Intell Environ. https://doi.org/10.1007/s40860-021-00157-y
9. Nabati M, Maadani M, Pourmina MA (2021) AGEN-AODV: an intelligent energy-aware routing protocol for heterogeneous mobile ad-hoc networks. Mob Netw Appl 27(2):576–587. https://doi.org/10.1007/s11036-021-01821-6
10. Wei Q, Bai K, Zhou L, Hu Z, Jin Y, Li J (2021) A cluster-based energy optimization algorithm in wireless sensor networks with mobile sink. Sensors 21(7):2523. https://doi.org/10.3390/s21072523
11. Yue H, Yin CQ, Wilson J (2020) Research on data aggregation and transmission planning with internet of things technology in WSN multi-channel aware network. J Supercomput 76(5):3298–3307. https://doi.org/10.1007/s11227-018-2565-5
12. Radhika M, Sivakumar P (2020) Energy optimized micro genetic algorithm based LEACH protocol for WSN. Wirel Netw 27(1):27–40. https://doi.org/10.1007/s11276-020-02435-8
13. Ullah I, Youn HY (2020) Efficient data aggregation with node clustering and extreme learning machine for WSN. J Supercomput 76(12):10009–10035. https://doi.org/10.1007/s11227-020-03236-8
14. Heinzelman W, Chandrakasan A, Balakrishnan H (2002) An application-specific protocol architecture for wireless microsensor networks. IEEE Trans Wirel Commun 1(4):660–670. https://doi.org/10.1109/twc.2002.804190
15. Qubbaj NNA, Taleb AA, Salameh W (2020) LEACH based protocols: a survey. Adv Sci, Technol Eng Syst J 5(6):1258–1266. https://doi.org/10.25046/aj0506150
16. Thi Quynh TP, Viet TN (2021) Improvement of LEACH based on K-means and Bat Algorithm. Int J Adv Eng Res Sci 8(2):031–035. https://doi.org/10.22161/ijaers.82.6
17. Saleh SS, Mabrouk TF, Tarabishi RA (2021) An improved energy-efficient head election protocol for clustering techniques of wireless sensor network (June 2020). Egypt Inform J 22(4):439–445. https://doi.org/10.1016/j.eij.2021.01.003
18. El Idrissi N, Najid A, El Alami H (2020) New routing technique to enhance energy efficiency and maximize lifetime of the network in WSNs. Int J Wirel Netw Broadband Technol 9(2):81–93. https://doi.org/10.4018/ijwnbt.2020070105
19. El Khediri S, Khan RU, Nasri N, Kachouri A (2020) Energy efficient adaptive clustering hierarchy approach for wireless sensor networks. Int J Electron 108(1):67–86. https://doi.org/10.1080/00207217.2020.1756454
20. El Aalaoui A, Hajraoui A (2020) Energy efficiency of organized cluster election method in wireless sensor networks. Indones J Electr Eng Comput Sci 18(1):218. https://doi.org/10.11591/ijeecs.v18.i1.pp218-226
21. Akleek FA, Alquraan R, Shurman M (2020) Enhancement of WSN network lifetime. In: 2020 32nd international conference on microelectronics (ICM). https://doi.org/10.1109/icm50269.2020.9331817
22. Kumar N, Desai JR, Annapurna D (2020) ACHs-LEACH: efficient and enhanced LEACH protocol for wireless sensor networks. In: 2020 IEEE international conference on electronics, computing and communication technologies (CONECCT). https://doi.org/10.1109/conecct50063.2020.9198666
23. Us Sama N, Bt Zen K, Ur Rahman A, BiBi B, Ur Rahman A, Chesti IA (2020) Energy efficient least edge computation LEACH in wireless sensor network. In: 2020 2nd international conference on computer and information sciences (ICCIS). https://doi.org/10.1109/iccis49240.2020.9257649
24. Panchal A, Singh L, Singh RK (2020) RCH-LEACH: residual energy based cluster head selection in LEACH for wireless sensor networks. In: 2020 international conference on electrical and electronics engineering (ICE3). https://doi.org/10.1109/ice348803.2020.9122962
25. Sahib HA, Kurnaz S, Mohammed AH, Sahib ZA (2020) Network of low energy adaptive clustering protocols. In: 2020 4th international symposium on multidisciplinary studies and innovative technologies (ISMSIT). https://doi.org/10.1109/ismsit50672.2020.9254821

26. Umbreen S, Shehzad D, Shafi N, Khan B, Habib U (2020) An energy-efficient mobility-based cluster head selection for lifetime enhancement of wireless sensor networks. IEEE Access 8:207779–207793. https://doi.org/10.1109/access.2020.3038031
27. Takele AK, Ali TJ, Yetayih KA (2019) Improvement of LEACH protocol for wireless sensor networks. Commun Comput Inf Sci 250–259. https://doi.org/10.1007/978-3-030-26630-1_21
28. Zhang W, Wei X, Han G, Tan X (2018) An energy-efficient ring cross-layer optimization algorithm for wireless sensor networks. IEEE Access 6:16588–16598. https://doi.org/10.1109/access.2018.2809663
29. Dhami M, Garg V, Randhawa NS (2018) Enhanced lifetime with less energy consumption in WSN using genetic algorithm based approach. In: 2018 IEEE 9th annual information technology, electronics and mobile communication conference (IEMCON). https://doi.org/10.1109/iemcon.2018.8614754
30. Ghosh S, Misra IS (2017) Design and testbed implementation of an energy efficient clustering protocol for WSN. In: 2017 international conference on innovations in electronics, signal processing and communication (IESC). https://doi.org/10.1109/iespc.2017.8071864
31. Taj MBM, Kbir MA (2016) ICH-LEACH: an enhanced LEACH protocol for wireless sensor network. In: 2016 international conference on advanced communication systems and information security (ACOSIS). https://doi.org/10.1109/acosis.2016.7843949

Slow TCP Port Scan Detection Using Flow Data

N. Muraleedharan and B. Janet

Abstract In cybersecurity attacks, the attackers use the port scan to identify the open ports and live machines in the targeted networks. As the scanning activities are carried out in the pre-stage of the attacks, identifying the scanning attempt helps to block the attacks in the initial stage itself. However, the attackers use the slow scan to evade the port scan detection. In this paper, a slow TCP scan detection approach using flow data is proposed. The proposed system takes the single packet flow as the parameters to detect the potential scan attempt, and further, the slow scan is verified using entropy values of flow parameters. The detection capability of the system is evaluated using two benchmark datasets. The result obtained shows that the flow data can be used to detect slow scan attacks effectively.

Keywords Port scan · Slow scan · Network flow · Entropy

1 Introduction

Port scanning is an important step in a cybersecurity attack. The attacker sends traffic to the TCP or UDP port of the target machine to identify the live systems that accept traffic from outside. Once the attacker identifies the open ports in the targets, he can further query to understand the application details and consequently the vulnerabilities of the services running in the port. The attackers are also using the responses from open ports to understand the operating system and its version [1]. Depending on the output of the port scan, the attacker can narrow down the attack to target a specific vulnerability that leads to a successful attack. As the port scan is a pre-stage of the attack, accurate detection of the port scan helps to identify and mitigate complex attacks at the initial stage itself.

N. Muraleedharan (✉)
Centre for Development of Advanced Computing (C-DAC), Bangalore, India
e-mail: murali@cdac.in

B. Janet
National Institute of Technology (NIT), Tiruchirappalli, India
e-mail: janet@nitt.edu

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_2

A typical port scan tool generates several requests to the targeted system for a short duration. Hence, the normal scan is detected by measuring the number of requests to different ports into a specific IP address. However, the attackers are using evasion techniques to avoid the detection and blocking of scan traffic. In this approach, the attackers send the scan packets at a very low rate that evades the traditional port scan detection system. In addition, the scanning tools provide options to adjust the scanning speed that can avoid detection. Thus, adversaries use the low-rate attacks to evade detection.

The slow scan techniques are used for the targeted attacks on a high-profile network. Moreover, critical information infrastructures are one of the common targets for these stealth attacks [2]. However, traditional scan detection approaches find it challenging to detect slow scan activities. Hence, early identification of the slow scan attempt has paramount importance in blocking multistage attacks. The rest of the paper is organized as follows. Section 2 explains the background of TCP scan and related works to detect the slow scan. The proposed architecture and its components are explained in Sect. 3. The results obtained and their analysis are explained in Sect. 4 and the conclusion and future works are shown in Sect. 5.

2 Background and Related Works

The port scanning traffic is generated using TCP or UDP protocols. Depending on the TCP flags used in the scan request, various TCP port scanning techniques are available. Some of the common TCP port scan types are explained below.

2.1 TCP-Based Scan

SYN scan: In the SYN scan, the attacker sends an “SYN” flag-enabled TCP segment to the port of the victim machine. As per the RFC of TCP protocol [3], once it receives an SYN packet to an open port, SYN/ACK should return as a response. The closed port should return the RST packet as the response to the SYN request.

Connect scan: The connect scan establishes the three-way handshake to the victim machine’s port to verify the port’s status. Since it establishes the connection, this scan type provides better accuracy. However, compared to the SYN scan, it requires more packets to obtain the port status. Moreover, the victim machine generates log entries for all the established connections. Thus, there is a possibility of detecting the connect scan attempt using log analysis.

ACK scan: Unlike the standard scan output, the ACK scan cannot provide the status of the port. Instead, it is used to detect the presence of a stateful firewall in the targeted network. It sends the TCP packet with the ACK flag. Both the open and closed ports return the RST packet as a response to the ACK request.

XMAS, NULL, and FIN scan: As per the RFC of TCP protocols [3], any packet without the SYN, RST, or ACK bits will result in a returned RST from the closed port and no response from the open port. Based on this observation, multiple scan types are derived.

In the XMAS scan, the TCP packet with FIN, PSH, and URG flags is sent to the victim. Once the victim machine receives the XMAS scan packet to a closed port, it should return a TCP packet with the RST flag. The NULL scan sends a TCP packet without setting any flag value in it. The target system should send back a TCP packet with the RST flag as the response from the closed port. Upon receiving a TCP packet with the FIN flags, the machine should respond with an RST packet from the closed port. The attacker uses this information to derive the status of the port in the target machine.

2.2 Related Works

Network port scan detection has been a research topic for several years. Bhuyan et al. [4] survey the port scan attacks and compare port scan methods based on type, mode of detection, the mechanism used for detection, and other characteristics. Several approaches were derived to detect port scans in a network and host [1, 5]. However, compared to the normal port scan, slow port scan detection has not received much attention from the research community.

Nisa et al. [6] proposed a slow port scanning attack detection framework by detecting outliers. However, as they used packet-based approach, collection and analysis of packets in a high-speed network will be a challenge to apply this technique. Harang et al. [7] propose an evasion-resistant network scan detection by evaluating the behavior of internal network nodes in combination with the threshold random walk model.

A slow scan detection approach using the Convolutional Neural Network (CNN) using the features learned from the scanning traffic is presented in [8]. A flow-based slow scan detection approach is presented in [9]. In their approach, they used a two-stage approach, and in the first stage, a flow enriching process was used with additional knowledge about network structure. As part of it, a network information file derived by the domain expert is used. Further, they collect the flow over a time window to detect the slow scan attack. One of the limitations of this approach is that to derive the network information file, an understanding of the network and domain experts is required.

In our approach also we have used flow-based data for slow scan attacks. But we have used the number of single packet flow as the parameter for identifying the potential scan attempt. The single packet parameter can be derived without any flow enhancement and domain expertise. The novelty of our approach is the usage of single packet flow parameter for the potential scan attempt identification. Following are the advantages of our approach:

- It uses a two-stage approach for slow scan detection. In the initial stage, the potential scan attempts are identified using single packet flows. Further, the detailed analysis of the scan detection is carried out using the entropy-based approach. Hence, the processing delay and resource requirements are minimum.
- As it uses flow data, the data volume is less compared to the packet-level analysis. Hence, this approach is suitable for slow scan detection in high-speed network.
- It uses the default parameters available in the flow data. Hence, flow enhancement or customization is not required.

3 Proposed System

The proposed system architecture and its components are explained below.

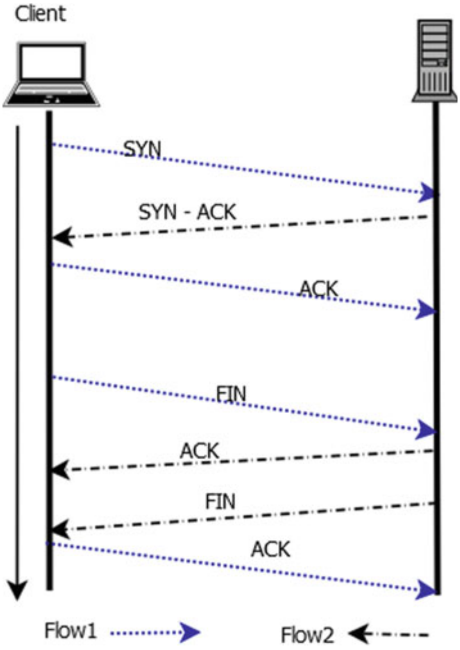
3.1 Flow Representation of Scan Traffic

A flow is defined as a set of packets passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties [10]. Flow-level data are generated by grouping the packets using the flow key, consisting of packet header fields, such as source IP address, destination IP address, source port number, destination port number, and transport layer protocols. The incoming packets are hashed based on the flow key and matched against existing flow entries. A match triggers an update in the flow record where the packet and byte counters are tallied. Packets that do not match any existing entry in the flow are identified as a new flow and create new entries in the flow record. The fields available in the flow records can be used for different insights related to the network traffic anomalies [11].

The flow-level representation of a typical TCP communication and scan traffic is explained below. Figure 1 shows the TCP connection establishment and release sequence. The TCP connection establishment process, known as a three-way handshake, requires three packets to share between the client and server. In connection release, four packets are needed to properly release the established connections. Hence, in TCP, a minimum of seven packets are needed to establish and release a connection. As the TCP connection is used for transferring data between the client and server, including the data transfer packets, the typical TCP connection consists of more than seven packets.

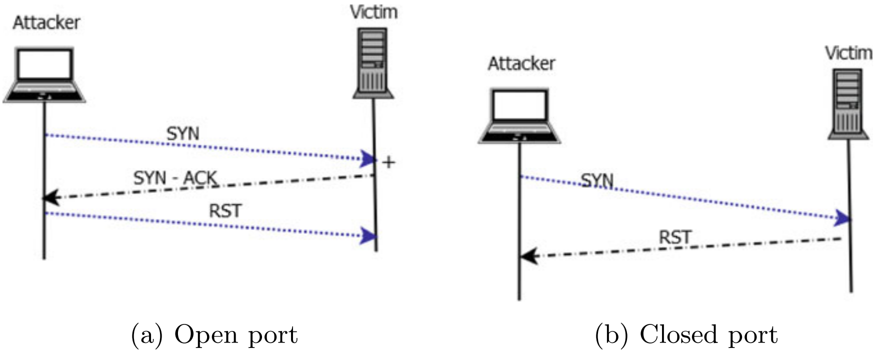
Let $P1$, $P2$, ... Pn be the total number of packets transferred between the client and server in a TCP connection. Then the "n" packets can be divided based on the direction of communication such as CS_{Pi} and SC_{Pj} where the CS_{P1}, CS_{P2}, ..., CS_{Pi} are the packets sent from the client to the server, and SC_{P1}, SC_{P2}, $\ldots SC_{Pj}$ are the packets sent from server to client.

Fig. 1 TCP connection establishment and connection release



If "i" represents the number of packets from client to server and "j" represents the number of packets from the server to clients, then the total packet transferred "n" can be represented as $n = i + j$, where $i >= 4$ and $j >= 3$.

The flow-level representations of the TCP SYN scan traffic are depicted in Fig. 2. By analyzing the scan traffic flow from Fig. 2a, we can observe that the value of "i" is two and "j" is one from the open port. Similarly, from Fig. 2b, we can observe that a closed port creates two flows, and each flow consists of only one packet. The



(a) Open port

(b) Closed port

Fig. 2 Flow representation of SYN scan

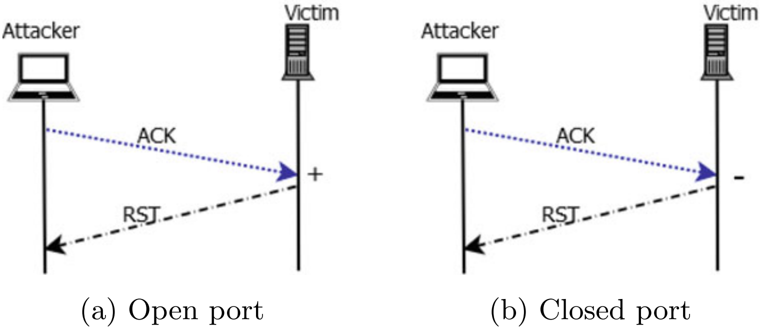


Fig. 3 Flow representation of ACK scan

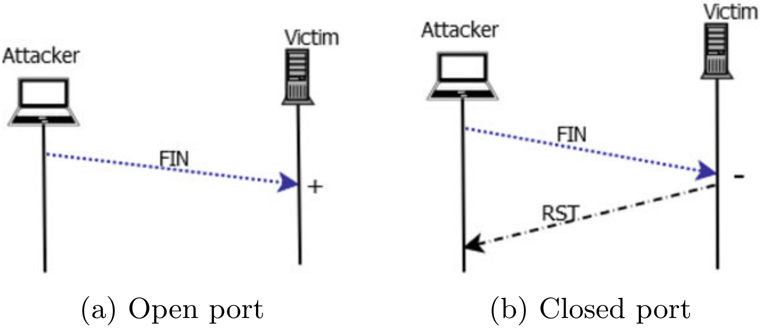
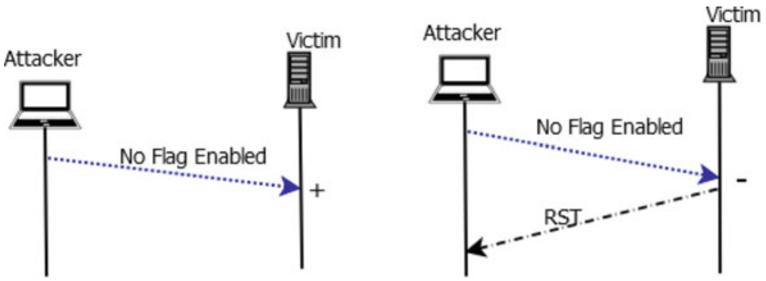


Fig. 4 Flow representation of FIN scan

flow representation of the ACK scan depicted in Fig. 3 shows that from both the open port and closed port, it returns an RST packet as the response. By analyzing the number of packets in the flow derived from the ACK scan, we observed that the source-to-destination and destination-to-source flows consist of a single packet.

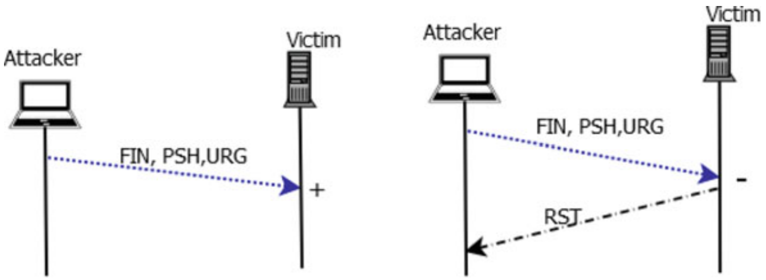
The flow representation of the open port communication in the scan traffic generated from “FIN”, “NULL”, and “XMAS” is depicted in Figs. 4a, 5a, and 6a respectively. It shows that this communication generates the only source-to-destination flow with a single packet in it.

The responses from the closed port of a victim machine for “FIN”, “NULL”, and “XMAS” are depicted in Figs. 4b, 5b, and 6b, respectively. From these figures, we can observe that during the “FIN”, “NULL”, and “XMAS” scan, the closed ports respond with “RST” packets and generate two flows and each flow consists of single packets. Moreover, the attacker sends probe packets to multiple ports in the victim machines, and the number of single packet flows during the port scan is high compared to the normal traffic. However, in the normal TCP communication, due to the three-way handshake and four-way connection release, the flows shall have more than one packet.



(a) Open port (b) Closed port

Fig. 5 Flow representation of NULL scan



(a) Open port (b) Closed port

Fig. 6 Flow representation of XMAS scan

From the flow-level representation of different TCP scan traffics, we can observe a significant increase in the number of flows with single packets during the scan. Hence, by collecting and analyzing the single packet flows, the port scan activities shall be identified.

3.2 Port Scan Detection using Flow data

The datasets and approach used for port scan detection using flow data are explained below.

Dataset used: We have used two datasets for our experiment and evaluation. The first dataset consists of the labeled flow data, including scan traffic and the second dataset consists of a network traffic dump in "pcap" format. The details of the datasets used are described below.

DARPA dataset: DARPA Intrusion Detection Datasets [12] is one of the popular benchmark datasets for intrusion detection collected and published by The Cyber

Systems and Technology Group of MIT Lincoln Laboratory. This dataset consists of 7 weeks of training data and 2 weeks of test data. More than 25 attack traffics were collected in this dataset. We have selected the third week Wednesday dataset with attack name “nmap” where the network mapping was done using the “nmap” [13] tool. This scan technique was stealthy where attempts were made to hide components of the attack in the sniffing or audit data by encryption, spreading the attack over multiple sessions, or other techniques.

Data pre-processing
The DARPA dataset is in the network packet dump format (pcap). Hence, we have converted the network packet dump into network flow data using the “NFStream” [14] tool. NFStream is a Python framework that generates statistical features from online and offline network data. It derives flow data from the network packet dump by aggregating the packets, and the derived flow can be exported into a comma-separated file (CSV) for analysis. The derived flow data consists of 28 parameters. With the help of the Source and Destination IP address used to generate the scan traffic and the time of the attack in the DARPA dataset, we labeled the flow records into benign and scan.

Port scan Dataset by Dalmazo et al. [15]: We have used the dataset titled “Public dataset for evaluating Port Scan and Slowloris attacks” published in 2019 by Dalmazo et al. [15]. In addition to the normal traffic, this dataset consists of port scan and slow HTTP DDoS attack data in network flow format. This dataset consists of 85 fields, including the label. The label field separates the benign and scans flow in the dataset. For our experiment, we have selected the port scan dataset.

3.3 System Architecture

The sequence of steps followed to detect scan attacks is depicted in Fig .7. As shown in the figure, the system consists of two major components called profiler and scan detector. The Profiler component consists of SPF (single packet flow) Counter and Entropy Calculator. The Scan detector component consists of an SPF counter, SPF threshold comparator, Entropy Calculator, Entropy Threshold Comparator, and Scan Type Analyzer. The details of these components are explained below:

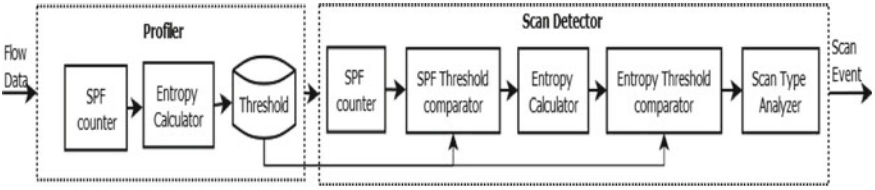


Fig. 7 System architecture

Profiler: The profiler component collects and analyzes the normal traffic to set the threshold values. It takes attack-free flow data and processes it using the SPF (single packet flow counter) and Entropy Calculator components to derive the threshold. The profiler reads a group of "N" flow records in a single iteration, and multiple such iterations are used to set the threshold. The value of "N" shall decide based on the traffic rate of the network. The number of iterations "i" shall be fixed based on the profile data. However, a higher value of "i" should capture the long-term behavior.

SPF counter: The flow representation of scan traffic reveals that the number of flows with single packet increases during the scan time during scan traffic. Hence, we have used the Single Packet Flow (SPF) as a parameter for the early detection of scan traffic. The SPF counter measures the number of single packet flows in each iteration and calculates the ratio R_{SPF} as shown in Eq. 1.

$$R_{SPF} = \left(\frac{SPF}{N}\right) * 100 \tag{1}$$

where N is the total number of records in a single iteration and SPF is the number of single packet flows. At the end of all profile iteration, the SPF threshold is derived as per Eq. 2.

$$T_{SPF} = M_{SPF} + 3 * \mu_{SPF} \tag{2}$$

where "M_{SPF}" is the mean value of all the single packet flow ratio and "μ_{SPF}" is the standard deviation of the SPF value.

Entropy calculator: Entropy measures the average amount of information needed to represent an event drawn from a probability distribution for a random variable. The entropy-based approaches have been used in cybersecurity for different attack detections, including network scans. The entropy for a probability distribution $p(X = x_i)$ of a discrete random variable "X" is defined in Eq. 3.

$$H_s(X) = \sum_{i=1}^{n} P(X_i) \log_a \frac{1}{P(X_i)} \tag{3}$$

where "X" is the feature that can take values $x_1, x_2 \ldots x_n$ and $P(X_i)$ is the probability mass function of outcome X_i. Depending on the logarithm base used, the value of "a" can take different units. The entropy value depends on randomness and the value of "n". High entropy values signify a more dispersed probability distribution, while low entropy values denote the concentration of a distribution.

We have calculated the entropy values of source IP, destination IP, source port, destination port, flow duration, number of packets in the flow, and packet length parameters available in the flow records. At the end of profiling, the entropy calculator calculates the average and standard deviation entropy of the parameters and derives the threshold value for the entropy of these parameters.

Scan detection: The scan detector component reads the flow data from the real-time traffic and analyzes it for identifying the attack. Upon detecting the scan attempt, it generates an event that includes the attacker and attack type. The details of scan detection components are explained below.

Initial Scan detection using SPF counter: The SPF counter in the scan detection component reads "N" flow records and counts the number of flows with the single packets (SPF). Further, it calculates the ratio of SPF in the total flow using Eq. 1. It compares the result with the SPF threshold set at the profile time. Suppose the calculated SPF ratio is greater than the profile threshold. In that case, it detects the scan attempt, and further verification of the attack is carried out using the entropy calculator and scan type analyzer. Suppose the calculated SPF ratio is less than the profile threshold then all single packet flows in the received flow records are stored in the SPF dataset. Though the SPF number is less in the received flow records, we are storing them for further analysis to detect the slow scan attempt.

As the slow scan tools send the scan packets in stealthy mode, the frequency of scan packets will be less compared to the normal scan traffic. However, similar to the normal scan, the slow scan tool generates a scanning packet with the same TCP flag values but at a lower rate. Hence, the slow scan traffic also contains a single packet flow. As the presence of SPF can be a potential scan attempt, we are storing the SPF for further analysis and detection of the slow scan.

Scan verification using entropy analysis: An entropy-based approach is used to verify the scan attacks upon identifying the potential scan attempt using the SPF counter. Like the entropy calculator in the profiler, the scan detection entropy calculator computes the entropy values of source IP, destination IP, source port, destination port, flow duration, number of packets in the flow, and packet length from the received flow records.

The entropy threshold comparator compares the calculated entropy with the profile entropy value of the corresponding parameters. Once it identifies the deviation from the threshold, it confirms the scan attack. The scan type analysis is carried out using the TCP flag value in the flow records.

The scan type analyzer checks the source IP, Destination IP, and TCP flag values in the flow records to detect the attacker, victim, and scan type. The top IP address in the sorted list of source IP addresses in the flow records shall be considered the attacker IP address. Similarly, the top destination IP address of the flow record indicates the victim IP address. The scan type can be identified by taking the count of flag values from the attacker's IP address. At the end of scan verification, an event with the timestamp, source IP address, victim IP address, and scan type is generated by the scan detector to indicate the scan attack.

Slow scan detection: The slow scan tools send the packets to the victim machine's port at a very low rate. Hence, it is not easy to detect the slow scan using the normal scan detection approaches. As we observed from the flow representation of scan traffic, the number of single packet flows is increasing during the scan attempt.

For example, a single request sent to an open port using SYN, ACK, FIN, NULL, or XMAS scan creates a minimum of two single packet flows. Similarly, from the closed port also, it generates a single packet flow as the response. Hence, any scan attempt to a single port shall generate single packet flows independent of the port and scan type status. Based on this observation, we have profiled all the single packet flows for detecting the slow scan activities.

In the slow scan, the time delay between two consecutive scan packets is increased. Hence, if we collect the flow records, all the scan packets need not be captured during the flow collection time. Moreover, as the delay between two consecutive scan packets is increased (it can be in hours), the tool may take days or weeks to complete the port scan of a single victim. However, collecting and analyzing weeklong network flows from a high-speed network require enormous storage and processing resources.

We have collected all the single packet flows and periodically analyze them for slow scan detection in our approach. Since the scan traffic generates single packet flows, collecting all the single packet flows helps detect the scan traffic with minimal resources. The slow scan analyzer periodically reads and analyzes the SPF data from the storage. The entropy values of the source IP, destination IP, source port destination port, flow duration, number of packets in the flow, and packet length parameters are calculated using the entropy calculator. The calculated entropy values are compared with the profile entropy value. Upon identifying any deviation, it verifies the slow scan. Further, the analysis is carried out to detect the attacker, victim, and type of scan. After the analysis, an event is generated with the attacker, victim, and type of attack details. The steps involved to detect the slow scan detection are shown in Algorithm 1.

Algorithm 1: Slow scan detector

```
Input: Flow records
Output: slow scan event
Data: $N$ flow records
for $i$=1 to $P$                                          // Profile Iteration
do
    SPF=0                                                 // Initilize SPF
    for $j$=1 to $N$ do
        Read flow records
        $pc$=Number of packets in the flow
        if $pc < 2$ then
            $SPF_j$=$SPF_j$+1                             // Count the SPF
    $R_{SPF}$=$SPF/N$                                     // Ratio of SPF
    $ENT_j$=entropy(IP address, port,duration, packet size)
    /* End of Profiling                                             */
$\mu_{SPF}=\sum_{i=1}^{n} R_{SPF}/P$                      // Average profile SPF
$\sigma_{SPF}$=STD of $R_{SPF}$                           // Standard Deviation
$T_{SPF}=\mu_{SPF} + 3 * \sigma_{SPF}$                    // Threshold
$\mu_{ENT}$=Avg of all profile $ENT$
$\sigma_{ENT}$=STD of all profile $ENT$
Set entropy threshold as $T_{ENT}$
/* Checking for Slow Scan                                           */
for Every n seconds do
    Read all SPF flow records
    $ENT_S$=entropy(IP address, Port, Duration, Packet size)
    if $ENT_D$ varies from $T_{ENT}$ then
        Slow Scan detected
    Generate slow scan event
```

4 Result and Result Analysis

The results obtained from the slow scan detection experiments and their analysis are explained in this section.

4.1 Single Packet Flow Ratio

For the initial detection of potential scan activities, we have calculated the distribution of single packet flows in the benign and scan network flow. In this analysis, we have segregated all the TCP flow in the dataset using the “protocol” fields in the flow records. Further, the count of single packet flow (SPF) and multi-packet flow (MPF) in the benign and scan traffic is taken from each dataset. The result obtained from this analysis is summarized in Table 1.

Table 1 Summary of the single packet flow

Dataset	Benign traffic			Scan traffic		
	#SPF	#MPF	%SPF	#SPF	#MPF	%SPF
Dataset1[12]	3327	9594	25.7	1008	24	97.7
Dataset2 [15]	19672	47662	29.2	158415	508	99.7

The number of single packet flows is represented by the column titled "#SPF", and the number of multi-packets flows observed is represented by the column titled "#MPF". The percentage of single packet flow in the total flow records is represented in the column titled "%SPF". From the table, we can observe that during the normal traffic, in both datasets, the number of single packet flows is less compared to the number of multi-packet flows. In Dataset1, the SPF is 25.7 % of the total flows. A similar distribution of SPF is observed in Dataset2, where 29.2 % of the total flow is SPF. By analyzing the scan traffic, we can observe that the number of single packet flows is increased significantly during the scan. In Dataset1, the percentage of SPF is 97.7, and in Dataset2, it is 99.7.

4.2 *The Entropy of Flow Parameters*

After identifying the potential scan, we calculated the entropy of flow parameters to verify port scan verification. The obtained entropy values of the flow parameters for benign and scan traffic using Dataset1 are depicted in Fig 8. From the figure, we can observe that the flow parameters' entropy values vary during the scan traffic. Compared to benign traffic, the entropy values of the Source, Destination IP address, and Source Port are decreased to zero during the scan traffic. The entropy value of the destination port is increased during the scan traffic. Similarly, a significant reduction in the entropy values of flow duration, forward packet, backward packet, forward packet length, and backward packet length is observed during the scan traffic. Figure 9 shows the entropy values of flow parameters obtained using Dataset2. Similar to Dataset1, we can observe that the entropy values of the destination port are increased during the scan traffic. It was observed that the entropy values of the source and destination IP addresses during the scan traffic are zero in Dataset2 as well. However, unlike the Dataset1 results, we cannot identify any major change in the entropy value of Source Port during the scan; it is equal to the entropy value of benign traffic. Similar to the results obtained from Dataset1, a significant reduction in entropy values is observed for flow duration, forward packet, backward packet, forward packet length, and backward packet length during the scan.

Threshold and scan detection: We have used 120000 benign flows to derive the profile threshold values. These flows are divided into 12 sets with 10000 flows in each set. As part of the profiling, the single packet flow ratio is derived for each

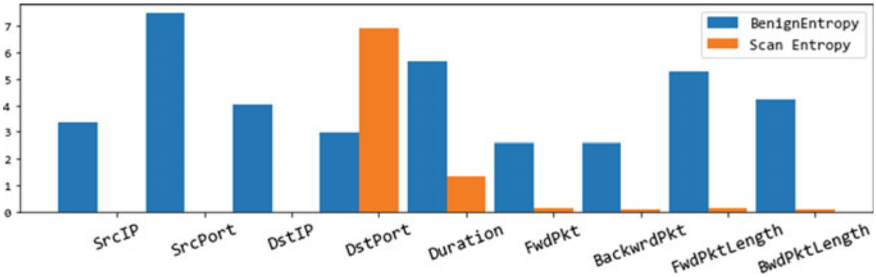


Fig. 8 Entropy value of flow parameters for Dataset1

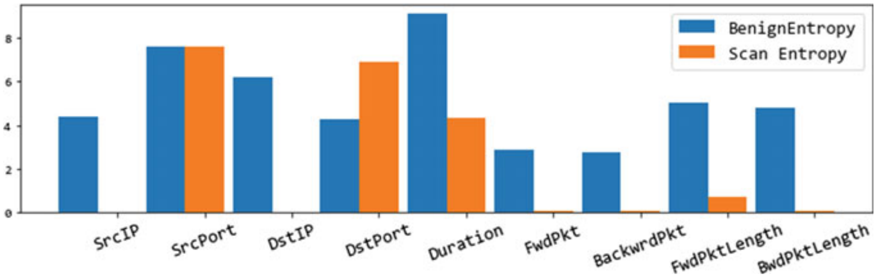


Fig. 9 Entropy value of flow parameters for Dataset2

iteration and entropy values of the identified flow parameters are also calculated. At the end of the profiling, the average and standard deviation of SPF and entropy are calculated to set the threshold values. The threshold values derived for Single Packet Flow (SPF) and entropy of different parameters are summarized in Table 2. The "SrcIP", "Src Port", "Dst IP", and "Dst port" indicate the entropy of the IP address and port number of the source and destination. The column titled "Dura" shows the flow duration. "Fwd Pkt" and "Bkd Pkt" show the entropy values of the number of packets observed in the forward and backward directions of the flow. Similarly, the "Fwd Len" and "Bkd Len" indicate the forward and backward packet lengths.

We have used the scan traffic to derive the SPF and entropy values during the attack. Three instances of the scan traffic with 10000 flows in each instance were selected and the SPF and entropy values were calculated. The obtained results are tabulated in Table 3. From the table, we can observe that, compared to the normal flow, the SPF ratio is increased during the scan where the threshold of SPF was 28.07. As the SPF value is above the threshold value, it indicates the potential scan attempt. Further, by comparing the entropy of scan traffic with the threshold value shown in Table 3, we can observe that the entropy value of all parameters, except the destination port, decreases during the scan which confirms the scan attack.

Table 2 Summary of the threshold values

Profile	SPF	Entropy values								
		SrcIP	Src Port	Dst IP	Dst port	Dura	Fwd Pkt	Bkd Pkt	Fwd Len	Bkd Len
Iteration1	18.96	3.75	6.20	5.12	3.68	7.51	2.85	2.75	4.66	4.18
Iteration2	24.00	3.62	6.37	5.33	3.53	7.65	2.85	2.74	4.74	4.33
Iteration3	20.31	3.26	6.55	4.83	3.07	7.87	2.58	2.52	4.49	3.65
Iteration4	18.32	3.64	6.29	5.17	3.46	7.58	2.80	2.70	4.59	4.15
Iteration5	20.21	3.84	6.25	5.31	3.47	7.52	2.94	2.85	4.77	4.32
Iteration6	19.22	3.79	6.22	5.24	3.68	7.56	2.80	2.72	4.61	4.09
Iteration7	19.15	3.72	6.27	5.27	3.62	7.45	2.82	2.71	4.57	4.14
Iteration8	19.25	3.75	6.21	5.20	3.64	7.41	2.86	2.78	4.62	4.18
Iteration9	22.68	3.77	6.44	5.11	3.99	7.24	2.79	2.67	4.47	4.10
Iteration10	22.45	3.76	6.32	5.01	3.89	7.35	2.78	2.66	4.57	3.97
Iteration11	21.11	3.96	6.15	5.20	3.60	7.53	2.75	2.65	4.54	4.02
Iteration12	26.05	3.72	6.28	5.29	3.44	7.48	2.91	2.88	4.82	4.42
Avg.	20.98	3.72	6.30	5.17	3.59	7.51	2.81	2.72	4.62	4.13
STD	2.37	0.17	0.11	0.14	0.23	0.16	0.09	0.09	0.11	0.20
Threshold	28.07	3.21	5.96	4.75	2.89	7.04	2.54	2.43	4.29	3.53

Table 3 SPF and entropy values during the scan

Scan	SPF	Entropy Values								
		SrcIP	Src Port	Dst IP	Dst port	Dura	Fwd Pkt	Bkd Pkt	Fwd Len	Bkd Len
Scan1	92.73	0	2.35	0	6.9	4.27	0.03	0	0.03	0.03
Scan2	71.76	1.22	4.35	1.36	2.64	3.07	1.21	1.31	1.33	1.47
Scan3	73.87	2.26	3.83	2.07	3.58	3.65	1.01	1.11	1.33	1.53

5 Conclusion and Future works

In this paper, a slow TCP scan detection using flow data is presented. Upon analysis, it was observed that the single packet flows marginally increase during the scan attempt. Hence, the potential scan traffic is identified using the ratio of single packet flow in the total collected flow records. Further, the scan verification is carried out using the entropy values of flow parameters. The proposed approach is verified using two benchmark datasets, and it was observed that the slow scan attack could be detected using the single packet flows and entropy values of flow parameters. We would like to extend this work by blocking the scan traffic upon detecting the scan attack as future works.

References

1. Bou-Harb E, Debbabi M, Assi C (2014) Cyber scanning: a comprehensive survey. IEEE Commun Surv Tutor 16(3):1496–1519
2. Cazorla L, Alcaraz C, Lopez J (2018) Cyber stealth attacks in critical information infrastructures. IEEE Syst J 12:1778–1792
3. Postel J (1981) Transmission Control Protocol, Technical Report. RFC0793, RFC Editor
4. Bhuyan MH, Bhattacharyya DK, Kalita JK (2011) Surveying port scans and their detection methodologies. Comput J 54:1565–1581
5. Muraleedharan N (2008) Analysis of TCP flow data for traffic anomaly and scan detection. In: 2008 16th IEEE international conference on networks. IEEE, New Delhi, pp 1–4
6. Nisa MU, Kifayat K (2020) Detection of slow port scanning attacks. In: 2020 international conference on cyber warfare and security (ICCWS). IEEE, Islamabad, Pakistan, pp 1–7
7. Harang RE, Mell P (2015) Evasion-resistant network scan detection. Secur Inf 4:4
8. Wang Y, Zhang J (2018) DeepPort: detect low speed port scan using convolutional neural network. In: Qiao J, Zhao X, Pan L, Zuo X, Zhang X, Zhang Q, Huang S (eds) Bio-inspired computing: theories and applications, vol 951, pp 368–379
9. Ring M, Landes D, Hotho A (2018) Detection of slow port scans in flow-based network traffic. PLOS ONE 13:e0204507
10. Claise B, Trammell B, Aitken P (2013) Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, Technical Report. RFC7011, RFC Editor
11. AlEroud AF, Karabatis G (2018) Queryable semantics to detect cyber-attacks: a flow-based detection approach. IEEE Trans Syst, Man, Cybern: Syst 48:207–223
12. Lippmann R, Fried D, Graf I, Haines J, Kendall K, McClung D, Weber D, Webster S, Wyschogrod D, Cunningham R, Zissman M (1999) Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation. In: Proceedings DARPA information survivability conference and exposition. DISCEX'00, vol 2. , IEEE Comput. Soc., Hilton Head, SC, USA, pp 12–26
13. “Nmap: the Network Mapper.” [Online]. Available: https://nmap.org/
14. “NFStream: Flexible Network Data Analysis Framework.” [Online]. Available: https://www.nfstream.org/docs/
15. Dalmazo BL, Deolindo VM, Nobre JC (2019) Public dataset for evaluating Port Scan and Slowloris attacks. Type: dataset

An Exploration of Digital Image Forensic Techniques: A Brief Survey

Divya P. Surve and Anant V. Nimkar

Abstract Digital image forensics deals with assessing whether an image is genuine or not. As an image may undergo several manipulations done to either improve its quality or to intentionally change its meaning it is very difficult to conclude if an image is forged or is genuine. In this paper, three important aspects of image forgery detection are explored. An in-depth discussion of image forgery detection, a technique which is based on assessing image features categorized under active and passive methods is present. Analysing both image features and device features to know the image capturing device while checking for forgery is explained in detail. Provenance analysis which is the entire derivation of image manipulation history is also expressed. Discussion regarding research directions in the domain of image forensics is mentioned in the paper.

Keywords Image forgery detection · Illumination estimation · Image source camera identification · Image provenance analysis

1 Introduction

Digital images are a great medium for conveying information through various communication mediums. Easy manipulation of images due to advance image editing tools leads to change in message sent. Image forgery detection or image manipulation detection identifies such manipulation done in the images in order to change its meaning.

Image forgery detection is a multi-faceted approach having multiple viewpoints. There are various active and passive methods to detect image forgery based on image features. Illumination direction estimation is a technique under passive image forgery detection. Image source camera identification is an important area where features of

D. P. Surve (✉) · A. V. Nimkar
Sardar Patel Institute of Technology, Mumbai, India
e-mail: divya.surve@spit.ac.in

A. V. Nimkar
e-mail: anant_nimkar@spit.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_3

both capturing device and images are analysed to know the source of an image [3]. Image provenance analysis builds a graph wherein the series of manipulation for an image is derived [9].

This review addresses the various concerns in detecting forgery by means of illumination-based technique which is a method under passive image forgery detection. In image forgery detection and source device identification, this paper focuses on devising mechanism that can distinguish well between genuine image enhancement operations of scaling, rotation with those of image manipulation operations well. In case of provenance analysis, this paper tries to show the importance of dealing with near-duplicate images, donor images of small size and use of metadata information.

Considering environmental lighting conditions is one possible alternative to the problems under passive physics-based method of image forgery detection. Image forgery detection with source camera identification giving equal importance to forgery detection and genuine manipulation detection is discussed in the paper. Techniques to deal with donor images of small sizes and near-duplicate images using contextual masks, metadata information, etc. is highlighted in the paper.

In order to experiment with issues in illumination-based forgery detection, distinguished object in an image needs to be identified. Illumination direction should be estimated for these objects including the background of an image. For dealing with the problem of having perceptual robustness, techniques for genuine image manipulation detection need to be experimented first before hash computation. For faster provenance analysis, metadata of an image database can be considered.

Our significant contribution in the domain of image forgery detection are as follows:

- Detailed discussion of the areas in digital image forgery detection which are active/passive forgery detection techniques, image forgery detection with source camera identification and image provenance analysis.
- Minutely stated problems related to all the three domains of passive illumination-based image forgery detection, image source device identification and provenance analysis.
- Possible solutions to the stated problems are also discussed in the paper.

The paper is organized as follows: Sect. 2 states background of the domain explaining basic terminologies and techniques under digital image forensics which are image forgery detection, image forgery detection with source device identification and provenance analysis. Section 3 discusses the motivation and related work, Sect. 4 discusses the various issues in three areas mentioned. Section 5 details possible area to explore in the area of digital image forensics. Section 6 states conclusion regarding learning derived from the overall review on digital image forensics. Section 7 discusses the future possible areas to explore in the domain of image forensics.

2 Background

The domain of digital image forensics has three important aspects of:

1. Image forgery detection.
2. Image forgery detection with source camera identification.
3. Image provenance analysis.

2.1 Image Forgery Detection

Image forgery detection methods check for properties of an image to decide if an image is genuine or forged. Image forgery detection method depends on the image storage formats. The forgery detection method for image depends on the type of image and the format of compression of an image. A detail of strategies employed for detection of JPEG and PNG images is discussed in [15–18].

There are multiple active and passive image forgery detection methods which are employed. Active image forgery detection techniques rely on use of proactive measures like watermarking, digital signature and texture analysis to know presence of any forged content in an image [19, 25]. Passive image forgery detection techniques use measures like image features based on comparison of pixel values, compression methods, camera properties, illumination environment and geometric features [7, 25].

Image forgery detection technique involves the following six steps, namely, pre-processing, image feature extraction, image feature matching, false match removal, result optimization and region localization [7, 25]. The details of every phase are depicted below:

- Pre-processing: This operation enhances image quality to be useful in the further phases of processing. Operations like noise removal, resizing and colour-space conversion, segmentation, etc. are carried out to make it suitable for training.
- Image feature extraction: Unique distinguishable features are extracted from the image. These features represent values that are used as an identifier for image rather than the entire set of pixel values. It uses techniques of transformation based on coding, hashing, LBP method, key point processing and histogram-based processing.
- Image feature matching: Features extracted from the query image are matched with the features of images in the database. If the resultant value of matching formula computed over the query image is within acceptable range, then the image is considered to be genuine else some manipulation is said to have occurred. Some popularly used image feature matching techniques are nearest neighbour technique, clustering and segmentation, thresholding, Manhattan distance, etc.
- False match removal: If multiple matches are detected from the database for a given query image, then removal of the images which are falsely detected as matching is

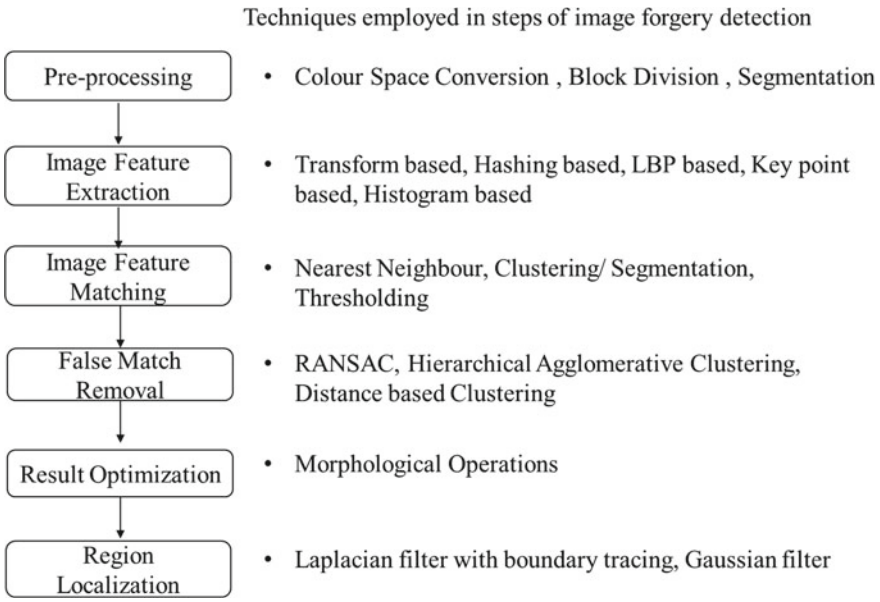

Techniques employed in steps of image forgery detection
Pre-processing
Image Feature Extraction
Image Feature Matching
False Match Removal
Result Optimization
Region Localization
• Colour Space Conversion , Block Division , Segmentation
• Transform based, Hashing based, LBP based, Key point based, Histogram based
• Nearest Neighbour, Clustering/ Segmentation, Thresholding
• RANSAC, Hierarchical Agglomerative Clustering, Distance based Clustering
• Morphological Operations
• Laplacian filter with boundary tracing, Gaussian filter

Fig. 1 Steps in image forgery detection

carried out in this phase. The techniques of RANSAC, hierarchical agglomerative technique and distance-based techniques are used to remove such false matches.

- Result optimization: The resultant images after removal of false match are passed through morphological operations to derive the structure of the forged content. It is used to optimize the resultant structure derived of the objects present in the image using operation of dilation, erosion, closing and opening operations.
- Region localization: Images derived from region optimization process are further processed in Region localization phase to get accurate boundaries of objects. Filters like Laplacian filter, Gaussian filter, etc. are used here to derive the boundary of the objects.

2.2 Image Forgery Detection using Source Camera Identification

Every digital image captured by a device will have properties as embedded by the capturing device. Set of images captured by a camera may induce distortion uniformly at same location in all images captured by that device. Such peculiar pattern of intensities can act as distinguishing feature of image source helping in identification of image forgery using source camera detection [1–6]. The features of colour filter array, photoresponse non-uniformity pattern [1], sensor pattern noise [1] and

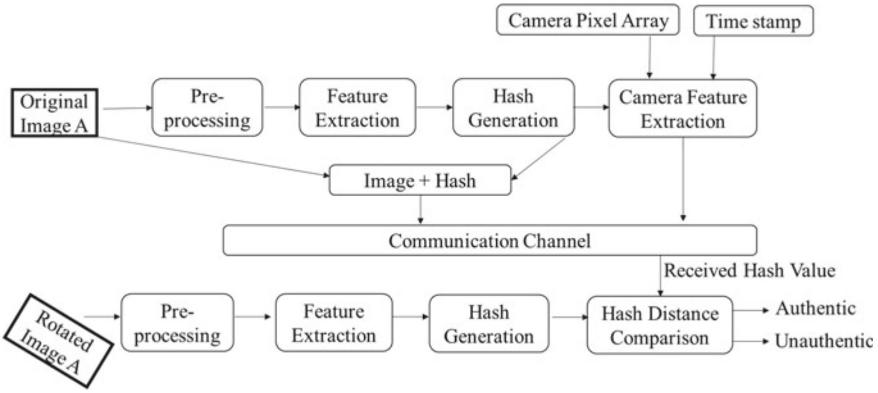


Fig. 2 Steps in image forgery detection with source device identification

compression scheme [5] are some of the features of the capturing device to check the type of device and the brand of it.

Image forgery detection with source camera identification involves the following phases as mentioned in [2, 3], namely, pre-processing, feature extraction, camera feature extraction, hash generation and hash distance comparison.

- Pre-processing: The image is checked to see if it is suitable for the further processing phases. If not operations of image enhancement are applied to make it apt for further phases of feature extraction.
- Feature Extraction: The aim here is to extract features from an image in order to identify image using minimum representative pixels. The extraction process can vary depending on the type of features to be extracted from the image.
- Camera Feature Extraction: Combined features from capturing device and image are extracted to have a minimum representative set of values for every image. Features include PRNU, SPN, colour filter array or compression scheme.
- Hash Generation: Hash value is generated using both the device features and the image features. The generated hash value is appended to the captured image and sent through the communication channel. A similar process in the reverse fashion is followed at the receiver's end.
- Hash Distance Comparison: If the generated hash value at the receiver's end is within the decided threshold then the image generated is said to be authentic else it is called to be tampered one.

2.3 Image Manipulation History Tracking

An image may undergo series of manipulation before it is ready to use for a particular application. In provenance analysis, identification of the entire set of contributing images for a given query image is carried out. This would ultimately help in under-

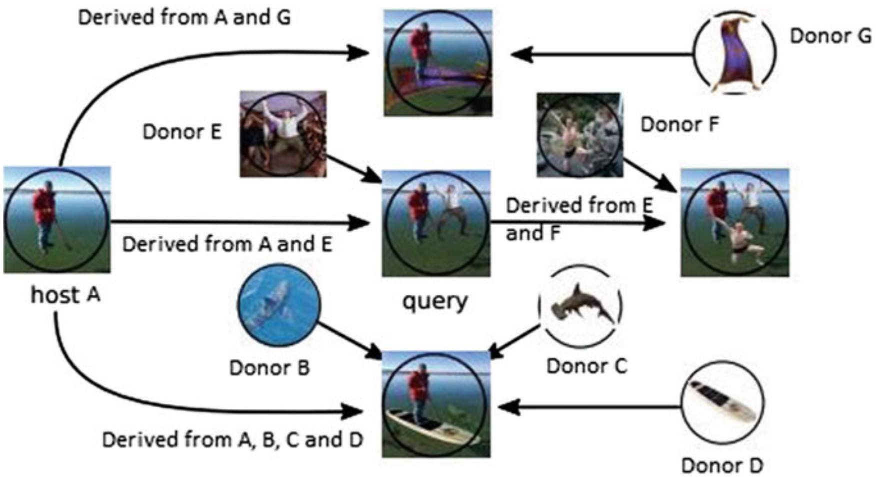
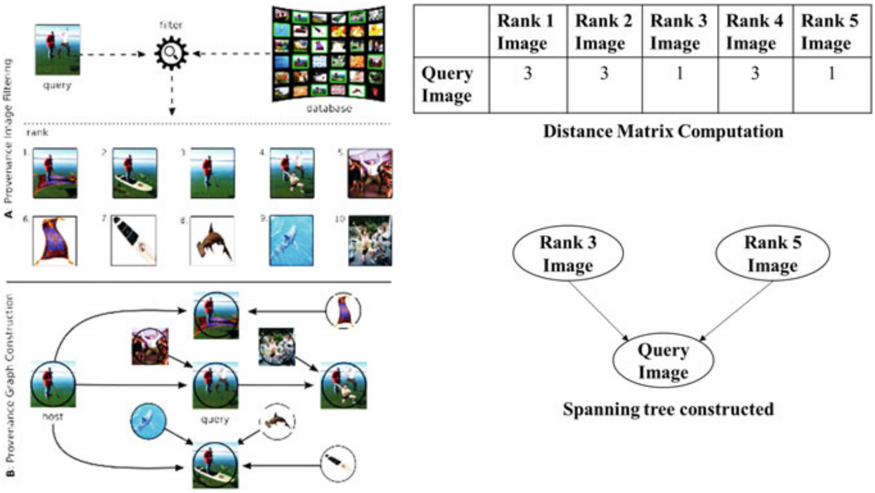


Fig. 3 Image source identification and provenance graph construction


A. Provenance Image Filtering
query
filter
database
rank
1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
B: Provenance Graph Construction
host
query
Distance Matrix Computation
Rank 3 Image
Rank 5 Image
Query Image
Spanning tree constructed

	Rank 1 Image	Rank 2 Image	Rank 3 Image	Rank 4 Image	Rank 5 Image
Query Image	3	3	1	3	1

Fig. 4 Steps in provenance analysis

standing the reason for manipulation in the query image. As shown in Fig 3, the central query image has donors from multiple images like image A and image E. Also multiple images can be derived from query image too. Hence, there can be a series of manipulation that an image can undergo. Figure 4 from [9] depicts detailed steps involved in provenance analysis:

- Provenance Image Filtering: A search for the extracted features from the query image and the database of images is carried out. The matched images are then ranked like in Fig. 4, so as to find the best suitable match in the database for the various objects present in the query image [9–11, 13, 14].

- Provenance Graph Construction: Once the images are filtered so as to find the best images from the database, a dissimilarity matrix is constructed between the query image and the best match images. This matrix is further converted to a graph using minimum spanning tree algorithm to get the history of manipulation for the query image [9, 10, 23, 24]. As depicted in Fig. 4, ten images having rank 1–10 are assessed during graph construction phase. The distance between query image and the top 5 best ranked images is mentioned in the matrix. Rank 3 image is the closest to the query image having maximum content derived from it and hence named as host image. Rank 5 image has some content adopted in the query image and has the next least distance from the query image. The query image is thereby derived from the Rank 3 and Rank 5 images, respectively. Similarly, in the distance matrix we can consider images till Rank 10 as well based on algorithmic thresholds placed.

3 Motivation and Related Work

This section provides an elaborate detail of various techniques under image forgery detection. A detailed comparison of techniques under image forgery detection with source device identification is provided for reference. Investigation of techniques under provenance analysis with varied donor sizes is also expressed.

In Table 1, details regarding various techniques under digital image forensics based on pixel values, compression method, camera properties, physics of lighting condition and geometric properties of image capturing device are mentioned.

Table 2 provides a comparison regarding various image forgery detection techniques based on source camera identification. Techniques here are compared based on parameters of perceptually robust operations of rotation and scaling. Other parameters of comparison are whether tamper detection, device authentication are possible. It can be observed that there is a need to attain better accuracy level where genuine image manipulation is well differentiated to that of tamper operation.

Table 1 Image forgery detection techniques

Techniques	Methods
Pixel based	Copy-Move, Splicing, Resampling, Retouching
Compression based	JPEG Quantization, Double JPEG, Multiple JPEG, JPEG blocking
Camera based	Chromatic Aberration, Source Camera Identification, Pixel Array, Sensor Noise
Physics based	2D and 3D Light Direction, Light Environment
Geometric based	Camera Intrinsic Parameters, Multi-view geometry

Table 2 Comparison of various image forgery detection techniques with image source identification

Techniques	Rotation	Scaling	Tamper detection	Device authentication
[4]	80.02	1	Yes	No
[1]	No	No	Yes	Yes
[2]	No	No	Yes	Yes
[3]	96.25	90.42	95.42	Yes

Table 3 Comparison of image provenance analysis techniques based on various size donors

Paper	SD	SDR	MD	MDR	LD	LDR
[9]	195	28.3	265	56.8	286	67.0
[28]	195	33.3	265	72.6	286	76.8
[23]	195	**55.3**	265	**75.2**	286	**78.0**

Table 3 [23] provides an analysis of various provenance analysis techniques proposed in [9, 23, 28]. The terms #SD, #MD and #LD mean count of small donor, medium donor and large donor. The terms SDR, MDR and LDR mean small, medium and large donor recall rate. The dataset MFC19EP1 is being considered for evaluating these parameters. Donor images having spliced region less than 1 percent of its image size are classified as small donors. Spliced images greater than 10% of its size are classified as large donor and the others are considered as medium size donors. Same number of samples under various categories of small, medium and large when compared attains a recall rate of around 78% for donors of large size, however it can attain only 55% of recall rate for small size donors. The observation is similar for other provenance-based datasets like MFC18EP1, MFC17EP1 and Reddit real time. This emphasizes the need to improve on detection of small spliced regions while building provenance graph.

4 Discussion

In this section problems associated with every image forgery detection scheme is discussed in depth.

4.1 Image Forgery Detection Using Illumination-Based Methods

Illumination-based methods of image forgery detection come under the category of physics-based methods for detection of fraud image. In [8, 21] technique, the illumination pattern of the objects in the scene is analysed to check if there is any false content present. The falsification could be because of splicing of multiple images or using small cropped objects from the same image. Detection of spliced objects based on colour illumination inconsistencies is discussed in [20]. In spliced images where there is a seamless integration of images present it is difficult to find the difference between the objects at the first glance. However, analysing them thoroughly using methods of illumination detection can reveal such forged content.

Figure 5 from [8] gives a good example of illumination direction estimation for forgery detection. There are two parts in the image where the top image is the coloured image seamlessly spliced from multiple source. The bottom part is the illumination estimated for the image on top. If observed carefully one can see in the bottom black and white image that the dominant illumination direction estimated for two people on the left is towards left while for the three people on the right is towards right. Hence, analysing the illumination pattern of objects in a scene provides a good intuition about image forgery. However, there exists certain area of concerns in such methods as stated below:

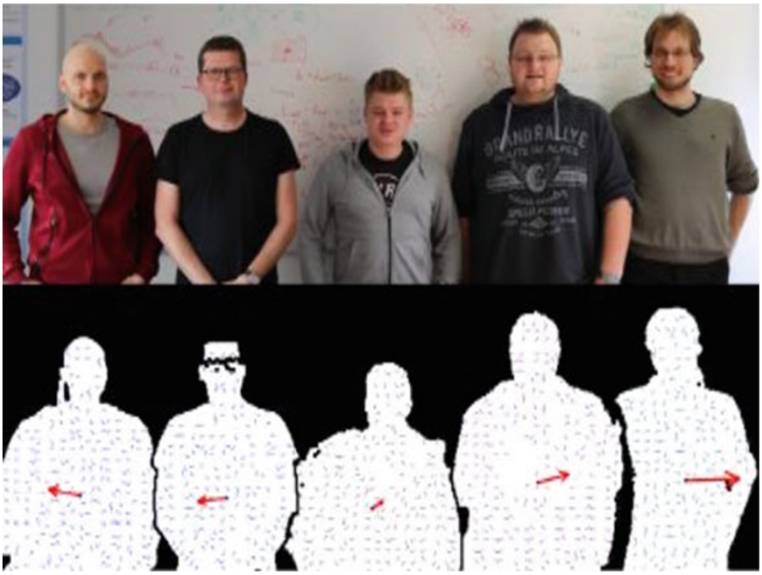


Fig. 5 Illumination direction estimation of scene objects [8]

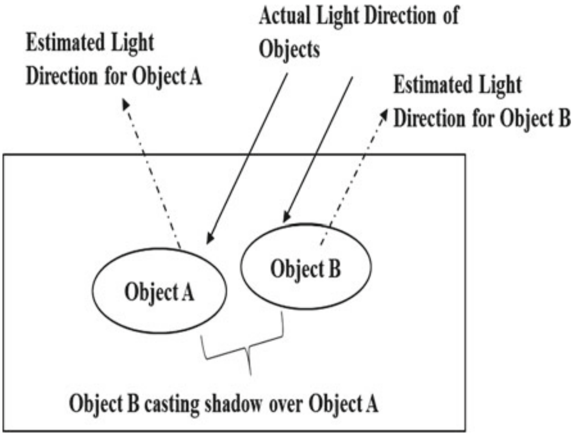

Estimated Light Direction for Object A
Actual Light Direction of Objects
Estimated Light Direction for Object B
Object A
Object B
Object B casting shadow over Object A

Fig. 6 Incorrect Illumination direction estimation due to shadow effect

- Incorrect illumination direction estimation due to shadow effect:
  An incorrect estimation of source light occurs when objects in the scene cast their shadow over the other objects present leading to a misinterpretation that these objects are illuminated by different light sources but in actual they might be illuminated from the same source itself. Consider Fig. 6, where an image has two objects A and B, they are illuminated by the same light source shown by plain arrow and their estimated light directions in dashed arrow. However, object B casts its shadow on object A which changes the illumination direction estimation of object A. Even though objects A and B belong to the same image they are concluded to belong to different images and are forged. Hence, appropriate estimation of illumination direction considering the effect of shadow from objects becomes important.
- Incorrect illumination direction estimation due to multiple spliced objects from same source:
  A spliced image is generated using image from different source. Two objects copied and pasted from same source and pasted on a different image will have same kind of illumination pattern. This creates a problem as the image under consideration though being fabricated image generated using spliced objects from a same source is treated as genuine. This leads to a false positive that the image is genuine even though it is manipulated.
  As can be seen in Fig. 7, objects C and D are spliced from same source image A into the image B and have the same illumination direction estimation. The image B on checking for forgery is detected as genuine image as both the objects C and D exhibit the same illumination pattern but actually this is a case of image forgery. Hence, checking of illumination direction of objects present in the image is insufficient and an enhancement in the technique is expected.

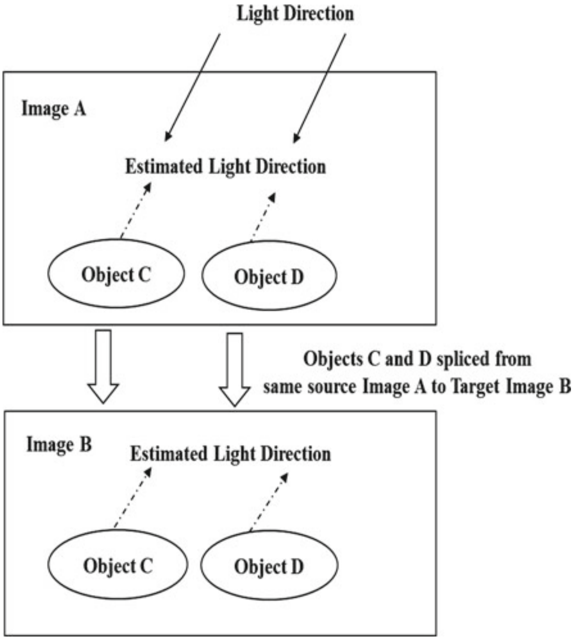

Light Direction
Image A
Estimated Light Direction
Object C
Object D
Objects C and D spliced from
same source Image A to Target Image B
Image B
Estimated Light Direction
Object C
Object D

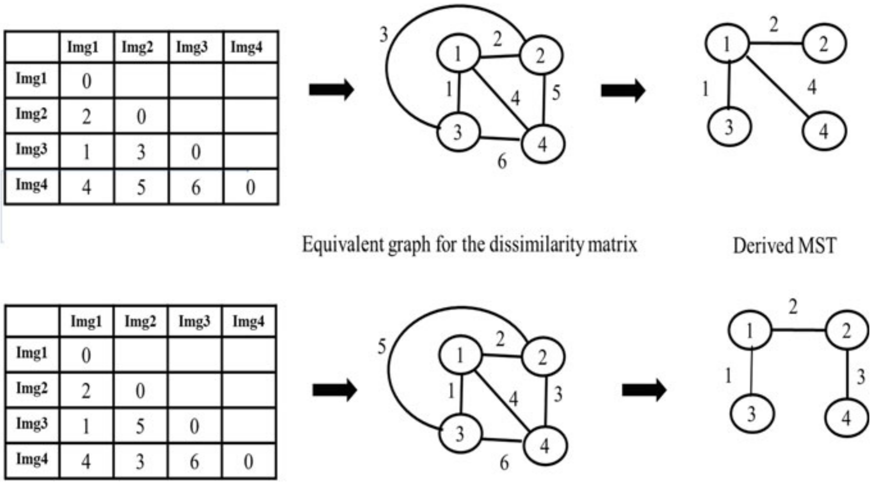
Fig. 7 Incorrect illumination direction estimation due to spliced objects from same source

4.2 Image Forgery Detection and Source Camera Identification

In this case, a hash comprising of both device features and image features is generated. There are techniques proposed for detection of forgery and source device. However, there is a need to distinguish between genuine perceptually robust image manipulation operation like rotation and scaling with those of forgery image manipulation operation while examining source camera. On combining approaches related to detection of source camera and image forgery, there is an increase in false alarms where genuine image editing operations are detected as manipulation [22]. There are techniques that can attain perceptual robustness of around 99% working independent of source device identification. Incorporating them with those of source device identification and manipulation detection is required. As currently employed techniques that can detect image forgery and distinguish between perceptual robust operation too are only around 90% which can be further tried for improvement.

4.3 Image Provenance Analysis for Tracking Image History

Provenance graph gives a set of all images related to a given query image and possible derivation tree for that image. The donor images could be of varied sizes and can



	Img1	Img2	Img3	Img4
Img1	0			
Img2	2	0		
Img3	1	3	0	
Img4	4	5	6	0

	Img1	Img2	Img3	Img4
Img1	0			
Img2	2	0		
Img3	1	5	0	
Img4	4	3	6	0

Fig. 8 Effect of near-duplicate images on conversion of dissimilarity matrix to provenance graph

represent multiple regions in the given query image. Identifying the related images from a huge database of images is a challenging task as the amount of comparisons increase. Following are the challenges associated to the study in this domain:

- Small donor identification in provenance analysis for image forgery detection: When the contributing donors for a query image become small in size accounting for size as less as 1 to 10% of total image size it becomes difficult tracing its features and matching them with related images from the database. The recall rate for donors of small sizes is approximately in the range 55–58% [23]. In comparison to donors of medium and large size that have recall rate of small donor between 75 and 80% the recall rate of small donors needs to be improved. Hence, a check on appropriate analysis of small donors is important in the process of provenance analysis.
- Improving the dissimilarity matrix construction by considering noise from near-duplicate images:

  Provenance graph is generated using the minimum spanning tree algorithm computed over the dissimilarity matrix. If there is a minor change in the values of the dissimilarity matrix due to noise from near-duplicate images it would change the entire derivation process of graph as the spanning tree would vary. It is thereby important to extract features of the images that are near duplicates with care in order to distinguish them properly and derive appropriate graph. Figure 8 shows the actual distance between images Img1, Img2, Img3 and Img4 and a slightly modified dissimilarity matrix due to noise from the near-duplicate images. An equivalent graph for the stated matrix and the spanning tree is also stated for both original and modified dissimilarity matrices. The images Img2, Img3 and Img4 are all derived from Img1. A slight modification in the matrix changes the entire derivation process of the images. As can be seen that Img1 is the prominent root Image from which Img2 and Img3 are derived same as previous tree. However, Img4 is derived from Img2 which is different from the previous case of original tree. Hence, a small change in the values of dissimilarity matrix can change the entire provenance graph constructed. This signifies the importance of dealing with near-duplicate images efficiently.

- Provenance graph construction using features other than image properties: Graphs constructed relying on only image features lose on certain important features which can help build provenance graph quickly. Using metadata present in images rather than merely image features can be useful in reducing the time associated for the entire provenance analysis.

5 Research Direction

Possible area of research for the stated research gaps in the discussion section is mentioned below:

- Passive image forgery detection using illumination-based techniques: Analysis of images using passive forgery detection mechanism requires detection and estimation of illumination direction of various objects present in the image. This illumination direction estimation can be erroneous if objects cast shadow over each other. This leads to the problem of concluding that an image is forged despite being genuine and raising false alarms. Also, an image generated using components from same donors will be estimated to have same illumination direction. Hence, checking merely the illumination direction of objects in the image will be insufficient. Checking of background illumination could be a possible alternative.
- Image Forgery Detection using Source Camera Identification: Techniques for image forgery detection using source camera identification are based on detection of features from images and camera or capturing device. The features extracted from this technique should be able to well distinguish between operations that are genuine and those that have manipulated the image content. Some of the operations that are performed over images to improve their quality are rotation and scaling. If operations that are genuine are identified as forgery it will lead to unnecessary false alarms. Hence, there is a need to check the nature of manipulation while checking for the source of forgery. A technique suitable in both the cases needs to be devised.
- Improving Image Provenance Analysis Process: Provenance graphs constructed using the phases of image filtering and provenance graph construction require searching huge database of images and deducing relationship between images. This process gets difficult as the search space is very large and there could be multiple objects of varied sizes in the query image. Small donors affect the accuracy of the approaches used as slight modifications like converting digit 0–8 or 1–7 in an

image is not easy to identify. Hence, there is a need to improvise provenance graph construction for small-sized donors.

The search and comparison phases in case of provenance graph construction are very large. Image metadata provide useful information like date, compression strategy, etc. which can be helpful in the provenance graph construction [12]. Rather than merely relying on the image features other complimentary aspects that come with an image need to be analysed, which may help speed up the process of provenance graph generation.

Provenance graph is built using minimum spanning tree algorithm from the dissimilarity matrix. If there is variation in the dissimilarity matrix the provenance graph will too vary. The chances of variation increase when the images are near duplicate of each other. Hence, near-duplicate images need to analysed before building the provenance graph.

6 Conclusion

A detailed review on passive method of detecting forgery through illumination detection is discussed in the paper. Areas where both image properties and capturing device properties are given attention to check for information contributing in detection of forged images is also a topic of discussion in this paper. A discussion regarding provenance analysis for building derivation tree for entire image manipulation process is mentioned in detail. Research direction and areas to explore in the field on digital image forensics are elaborated well in the paper.

7 Future Scope

In future, the problem of image forgery detection can be used to address issues like considering societal impact on a particular forgery. This cultural trend will help understand the reason for a particular manipulation better. Detection of video manipulation which is also a mode of multimedia information transfer can turn deceptive if modification of sequence of images present in the video is carried out. These further areas of research can be fruitful broad domains of study.

References

1. Cao Y, Zhang L, Chang C (2016) Using image sensor PUF as root of trust for birthmarking of perceptual image hash. In: 2016 IEEE Asian hardware-oriented security and trust (AsianHOST)
2. Zheng Y, Dhabu S, Chang C (2018) Securing IoT monitoring device using PUF and physical layer authentication. In: 2018 IEEE international symposium on circuits and systems (ISCAS)

3. Zheng Y, Cao Y, Chang C (2020) A PUF-based data-device hash for tampered image detection and source camera identification. IEEE Trans Inf Forensics Secur 15:620–634
4. Davarzani R, Mozaffari S, Yaghmaie K (2016) Perceptual image hashing using center-symmetric local binary patterns. Multimed Tools Appl 75:4639–4667
5. Roy A, Chakraborty R, Sameer U, Naskar R (2017) Camera source identification using discrete cosine transform residue features and ensemble classifier. In: *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*
6. Sameer V, Sarkar A, Naskar R (2017) Source camera identification model: Classifier learning, role of learning curves and their interpretation. In: *2017 international conference on wireless communications, signal processing and networking (WiSPNET)*
7. Teerakanok S, Uehara T (2019) Copy-move forgery detection: a state-of-the-art technical review and analysis. IEEE Access 7:40550–40568
8. Matern F, Riess C, Stamminger M (2020) Gradient-based illumination description for image forgery detection. IEEE Trans Inf Forensics Secur 15:1303–1317
9. Moreira D, Bharati A, Brogan J, Pinto A, Parowski M, Bowyer K, Flynn P, Rocha A, Scheirer W (2018) Image provenance analysis at scale. IEEE Trans Image Process 27:6109–6123
10. Pinto A, Moreira D, Bharati A, Brogan J, Bowyer K, Flynn P, Scheirer W, Rocha A (2017) Provenance filtering for multimedia phylogeny. In: 2017 IEEE international conference on image processing (ICIP)
11. Bharati A, Moreira D, Pinto A, Brogan J, Bowyer K, Flynn P, Scheirer W, Rocha A (2017) U-Phylogeny: undirected provenance graph construction in the wild. In: *2017 IEEE international conference on image processing (ICIP)*
12. Shichkina Y, Tishchenko V, Fatkieva R (2020) Synthesis of the method of operative image analysis based on metadata and methods of searching for embedded images. In: *2020 9th mediterranean conference on embedded computing (MECO)*
13. Bharati A, Moreira D, Brogan J, Hale P, Bowyer K, Flynn P, Rocha A, Scheirer W (2019) Beyond pixels: image provenance analysis leveraging metadata. In: 2019 IEEE winter conference on applications of computer vision (WACV)
14. Bharati A, Moreira D, Flynn P, Rezende Rocha A, Bowyer K, Scheirer W (2021) Transformation-aware embeddings for image provenance. IEEE Trans Inf Forensics Secur 16:2493–2507
15. Fernandez J, Pandian N (2018) JPEG metadata: a complete study. In: 2018 international conference on recent trends in advance computing (ICRTAC)
16. McKeown S, Russell G, Leimich P (2017) Fast filtering of known PNG files using early file features
17. Gloe T (2012) Forensic analysis of ordered data structures on the example of JPEG files. In: 2012 IEEE international workshop on information forensics and security (WIFS)
18. Mullan P, Riess C, Freiling F (2019) Forensic source identification using JPEG image headers: the case of smartphones. Digit Investig 28:S68–S76
19. Rhee K (2020) Detection of spliced image forensics using texture analysis of median filter residual. IEEE Access 8:103374–103384
20. Sekhar P, Shankar T (2021) Splicing forgery localisation using colour illumination inconsistencies. Int J Electron Secur Digit Forensics 13:346
21. Kumar S, Kasiselvanathan, Vimal (2021) Image splice detection based on illumination inconsistency principle and machine learning algorithms for forensic applications. In: 2021 smart technologies, communication and robotics (STCR)
22. Tang Z, Zhang X, Li X, Zhang S (2016) Robust image hashing with ring partition and invariant vector distance. IEEE Trans Inf Forensics Secur 11:200–214
23. Zhang X, Sun Z, Karaman S, Chang S (2020) Discovering image manipulation history by pairwise relation and forensics tools. IEEE J Sel Top Signal Process 14:1012–1023
24. Castelletto R, Milani S, Bestagini P (2020) Phylogenetic minimum spanning tree reconstruction using autoencoders. In: ICASSP 2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)

25. Thakur R, Rohilla R (2020) Recent advances in digital image manipulation detection techniques: a brief review. Forensic Sci Int 312:110311
26. Yao H, Xu M, Qiao T, Wu Y, Zheng N (2020) Image forgery detection and localization via a reliability fusion map. Sensors (Basel) 20:6668
27. Kadam K, Ahirrao S, Kotecha K (2022) Efficient approach towards detection and identification of copy move and image splicing forgeries using Mask R-CNN with MobileNet V1. Comput Intell Neurosci 2022:6845326
28. Tolias G, Jégou H (2014) Visual query expansion with or without geometry: refining local descriptors by feature aggregation. Pattern Recognit 47:3466–3476

Personality-Based Friends Recommendation System for Social Networks

Harshit Garg, Mradul Tiwari, Mehul Rawal, Kaustubh Gupta, Ranvijay, and Mainejar Yadav

Abstract Social media platforms have created a buzz among people in the last couple of decades. These platforms offer a chance to connect with each other irrespective of the geographic distance between them. These platforms help in connecting with others by recommending new users to them. These recommendations are generally based on the user's profile, mutual friends, interests, and preferences. But the current social media platforms do not consider the kind of a person and his personality. This work considers the user's personality along with other parameters like mutual friends, age group, and interests. The personality of a person has been predicted from the posts that person has posted on his social media account.

Keywords Recommendation system · Personality prediction · Myers–Briggs-type indicator (MBTI) · TF-IDF · SVM · RBF · Logistic regression

1 Introduction

With the advancement in technology and internet now being available to most people, social media platforms have witnessed a great increase in their user base. Social media has brought people closer to each other from all over the globe. Sharing ideas and information in various network groups and communities have really helped a lot in connecting people worldwide. These platforms helped a lot in strengthening the relationship with friends and other people with whom we are unable to meet personally.

People are becoming increasingly interested in connecting with others, so it has become significant that the recommendation systems suggesting friends must be more precise and understand a user's personality. Most of the existing friend recommendation systems are based on an individual's social network (i.e., their mutual

H. Garg (✉) · M. Tiwari · M. Rawal · K. Gupta · Ranvijay
Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India
e-mail: harshitgarg30920@gmail.com

M. Yadav
Rajkiya Engineering College Sonbhadra, Markundi, India

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_4

connections), interests, etc. But they do not consider the user's personality traits and what kind of a person the user is. Over the years, various machine learning techniques have been introduced, which can be used to our advantage to predict a user's personality. This research paper aims to understand how a person's personality can be identified by the content they share on the social media platforms, which can then be leveraged to build a better friend recommendation system based on personality matching. Over the years, many different types of social media recommendation systems have been designed. Our goal of this research is an attempt to implement a friend recommendation system where users are recommended new connections which could be relevant in some way to them. We considered a variety of parameters like a person's personality, age group, interests, hobbies, and existing friends. On these factors, we have designed a recommendation system that uses personality prediction as its core to suggest better connections.

The remaining part of this paper is as follows; Sect. 2 describes the related work. A discussion about the used dataset is given in Sect. 3. Section 5 describes the methodology of the proposed work. The experimental results and discussion is given in Sect. 5. Section 6 shows a brief conclusion.

2 Related Work

Over the past few years, many studies and surveys have been conducted in order to understand the proper mechanism of recommending friends to a person. The result of these studies suggests that the majority of the social networking platforms show friend recommendations using closeness of people in their social connections graph [1]. But it has been concluded that this is not the right strategy to recommend friends. Friend recommendations must be more centric towards the way in which people make friends in their real life [2].

In some of the previous research, it has been proven that the data reflect a person's personality they post online, and that data is of great value and can be further used to predict an individual's personality [3] effectively. Also, what people express through their posts and status updates can be used for research in studying human personality and behavior [4]. Some feature reduction techniques can also be used which may prove to be beneficial because they reduce the input data size which helps in improving the prediction accuracy [5]. It has been shown that the collaborative filtering approaches using social information and mutual understanding data among people in existing social network connections have also produced significant results [2]. It has been observed that recurrent neural networks can capture the sequential information from the text data [6].

Various personality tests have been in use for a very long time now. The usage of these tests can be traced back to the time of World War I [7]. This test was first introduced by Katharine Briggs and Isabel, who began their research into personality in 1917 [8]. The Big 5 model was given by D. W. Fiske and expanded by Norman, Goldberg, and McCrae [9]. Anandhan et al. [10] described the various challenging issues and deep literature survey related to Social Media Recommender Systems. Michael et al. [11] introduced a personality prediction approach. This method used Facebook's human behavior and the Big 5 model. The accuracy achieved by this approach is 74.2%. Kamalesh and Bharathi [12], have introduced a personality prediction model. This method used the Big 5 traits and machine learning Technique. The accuracy of this model is 78.34%.

3 Dataset

The dataset used in this work consists of posts from 8675 users. It contains around 50 most recent posts for each user. Each post for a user is separated by using three pipes(|||). The dataset also contains the user's personality type from one of the 16 MBTI types. Each personality type is made up of a combination of four characters, and each character is contributed by the first or second letter of the traits from the four MBTI classes. The dataset has the following distribution of the various MBTI traits in each category:

Introversion(I): 6676; Extraversion(E): 1999
Sensing(S): 7478; Intuition(N): 1197
Thinking(T): 4694; Feeling(F): 3981
Judging(J): 5241; Perceiving(P): 3434.

The MBTI assessment is based on the fact that people have some preferred modes of judgment (thinking or feeling) and perception (sensing or intuition), as well as their orientation to the outer world (judging or perceiving). One other important aspect is the attitude about how people build energy (extroversion or introversion).

4 Methodology

Our proposed approach recommends friends using similarities in personality traits and their existing social connections. It also considers the user's age group and their various interest fields. For a given user, our system will recommend a list of users with the highest recommendation score that could become potential friends to that user. The workflow diagram in Fig. 1 shows the overall functioning and various phases of the proposed system.

4.1 Personality Prediction

The following subsection describes the procedure followed to obtain an individual's personality type.

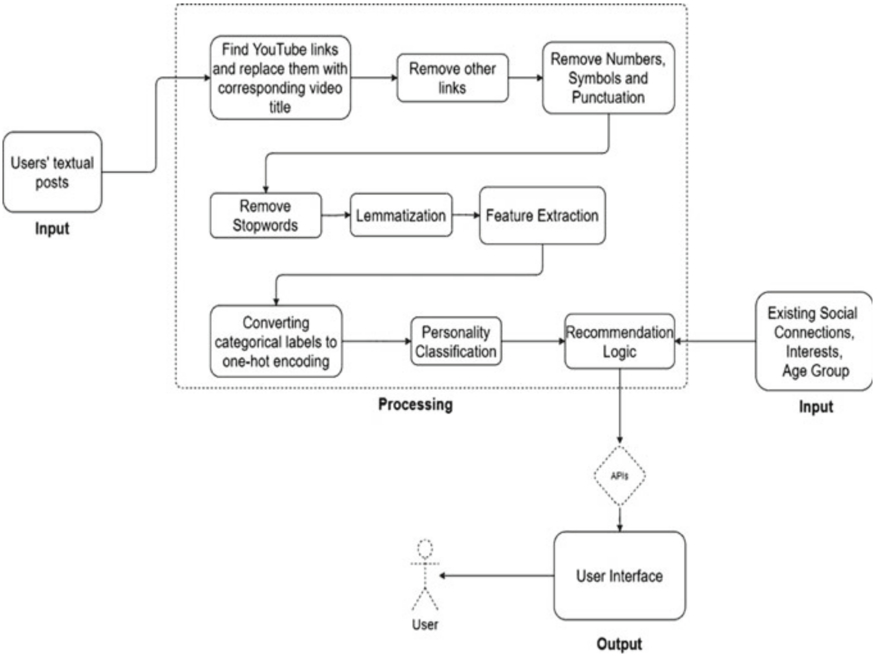

Users' textual posts
Input
Find YouTube links and replace them with corresponding video title
Remove other links
Remove Numbers, Symbols and Punctuation
Remove Stopwords
Lemmatization
Feature Extraction
Converting categorical labels to one-hot encoding
Personality Classification
Recommendation Logic
Processing
Existing Social Connections, Interests, Age Group
Input
APIs
User Interface
User
Output

Fig. 1 Workflow of proposed work

Data Preprocessing In this work, we have followed a series of steps which are described below in a stepwise manner.

> The user posts are textual, but they also contain URLs to various other websites for images or videos. We are only considering youtube URLs in this work. Firstly, extract the youtube URLs using regex (i.e., regular expressions) and replace those URLs with their corresponding video titles. All other URLs are removed from the user posts using the same regex method.
> User posts also contain symbols and punctuation marks like periods, commas, apostrophes, etc. These do not include any meaning and are hence removed from the text. Numbers also do not have any sense; hence they are removed. Extra spaces between words are also removed during this step. Stopwords are then removed from the posts as they do not add much meaning or information to a sentence or text. These words can be simply ignored while considering the text without any loss in the sentence's meaning. Lemmatization is then applied on the text to convert the words to their root form as different forms of a word convey the same meaning.

Feature Extraction Feature extraction is done to convert the raw text data into vectors of decimal numbers. One common technique for extracting features is TD-IDF (i.e., Term Frequency-Inverse Document Frequency). This calculates the relative importance of words within the document.

Model Training The task at hand is to classify the user to 1 of the 16 MBTI personality types based on his processed posts. Looking closely at the MBTI dataset and the output labels, it can be observed that there are four different classes of MBTI. Each class consists of two personality traits, and either one can be chosen for a person. Instead of classifying a person directly to 1 of the 16 types, we have divided this task into 4 steps corresponding to each of the 4 MBTI classes. The above problem is now converted to a multi-output binary classification problem. The two traits of each of the four classes are assigned either 0 or 1. Support Vector Machine (SVM) is trained on the data. A nonlinear Radial Basis Function (RBF) is used as a kernel to train the SVM classifier. Logistic Regression is similarly trained on the data. Neural network is also trained on the data. The input layer consists of 5000 nodes (i.e. equal to feature vector size) followed by 7 hidden layers with 256, 200, 160, 120, 80, 16, and 6, respectively. The output layer has two nodes for two personality traits. The output layer uses softmax, whereas hidden layers use ReLu as their activation functions in neural networks. Various classifiers like Multinomial Naive Bayes, K-Nearest Neighbors, Random Forests, and Decision Trees are also trained on the dataset.

4.2 Recommendation Logic

Finally, after a person's personality is predicted using the above-trained models, the final step is to recommend friends to a user. Since we are considering a number of other parameters (other than personality) like mutual connections, age group, and user interests, we have to assign a certain weightage to each parameter and calculate an overall recommendation score for each user. The percentage weightage of various parameters is as follows: Personality—30%, Mutual connections—30%, Age group—20%, and User interests—20%

To calculate the recommendation score for users, follow the calculations as given in Eqs. (1)–(5).

$$\text{Personality score} = \frac{\text{Number of personality traits matching for two users}}{4} \tag{1}$$

Calculating mutual connections score is a two-step process. In the first step, a value is calculated for a new user based on how frequent that user is in the friends list of the user's friends.

$$\text{Users score} = \sum_{i=1}^{d} \frac{\text{Number of common friends at depth}\, i}{i}$$

where d is the depth of mutual connections to be considered.

In the second step, the values of the user's score calculated in the first step are normalized using the below formula.

$$\text{Mutual Connections score} = \frac{\text{Users score}}{\text{Maximum of all users score calculated}} \tag{2}$$

$$\text{Age Group score} = 1 - \frac{\text{Absolute difference of two users ages}}{100} \tag{3}$$

$$\text{Tags score} = \frac{\text{Number of common tags for two users}}{\text{Minimum of number of tags of both users}} \tag{4}$$

So, the final recommendation score for a user can be calculated as

$$\begin{aligned}\text{User Recommendation Score} = {} & (\text{Personality weightage} * \text{Personality score}) \\ & + (\text{Mutual connections weightage} * \text{Mutual connections score}) \\ & + (\text{Age group weightage} * \text{Age group score}) \\ & + (\text{Tags weightage} * \text{Tags score})\end{aligned} \tag{5}$$

5 Results and Discussions

Various supervised machine learning classification algorithms were trained on the MBTI dataset. These models were compared based on prediction accuracy and evaluated based on other metrics like precision, recall, and F1-score. This section summarizes our observations and the results obtained from different classifiers. The personality prediction accuracy scores achieved from various models are given in Table 1. It was observed from Table 1 that support vector machines are the most

Table 1 Accuracies of classification models

Classifier	I/E (%)	S/N (%)	T/F (%)	J/P (%)	Average (%)
Support vector machine	85.5	89.7	85.4	80.1	85.2
Logistic regression	82.6	87.1	85.8	79.4	83.7
Neural network	77.2	87.2	81.5	77.9	81.0
Multinomial Naive Bayes	77.0	86.2	78.0	66.6	77.0
Decision tree	81.6	87.7	77.3	73.4	80.0
Random forest	76.8	86.2	80.2	63.5	76.7
K-nearest neighbors	80.4	87.4	68.2	68.9	76.2

Table 2 Evaluation metrics for SVM, logistic regression, and neural network

Personality traits	SVM		Logistic Regression		Neural network	
	Precision	Recall	Precision	Recall	Precision	Recall
Extroversion(E)	0.82	0.48	0.82	0.32	0.51	0.72
Introversion(I)	0.86	0.97	0.83	0.98	0.90	0.79
Intuition(N)	0.90	0.99	0.87	0.99	0.93	0.92
Sensing(S)	0.82	0.33	0.72	0.11	0.54	0.57
Feeling(F)	0.87	0.86	0.86	0.88	0.83	0.83
Thinking(T)	0.84	0.84	0.85	0.83	0.80	0.80
Judging(J)	0.81	0.65	0.83	0.60	0.72	0.72
Perceiving(P)	0.80	0.90	0.78	0.92	0.82	0.82

accurate model for classification with an average accuracy of 85.2%. Support vector machine also leads in accuracy for the three classes (*I*/*E*, *S*/*N*, and *J*/*P*) as compared to other models. Logistic regression performed comparatively better than SVM for *T*/*F* class. Other classifiers, such as logistic regression and neural networks, achieved accuracy above 80%. The precision and recall values, as observed on the test data for the three best classification models, i.e., support vector machine, logistic regression, and neural network, are summarized in Table 2. In order to compare and visualize the performance of various classification models, receiver operating characteristics curves have been generated for different MBTI categories, which can be seen in Figs. 2, 3, 4, and 5. ROC curves are interpreted using the AUC (Area Under Curve)

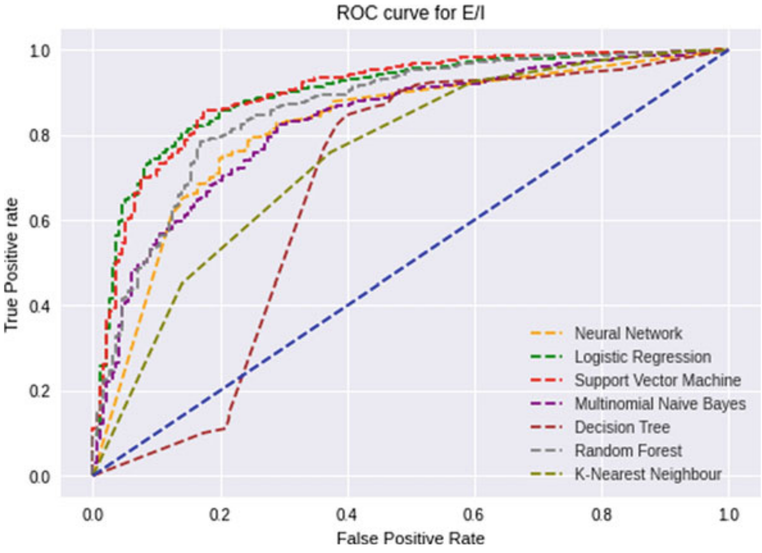


Fig. 2 ROC curve for *E*/*I*

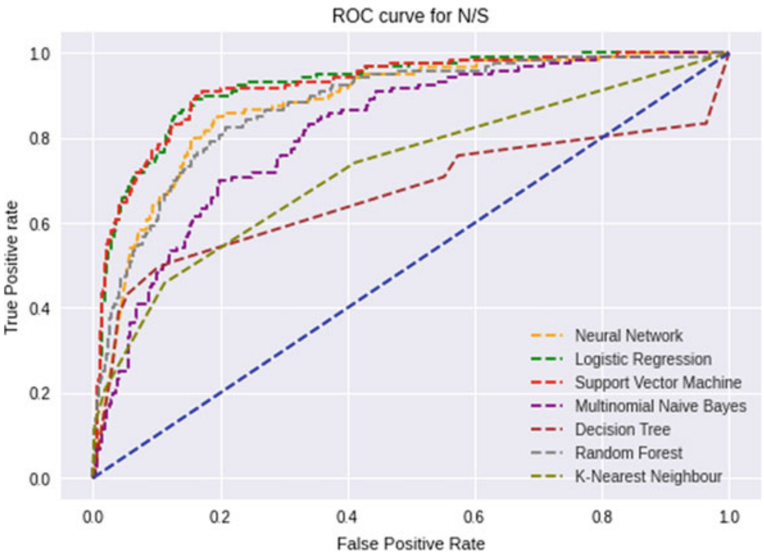


Fig. 3 ROC curve for *N*/*S*

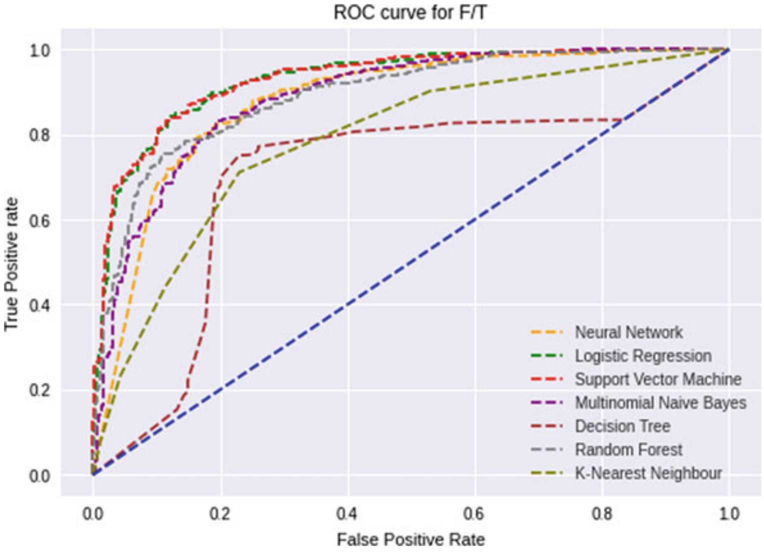


Fig. 4 ROC curve for *F*/*T*

values. The more the AUC, the better the model separates the classes. It can also be observed from the ROC curves that support vector machines and logistic regression performed comparatively better than other classifiers for all four MBTI categories.

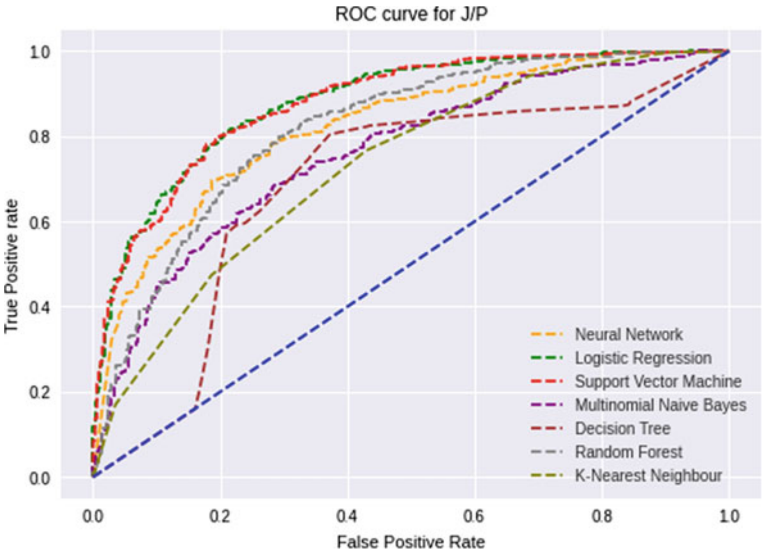


Fig. 5 ROC curve for *J*/*P*

6 Conclusion

The main objective of this paper is to build a friend recommendation system that uses personality prediction along with other parameters. The personality is predicted using users' textual posts. The text was preprocessed and cleaned to generate the feature vector, which was further used as input to various machine learning models.

It was observed that increasing the number of hidden layers increases the accuracy of the Neural Network model. But increasing the hidden layers beyond a certain threshold does not increase the performance but only increases the overhead of the prediction. In the case of K-nearest neighbors, the prediction was very slow as it is a lazy learning algorithm that evaluates the whole dataset every time it has to make a prediction. The best accuracy is obtained using support vector machine with a nonlinear kernel (i.e., radial basis function) to generate feature vectors. In this work, an accuracy of up to 85.2% is achieved.

References

1. Cheng S, Zou G, Huang M, Zhang B (2019) Friend recommendation in social networks based on multi-source information fusion. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-017-0778-1
2. Shahane A. Galgali R (2016) Friend recommendation system for social networks. IOSR J Comput Eng (IOSR-JCE), pp 37–41

3. Alsadhan N, Skillicorn D (2017) Estimating personality from social media posts. In: IEEE International conference on data mining workshops, pp 350–356
4. Tandera T, Suhartono D, Wongso R, Prasetio YL (2017) Personality prediction system from facebook users. In: (ICCSCI)
5. Terol RM, Reina AR, Ziaei S, Gil D (2020) A machine learning approach to reduce dimensional space in large datasets. University of Alicante
6. Hernandez R, Knight IS (2017) Predicting myers-briggs type indicator with text classification. In: Conference on neural information processing systems (NIPS)
7. Gibby RE, Zickar MJ (2008) A history of the early days of personality testing in American industry: An obsession with adjustment. In: History of psychology. https://doi.org/10.1037/a0013041
8. King SP, Mason BA (2020) Myers briggs type indicator. In: Wiley encyclopedia of personality and individual differences: vol. II, measurement and assessment
9. Johnson J (2017) Big-five model. In: Encyclopedia of personality and individual differences, pp 1–16. https://doi.org/10.1007/978-3-319-28099-8_1212-1
10. Anandhan A, Shuib L, Ismail MA, Mujtaba G (2018) Social media recommender systems: review and open research issues. IEEE Access 6:15608–15628. https://doi.org/10.1109/ACCESS.2018.2810062
11. Tadesse MM, Lin H, Xu B, Yang L (2018) Personality predictions based on user behavior on the facebook social media platform. IEEE Access 6:61959–61969
12. Kamalesh MD, Bharathi B (2022) Personality prediction model for social media using machine learning technique. Comput Electr Eng 100

Analysis of Different Sampling Techniques for Software Fault Prediction

Sanchita Pandey and Kuldeep Kumar

Abstract The process of predicting whether or not a software module is faulty based on specific metrics is known as software fault prediction. Software faults predicted in prior stages help in the management of resources and time required during software testing and maintenance. Over the years, various supervised machine learning-based techniques for fault prediction have been suggested. The models are trained using a labeled dataset that consists of multiple independent variables like lines of codes, the complexity of the software, the size of the software, etc., and a dependent binary variable that is either true or false. Recent research in software fault prediction focuses on data quality. In this paper, we have mainly focused on the class imbalance problem. In imbalanced data, one of the class labels has a higher number of observations, while the other class label has a lower number of observations. Different over-sampling and under-sampling techniques are used to tackle the class imbalance problem. In this paper, random under-sampling, random over-sampling, and SMOTE are applied to six PROMISE datasets. A decision tree classifier is used to create the model. The efficiency of different sampling techniques is compared using the ROC-AUC score and F1-score.

Keywords Software fault prediction · Random under-sampling · Random over-sampling · SMOTE · Decision tree classifier · ROC-AUC score · F1-score · Data imbalance

1 Introduction

With the growth in size and complexity of software [1], detecting faults becomes a laborious task. Therefore, detecting faults in the prior stages of software development is preferable. This reduces the efforts required in the testing and maintenance stages. Prior detection of faulty modules helps in the optimal distribution of resources for the

S. Pandey (✉) · K. Kumar
Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Punjab, India
e-mail: pandeysanchita11@yahoo.com

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_5

timely development of the project. Software Quality assurance helps in maintaining the quality and quantity of resources utilized in the software development process.

Software fault prediction (SFP) is detecting faults in a software module [1, 2]. It is a notably significant topic in software engineering. Detection of faults in software modules helps developers to focus on a particular module to rectify it on time. SFP eases the work of the testing team by identifying the modules containing faults. Now, one has to look into specific modules instead of checking the complete software. With the increase in complexity of the software, finding faults is getting more difficult each day. But with a good SFP model, detecting faults has become a more straightforward job.

SFP is used in not only classifying the modules as faulty or non-faulty but also helps in finding the number of faults and the severity of these faults. But major studies are done on the classification of software modules [3]. A model is built using different algorithms, which are then trained using a labeled fault dataset [4].

As per the research, the fault dataset suffers from the problem of class imbalance (CI). CI dataset contains one of the class labels having a higher number of observations, while the other class label has a lower number of observations [5, 6]. A model prepared using an imbalanced dataset gives biased results, resulting in inaccurate system evaluation. Over time, various algorithms have been proposed to overcome the CI problem [7]. Some of the most used data sampling algorithms are SMOTE [8], random over-sampling, random under-sampling, cluster-based clustering, etc. Researchers have applied most of these sampling techniques to different fault datasets and calculated the efficiency of the models. In this paper, we have applied ensemble models and deep learning models to six different PROMISE datasets. The ROC-AUC and F1-score is used to compare the sampling techniques.

The rest of the paper is as follows: Sect. 2 gives background knowledge about issues in SFP and the data sampling techniques used in this paper. Section 3 explains the step-by-step process followed in SFP. The literature survey is discussed briefly in Sect. 4 followed by Sect. 5 which contains information about the dataset used in this study. Section 6 contains the results followed by the conclusion and future work.

2 Background

2.1 Data Quality Issues in SFP Dataset

The datasets used in SFP suffer from various data quality issues. These are

- class overlapping,
- class imbalance,
- incompleteness,
- outliers,
- redundancy,
- inconsistency,

- noise,
- data quality metadata, etc.

To build an efficient model for SFP, one needs to tackle the above-given issues. Otherwise, the reliability of the model will decrease resulting in inaccurate outcomes.

This paper specifically focuses on the class imbalance problem in the fault dataset. An imbalanced dataset contains an asymmetrical distribution of data across the different classes. A model built on imbalanced data results in biased results. This makes the model incorrect.

2.2 Class Imbalance Problem in SFP

There are numerous factors that affect the working of SFP models. These are

- over-fitting of the model,
- noise in the dataset,
- cost parameters,
- software metrics,
- imbalanced dataset,
- feature selection, etc.

One of the most serious issues with fault datasets is the issue of class imbalance (CI) [5, 6]. A biased dataset with an unequal distribution of classes is known as an imbalanced dataset. One class data is in the majority, whereas another class data is in the minority. When a model is applied to this dataset, it produces biased results, resulting in an inaccurate system evaluation.

To solve this issue, various sampling algorithms are used [6, 7]. These sampling algorithms are combined with pattern learning algorithms to create high-performance SFP models. Data sampling procedures such as over-sampling and under-sampling are widely used [5, 9] to overcome CI problems. Both strategies aim to enhance the working of SFP.

2.3 Data Sampling Techniques

On datasets with skewed class distributions, models tend to overestimate their performance. It's possible to improve the class distribution by balancing or randomizing a training dataset using data sampling. After balancing the dataset, standard machine learning algorithms can be used to train it. Figure 1 shows the different types of data sampling techniques.

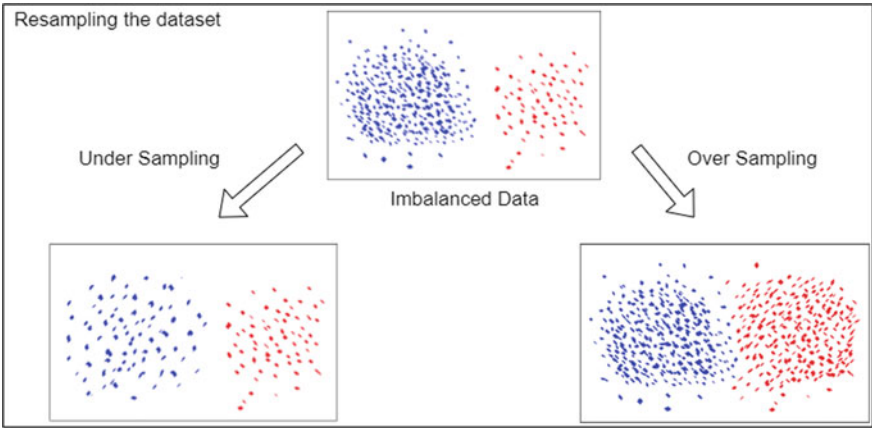


Fig. 1 Different data sampling techniques were used to balance the dataset

There are various sampling techniques used in the field of SFP. These are

(1) **Over-Sampling**: To achieve data equality, over-sampling techniques either duplicate existing instances from the minority class or artificially generate new samples from the minority class.
(2) **Under-Sampling**: Under-sampling methods remove or select a subset of the majority class's examples to balance the dataset.

In this study, we have used random over-sampling, random under-sampling, and SMOTE. A machine learning model is built using a decision tree classifier. ROC-AUC score and F1-score are used to compare the efficiency of these sampling techniques.

2.4 Random Under-Sampling

One of the most straightforward methods for sampling is random under-sampling. The ratio of minority to majority class data in the training set is adjusted by randomly removing majority class data. It is theoretically difficult to control what information about the majority class is discarded when using random under-sampling. It has been demonstrated empirically that random under-sampling is one of the most effective re-sampling methods, in spite of its apparent simplicity.

2.5 Random Over-Sampling

In this case, random members of the minority class are selected and then added to the new training set; these random members are then duplicated.

When randomly over-sampling, two things should be kept in mind. In the first step, a random selection of data is made from the original training set and not the new one. In addition, one always over-samples with replacement. Over-sampling without a replacement would quickly deplete the minority class, preventing it from achieving the desired level of minority–majority balance.

2.6 Smote

The Synthetic Minority Over-sampling Technique (SMOTE) is a widely used over-sampling technique. SMOTE was given by [8] in 2002. This algorithm facilitates tackling the over-fitting issue caused by random over-sampling. To create new instances, the algorithm interpolates between pairs of positive examples that are spatially close to one another in the feature space.

2.7 Decision Tree

A decision tree (DT) is a supervised learning algorithm having a tree-like structure where dataset attributes are depicted by internal nodes. Branches are used for decision-making and leaf nodes are used for classification results. It is a graphical representation for finding every solution to a problem based on predefined parameters. It is a simple algorithm and anyone can visualize it because of its tree-like structure.

The decision tree algorithm is used in the experimental work because it is an easy-to-understand algorithm and also requires less data cleaning as compared to other algorithms. There are different data cleaning techniques. But, in this paper, the main focus is on the class imbalance problem. Therefore, the decision tree is a better choice as less data cleaning is required in it.

3 Software Fault Prediction Process

Figure 2 shows the generic seven-step process followed by a software fault prediction model

1. ***Data collection***: There are various public datasets available for fault prediction. In this step, a dataset is either created or publicly available datasets are used.
2. ***Data pre-processing***: The fault prediction datasets may suffer from the problem of noise, outliers, the presence of irrelevant data, and class imbalance problems. In this step, various noise removal techniques along with sampling techniques are applied to the dataset.

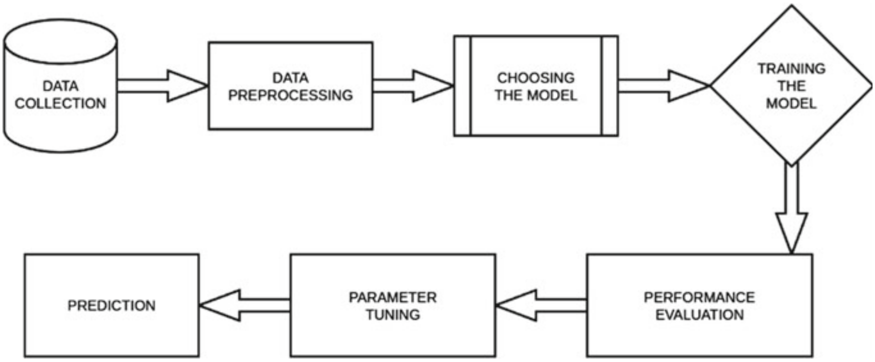


Fig. 2 Software fault prediction process

3. ***Choosing the model***: Based on the dataset and the objective, a machine learning model is selected. Either traditional models can be selected, or one can modify an existing machine learning model or create a new model.
4. ***Training the model***: In this step, the model is prepared using a fault prediction dataset. The model trains itself using the training set with the specified algorithm. The dataset is split into training and testing sets.
5. ***Evaluation***: After the training stage, the model is tested using the testing data. The system's ability to correctly predict faulty software modules is measured using parameters such as precision, accuracy, and the F1-score. User-defined values are also fed as input to test the efficiency of the model.
6. ***Parameter Tuning***: In this step, the variables of the model are modified to get better performance and model accuracy. Once the parameters are tuned, the model is again evaluated.
7. ***Prediction***: The fault prediction model is evaluated against real-world data and the results are recorded.

4 Literature Review

In the domain of software engineering, SFP has become the most prominent area of study. SFP not only detects faulty software modules but also aids in software quality improvement by eliminating the fault during the initial phases of the project development. In this section, recent work done by researchers in the field of SFP is discussed. The studies related to machine learning (ML), deep learning (DL), and ensemble learning (EL) methods in the fault prediction domain are discussed below.

Rathore et al. [10] have introduced three separate generative over-sampling methods, namely, Conditional GAN (CTGAN), Vanilla GAN, and Wasserstein GAN with Gradient Penalty (WGANGP). The experiment is conducted on the PROMISE, JIRA, and Eclipse datasets. When these sampling methods are used with baseline models in tests on fault datasets, the results of the baseline models are greatly improved.

For SFP in intra-release and cross-release models, Singh and Rathore [11] adopted non-linear and linear Bayesian regression approaches. SMOTE data sampling methods are applied along with Random Forest (RF), Support Vector Machine (SVM), Linear Regression (Lr), Linear Bayesian Regression (LBr), and Non-linear Bayesian Regression (NLBr). Bayesian non-linear regression surpassed linear regression approaches on a sample containing 46 different software projects. Mean absolute error (MAE), root mean square error (RMSE), and fault percentile average (FPA) are used as evaluation metrics.

SHSE was put forward by Tong et al. [12]. It is a combination of different sampling, feature subspace, and ensemble learning. Subspace hybrid sampling is used to tackle data imbalance issues which is a combination of SMOTER and random under-sampling (RUS). On performing experiments on 27 datasets, SHSE performed better than other software fault number prediction techniques. DTR (decision tree regressor) gives the best results when combined with SHSE. The fault percentile average (FPA) is used as an evaluation metric.

Goyal [13] proposed a novel sampling approach, Neighborhood-based under-sampling (N-US) to encounter the data imbalance problem in SFP. ANN, DT, KNN, SVM, and NB classifiers are used to construct the model. PROMISE dataset is used in the study. Accuracy, AUC, and ROC are used for measuring the performance of the model. When applying the N-US approach, the classifiers' accuracy improves. The Imbalanced Ratio (IR) is reduced by 19.73% by using the N-US approach.

Pandey et al. [14] performed SFP on NASA and the PROMISE repository. SMOTE is used to make the dataset equal. When compared to NB, LR, Multi-Layer Perceptron Neural Network (MLP), SVM, and conventional Principal Component Analysis-Extreme Learning Machine (PCA-ELM)-based fault prediction models. Extreme learning machine has higher Receiver Operating Characteristic curve (ROC) values when paired with K-PCA and SMOTE approaches. The proposed strategy yields more objective outcomes.

A defect prediction model via an attention-based recurrent neural network (DP-ARNN) is proposed by Fan et al. [15]. For data imbalance, over-sampling is used, and the suggested model combines Bidirectional Long-Short Term Memory (Bi-LSTM). The experiment is carried out on Apache Java projects. When compared to baseline approaches, the F1-measure increases by 14%, and Area Under the Curve (AUC) increases by 7%.

Yedida and Menzies [16] proposed fuzzy sampling which is a novel over-sampling technique to tackle the data imbalance problem. An SFP model is built using Deep Belief Network (DBN). The experiment is carried out on the PROMISE dataset. Recall, AUC, RUC, and false alarm rate (FAR) are used as performance metrics. The authors conclude that over-sampling is required before applying deep learning for SFP.

Pandey et al. [17] performed experiments on the NASA dataset to detect software faults. SMOTE technique is implemented on the dataset to overcome the data imbalance problem. SqueezeNet and Bottleneck DL models are applied to the balanced dataset. The F-measure on applying SqueezeNet and Bottleneck model is 0.93 ±

0.014 and 0.90 ± 0.013, respectively. However, the computational cost of both these methods is high in terms of training time.

Malhotra and Kamal [18] implemented five over-sampling methods: Safe-Level SMOTE, SMOTE, Selective Preprocessing of Imbalanced Data (SPIDER), SPIDER2, and Adaptive Synthetic Sampling Approach (ADASYN) on five ML classifiers: NB, AdaBoost (AB), J48, RF, and Bagging (BG), SPIDER2, and SPIDER3 are also put forward in this paper. When these over-sampling methods were applied with different ML techniques, the average AUC was 0.94 and the average precision value was 0.93 for the NASA dataset. ADASYN's over-sampling method gave the best results. The proposed SPIDER3 method also gives better results than SPIDER2 and SPIDER methods. ADASYSN performed better than MC learners.

Tantithamthavorn et al. [5] applied four class balancing techniques, namely, over-sampling (OS), under-sampling (US), SMOTE, and Random Over-Sampling Examples (ROSE) along with NB, AVNNet, xGBTree, C5.0, RF, LR, and Gradient Boosting Method (GBM) classification techniques. The experiments showed that AUC can be improved by optimizing the parameters of SMOTE.

For fault prediction, Nitin et al. [19] tested four ensemble methods which are random forest, bagging, random subspace, and boosting, along with SMOTE for data imbalance. The ensemble approaches use DT, LR, and KNN as baseline learners. Fifteen datasets from the Eclipse and PROMISE repositories are used in the experiment. The Wilcoxon test revealed that bagging performed statistically divergent from the other ensemble techniques. In terms of ROC-AUC score and recall, bagging performed best, while random forest performed worst.

Balaram and Vasundra [20] proposed a model based on Butterfly Optimization Algorithm (BOA) with Ensemble Random Forest (E-RF-ADASYN) along with ADASYN. PROMISE dataset is used in the study. BOA is implemented to overcome the problem of over-fitting and ADASYN (Adaptive Synthetic Sampling) is applied to overcome class imbalance problems. In comparison to KNN and DT classifiers, E-RF-ADASYN achieved better results in specificity, AUC, and sensitivity.

Table 1 gives a review of the literature survey.

5 Dataset Used

Inspired by the UCI machine learning repository, the PROMISE repository was developed to encourage researchers to study the field of software engineering. PROMISE stands for PRedictOr Models In Software Engineering. The PROMISE repository consists of multiple publicly available datasets.

In this paper, six PROMISE datasets are used. These are

- CM1
- KC1
- PC1
- PC2

Table 1 Summary of software fault prediction techniques discussed in Sect. 4

Author	Sampling techniques	ML algorithms	Dataset
Goyal, 2021	Neighborhood-based under-sampling (N-US)	ANN, DT, SVM, NB	NASA
Rathore et al., 2022	GAN, WGANGP, CTGAN	RF, LR, NB, KNN, DT	PROMISE, JIRA, ECLIPSE
Singh and Rathore, 2022	SMOTE	RF, SVR, Lr, LBr, NLBr	AEEEM, JIRA, PROMISE
Tong et al., 2022	RUS + SMOTER	DTR	AEEEM, MetricsRepo
Fan et al., 2019	Over-sampling (OS)	LSTM	Java projects in APACHE
Pandey et al., 2020	SMOTE	K-PCA-ELM, MLP, LR, SVM, NB	PROMISE, NASA
Yedida and Menzies, 2011	Fuzzy sampling	DBN	PROMISE
Pandey et al., 2020	SMOTE	SqueezeNet, BottleNeck	NASA
Malhotra et al., 2019	ASASYN, SPIDER, SPIDER3, SMOTE	J48, NB, AB, BG, RF	NASA
Tantithamthavorn et al., 2020	OS, US, SMOTE, ROSE	RF, LR, NB, NN, boosting	NASA, PROMISE
Nitin et al., 2020	SMOTE	Ensemble learning	PROMISE, ECLIPSE
Balaram and Vasundra, 2022	ADASYN	E-RF, E-RF-ADASYN	PROMISE

- PC3
- PC4

The value of a dependent variable is predicted using a set of independent variables in each of these datasets. The criterion for evaluation is a binary one.

Independent variables of the fault dataset are the no. of unique operators, no. of unique operands, no. of total operators, no. of total operands, no. of flow graphs, McCabe's line count of code [21], cyclomatic complexity, design complexity, Halstead volume [22], Halstead's count of blank lines, Halstead's count of lines of comments, no. of flow graph, no. of unique operators and operands, etc.

6 Result

In this paper, ROS, RUS, and SMOTE data sampling techniques are applied to CM1, KC1, PC1, PC2, PC3, and PC4 datasets to overcome the data imbalance problem. After balancing the dataset, a decision tree classifier is used to build a model. ROC-AUC score and F1-score for the datasets are calculated.

(1) ROC-AUC score—ROC curves can be summarised by the area under the curve (AUC), which measures a classifier's ability to distinguish between classes. There are fewer false positives and more correct predictions when the AUC is higher.

(2) F1-score—Precision and Recall are weighted together to produce the F1-score. As a result, this score takes into account both incorrect positive and incorrect negative results. Even though F1 is more difficult to grasp intuitively than accuracy, it is often more useful in situations where the distribution of classes is uneven. In this paper, the PROMISE dataset is used which is an imbalanced dataset. Therefore, F-measure is an efficient parameter for uneven class distribution.

To calculate the F1-score, precision, and recall are used which are discussed below:

- Precision—Correctly predicted positive observations as a percentage of all positive predictions is precision. It denotes how often the classifier is correct when it predicts yes.
- Recall—It is the proportion of correctly predicted yes observations to all yes observations in the actual class. It denotes when the answer is actually yes and how frequently the classifier predicts yes.

Table 2 provides a detailed summary of the results obtained in this experiment.

A graphical representation of the results is given. Figure 3 compares the ROC-AUC score and F1-score using a random over-sampling technique.

Figure 4 compares the ROC-AUC score and F1-score using a random under-sampling technique.

Table 2 Summary of results of over-sampling, under-sampling, and SMOTE

	Over-sampling		Under-sampling		SMOTE	
Dataset	ROC-AUC score	F1-score	ROC-AUC score	F1-score	ROC-AUC score	F1-score
CM1	0.95	0.95	0.63	0.59	0.85	0.85
KC1	0.83	0.84	0.56	0.55	0.72	0.73
PC1	0.96	0.97	0.58	0.51	0.91	0.9
PC2	0.96	0.97	0.6	0.66	0.96	0.96
PC3	0.95	0.96	0.68	0.65	0.82	0.83
PC4	0.94	0.95	0.79	0.77	0.88	0.89

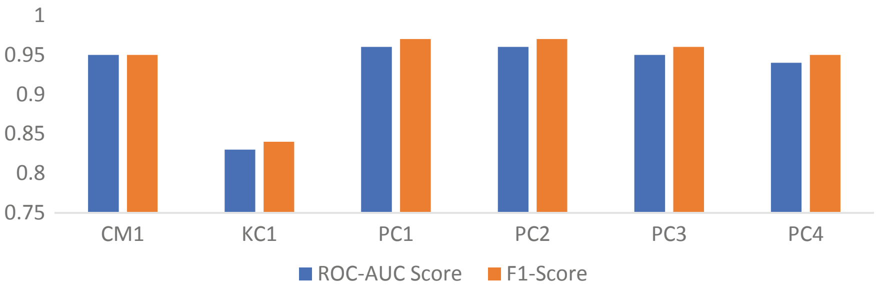


Fig. 3 Comparison of ROC-AUC score and F1-score using random over-sampling

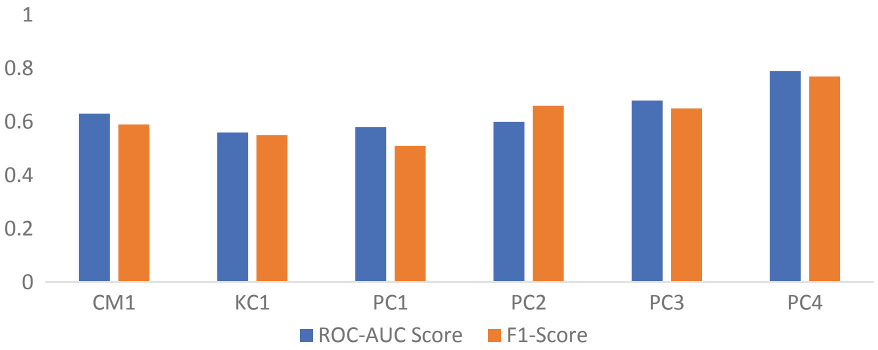


Fig. 4 Comparison of ROC-AUC score and F1-score using a random under-sampling

Figure 5 compares the ROC-AUC score and F1-score using SMOTE data sampling technique.

On comparing ROC-AUC score and F1-score for all three data sampling techniques, random over-sampling outperforms random under-sampling and SMOTE technique.

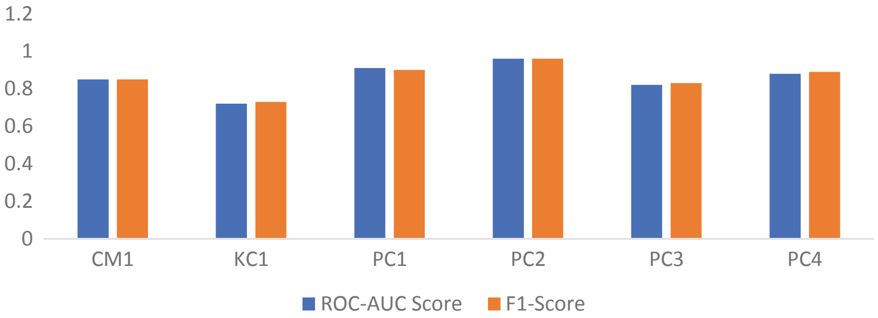


Fig. 5 Comparison of ROC-AUC score and F1-score using SMOTE data sampling

7 Conclusion

Fault prediction is an important topic in every domain. In software engineering, SFP plays a very crucial role as it helps in detecting faulty modules. If the faulty modules are rectified in the initial stages of software development, time and resources can be managed easily in the development of the software.

But the fault prediction dataset suffers from the problem of class imbalance. In this, one of the class data is present in the majority, while the other class data are present in the minority. This hinders the prediction efficiency of the models.

In this paper, we have applied following sampling techniques to overcome class imbalance problem—random over-sampling, random under-sampling, and SMOTE. Along with these techniques a decision tree classifier is built and trained on CM1, KC1, PC1, PC2, PC3, and PC4 datasets.

As per the experiment, random over-sampling performs better than random under-sampling and SMOTE.

References

1. Wan Z, Xia X, Hassan AE, Lo D, Yin J, Yang X (2018) Perceptions, expectations, and challenges in defect prediction. IEEE Trans Software Eng 46(11):1241–1266
2. Sherer SA (1995) Software fault prediction. J Syst Softw 29(2):97–105
3. Lessmann S, Baesens B, Mues C, Pietsch S (2008) Benchmarking classification models for software defect prediction: a proposed framework and novel findings. IEEE Trans Software Eng 34(4):485–496
4. Shepperd M, Song Q, Sun Z, Mair C (2013) Data quality: some comments on the NASA software defect datasets. IEEE Trans Software Eng 39(9):1208–1215
5. Tantithamthavorn C, Hassan AE, Matsumoto K (2018) The impact of class rebalancing techniques on the performance and interpretation of defect prediction models. IEEE Trans Software Eng 46(11):1200–1219
6. Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study. Intell Data Anal 6(5):429–449
7. Abd Elrahman SM, Abraham A (2013) A review of class imbalance problem. J Netw Innov Comput 1(2013):332–340
8. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority oversampling technique. J Artif Intell Res 16:321–357
9. Song Q, Guo Y, Shepperd M (2018) A comprehensive investigation of the role of imbalanced learning for software defect prediction. IEEE Trans Software Eng 45(12):1253–1269
10. Rathore SS, Chouhan SS, Jain DK, Vachhani AG (2022) Generative oversampling methods for handling imbalanced data in software fault prediction. IEEE Trans Reliab
11. Singh R, Rathore SS (2022) Linear and non-linear Bayesian regression methods for software fault prediction. Int J Syst Assur Eng Manag 1–21
12. Tong H, Lu W, Xing W, Liu B, Wang S (2022) SHSE: a subspace hybrid sampling ensemble method for software defect number prediction. Inf Softw Technol 142:106747
13. Goyal S (2022) Handling class-imbalance with KNN (neighbourhood) undersampling for software defect prediction. Artif Intell Rev 55(3):2023–2064
14. Pandey SK, Rathee D, Tripathi AK (2020) Software defect prediction using K-PCA and various kernel-based extreme learning machine: an empirical study. IET Software 14(7):768–782

15. Fan G, Diao X, Yu H, Yang K, Chen L (2019) Software defect prediction via attention-based recurrent neural network. Sci Program 2019
16. Yedida R, Menzies T (2021) On the value of oversampling for deep learning in software defect prediction. IEEE Trans Software Eng
17. Haldar A, Pandey SK, Tripathi AK Is deep learning good enough for software defect prediction? SSRN 4089137
18. Malhotra R, Kamal S (2019) An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data. Neurocomputing 343:120–140
19. Nitin, Kumar K, Rathore SS (2021) Analyzing ensemble methods for software fault prediction. In: Hura GS, Singh AK, Siong Hoe L (eds) Advances in communication and computational technology. ICACCT 2019. Lecture notes in electrical engineering, vol 668. Springer, Singapore. https://doi.org/10.1007/978-981-15-5341-7_95
20. Balaram A, Vasundra S (2022) Prediction of software fault-prone classes using ensemble random forest with adaptive synthetic sampling algorithm. Autom Softw Eng 29(1):1–21
21. McCabe TJ (1976) A complexity measure. IEEE Trans Software Eng 4:308–320
22. Halstead MH (1977) Elements of software science (operating and programming systems series). Elsevier Science Inc., USA

Mitigation of Trust-Related Issues in Cryptocurrency Payments Using Machine Learning: A Review

Harshal Shridhar Kallurkar and B. R. Chandavarkar

Abstract Cryptocurrency is a type of fiat currency in digital form, unlike the physical money that is commonly used for daily purposes. A blockchain is a base on which a cryptocurrency operates, i.e., it is a growing list of records of transactions happening in a particular cryptocurrency. Trust in a cryptocurrency comes into the picture when two stakeholders, virtually unknown to each other, are confident or not about each other's reliability in the context of whether each one is getting the service they intended to get. Trust in cryptocurrency can exist between any two stakeholders, such as users, merchants, government agencies, and blockchain technology, who are a part of cryptocurrency transactions. Furthermore, direct or indirect involvement of different stakeholders in cryptocurrency transactions results in issues such as lack of transparency, ease of use, regulations of the government, privacy, security of users, etc. Traditional approaches to anomaly detection in blockchain primarily use machine learning methods because they can infer patterns from historical data to give decent accuracy on test data. This survey presents trust in a cryptocurrency payment and its issues. Furthermore, it also shows the mitigation approaches which use machine learning techniques to address these issues.

Keywords Trust · Cryptocurrency · Blockchain · Transaction

1 Introduction

In a typical physical transaction, there is a guarantee of some regulating authority that closely monitors the cash flow, currency circulation, etc. However, in cryptocurrencies,[1] there is no existence of such kind of authority. The technology on which they operate is known as the blockchain. The transactions in the blockchain are main-

[1] https://www.coinbase.com/learn/crypto-basics/what-is-cryptocurrency.

H. Shridhar Kallurkar (✉) · B. R. Chandavarkar
Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore 575025, India
e-mail: harshalshridharkallurkar.213is001@nitk.edu.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_6

tained as a list of growing records in which the current block is linked to the previous one using a cryptographic hash. The transactions using cryptocurrencies are maintained in a decentralized distributed publicly available ledger. This ledger is publicly available for any cryptocurrency. According to the Merriam-Webster dictionary, trust is a guaranteed assurance of character and reliability of someone.[2] Users' trust in cryptocurrency transactions is affected by lack of transparency, no central authority to look over, security of assets, etc.

When users transact via cryptocurrencies, both parties need to know how they operate. Having details of traders and miners, the security model of the cryptocurrency increases users' trust in cryptocurrencies. One striking fact about these cryptocurrencies is that only those with adequate working knowledge of these technologies are willing to take risks. So, to further increase users' trust in them, cryptocurrency wallets/exchanges should extensively provide tutorials and classes in this context.

This paper focuses on the general working of a cryptocurrency transaction. It also describes how users' trust is established by looking at various solutions which use machine learning and deep learning methods. These methods provide an explicit measure for a user, which can show the trustworthiness of other users beforehand. It also looks into trust in cryptocurrency payments concerning four main stakeholders, focusing more on the users' trust part [39].

Anomaly detection in blockchain-based cryptocurrencies has gained much attention in the past few years, given the unstable nature of their market cap value, their generation rate in the blockchain, the increase in illegal payments, illicit users, etc. This issue has been looked into through various works which primarily use machine learning methods, like [6, 24, 33, 36]. Considering these approaches and the issues of trust in cryptocurrency payment, our contribution to this paper is answers to the questions below:

- What constitutes trust in cryptocurrency?
- What kind of trust exists between different stakeholders in a cryptocurrency transaction?
- How machine learning approaches have been used to address the trust issues of various stakeholders in a cryptocurrency payment?

The rest of this paper is organized as follows: Sect. 2 contains a brief introduction to trust in a cryptocurrency transaction, Sect. 3 describes, in brief, the working of a cryptocurrency transaction, Sect. 4 lists trust issues in cryptocurrency payments, Sect. 5 lists existing solutions to user trust establishment, which use machine learning, Sect. 6 mentions recommendations for future work and conclusion along with a comparison of existing solutions, and references are cited at the end.

[2] https://www.merriam-webster.com/dictionary/trust.

2 Transactions in Cryptocurrency

The first open-source cryptocurrency available is Bitcoin, introduced in the year 2009. Its concept was put through in the form of a white paper by an anonymous Satoshi Nakamoto [29]. And since the introduction of Bitcoin, many other cryptocurrencies (most commonly used are Ethereum[3] and Binance coin[4]) have been developed on its basis.

In a blockchain, a node is a computer that runs the cryptocurrency client on its machine. Each block holds a specific number of transaction records and is made secure by including in it, the hash of its immediate predecessor block. Every user of the particular cryptocurrency concerning that blockchain has its own copy of the distributed ledger. If a new transaction is being added to a specific block, information regarding it is added to every user's ledger. The remaining part of this section presents the general overview of the stages involved in a cryptocurrency transaction.

Figure 1 presents the flow of the cryptocurrency transaction. Consider two users, Alice and Bob, intending to do a transaction of 1 unit of bitcoin(BTC). The three main stages in the transaction and their working is described below [3]:

- **Signing stage**: A message is created by the wallet or an exchange [23]. The message's contents are Alice's address, Bob's address, and the amount to be sent. Then a digital signature is created for this message. Note that each transaction has a different digital signature [45]. The wallet then combines the digital signature and the message into one file.
- **Broadcasting stage**: The file created in stage I is broadcasted to all the nodes in the blockchain. They verify the digital signature of Alice using her public key and add it to their *mempool* [2, 16].
- **Confirmation stage**: Miners [30] add the transaction to the block by solving a complex mathematical problem, also known as *Proof-of-work* (PoW) [15]. Every node in the blockchain network competes against each other to gain an advantage of adding a block to the blockchain and gaining rewards. This computationally expensive operation ensures randomness in block addition, fraudulent users do not get control over the blockchain network, and fairness in rewarding miners. A transaction is supposed to wait for a certain amount of time before it gets added to a particular block in the blockchain, which is the time when miners are doing PoW. For example, it takes 10 min, on average, in the case of Bitcoin cryptocurrency.

3 Scenario of Trust in Cryptocurrency

With respect to Fig. 1, there are four major stakeholders in a blockchain cryptocurrency ecosystem [14]:

[3] https://ethereum.org/en/.

[4] https://www.binance.com/en.

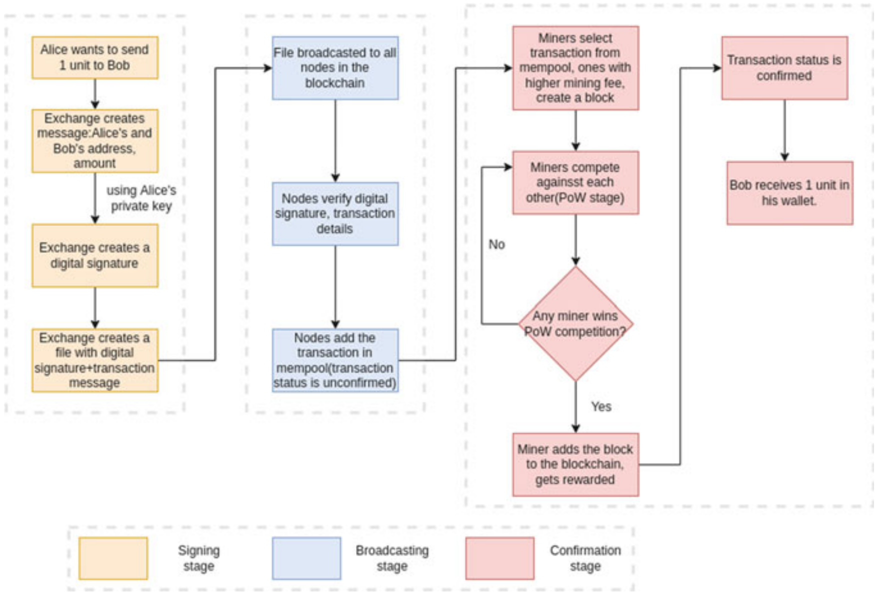

Alice wants to send 1 unit to Bob
Exchange creates message:Alice's and Bob's address, amount
using Alice's private key
Exchange creates a digital signature
Exchange creates a file with digital signature+transaction message
File broadcasted to all nodes in the blockchain
Nodes verify digital signature, transaction details
Nodes add the transaction in mempool(transaction status is unconfirmed)
Miners select transaction from mempool, ones with higher mining fee, create a block
Miners compete againsst each other(PoW stage)
No
Any miner wins PoW competition?
Yes
Miner adds the block to the blockchain, gets rewarded
Transaction status is confirmed
Bob receives 1 unit in his wallet.
Signing stage
Broadcasting stage
Confirmation stage

Fig. 1 Flow of a cryptocurrency transaction

i. Users: Using cryptocurrencies for trading and exchange of goods.
ii. Miners: Responsible for Proof-of-work solving and gaining rewards from the same.
iii. Cryptocurrency exchanges/wallets: Secure maintenance of user's cryptocurrencies and private keys.
iv. Government and financial institutions: Regulating authority for banking services (e.g., Reserve Bank of India (RBI),[5] Ministry of Finance[6])

Figure 2 presents different kinds of trust that exist between stakeholders, namely the primary user, miners, cryptocurrency exchanges, and government agencies. The three types of trust described here are social, institutional, and technological trust [39].

4 Issues of Trust in Cryptocurrency

As cited in this paper by Rehman et al., some of the common trust issues with the payments using cryptocurrencies are *lack of transparency, ease of use, government*

[5] https://www.rbi.org.in/.

[6] https://www.finmin.nic.in/.

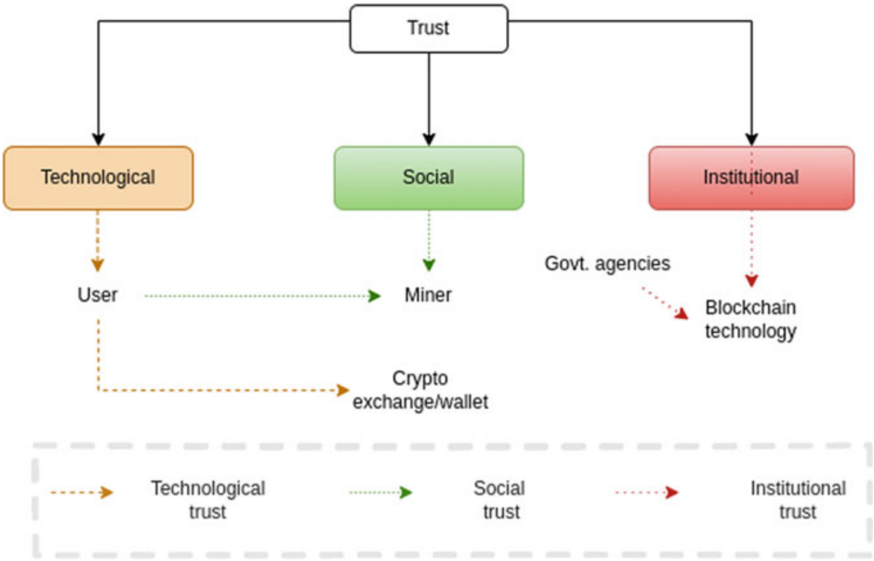


Fig. 2 Stakeholders and types of trust [39]

regulations, privacy and security of users [38]. A brief overview of the above issues is summarized below:

i. **Lack of transparency**: The frequency over which the price of any cryptocurrency is changing, rules of trading and transaction regulations, and the time it takes to process any given transaction in the blockchain are some of the main attributes which are available in the blockchain ledger, but very few people understand the significance of it.

ii. **Government regulations and policies**: Though the decentralized nature is an advantage in the context of making it more secure and having no centralized authority, it creates an impasse as to the number of people having the actual knowledge of the working of blockchain technology [5]. In line with this argument, having prior information about the person to whom the money is being sent not only increases users' trustworthiness but also that of the other stakeholders in that cryptocurrency transaction. The absence of cryptocurrency regulation laws and increasing disinclination of the financial institutions in adopting cryptocurrencies are some of the issues which are making the cryptocurrency, as an ecosystem, less trustworthy.

iii. **Ease of use**: The design and ease of use play a crucial role in increasing trust among the new users of cryptocurrency. An example of this scenario is the complex management of the exchange's public and private keys, the inability to predict the confirmed transaction time, and the dynamic variation in transaction fees to be paid upfront before initiating a transaction.

iv. **Privacy and security of users**: In this ecosystem, it is comparatively easy to track a particular user by analyzing their transaction history, public keys, and IP address [25]. So, the wallets and the exchanges providing cryptocurrency services must build attack-prone systems. Some of the attacks supporting this fact are mentioned in [11, 18, 19, 22].

Though the above-mentioned issues are common to all the stakeholders of the cryptocurrency ecosystem, the next section presents solutions to mitigate trust issues with respect to the user's side only.

5 Existing Solutions to Mitigate Trust Issues in Cryptocurrency

A well-known approach to anomaly detection problems is the use of machine learning-based techniques. The advantage of these techniques is that they tend to draw out patterns from already-seen data and hence derive inferences from them. This results in a reduction of the false positive rate of the model with respect to the false negative rate [43].

5.1 Machine Learning Methods

A transaction can only be considered successful between two parties when both of them get what they want before the initiation of the transaction. This enhances the interpersonal trust between the users. Monica Catherine et al. [21] demonstrate this by predicting the clusters [17] of malicious nodes in a given blockchain system, using dynamic time warping, which is a technique to find similarities between two sequences that are time-dependent [1]. The authors proposed a major change to the existing k-means clustering algorithm. Instead of using the **mean** value of each node to determine the centroid, select the sequence which has the minimum distance from its $(\frac{n}{k})$th nearest node among the complete neighbors set.

S. Sayadi et al. proposed a two-stage machine learning model to detect anomalies in blockchain transactions. The first stage used One-class SVM [8] to detect outliers, then stage two used the K-means algorithm to cluster similar kinds of attacks [40]. The bitcoin transaction data was obtained from this source.[7] Since the attack data concerning Bitcoin is very scarce, they have created it manually to validate it later. The three types of attacks present in the dataset are DDoS attacks, 51% vulnerability attacks, and the double spending attack, in which it is possible to spend a digital token more than once because the token consists of a digital record that can easily be duplicated [10]. However, at the same time, Signorini et al. have pointed out that

[7] https://www.blockchain.com/charts.

this solution does not consider the underlying hidden data, which helps in the better identification of anomalies [41]. It also does not consider the property by which default blockchain data, along with additional information, could help other nodes better identify anomalies.

Considering the recent regulations that the Central Government has imposed in the context of cryptocurrency assets, it is evident that a user ought to have some hesitancy about using cryptocurrencies. Hyochang Back et al. present this concept of strengthening institutional trust in Bitcoin by the use of the Expectation-maximization algorithm, which is a statistical method used to find maximum likelihood values for parameters in the model which is dependent on hidden variables [27] to cluster the dataset of Ethereum wallets[4]. Then, binary classification was done using the Random Forest [12] algorithm, with wallets having anomalous transactions labeled as "1", otherwise "0". This model could help stakeholders in the finance sector look into illegal activities in cryptocurrency payments. Wen et al. mention an issue with this approach that, even though machine learning models have given good results in terms of detection, their credibility is still not verified in case of malicious attacks [44].

B. Podgorelec et al. have proposed using artificial intelligence on top of blockchain to implement automated signing of transactions, and anomaly detection for each user, thus enhancing the user's trust in blockchain transactions [35]. The results from anomaly detection are stored locally and not in a centralized server/location. M. Ostapowicz et al. presented a model which uses supervised learning applied on the Ethereum blockchain using Random Forest, SVM [42], and XGBoost [9] classifiers to classify the cryptocurrency accounts on a particular exchange into fraud/anti-fraud category [32]. This model increases exchanges' and users' trust in the blockchain system.

The gas acts as a surcharge which quantitatively represents the amount needed to successfully conduct a transaction on Ethereum Virtual Machine (EVM). Gas limit is a measure of the maximum amount of gas required for executing a transaction (which is typically set to 21,000 for transactions between externally owned accounts and more significant for transactions between a user and smart contract). Oliveira et al. proposed using Machine learning models, which were trained using an unbiased dataset of failed/non-failed transactions, to predict confirmation of a transaction on the Ethereum blockchain [31]. Their proposed framework starts with the collection of transaction data, dividing it into blocks containing several transactions. Since the data set was highly unbalanced, in order to have an unbiased computation, they have used techniques like undersampling (examples from the class which are very high compared to class with a lower count are selected on a random basis, for deletion from the training set.) [26] and oversampling(selecting examples from the class which is very low compared to class with higher count, to be added to the training set on a random basis). Using machine learning techniques for classification like decision trees [37], random forest [20], logistic regression [13], and the support vector machine [34]. Random Forest algorithm, along with undersampling, was found to be better than other techniques in terms of area under the Receiver Operating Characteristic (ROC) curve. ROC is a two-dimensional graph with *true-positive*

Table 1 Summary of approaches

Approach	Advantages	Limitations	Trust issues addressed
Malicious cluster prediction in a blockchain [21]	Use of Dynamic Time Warping instead of Euclidean distance as a similarity measure because of variety in block length	Temporal activity of blockchain not taken into consideration	Enhances technological trust of the user in blockchain by early detection of malicious activity
Machine learning for anomaly detection [40]	Efficient classification into attacks like DDoS, 51% vulnerability, etc.	Higher order association between accounts and Bitcoin addresses not included in the dataset	Addresses users' trust in blockchain by early prediction of different types of anomalies in Bitcoin transactions.
Anomalous transaction detection in Binance using Random forests [4]	Use of Expectation Maximization technique for cluster formation	Individual users' trust is not established	Addresses technological and institutional trust for government agencies by labeling wallets with suspicious transactions
Machine learning-based transaction signing and user-focussed anomaly detection [35]	Personalized anomaly detection for a sender w.r.t to a particular receiver address	Success rate of a transaction is not presented in the anomaly detection system	Focusses on improving users' trust in blockchain technology by developing a machine learning model for detecting fraudulent transactions in Ethereum
Fraudulent account detection on Ethereum [32]	efficient detection of fraud accounts on Ethereum along with feature importance presented	Usage of big datasets with class imbalance leads to precision–recall tradeoff	Addresses technological and institutional trust of users and govt agencies by early detection of fraudulent accounts on Ethereum blockchain
Classification of the transaction into confirmed/unconfirmed status in Ethereum [31]	Enhances user's confidence by giving a confirmation status using machine learning models	Time-series nature of data, along with data imbalance leading to low precision and high recall values	Enhances user's trust in blockchain technology by early prediction of transaction status in Ethereum

observations on the Y-axis and *false-positive* observations on the X-axis, which represents the performance of the model, on every cusp of the classification [7].

After the comparison of the models mentioned above, one significant conclusion the authors came to is that "**gas**"[8] is a very crucial attribute in deciding the success/failure of the transaction. Their analysis of transaction confirmation using a decision tree[28] shows that "**gas**" has the highest Gini impurity index value,[9] which

[8] https://ethereum.org/en/developers/docs/gas/.

[9] https://www.learndatasci.com/glossary/gini-impurity/.

makes it the deciding root of the tree and hence an essential attribute among other attributes of a cryptocurrency transaction. Extracted dataset of Ethereum transactions between April 26 and July 15, 2019, which was used as a base dataset in this paper, is available here.[10] Having such kind of information beforehand has a positive effect on the users and the institutional trust in the blockchain system. However, at the same time, gas may not be the only attribute that affects the user's trust in blockchain technology. It is also equally dependent on other factors, such as network congestion (i.e., the total number of pending transactions) at a given time in the Ethereum network, the approximate time to get final confirmation of the transaction, etc. Table 1 provides a summary of the approaches discussed in this section, along with their merits and shortcomings.

6 Conclusion and Future Work

This review focused specifically on the trust side of payments in cryptocurrencies. While some papers look into the philosophical side of trust, others have worked extensively on the technological side of it, i.e., some work has been done in anomaly detection using machine learning techniques. From this review, it can be concluded that while dealing with cryptocurrencies, an ordinary user who is not much familiar with the working of blockchain technology is less likely to do transactions using cryptocurrencies as compared to a person who has got themselves acquainted with using cryptocurrencies. Factors like interpersonal trust, users' trust in blockchain technology, and fear of restrictions on cryptocurrency trading by the regulating authority play an important role in deciding the popularity of cryptocurrencies among users. Most of the literature contains work about how an anomalous transaction/node can be detected in the blockchain using machine learning methods. Future work in trust can be done by using machine learning or deep learning method to predict certain attributes related to a cryptocurrency transaction dynamically, primarily because these methods give better results when structured data (like cryptocurrency transaction data) is used for training the model. A subtle point with respect to using machine learning and deep learning methods on cryptocurrency transaction data is that it is essential to consider the data's time series nature. Also, the built models need to be trained on real-time transaction data such that changes in the network congestion would also get incorporated accordingly.

References

1. (2007) Dynamic time warping. Springer, Berlin, Heidelberg, pp 69–84. https://doi.org/10.1007/978-3-540-74048-3_4
2. (2021) Mempool in blockchain. Accessed 02 Dec 2021
3. (2021) How deos a tranaction get into blockchain?. Accessed 17 Nov 2021

[10] http://netlab.ice.ufjf.br/ethereum.

4. Baek H, Oh J, Kim C, Lee K (2019) A model for detecting cryptocurrency transactions with discernible purpose. In: ICUFN 2019-11th international conference on ubiquitous and future networks. IEEE Computer Society, ICUFN, pp 713–717. https://doi.org/10.1109/ICUFN.2019.8806126
5. Birch DG, Parulava S (2018) Chapter 17-ambient accountability: Shared ledger technology and radical transparency for next generation digital financial services. In: Lee Kuo Chuen D, Deng R (eds) Handbook of blockchain, digital finance, and inclusion, vol 1. Academic Press, pp 375–387. https://doi.org/10.1016/B978-0-12-810441-5.00017-8. https://www.sciencedirect.com/science/article/pii/B9780128104415000178
6. Bogner A (2017) Seeing is understanding: anomaly detection in blockchains with visualized features. Association for Computing Machinery, New York, NY, USA, UbiComp '17, pp 5–8. https://doi.org/10.1145/3123024.3123157
7. Bradley AP (1997) The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognit 30(7):1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2, https://www.sciencedirect.com/science/article/pii/S0031320396001422
8. Brownlee J (2020) One-class classification algorithms for imbalanced datasets. Accessed 06 Feb 2022
9. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. Association for Computing Machinery, New York, NY, USA, KDD '16, pp 785–794. https://doi.org/10.1145/2939672.2939785
10. Chohan UW (2021) The double spending problem and cryptocurrencies. SSRN 3090174
11. Conti M, Sandeep Kumar E, Lal C, Ruj S (2018) A survey on security and privacy issues of bitcoin. IEEE Commun Surv Tutor 20(4):3416–3452. https://doi.org/10.1109/COMST.2018.2842460
12. Cutler A, Cutler DR, Stevens JR (2012) Random forests. In: Ensemble machine learning. Springer, pp 157–175
13. Dreiseitl S, Ohno-Machado L (2002) Logistic regression and artificial neural network classification models: a methodology review. J Biomed Inform 35(5):352–359. https://doi.org/10.1016/S1532-0464(03)00034-0, https://www.sciencedirect.com/science/article/pii/S1532046403000340
14. Filippi PD, Mannan M, Reijers W (2020) Blockchain as a confidence machine the problem of trust and challenges of governance. Technol Soc 62(101):284. https://doi.org/10.1016/j.techsoc.2020.101284, https://www.sciencedirect.com/science/article/pii/S0160791X20303067
15. Frankenfield J (2021) Proof of work (PoW) definition. Accessed 05 Dec 2021
16. Garfinkel SL (1996) Public key cryptography. Computer 29(6):101–104
17. Hartigan JA, Wong MA (1979) Algorithm as 136: a k-means clustering algorithm. J R Stat Soc Ser C (Appl Stat) 28(1):100–108. http://www.jstor.org/stable/2346830
18. Hasanova H, Baek Uj, Shin M, Cho K, Kim MS (2019) A survey on blockchain cybersecurity vulnerabilities and possible countermeasures. Int J Netw Manag 29(2). https://doi.org/10.1002/nem.2060
19. Konoth RK, van Wegberg R, Moonsamy V, Bos H (2019) Malicious cryptocurrency miners: status and outlook. https://doi.org/10.48550/ARXIV.1901.10794, https://arxiv.org/abs/1901.10794
20. Kulkarni VY, Sinha PK (2012) Pruning of random forest classifiers: a survey and future directions. In: 2012 international conference on data science & engineering (ICDSE), IEEE, pp 64–68
21. Kumari R, Catherine M (2018) Anomaly detection in blockchain using clustering protocol. Int J Pure Appl Math 118(20):391–396
22. Li X, Jiang P, Chen T, Luo X, Wen Q (2020) A survey on the security of blockchain systems. Futur Gener Comput Syst 107:841–853. https://doi.org/10.1016/j.future.2017.08.020, https://www.sciencedirect.com/science/article/pii/S0167739X17318332
23. Little K (2021) What are crypto exchanges? Accessed 01 Dec 2021

24. Martin K, Rahouti M, Ayyash M, Alsmadi I (2022) Anomaly detection in blockchain using network representation and machine learning. Secur Priv 5(2):e192
25. Meiklejohn S, Pomarole M, Jordan G, Levchenko K, McCoy D, Voelker GM, Savage S (2016) A fistful of bitcoins: characterizing payments among men with no names. Commun ACM 59(4):86–93. https://doi.org/10.1145/2896384
26. Mohammed R, Rawashdeh J, Abdullah M (2020) Machine learning with oversampling and undersampling techniques: overview study and experimental results. In: 2020 11th international conference on information and communication systems (ICICS), pp 243–248. https://doi.org/10.1109/ICICS49469.2020.239556
27. Moon TK (1996) The expectation-maximization algorithm. IEEE Signal Process Mag 13(6):47–60
28. Myles AJ, Feudale RN, Liu Y, Woody NA, Brown SD (2004) An introduction to decision tree modeling. J Chemom: J Chemom Soc 18(6):275–285
29. Nakamoto S (2021) Bitcoin: a peer-to-peer electronic cash system. Accessed 16 Nov 2021
30. O'Dwyer KJ, Malone D (2014) Bitcoin mining and its energy footprint
31. Oliveira VC, Almeida Valadares J, Sousa A, JE, Borges Vieira A, Bernardino HS, Moraes Villela S, Dias Goncalves G (2021) Analyzing transaction confirmation in ethereum using machine learning techniques. SIGMETRICS Perform Eval Rev 48(4):12–15. https://doi.org/10.1145/3466826.3466832
32. Ostapowicz M, Żbikowski K (2019) Detecting fraudulent accounts on blockchain: a supervised approach. 1908.07886
33. Pham T, Lee S (2016) Anomaly detection in bitcoin network using unsupervised learning methods. arXiv:1611.03941
34. Pisner DA, Schnyer DM (2020) Chapter 6-support vector machine. In: Mechelli A, Vieira S (eds) Machine learning. Academic Press, pp 101–121. https://doi.org/10.1016/B978-0-12-815739-8.00006-7, https://www.sciencedirect.com/science/article/pii/B9780128157398000067
35. Podgorelec B, Turkanović M, Karakatič S (2020) A machine learning-based method for automated blockchain transaction signing including personalized anomaly detection. Sensors 20(1). https://doi.org/10.3390/s20010147, https://www.mdpi.com/1424-8220/20/1/147
36. Poursafaei F, Hamad GB, Zilic Z (2020) Detecting malicious ethereum entities via application of machine learning classification. In: 2020 2nd conference on blockchain research and applications for innovative networks and services (BRAINS), pp 120–127. https://doi.org/10.1109/BRAINS49436.2020.9223304
37. Priyanka Kumar D (2020) Decision tree classifier: a detailed survey. Int J Inf Decis Sci 12(3):246–269
38. Rehman MHu, Salah K, Damiani E, Svetinovic D (2020) Trust in blockchain cryptocurrency ecosystem. IEEE Trans Eng Manag 67(4):1196–1212. https://doi.org/10.1109/TEM.2019.2948861
39. Sas C, Khairuddin IE (2015) Exploring trust in bitcoin technology: a framework for HCI research. In: Proceedings of the annual meeting of the australian special interest group for computer human interaction. Association for Computing Machinery, New York, NY, USA, OzCHI '15, pp 338–342. https://doi.org/10.1145/2838739.2838821
40. Sayadi S, Rejeb S, Choukair Z (2019) Anomaly detection model over blockchain electronic transactions, pp 895–900. https://doi.org/10.1109/IWCMC.2019.8766765
41. Signorini M, Pontecorvi M, Kanoun W, Di Pietro R (2020) Bad: a blockchain anomaly detection solution. IEEE Access 8:173,481–173,490. https://doi.org/10.1109/ACCESS.2020.3025622
42. Suthaharan S (2016) Support vector machine. Springer US, Boston, MA, pp 207–235. https://doi.org/10.1007/978-1-4899-7641-3_9
43. Vassallo D, Vella V, Ellul J (2021) Application of gradient boosting algorithms for anti-money laundering in cryptocurrencies. SN Comput Sci 2(3):1–15
44. Wen H, Fang J, Wu J, Zheng Z (2021) Transaction-based hidden strategies against general phishing detection framework on ethereum. In: 2021 IEEE international symposium on circuits and systems (ISCAS), pp 1–5. https://doi.org/10.1109/ISCAS51556.2021.9401091
45. Zuidoorn M (2020) The magic of digital signatures on Ethereum. Accessed 02 Dec 2021

Secured Workflow Scheduling Techniques in Cloud: A Survey

Sarra Hammouti, Belabbas Yagoubi, and Sid Ahmed Makhlouf

Abstract Scheduling a scientific workflow in a cloud computing environment is a critical issue that is widely discussed in the literature, and since security is one of the most important user concerns, researchers were constantly looking for the most efficient techniques to solve the problem while taking into account the security and privacy of data and tasks. Thus we introduce in this paper, a literature review, taxonomy and comprehensive analysis of the research works that investigate the mentioned problem. Additionally, we highlight and discuss limitations, gaps and problem aspects that are not sufficiently investigated in the literature, as well as future research challenges.

Keywords Cloud computing · Scientific workflow · Scheduling · Optimization · Security · Privacy

1 Introduction

In a cloud computing environment, scheduling a scientific workflow is a crucial problem that is extensively covered in the literature. The issue attracted a lot of attention due to the powerful features provided by cloud environments, which encourage businesses to carry out their scientific workflows in such environments. Among these characteristics, we distinguish the ability of cloud service providers to offer a pool of resources such as computing power, storage, and bandwidth in the form of on-demand services that users can rent via the Internet. Additionally, the cloud computing platform provides rapid elasticity, extensive network access, and measured services [32].

The problem was designated as multi-objective and multi-constrained due to the varying user needs, e.g., some users need to optimize the workflow execution time, while others are concerned with cost optimization or other objectives such as resource

S. Hammouti · B. Yagoubi · S. A. Makhlouf (✉)
L.I.O. Laboratory, Department of Computer Science, University of Oran1-Ahmed Ben Bella, P.O. Box 1524, El M'naouer, Oran, Algeria
e-mail: sidahmed.makhlouf@gmail.com

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_7

utilization and reliability of the system. In addition, a user can express one or more constraints like deadline, budget, security, and others.

Since security is one of the most important user concerns, researchers were constantly looking for the most efficient techniques to prevent security-related issues, ensure privacy and data confidentiality when scheduling. Thus, in the present paper, we introduce a literature review, taxonomy, and comparative analysis of the different research works that investigate the mentioned problem.

The remainder of this paper is structured as follows. Related work is presented in Sect. 2. In Sect. 3, we provide details about our systematic review process. The main aspects surrounding the workflow scheduling problem are presented in Sect. 4. Additionally, the proposed taxonomy is illustrated in Sect. 5. Then, we summarize the existing secure cloud workflow scheduling approaches in Sect. 6, and we present a comprehensive analysis in Sect. 7 with a highlight of the open issues and challenges in Sect. 8. Finally, Sect. 9 brings the paper to a close.

2 Related Work

Scheduling a scientific workflow in a cloud computing environment is a critical issue that is widely discussed in the literature. In fact, some researchers have tended to provide various heuristic, meta-heuristic, and hybrid approaches to solve the problem while considering different objectives and QoS requirements. However, other researchers have tackled the problem by reviewing, classifying, and analyzing existing solutions.

In this field, certain systematic studies define and assess a wide range of workflow scheduling techniques, classifying them in accordance with various criteria [2, 9, 22, 23, 36]. For example, in [17, 51] The authors concentrated on some of the key workflow scheduling techniques and divided them into static and dynamic scheduling strategies, while in [44] the categorization was done in terms of scheduling criteria, schedule generation, and task-resource mapping. Furthermore, in [31] the authors divided the existing scheduling schemes into task scheduling, workflow scheduling, and task and workflow scheduling schemes. After concentrating on the workflow scheduling schemes, the authors divided them into heuristic and meta-heuristic techniques, and then further divided them based on the type of scheduling algorithms. The existing schemes were similarly categorized into heuristic, meta-heuristic, and hybrid methods by [25]. However, the authors' focus in [21, 43] was solely on meta-heuristic-based methods for cloud task scheduling. While [4] limited their survey on the existing cost optimization scheduling approaches.

Since security is one of the most important user concerns, researchers were constantly looking for the most efficient techniques to solve the problem under security and privacy considerations. However, till now, no study in the literature summarizes and classifies all the existing secure scheduling schemes, which can make it easier for Cloud Service Provider (CSP) and users to choose the most adequate approach to their needs, it also allows researchers to know the gaps, the existing problems and thus provide more effective solutions. Except, [18] performed an investigation into secured workflow scheduling systems in the cloud setting. But, the work is limited in terms of the number of studied approaches, where they explored only 9 research works, as well as they didn't provide a critical literature review, and they didn't mention the limitations of each approach. Consequently, we introduce in the present paper a systematic literature review, taxonomy, and comprehensive analysis of almost all the research works that investigate the aforementioned problem over the last decade. Additionally, we highlight and discuss limitations, gaps, and problem aspects that are not sufficiently investigated in the literature, as well as future research challenges.

3 Systematic Review Process

We outline the procedure we used to conduct this review in this section. Where we first gathered different research papers that are published in international journals and conferences during the period (2010–2022), we based on various well-known digital libraries such as Google Scholar,[1] Springer LNCS,[2] ScienceDirect—Elsevier,[3] and IEEE eXplore.[4] We used different search keywords, e.g., secure workflow scheduling in the cloud, secure optimization, QoS parameters, etc. Second, we examined the collected papers and then deleted the ones that don't address our problem, or that are not published and indexed in well-known databases. Third, we studied and classified the selected papers, as well as we extracted the gaps of each reviewed paper as described in Sect. 6. Forth, we discussed and analyzed the reviewed approaches as illustrated in Sect. 7. Finally, we conducted the open issues and challenges surrounding the problem as mentioned in Sect. 8 (see Fig. 1).

4 Workflow Scheduling in Cloud

In this section, we describe the most important notions surrounding the cloud workflow scheduling problem.

Scientific Workflow: It is a group of computational tasks that reveal dependencies among them in order to address a scientific issue.

Cloud Environment: Cloud service providers can offer their services in public, private, community, or hybrid environments [32].

- **Public Cloud** is a computing environment that allows the public to use resources freely. A corporation, academic institution, government agency, or a combination

[1] (https://www.scholar.google.com).

[2] (https://www.springer.com/lncs).

[3] (https://www.sciencedirect.com).

[4] (https://ieeexplore.ieee.org).

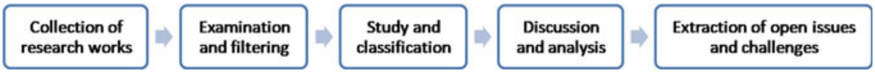

Collection of research works
Examination and filtering
Study and classification
Discussion and analysis
Extraction of open issues and challenges

Fig. 1 Systematic review process

of them may own, manage, and operate the cloud infrastructure on the premises of the cloud service provider [32].

- **Private Cloud** is a computing environment that allows one company to access resources privately. The business, a third party, or a combination of them may own, manage, and operate the cloud infrastructure, which is present on or off the cloud service provider's premises [32].
- **Community Cloud** is a computing environment that allows a community of organizations to access resources privately. The cloud infrastructure may be owned, managed, and maintained by one or more community groups, a third party, or a mix of them. It may exist on or off the cloud service provider's premises [32].
- **Hybrid Cloud** is a computing environment that combines two or more distinct cloud computing environments (public, private, or community) that are linked by standardized or proprietary technology that enables portability of data and applications [32].

Scheduling Strategy: Workflow scheduling strategies are categorized into three categories (static, dynamic, static and dynamic) depending on the workflow resources' information that was available during the scheduling process [36, 51].

- **Static Scheduling**: this type of schedule ends before the workflow execution begins. The advantage of this strategy is that it produces high-quality solutions as it has the possibility of comparing several feasible solutions to obtain the best fit. However, the disadvantage of this strategy is that it only makes estimates concerning missing information such as task execution time and communication time, which may be poor or not adaptable with the real system [36, 49, 51].
- **Dynamic Scheduling**: this type of scheduling is done during the execution of the workflow. The advantage of this strategy is that it adapts to real systems as it takes into account unexpected actions that may occur during execution, and relies on exact information about the workflow and resources. However, the disadvantage of the dynamic schedule is that it cannot produce high-quality solutions, because it has not the possibility of trying several feasible solutions and choosing the best one [36, 49, 51].
- **Static and Dynamic**: this category combines the two aforementioned scheduling strategies to achieve their advantages simultaneously, where a static mapping is performed before the start of execution based on approximate estimations. Then, during the execution, the assignment is adapted and redone if necessary [36, 51].

Figure 2 shows the scheduling strategies with their advantages and disadvantages.

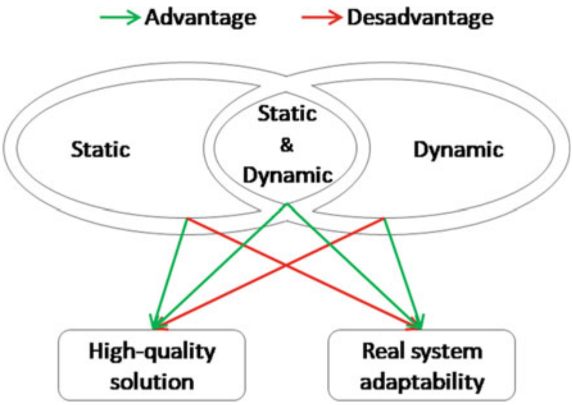


Fig. 2 Scheduling strategies

Workflow Scheduling Objectives and constraints: Usually, the workflow scheduling process has some objectives to achieve and QoS constraints to meet, where these objectives and constraints express the user's requirements. In the following, we explain these concepts.

4.1 Scheduling Objectives

In the literature, researchers have introduced various scheduling objectives, but the most studied are makespan and cost besides other objectives like resource utilization, energy consumption, load balancing, reliability of the system, throughput, etc.

- **Makespan** is the time passed between the beginning of the first workflow task's execution and the end of the last workflow task [10, 25].
- **Cost** is the price that the user has to pay to run his workflow [11].
- **Resource Utilization** refers to the optimal use of resources by minimizing time slots of resource inactivity [10].
- **Energy consumption** refers to the consumed energy by the cloud servers including the use of electricity and the carbon emissions [2].
- **Load Balancing** refers to the balanced distribution of workloads between different computing resources to avoid overloading one of the resources [5, 12].
- **Reliability** is the system's capacity to deliver service for a specific time frame without interruption or failure [39].
- **Throughput** is the number of tasks completed in a predetermined period [33].

4.2 User's Constraints

In the literature, researchers have introduced various scheduling QoS constraints, but the most expressed are deadline, budget, and security.

- **Deadline** is the maximum time a user can wait to complete the execution of his workflow.
- **Budget** is the maximum price a user can pay to run his workflow in a cloud environment
- **Security** defines the user's needs in terms of security services and scheduling policies to ensure high protection of his sensitive tasks and data.

Workflow Scheduling Algorithms: Generally, researchers have introduced heuristics, meta-heuristics, and hybrid algorithms to fix the cloud workflow scheduling issues.

– *Heuristics* generally used to quickly find satisfactory solutions.
– *Meta-Heuristics* generally used to find satisfactory solutions for large-scale problems [15].
– *Hybrid algorithms* combine two or more heuristic or meta-heuristic algorithms to gain more advantages simultaneously.

5 Taxonomy of Secured Workflow Scheduling Approaches

In this paper, we have reviewed about 32 scientific papers dealing with secure workflow scheduling problem in cloud computing, these papers were selected from the period (2010–2022) using various well-known digital libraries such as Google Scholar,[5] Springer LNCS,[6] ScienceDirect—Elsevier,[7] and IEEE eXplore.[8] While reviewing the papers, we noticed that they can be classified according to the aspects mentioned in the previous Sect. 4, in addition to the other two parameters that we have extracted while reviewing, where the first relates to the policy of security they used and the second to the security level they targeted. Thus, Fig. 3 illustrates the taxonomy we have proposed for the secure workflow scheduling approaches in cloud environments.

5.1 Security Policy

The research works we have reviewed use different policies to meet the user requirements in terms of security, among them we distinguish two policies: task/data placement policy and security-enhanced scheduling policy.

Task/Data Placement Policy: This category includes the set of studies assuming that the CSP offers secure resources, e.g., the CSP provides some virtual machines

[5] (https://www.scholar.google.com).

[6] (https://www.springer.com/lncs).

[7] (https://www.sciencedirect.com).

[8] (https://ieeexplore.ieee.org).

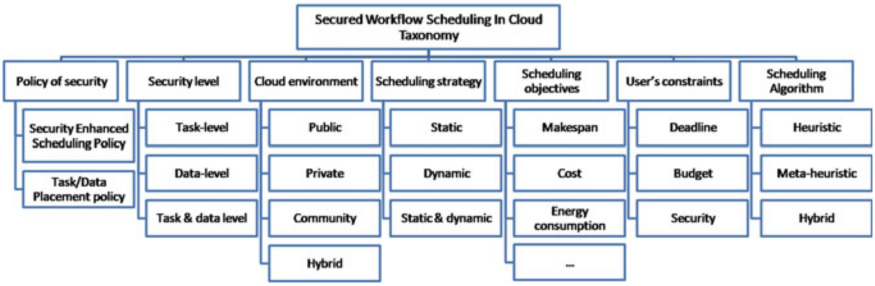


Fig. 3 Taxonomy of secured workflow scheduling techniques in cloud

that are pre-provisioned with a certain level of security services, even in the case of the hybrid cloud they assume that the private cloud is secure. Therefore, they propose data/task placement policies so that sensitive data/tasks will be mapped to secure resources.

Security-Enhanced Scheduling Policy: This category includes the set of studies assuming that the user has some security requirements, and the available cloud resources cannot meet these requirements, so they enhance the scheduling approach with some hash functions, encryption/decryption algorithms, and security services like authentication, confidentiality, and integrity services.

5.2 Security Level

The research works we have reviewed can be classified according to the level of security they target in three categories: task level, data level, and task and data level.

Task Level: In this category, the scheduler must preserve sensitive tasks, either by adding security services or by implementing placement strategies so that sensitive tasks will be protected.

Data Level: In this category, the scheduler must preserve sensitive data, either by adding security services or by implementing placement strategies so that sensitive data will be protected.

Task and Data Level: In this category, the scheduler must preserve both tasks and data simultaneously, it hybridizes the two aforementioned strategies.

6 Summary of Secured Workflow Scheduling Approaches

In this section, we list and summarize the existing secure workflow scheduling approaches. In addition, we classify them according to the taxonomy proposed in the previous section (Tables 1 and 2). Furthermore, we criticize the existing approaches and extract the disadvantages for each reviewed paper as shown in Table 3, in order to find the gaps and help researchers to improve and cover this research area and provide more effective solutions.

6.1 Task/Data Placement Policy

In the literature, various studies have proposed task/data placement strategies to secure workflow scheduling, so that sensitive tasks/data will be mapped to the most secure resources without affecting the temporal and monetary cost of the workflow scheduling, as described in Sect. 5.1. Table 1 summarizes and classifies the existing studies regarding the taxonomy proposed in the previous section.

- **Xiaoyong et al.** [52] offered a Security Driven Scheduling (SDS) algorithm for heterogeneous distributed systems in order to meet tasks security requirements with minimum makespan, where the SDS algorithm measures the trust level of system nodes dynamically, then provides secure scheduling list that considers security overhead and risk probability.
- **Liu et al.** [28] presented a Variable Neighborhood Particle Swarm Optimization (PSO) (VNPSO) to solve the scheduling issues for workflow applications in data-intensive distributed computing environments, where the goal of the schedule was to reduce the makespan while maintaining security requirements. To evaluate the proposed VNPSO, they compared it with Multi-Start Genetic Algorithm (GA) (MSGA) and Multi-Start PSO (MSPSO) and they found that VNPSO is the most feasible and efficient.
- **Marcon et al.** [30] proposed a scheduling approach in order to reduce the tenant cost while preserving the security and time requirements in a hybrid cloud.
- **Jianfang et al.** [24] They utilized a cloud model to assess the security level of tasks and resources as well as the user's degree of security satisfaction. They developed a Cloud Workflow Discrete PSO (CWDPSO) algorithm to overcome the security issues while scheduling workflows in the cloud. The scheduling goals were to accelerate the execution, reduce the monetary cost and preserve security requirements.
- **Liu et al.** [29] proposed a security-aware placement strategy based on the ACO algorithm, which uses dynamic selection to choose the best data centers for intermediate data while taking data transmission time into account.
- **Zeng et al.** [54] introduced the concept of an immovable dataset, which limits the transfer of some information for economic and security reasons. In order to offer higher security services and a quicker response time, they also recommended a security and budget-constrained workflow scheduling approach (SABA).
- **Chen et al.** [13] presented a Genetic Algorithm (GA)-based technique to reduce the processing cost and preserve data privacy while scheduling data-intensive workflow applications in the cloud, where they considered that the private data can

only be placed and scheduled in a specified data center, it cannot be transmitted to or duplicated from other datacenters during scheduling process.

- **Sharif et al.** [37] presented the MPHC algorithm with three policies (MPHC-P1/P2/P3) to address task/data privacy and time requirements while reducing workflow execution costs. Regarding privacy protection, they applied and studied two different policies Multi-terminal Cut and Bell-LaPadula.
- **Li et al.** [27] introduced a PSO-based security and cost-aware scheduling (SCAS) algorithm, where the scheduling objective was to reduce monetary cost and met the user's deadline and risk rate constraints.
- **Prince et al.** [35] proposed a hybrid workflow scheduling algorithm that combines the HEFT and SABA algorithms, in order to reduce the workflow execution time under deadline, budget, and security limitations.
- **Shishido et al.** [41] studied the impact of three meta-heuristic algorithms Practicle Swarm Optimisation (PSO), GA, and Multi Population GA (MPGA) on the cloud workflow scheduling problem; they aimed to minimize cost and met the user's deadline and risk rate constraints. The experimental results show that in terms of cost and time, GA-based algorithms outperform PSO-based algorithms.
- **Sujana et al.** [45] proposed a PSO-based approach to overcome the secure workflow scheduling issue. They used a Smart PSO algorithm to minimize cost and time and met security requirements, and they are based on a Variable Neighborhood PSO algorithm to fix the local optima issue.
- **Thanka et al.** [47] suggested a more effective Artificial Bee Colony (ABC) algorithm to reduce time and cost while maintaining the risk rate restriction. According to the experimental findings, the algorithm guarantees security and is better than other similar algorithms in terms of cost, time, and task migration throughout the schedule.
- **Bidaki et al.** [8] suggested an Symbiotic Organism Search (SOS)-based method to reduce the processing time and cost while maintaining security. The simulations demonstrate that in terms of cost, makespan, and level of security, the SOS-based method performs better than the PSO-based method.
- **Naidu and Bhagat** [34] suggested a Modified-PSO with a Scout-Adaption (MPSO-SA) scheduling approach to reduce workflow execution time under security restrictions. In order to preserve the security constraint, they schedule the workflow tasks in three modes (secure mode, risky mode, and gamma-risky mode) according to the task security requirements. The simulation findings demonstrate that MPSO-SA offers a more cost-effective, low-security risk alternative to GA, PSO, and VNPSO.
- **Arunarani et al.** [6] presented the FFBAT algorithm that hybridizes the Firefly and BAT algorithms to reduce the cost of workflow execution while satisfying deadline and risk restrictions. The simulation results demonstrate that, in terms of cost, time, and risk level, FFBAT is preferable to Firefly and BAT algorithms.
- **Xu et al.** [53] proposed data placement method to save costs and energy consumption in the hybrid cloud environment, where the scheduling goals were to retain sensitive data while reducing energy use in the private cloud and financial costs in the public cloud.

- **Shishido et al.** [42] suggested a Workflow Scheduling Task Selection Policy (WS-TSP), where they used a MPGA to reduce the execution cost while maintaining the deadline.
- **Swamy and Mandapati** [46] proposed a fuzzy logic algorithm to address the dynamic cloud workflow scheduling problem while minimizing energy consumption and preserving the user's security requirements. According to the simulation findings, the algorithm outperforms Max-Min and Min-Min algorithms in terms of makespan, resource utilization, and degree of imbalance.
- **Wen et al.** [50] proposed a Multi-Objective Privacy-Aware (MOPA) workflow scheduling system, which intended to reduce cost while maintaining the privacy protection requirement. The algorithm uses the Pareto optimality technique to determine the trade-off between the scheduling objectives.
- **Abdali et al.** [3] suggested a hybrid meta-heuristic scheduling technique that combines the Chaotic PSO algorithm and the ac GA algorithm to reduce risk rate and user limitations while minimizing cost and load balance deviation.

6.2 Security-Enhanced Scheduling Policy

In the literature, various studies have proposed security-enhanced scheduling strategies, where they proposed optimized scheduling schemes that meet the scheduling objectives and QoS constraints, then they add some security service to preserve the task/data privacy as described in Sect. 5.1. Table 2 summarizes and classifies the existing studies regarding the taxonomy proposed in the previous section.

- **Zhu et al.** [55] provided a Security-Aware Workflow Scheduling algorithm (SAWS) to save cost and time, enhance resource usage of virtual machines, and maintain security measures. To achieve these objectives and meet the constraints, the approach makes advantage of task slack time, intermediate data encryption, and selective task duplication to fill empty slots.
- **Chen et al.** [14] extended the Zhu et al. [55] work, where they presented a scheduling approach with selective tasks duplication, named SOLID, which is a developed version of the SAWS algorithm. The simulation results show that SOLID is more efficient in terms of makespan, monetary costs, and resource efficiency compared to existing similar algorithms.
- **Hammouti et al.** [20] established a new workflow scheduling strategy for hybrid cloud to provide clients high security systems at minimal cost and time. The proposed strategy consists of three modules: the Pre-Scheduler, which assigns each task or dataset to be executed or stored in either the private or public cloud; the Security Enhancement Module, which adds the dataset's necessary security services while minimizing the generated cost overhead; and the Post-Scheduler, which assigns each task or dataset to be executed or stored in the appropriate VM. The results of the experiment demonstrate that the suggested technique maintains the same cost but increases the execution time.

- **Dubey et al.** [16] created a new Management System for supplying a Community cloud with multiple organizations (MSMC), where they proposed a new cloud framework enhanced with some security services in addition to three algorithms, where the first is an allocation method that effectively divides up the available VMs across different companies with a variety of employees, the second is scheduling algorithm, called Ideal Distribution Algorithm (IDA) that takes into account cost and time restrictions and the third is an Enhaced IDA (EIDA) algorithm to improve the workload balance.
- **Abazari et al.** [1] developed a scheduling technique to reduce makespan and satisfy the task's security criteria. Moreover, they enhanced the system security by introducing a novel attack response strategy to lessen some cloud security vulnerabilities.
- **Hammed and Arunkumar** [19] proposed a new cloud workflow scheduling approach to minimize cost and time and maintain sensitive data. The proposed approach consists of four phases, categorization of tasks, scheduling of tasks, resource allocation, and security provisioning. Where the first phase categorizes the workflow tasks into high sensitive and less sensitive based on user requirements, the second and third phases are concerned with the workflow scheduling and resource allocation with minimum cost and time, and the fourth phase offers a security provisioning only for sensitive tasks resulted from the first phase.
- **Wang et al.** [48] proposed a Task Scheduling method concerning Security (TSS) in hybrid clouds, where the scheduling objectives are the completed task number maximization and minimizing the total cost of renting public resources while meeting with user's security and deadline requirements. Concerning security, they assume that the private cloud is secure, but for tasks assigned to the public cloud, they provide some authentication, integrity, and confidentiality services with different levels according to the user requirements. In addition, the TSS algorithm can control the overhead generated by the security services and minimize the total cost and meet the deadline constraint.
- **Shishido et al.** [40] proposed a scheduling strategy employing a multi-population genetic algorithm to save costs and preserve the user's defined deadline. In addition, they introduced a user annotation technique for workflow tasks according to sensitiveness, subsequently looked at the effects of using security services for delicate tasks. The simulation findings demonstrate that, when compared to existing techniques in the literature, the proposed method can more effectively preserve sensitive tasks at a lower cost.
- **Lei et al.** [26] discussed that a hybrid encryption technique is developed using hash functions to maintain data security while data moves between multiple clouds, and a novel privacy- and security-aware scheduling model is also offered. Then, in order to minimize the cost within the restrictions of deadline and privacy, they introduced a simulated annealing method and a privacy- and security-aware list scheduling algorithm.
- **Bal et al.** [7] proposed a combined Resource Allocation security with an efficient Task Scheduling algorithm in cloud computing using a Hybrid Machine learning (RATS-HM) technique; it consists of scheduling tasks with minimum makespan

and maximum throughput using an improved cat swarm optimization algorithm. Then, allocating resources under bandwidth and resource load constraints using a group optimization-based deep neural network. Finally, an authentication method is suggested for data encryption to secure data storage.

Table 3 summarizes the gaps and disadvantages we found while reviewing the papers, which may help researchers to develop, improve, and cover this research field and provide more effective solutions.

7 Analysis and Discussion

In the present paper, we reviewed about 32 secure scheduling approaches in cloud computing as shown in Tables 1 and 2. In this section, we provide a comprehensive analysis and discussion of the reviewed approaches regarding different aspects, including cloud environment, scheduling strategy, security levels, proposed algorithms, and scheduling objectives as illustrated in Fig. 4.

Cloud Environment. Figure 4a illustrates that the public cloud gains the interest of about 75% of the approaches reviewed, while 22% focus on the hybrid cloud. However, the community cloud is not yet sufficiently studied.

Scheduling Strategy. Figure 4b indicates that most of the studied researches (84%) use static scheduling strategies, while 13% use dynamic strategies and only 3% use static and dynamic strategies, despite the hybrid strategy is more useful than the others.

Security Level. Figure 4c shows that 58% of the reviewed researches aim to secure tasks, while 35% aim to secure data and only 07% aim to secure both task and data simultaneously.

Proposed Algorithm. Figure 4d demonstrates that 50% of the reviewed researches proposed meta-heuristic algorithms to solve the problem, while 41% of them proposed heuristic algorithms and only 9% proposed hybrid algorithms.

Optimization problem and Scheduling Objectives. Among Fig. 4e we notice that 62% of studied researches investigated single-objective optimization problems, while 38% considered multi-objective optimization problems. Concerning the scheduling objectives, we observe that Cost and Makespan are the most targeted objectives.

Figure 5 indicates that the problem of secure workflow scheduling has not been discussed sufficiently in the past decade and that most of the papers reviewed were published in 2017. However, it didn’t receive much attention during 2010–2014 and 2020–2022.

Table 1 Secured workflow scheduling approaches with task/data placement policy

Authors and year	Approach	Algorithm	Objectives	Constraints	Strategy	Level	Environment
Xiaoyong et al. (2010) [52]	HDS	Heuristic	M	S	Dynamic	Task	HDS
Liu et al. (2012) [28]	VNPSO	Meta-heuristic	M	S	Static	Task	Cloud, grid
Marcon et al. (2013) [30]	AltroStatus	Heuristic	C	D, S	Static	–	Hybrid cloud
Sharif et al. (2013) [38]	MPHC	Heuristic	C	D, P	Static	Task	Hybrid cloud
Jianfang et al. (2014) [24]	CWDPSO	Meta-heuristic	M, C	S	Static	Task	Cloud
Liu et al. (2014) [29]	ACO-based	Meta-heuristic	M	S -	Dynamic	T & D	Cloud
Zeng et al. (2015) [54]	SABA	Heuristic	M	B, S	Sta. and Dyn.	Task	Cloud
Chen et al. (2015) [13]	CP-GA	Meta-heuristic	C	P	Static	Data	Cloud
Sharif et al. (2016) [37]	MPHC-P1/P2/P3	Heuristic	C	D, P	Static	T & D	Hybrid cloud
Li et al. (2016) [27]	SCAS	Meta-heuristic	C	D, RR	Static	Task	Cloud
Prince et al. (2016) [35]	HEFT-SABA	Heuristic	M	D, B, S	Static	Task	Cloud
Shishido et al. (2017) [41]	PSO, GA, MPGA	Meta-heuristic	C	D, RR	Static	Task	Cloud
Sujana et al. (2017) [45]	PSO-based	Meta-heuristic	M, C	S	Dynamic	Task	Cloud
Thanka et al. (2017) [47]	IE-ABC	Meta-heuristic	M, C	S	Static	Task	Cloud
Bidaki et al. (2017) [8]	SOS-based	Meta-heuristic	M, C	S	Static	Task	Cloud
Naidu and Bhagat (2017) [34]	MPSO-SA	Meta-heuristic	M	S	Static	Task	Cloud
Arunarani et al. (2017) [6]	FFBAT	Hybrid	C	D, RR	Static	Task	Cloud
Xu et al. (2017) [53]	CEDP	Heuristic	C, E	P	Static	Data	Hybrid cloud
Shishido et al. (2018) [42]	WS-TSP	Meta-heuristic	C	D, RR	Static	Data	Cloud
Swamy and Mandapati (2018) [46]	Fuzzy	Meta-heuristic	E	S	Dynamic	Data	Cloud
Wen et al. (2018) [50]	MOPA	Meta-heuristic	M, C	P	Static	Task	Cloud
Abdali et al. (2019) [3]	CPSO-GA	Hybrid	C, LB	D, RR	Static	Task	Cloud

M: Makespan, C: Cost, E:energy, S: Security, RR: Risk Rate D deadline, B: Budget, P: Privacy, LB:Load Balancing

Table 2 Secured workflow scheduling approaches with security-enhanced scheduling policy

Authors and Year	Approach	Algorithm	Objectives	Constraints	Strategy	Level	Environment
Zhu et al. (2016) [55]	SAWS	Heuristic	M, C, RU	S	Static	Data	Cloud
Chen et al. (2017) [14]	SOLID	Heuristic	M, C ,RU	S	Static	Data	Cloud
Hammouti et al. (2020) [20]	SLp	Heuristic	M, C	D, B, S	Static	Data	Hybrid cloud
Dubey et al. (2019) [16]	MSMC (IDA+EIDA)	Heuristic	C	D, S	Static	Data	Community cloud
Abazari et al. (2019) [1]	MOWS	Heuristic	M	S	Static	Task	Cloud
Hammed and Arunkumar (2019) [19]	MPGA-based	Meta-heuristic	M, C	S	Static	Data	Cloud
Wang et al. (2021) [48]	TSS	Heuristic	C, Number of finished tasks	D, S	Static	Task	Hybrid cloud
Shishido et al. (2021) [40]	SAST	Meta-heuristic	C	D, S	Static	Task	Cloud
Lei et al. (2022) [26]	PSLS and PSSA	Hybrid	S	D, P	Static	Data	Hybrid cloud
Bal et al. (2022) [7]	RATS-HM	Meta-heuristic	M, T	BW, RL, S	Static	Data	Cloud

M: Makespan, C: Cost, RU; Resource utilization, S: Security, D deadline, B:Budget, P: Privacy, BW: Bandwidth, RL:Resource Load, T: Throughput

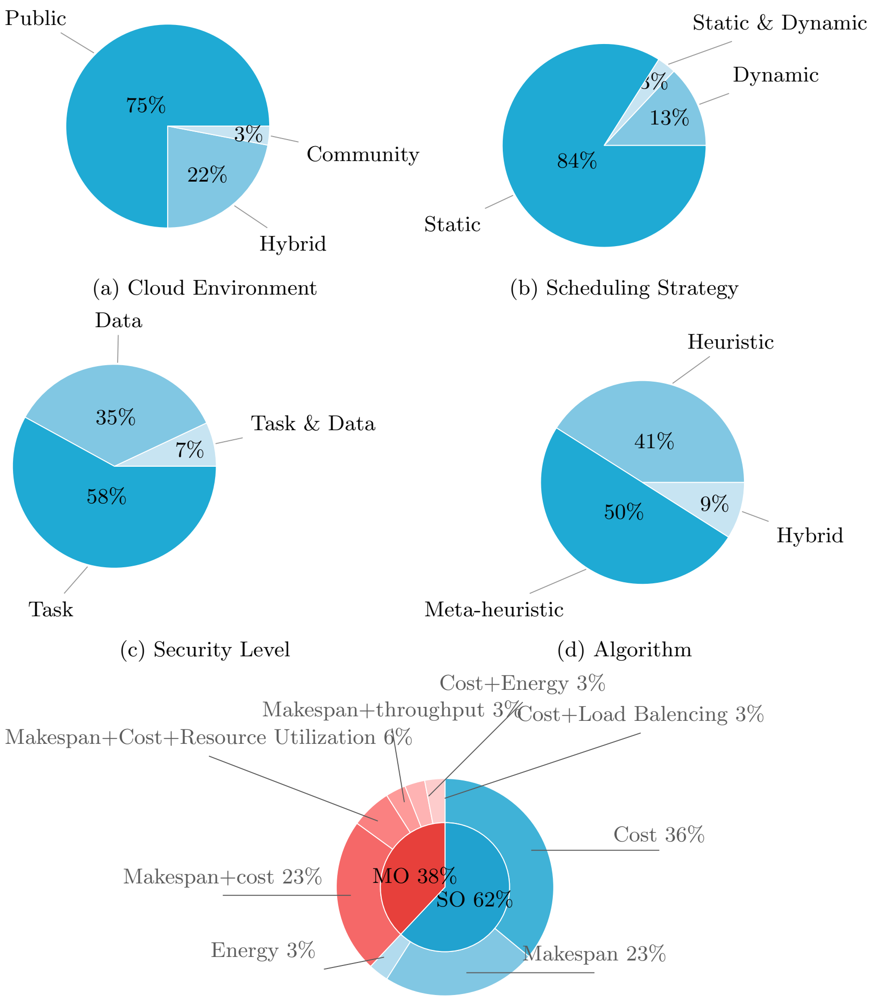
Table 3 Gaps of secure workflow scheduling papers

Work	Gaps
[1, 3, 6–8, 14, 16, 20, 24, 26, 27, 34, 35, 40–42, 48, 50, 53, 55]	Does not consider the dynamic scheduling strategy
[3, 8, 28, 47, 52, 53]	Does not secure the intermediate data transfer
[3, 30, 38]	Does not consider the execution time and other scheduling objectives
[1, 7, 46]	Does not consider cost and other scheduling objectives
[29]	Does not consider the workflow total execution cost and time
[54]	Does not consider security in dynamic scheduling strategy
[13]	Adding the privacy constraint negatively affects the scheduling cost
[20]	Adding security services affects negatively the execution time
[37]	In case of loose deadline scenarios, MPHC-P2/P3 are less efficient
[45]	Does not consider other scheduling objectives and constraints
[28, 47]	The proposed algorithm was tested using a very small-scale scheduling problem.
[19]	They compared the proposed approach with other approaches that offer security provisioning for all tasks, which is unfair because it is clear that securing all workflow tasks results in more overhead than securing only sensitive tasks
[40]	Does not consider the identification of sensitive tasks in automated ways Does not consider the trade-off between cost and makespan

8 Open Issues and Challenges

Among the previous sections, we conclude that the problem of secure workflow scheduling in cloud computing is not yet sufficiently discussed in the literature. Consequently, we extract the following issues and challenges:

- Introduce dynamic scheduling strategies, as they are more adaptable to real systems and take more parameters into consideration.
- Introduce hybrid scheduling strategies (static and dynamic), as they provide high-quality solutions and they are adaptable to the real systems at the same time.



(a) Cloud Environment

(b) Scheduling Strategy

(c) Security Level

(d) Algorithm

(e) Optimization Problem & Scheduling Objectives[a]

[a] SO: Single-Objective; MO:Multi-Objective

Fig. 4 Secured cloud scheduling approaches

- Provide scheduling approaches that aim to secure both task and data, in order to improve and ensure the security of all the workflow.
- Provide more secure scheduling approaches that address the hybrid and community clouds as they are insufficiently studied.
- Introduce secure scheduling approaches that address more scheduling objectives, e.g., energy, load balancing, etc.

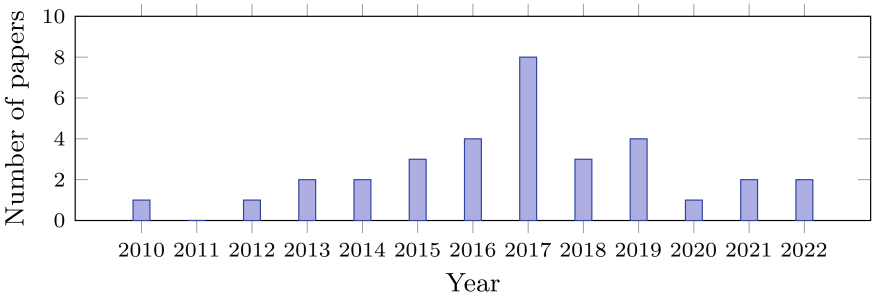


Fig. 5 Secured cloud scheduling papers per year

- Focus on multi-objective optimization approaches, as they aim to provide trade-off solutions between different scheduling objectives.
- Provide more robust scheduling security-enhanced policies to keep tasks and data secure simultaneously.

9 Conclusion

In this paper, we reviewed the existing secure workflow scheduling strategies, we outlined important aspects surrounding the problem, and then we introduced a taxonomy to categorize existing research papers. Moreover, we provided a comprehensive analysis, and hence we derived different gaps, issues, and challenges that can help researchers to improve and cover this research area and provide more effective solutions. In future studies, we can investigate the aforementioned issues and challenges.

References

1. Abazari F, Analoui M, Takabi H, Fu S (2019) Mows: multi-objective workflow scheduling in cloud computing based on heuristic algorithm. Simul Model Pract Theory 93:119–132
2. Adhikari M, Amgoth T, Srirama SN (2019) A survey on scheduling strategies for workflows in cloud environment and emerging trends. ACM Comput Surv (CSUR) 52(4):1–36
3. Ali Abdali S (2019) A new optimization method for security-constrained workflow scheduling. Indian J Comput Sci Eng (IJCSE) 10(1)
4. Alkhanak EN, Lee SP, Rezaei R, Parizi RM (2016) Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: a review, classifications, and open issues. J Syst Softw 113:1–26
5. Anju Baby J (2013) A survey on honey bee inspired load balancing of tasks in cloud computing. Int J Eng Res Technol 2(12):1442–5
6. Arunarani A, Manjula D, Sugumaran V (2017) FFBAT: a security and cost-aware workflow scheduling approach combining firefly and bat algorithms. Concurr Comput Pract Exp 29(24):e4295

7. Bal PK, Mohapatra SK, Das TK, Srinivasan K, Hu YC (2022) A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques. Sensors 22(3):1242
8. Bidaki M, Tabbakh SRK, Yaghoobi M, Shakeri H (2017) Secure and efficient SOS-based workflow scheduling in cloud computing. Int J Secur Its Appl 11:41–58
9. Bittencourt LF, Goldman A, Madeira ER, da Fonseca NL, Sakellariou R (2018) Scheduling in distributed systems: a cloud computing perspective. Comput Sci Rev 30:31–54
10. Blythe J, Jain S, Deelman E, Gil Y, Vahi K, Mandal A, Kennedy K (2005) Task scheduling strategies for workflow-based applications in grids. In: CCGrid 2005. IEEE international symposium on cluster computing and the grid, vol 2, pp 759–767. IEEE
11. Buyya R, Murshed M (2002) A deadline and budget constrained cost-time optimisation algorithm for scheduling task farming applications on global grids. arXiv preprint cs/0203020
12. Cao J, Spooner DP, Jarvis SA, Saini S, Nudd GR (2003) Agent-based grid load balancing using performance-driven task scheduling. In: Proceedings international parallel and distributed processing symposium. IEEE, pp 10–pp
13. Chen C, Liu J, Wen Y, Chen J, Zhou D (2015) A hybrid genetic algorithm for privacy and cost aware scheduling of data intensive workflow in cloud. In: International conference on algorithms and architectures for parallel processing. Springer, pp 578–591
14. Chen H, Zhu X, Qiu D, Liu L, Du Z (2017) Scheduling for workflows with security-sensitive intermediate data by selective tasks duplication in clouds. IEEE Trans Parallel Distrib Syst 28(9):2674–2688
15. Desale S, Rasool A, Andhale S, Rane P (2015) Heuristic and meta-heuristic algorithms and their relevance to the real world: a survey. Int J Comput Eng Res Trends 351(5):2349–7084
16. Dubey K, Shams MY, Sharma SC, Alarifi A, Amoon M, Nasr AA (2019) A management system for servicing multi-organizations on community cloud model in secure cloud environment. IEEE Access 7:159535–159546
17. Fakhfakh F, Kacem HH, Kacem AH (2014) Workflow scheduling in cloud computing: a survey. In: 2014 IEEE 18th international enterprise distributed object computing conference workshops and demonstrations. IEEE, pp 372–378
18. Francis AO, Emmanuel B, Zhang D, Zheng W, Qin Y, Zhang D (2018) Exploration of secured workflow scheduling models in cloud environment: a survey. In: 2018 sixth international conference on advanced cloud and big data (CBD). IEEE, pp 71–76
19. Hammed SS, Arunkumar B (2019) Efficient workflow scheduling in cloud computing for security maintenance of sensitive data. Int J Commun Syst 35(2)
20. Hammouti S, Yagoubi B, Makhlouf SA (2020) Workflow security scheduling strategy in cloud computing. In: International symposium on modelling and implementation of complex systems, Springer, pp 48–61
21. Houssein EH, Gad AG, Wazery YM, Suganthan PN (2021) Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends. Swarm Evol Comput 62:100841
22. Ibrahim IM et al (2021) Task scheduling algorithms in cloud computing: a review. Turk J Comput Math Educ (TURCOMAT) 12(4):1041–1053
23. Jain S, Meena J (2019) Workflow scheduling algorithms in cloud computing: an analysis, analogy, and provocations. In: Innovations in computer science and engineering. Springer, pp 499–508
24. Jianfang C, Junjie C, Qingshan Z (2014) An optimized scheduling algorithm on a cloud workflow using a discrete particle swarm. Cybern Inf Technol 14(1):25–39
25. Kaur S, Bagga P, Hans R, Kaur H (2019) Quality of service (QoS) aware workflow scheduling (WFS) in cloud computing: a systematic review. Arab J Sci Eng 44(4):2867–2897. http://link.springer.com/10.1007/s13369-018-3614-3
26. Lei J, Wu Q, Xu J (2022) Privacy and security-aware workflow scheduling in a hybrid cloud. Futur Gener Comput Syst 131:269–278
27. Li Z, Ge J, Yang H, Huang L, Hu H, Hu H, Luo B (2016) A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds. Futur Gener Comput Syst 65:140–152

28. Liu H, Abraham A, Snášel V, McLoone S (2012) Swarm scheduling approaches for work-flow applications with security constraints in distributed data-intensive computing environments. Inf Sci 192:228–243
29. Liu W, Peng S, Du W, Wang W, Zeng GS (2014) Security-aware intermediate data placement strategy in scientific cloud workflows. Knowl Inf Syst 41(2):423–447
30. Marcon DS, Bittencourt LF, Dantas R, Neves MC, Madeira ER, Fernandes S, Kamienski CA, Barcelos MP, Gaspary LP, da Fonseca NL (2013) Workflow specification and scheduling with security constraints in hybrid clouds. In: 2nd IEEE Latin American conference on cloud computing and communications. IEEE, pp 29–34
31. Masdari M, ValiKardan S, Shahi Z, Azar SI (2016) Towards workflow scheduling in cloud computing: a comprehensive analysis. J Netw Comput Appl 66:64–82. https://linkinghub.elsevier.com/retrieve/pii/S108480451600045X
32. Mell P, Grance T (2011) The NIST definition of cloud computing, p 7
33. Mohialdeen IA (2013) Comparative study of scheduling algorithms in cloud computing environment. J Comput Sci 9(2):252–263
34. Naidu PS, Bhagat B (2017) Secure workflow scheduling in cloud environment using modified particle swarm optimization with scout adaptation. Int J Model Simul Sci Comput 9(01):1750064
35. Prince PB, Ruphavathani DA, Lovesum SJ (2016) A security aware resource allocation model for cloud based healthcare workflows. Indian J Sci Technol 9(45):2–6
36. Rodriguez MA, Buyya R (2017) A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments. Concurr Comput Pract Exp 29(8):e4041
37. Sharif S, Watson P, Taheri J, Nepal S, Zomaya AY (2016) Privacy-aware scheduling SaaS in high performance computing environments. IEEE Trans Parallel Distrib Syst 28(4):1176–1188
38. Sharif S, Taheri J, Zomaya AY, Nepal S (2013) Mphc: preserving privacy for workflow execution in hybrid clouds. In: 2013 international conference on parallel and distributed computing, applications and technologies. IEEE, pp 272–280
39. Sharma Y, Javadi B, Si W, Sun D (2016) Reliability and energy efficiency in cloud computing systems: survey and taxonomy. J Netw Comput Appl 74:66–85
40. Shishido HY, Estrella JC, Toledo CF, Reiff-Marganiec S (2021) Optimizing security and cost of workflow execution using task annotation and genetic-based algorithm. Computing 103(6):1281–1303
41. Shishido HY, Estrella JC, Toledo CFM, Arantes MS (2017) Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds. Comput Electr Eng 69:378–394
42. Shishido HY, Estrella JC, Toledo CFM, Reiff-Marganiec S (2018) Tasks selection policies for securing sensitive data on workflow scheduling in clouds. In: 2018 IEEE international conference on services computing (SCC). IEEE, pp 233–236
43. Singh P, Dutta M, Aggarwal N (2017) A review of task scheduling based on meta-heuristics approach in cloud computing. Knowl Inf Syst 52(1):1–51
44. Smanchat S, Viriyapant K (2015) Taxonomies of workflow scheduling problem and techniques in the cloud. Futur Gener Comput Syst 52:1–12
45. Sujana J, Revathi T, Priya T, Muneeswaran K (2017) Smart PSO-based secured scheduling approaches for scientific workflows in cloud computing. Soft Comput 23(5):1745–1765
46. Swamy SR, Mandapati S (2017) A fuzzy energy and security aware scheduling in cloud. Int J Eng Technol 7(2):117–124
47. Thanka MR, Uma Maheswari P, Edwin EB (2019) An improved efficient: artificial bee colony algorithm for security and QoS aware scheduling in cloud computing environment. Clust Comput 22(5):10905–10913
48. Wang B, Wang C, Huang W, Song Y, Qin X (2021) Security-aware task scheduling with deadline constraints on heterogeneous hybrid clouds. J Parallel Distrib Comput 153:15–28
49. Wang Y, Guo Y, Guo Z, Liu W, Yang C (2019) Securing the intermediate data of scientific workflows in clouds with ACISO. IEEE Access 7:126603–126617

50. Wen Y, Liu J, Dou W, Xu X, Cao B, Chen J (2018) Scheduling workflows with privacy protection constraints for big data applications on cloud. Futur Gener Comput Syst 108:1084–1091
51. Wu F, Wu Q, Tan Y (2015) Workflow scheduling in cloud: a survey. J Supercomput 71(9):3373–3418
52. Xiaoyong T, Li K, Zeng Z, Veeravalli B (2010) A novel security-driven scheduling algorithm for precedence-constrained tasks in heterogeneous distributed systems. IEEE Trans Comput 60(7):1017–1029
53. Xu X, Zhao X, Ruan F, Zhang J, Tian W, Dou W, Liu AX (2017) Data placement for privacy-aware applications over big data in hybrid clouds. Secur Commun Netw 2017
54. Zeng L, Veeravalli B, Li X (2015) Saba: a security-aware and budget-aware workflow scheduling strategy in clouds. J Parallel Distrib Comput 75:141–151
55. Zhu X, Zha Y, Jiao P, Chen H (2016) Security-aware workflow scheduling with selective task duplication in clouds. In: Proceedings of the 24th high performance computing symposium, pp 1–8

A New BERT-Inspired Knowledge Distillation Approach Toward Compressed AI Models for Edge Devices

Suryabhan Singh, Kirti Sharma, Brijesh Kumar Karna, and Pethuru Raj

Abstract In the natural language processing (NLP) domain [7], the pre-trained models for tackling a variety of language understanding tasks (text-to-speech, speech-to-text, text summarization, etc.) are typically large in size. Because of their larger size, accommodating and running them in resource-constrained IoT edge devices are becoming a tough assignment. Large models generally consume a lot of computing and storage resources, waste precious power energy, and dissipate more heat into our fragile environment. Therefore, the concept of knowledge distillation has picked up fast as a viable and venerable method for model optimization. In this paper, we showcase a robust distillation approach for producing small-sized pre-trained models that can run on edge devices comfortably. For this unique distillation process, we have considered several pre-trained models and used BERT as a base model, and presented the results obtained through a few practical experiments. Distillation models are fine-tuned with good performance for a wide range of applications. Here, we have considered all 24 types of BERT models with different layers and included the hidden layers for distillation. This has given a performance improvement over the base model performance by employing fine-tuning for all types of models at various stages. We have reduced the model size by a margin of greater than 40%, while simultaneously retaining its original model language understanding capabilities and speed. The proposed model is smaller, faster, and cheaper to train and run.

Keywords AI model optimization · BERT · Transformers · Edge deployment · Model performance and accuracy

S. Singh (✉) · K. Sharma · B. K. Karna · P. Raj
Edge AI Division, Reliance Jio Platforms Ltd., Avana Building, Bangalore 560103, India
e-mail: Suryabhan2.Singh@ril.com

K. Sharma
e-mail: kirti4.sharma@ril.com

B. K. Karna
e-mail: brijesh.karna@ril.com

P. Raj
e-mail: Pethuru.chelliah@ril.com

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_8

1 Introduction

Deep neural networks (DNNs) are contributing immensely across industry verticals. Especially DNNs are extremely beneficial for domains such as computer vision (face recognition, object detection, and tracking), speech recognition, and language processing such as text translation and summarization. Autonomous vehicles and surveillance cameras extensively use the latest advancements being attained in the hugely popular DNN space. State-of-the-art models are being derived and leveraged to artistically accomplish the above-mentioned everyday tasks. The problem with DNNs is the participation of an enormous number of parameters (some DNN architectures comprise billions of parameters). This makes DNNs process-intensive. For example, AlexNet is a convolutional neural network (CNN) consisting of 60 million parameters with an architecture comprising five convolutional layers, several max-pooling layers, three fully connected layers, and a final softmax layer. GPUs are often used to train and use deep neural networks because they are able to deliver the highest peak arithmetic performance when compared with central processing units (CPUs) and field programmable gate arrays (FPGAs).

It is widely accepted that deep neural network (DNN) models carry a lot of redundancy. Most parameters contribute little or nothing to the final output. By eliminating irrelevant parameters and sacrificing a bit of precision and performance, it is possible to arrive at highly optimized DNN models for natural language understanding. In a nutshell, removing non-contributing parameters makes DNN models slim and sleek. That is, highly optimized yet sophisticated models can be made to run on resource-constrained and battery-operated devices. There are several proven and powerful methods for compressing DNNs such as making a small compromise on precision, removing redundant parameters, or the structure, and transferring knowledge from large models to smaller models. The aim of model compression is usually to reduce the hardware footprint of a model while sacrificing its inference accuracy a bit. DNNs are elegantly used for NLP tasks such as question answering, machine translation, reading comprehension, and summarization.

We have developed a framework, which enlightens knowledge distillation (KD) on multiple BERT models (known as the students' model). Student models neatly capture the general and task-specific knowledge in BERT, which is known as the teacher model. The knowledge gets distilled from the teacher model to the student model. The student model with many layers is empirically effective and has achieved more than 96.8% in performance compared to its teacher model (BERTBASE) as per the GLUE benchmark. The student model is around 8 times smaller and 9.5 times faster in some cases. The paper includes explanations about the Tiny BERT model, which is also a student model, which is better than a 4-layer state-of-the-art baseline. We have discussed other BERT models such as the BERT Small, BERT Medium, and BERT Mini, which are student models, and shown their experiments and results as well.

2 Demystifying the Aspect of Knowledge Distillation

The success of deep neural networks (DNNs) generally relies upon the comprehensive design of DNN architectures. Especially for complex and large-scale machine learning tasks such as face and speech recognition, most DNN-based models are over-parameterized to extract the most salient features and to ensure greater generalization. Such deep models are bound to waste a tremendous amount of computing and storage resources for training. Running such processes and memory-intensive models on edge devices turns out to be a difficult affair. Also, training such big models consumes a lot of time thereby real-time training is a tough assignment. Therefore, the research community has swung into action in a cogent and concerted fashion. The aim of producing smaller models has rekindled the interest in the minds and hearts of AI researchers across the globe. Smaller models can be quickly trained and made to run on resource-constrained edge devices.

The performance of DNN models heavily depends on correct labels for training datasets. The prime challenge is to get a sufficient amount of labeled data. A way forward for surmounting such a lack of data is to transfer the knowledge from one source model to a target model. One well-known example is none other than semi-supervised learning. Here, a model is trained with only a small set of labeled data and a large set of unlabeled data. In this case, knowledge is transferred within the model that assumes a twin role as teacher and student. For the unlabeled data, the student learns as before. However, the teacher generates target data, which are then used by the student for further learning. Thus, knowledge gets distilled to the student model from the teacher model. Thereby a new model need not be produced from the ground up. Instead, the knowledge learned by the teacher model gets distilled into student models.

Knowledge distillation is widely regarded as a primary mechanism that enables humans to quickly learn new complex concepts when given only small training sets. In deep learning, KD is widely used to transfer information from one network to another network when getting trained. KD has been broadly used for model compression and knowledge transfer. For model compression, a smaller student model is trained to mimic a pre-trained larger model or an ensemble of models. The model providing knowledge is called the teacher, while the model learning the knowledge is called the student.

3 The Latest Trends in the Knowledge Distillation Domain

Generally, the size of deep neural networks (DNNs) is enormous with millions and even billions of parameters. Training and running such large-scale networks mandate a tremendous amount of IT infrastructure resources. For real-time enterprises, edge computing with proximate data processing capability is needed. Therefore, there is an insistence on producing ultra-light DNNs with a few thousand parameters. This is where the concept of KD comes in. The teacher and student models are the two deep neural networks. The teacher network is actually a combination of separately trained models or a single model. The student network is comparatively a smaller model, which hugely depends on the teacher network.

The idea is to use the distillation technique to transfer knowledge from the large teacher model to the smaller student model. Several everyday problems are being solved through the distinct advancements occurring in the natural language processing (NLP) space. There are many deep learning (DL) algorithms empowering the NLP domain substantially. However, data is the prime input and an ingredient for creating flexible and fabulous DL models. Luckily, there is plenty of labeled and unlabeled data readily available online. This data can be used to train a baseline model that can be reused and refined across NLP tasks. Bidirectional Encoder Representations from Transformers (BERT) is one such shared and base model to be readily modified as per the problem at hand. BERT is pre-trained using unlabeled data to be used for language modeling tasks. For specific NLP tasks, the pre-trained model can be customized for meeting the goals of that task. BERT is an evolution of self-attention and transformer architecture, which is becoming hugely popular for neural network models. BERT is an encoder-only transformer. It is deeply bidirectional. That is, it uses both left and right context details in all the layers.

BERT primarily involves two stages: unsupervised pre-training followed by supervised task-specific training and learning. Once a BERT model is pre-trained, it can be shared across. This is for enabling further training on a smaller dataset toward fulfilling specific needs. That is, with a single shared baseline model, it is possible to produce a variety of specific models. While pre-training takes a few days on many cloud tensor processing units (TPUs), the act of fine-tuning takes only 30 min on a single cloud TPU. For fine-tuning, one or more output layers are being added to BERT. For the question and answering task, an input sequence contains the question and the answer while the model is trained to learn the start and end of the answers. For a problem of classification, the [CLS] token at the output is fed into a classification layer. Unlike word embeddings, BERT produces contextualized embedding. That is, BERT produces many embeddings of a word. Each embedding represents the context for the word. For example, the word2vec embedding for the word “bank” would not consider any difference between the phrases “bank account” and “bank of the river”. However, BERT knows the difference.

The Emergence of Edge AI

In the recent past, we hear and read about the concept of edge AI. The soaring demand for real-time and intelligent IoT services and applications in our everyday environments (home, hotel, hospital, manufacturing floor, retail store, etc.) calls for expertly deploying and running pioneering DNN models on IoT edge devices, which are being increasingly deployed in our locations. However, the outstanding contributions of sophisticated DNN models are being achieved through large-scale compute resources, which are found to be insufficient in IoT edge devices. Our places are stuffed with a variety of connected medical instruments and scanners, defense equipment, types of machinery on manufacturing floors, robots at retail stores, drones for product delivery, surveillance cameras at airports and other secure environments, multifaceted sensors and actuators in expressways, tunnels, bridges, etc. There are mobiles, wireless, nomadic, fixed, portable, handheld, and implantable gadgets and gizmos. It is indisputably clear that every entity gets methodically digitized and connected in order to design and deliver situation-aware digital services and applications. Besides the wastage of enormous compute resources, there is another noteworthy issue that can't be easily sidestepped. The privacy of devices and people's data has to be fully ensured at any cost.

Service providers collect, cleanse, and crunch a large volume of users' data in order to do sentiment analysis and other deeper analytics to understand more about their customers and their complaints and concerns. Data is indispensable for training and refining DNN models to solve a variety of operational issues. Directly deploying these models on each edge device may lead to data privacy challenges. Also, such a setup is bound to waste a lot of precious computing, memory, and storage resources. To benefit from the on-device deep learning without the capacity and privacy concerns, there are a few researchers who have cogently teamed up and designed a private model compression framework RONA. Following the knowledge distillation paradigm, they have jointly used hint learning, distillation learning, and self-learning to train a compact and fast neural network. More details about this privacy-preserving framework can be found in this research paper (https://arxiv.org/abs/1811.05072). With the faster maturity and stability of the machine and deep learning (ML/DL) algorithms and powerful and AI-specific processors (GPUs, TPUs, VPUs, etc.), setting up and sustaining cognitive systems and services have gained speed.

Learning new things and proposing fresh theories/hypotheses from datasets automatically are being continuously strengthened through a bevy of cutting-edge technologies and tools. Learning from data to come out with timely and accurate predictions gets a boost lately. However, with the aspect of edge computing gaining a lot of minds and market shares, highly complex learning models ought to be transmitted to edge devices to make real-time insights available with clarity and alacrity. Due to the lack of high-end GPUs in edge devices, running deep learning models on edge devices faces a challenge. To surmount this, researchers have introduced many model compression techniques. Knowledge distillation is the process of converting a hugely complex model into a smaller one to be made to run in edge devices.

The insensitive and repetitive parameters are meticulously found and eliminated to arrive at highly optimized AI models. As indicated above, the knowledge distillation (KD) technique utilizes a pre-trained teacher model for training a student network. Pre-trained transformer-based encoders such as BERT have the capability to achieve state-of-the-art performance on numerous NLP tasks [13]. Despite their success, BERT-style encoders are large and have high latency during inference, especially on CPU machines. This is found to be a hurdle for realizing real-time applications. The recently incorporated compression and distillation methods have provided effective ways to alleviate this shortcoming. In short, knowledge distillation has been successfully implemented in numerous domains like computer vision (CV), NLP, etc. In NLP, taking a pre-trained model and then fine-tuning the same for the desired purpose is the common practice. Pre-trained Language Models (PLMs), such as BERT [1], XLNet [2], RoBERTa [3], ALBERT [4], T5 [5], and ELECTRA [6], are used for overcoming the challenges of multitasking and for capturing their base model behavior.

The pre-trained base model is chosen as a teacher model. This has a large number of parameters and requires a higher computation time. This makes it very tough to deploy the model on any edge devices. Researchers such as [8, 9] have recently demonstrated that there is some redundancy in pre-trained base models. To retain their performance level, it is important to reduce their computation cost. There are several model compression techniques in the literature. The most commonly used techniques are Pruning [10], Quantization [11], and KD [12]. In this paper, we will shed light on the KD process. In knowledge distillation, a student model will imitate the teacher model. To investigate a large-scale pre-trained model, we have used the BERT technique.

Figure 1 illustrates tiny BERT and task-specific distillation. The Tiny BERT model is trained on an unsupervised text corpus and fine-tunes it on a task-specific dataset. For effective KD, there is a need to decide on a good strategy for training. To propose any distillation on the BERT base model (teacher), the distillation process has to be accordingly changed to easily extract the embedded information from the teacher model.

If we consider Tiny BERT as our distillation BERT model, then the contribution of this work is as follows: (1) We have proposed a new Transformer distillation method to adequately facilitate the transfer of all linguistic knowledge present in the teacher BERT base model to the student Tiny BERT model. (2) We have proposed a novel two-stage framework for learning and performing Transformer distillation at both pre-training and fine-tuning stages. This ensures that the distillation model of BERT (Tiny BERT) can absorb both general and task-specific knowledge of BERT. (3) In this, we have presented the experiments on the BERT distillation process considering

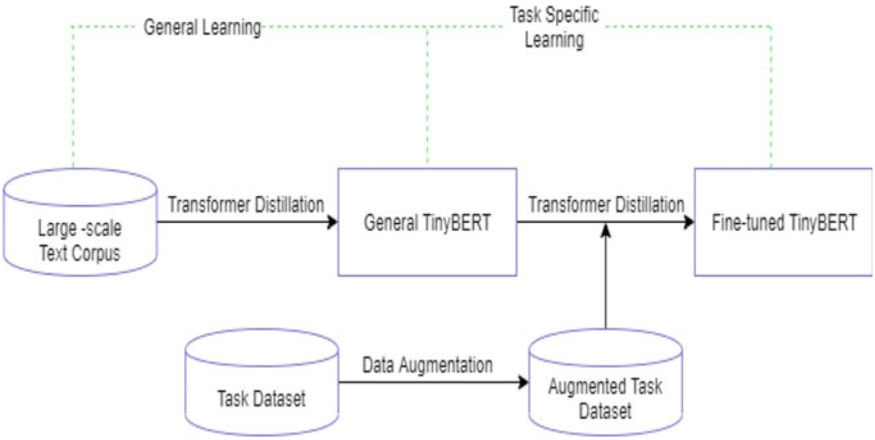


Fig. 1 General distillation and task-specific distillation

all the various types of BERT as a student and BERT Base as a teacher and measured the performance on GLUE task in terms of speed, accuracy, and inference time.

3.1 Knowledge Distillation

Knowledge distillation is a strategy for reducing the size of the teacher model. KD is used to teach the student model. As a part of the KD process, the student model learns to imitate the larger model's behaviors. Essentially, KD uses the Transformer-based architecture for NLP tasks to make the model efficient and effective. It also assists in developing an efficient model in NLP, for example, machine language translation [14] is a popular one. A classification model is trained to predict the class by maximizing the estimated probability of a given label in the case of supervised learning. In the training case, our objective is to minimize the cross-entropy between the model's predicted distribution and one hot empirical distribution of training labels. If a model performs well on the given training set, then it will predict an output distribution with a high probability on the correct class and with a near-zero probability on other classes. Occasionally, near-zero probabilities larger than the other will reflect the generalization capabilities of the model.

3.2 Training Loss

In training loss, the student model is trained with soft target probabilities with distillation loss of the teacher model represented as Lce = Pi ti log (si) where ti (resp. si) is a probability estimated by the teacher (resp. the student). We used a softmax-temperature: pi = P exp(zi/T) j exp(zj/T) where T holds the power to control the fine-tuning process and in turn control the level of smoothness of the output distribution and zi is used for the model score for the class i. The same temperature T is applicable for the student model and teacher model at the time of training, but at the time of inferences, T is set to 1 to recover a standard softmax. The final objective of model training is to linearly mix the distillation loss with Lce and the supervised training loss. We discovered that a cosine embedding loss (Lcos) leaning in the direction of the student and teacher hidden state vector should be added.

4 Proposed Solution Using Knowledge Distillation

4.1 Distillation Methods

The primary objective of the KD process is to create the smallest possible student model while maintaining the accuracy of the teacher model. As a result of our study and experiments on a variety of natural language processing tasks, we can conclude that the optimal student model's capacity to maintain accuracy may change with task difficulty. Therefore, to learn and investigate these scenarios in depth, we experimented with various sizes of distillation models, and in the end, we select the best and smallest among all the distilled models, which offers better accuracy than the original BERT base, for each specific task. After conducting all the experiments, we are able to observe that distilled model does not work well when it is distilled from a different model type. Therefore, we avoid performing distillation of RoBERT to BERT and vice versa. The distinction between these models is the input token embedding. Therefore, the disparity between input and output embedding spaces is the reason why knowledge transfer between various spaces is ineffective. Here, we propose an end-to-end Task-Specific KD on text classification using Transformers, PyTorch, and, if necessary, Amazon Sagemaker. Distillation is a process of training a small "Student model" to mimic a larger "Teacher model". Here, for these experiments, we use BERT Base as the teacher model and other BERT models (BERT Tiny, BERT small, BERT medium, etc. with different layers and hidden layers) as the Student model.

For training, we employ the Stanford Sentiment Treebank v2 (SST-2) dataset which functions as Task-Specific KD for Text Classification. Figures 2 and 3 depict two distinct types of KD: Task-Agnostic Distillation (right) and Task-Specific KD (left). Experiments are conducted utilizing Task-Specific KD in the present work. In the second step of distillation in Task-Specific KD, we employed "fine-tuning". This idea originates from the Distill BERT model [15] in which it was demonstrated that a student model can perform much better by simply fine-tuning the distilled language model. By refining the distillation process on BERT, we also investigated whether we may add additional distillation steps during the adaption phase. In this situation, there are two steps of distillation: one during the training phase of the distillation process and one during the adaptation phase of the distillation process. Utilizing BERT Base and other BERT models as Students, we can get good performance for distillation.

4.2 Dataset and Preprocessing

This proposed work utilizes the Stanford Sentiment Treebank v2 (SST-2) dataset, which is used in text categorization for sentiment analysis and includes the GLUE Benchmark. This dataset is an adaptation of the dataset presented by Pang and

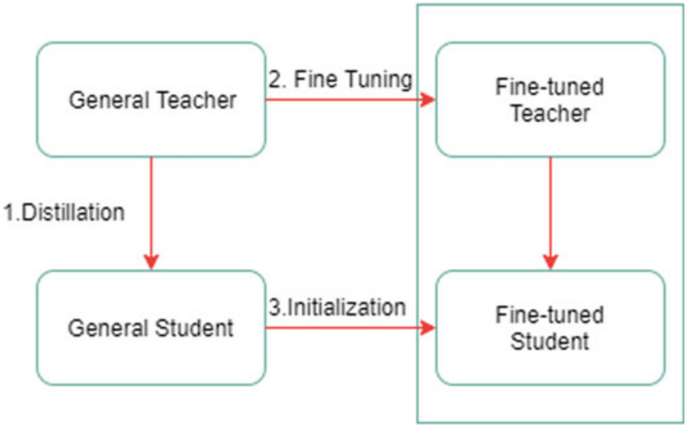


Fig. 2 Task-specific distillation

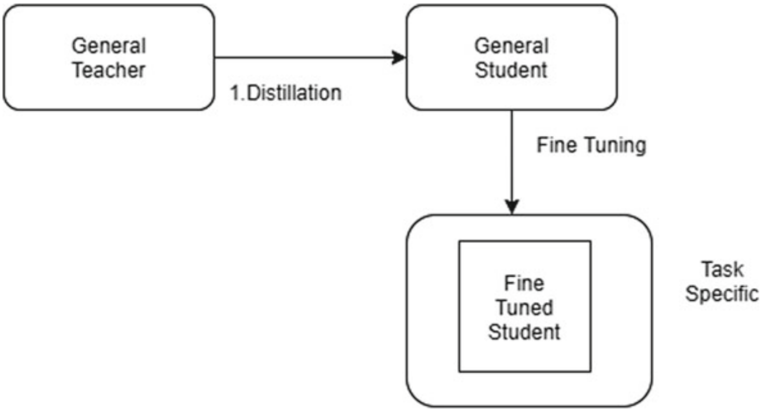


Fig. 3 Task-agnostic distillation

Lee [16] and consists of 11,855 single sentences extracted from movie reviews. This dataset was parsed with the Stanford parser and contains a total of 215,154 distinct phrases extracted from the parse tree, each of which was annotated by three human judges. It utilizes a two-way (positive/negative) class split with sentence-level labeling. We use the load dataset function from the dataset package to load the ss2 dataset. For the distillation procedure, we must transform "Natural Language" into token identifiers. This transformation is performed using a transformer tokenizer, which accepts tokens as input (including converting all tokens with their corresponding IDs in train vocabulary). We are taking this from the hugging face interface and will use it as the tokenizer for the teacher, as both produce the same result as the Student tokenizer.

4.3 Distilling the Model

Here, we perform distillation using PyTorch and Distill trainer function. Now that our dataset is already processed, we can distill it. Typically, when optimizing a transformer model using PyTorch, we must use their trainer API. In this Trainer class, PyTorch will provide an API for comprehensive training for the majority of standard use cases. We did not employ a trainer out of the box in our proposed work because it requires two models, the student model, and the teacher model, and must compute the loss for both. However, we can investigate using a trainer to create a Distillation Trainer that is capable of handling this and will only need to override the compute loss and init methods. Following this, we must add the subclass Training Argument to our distillation's hyper parameter. Here, we create a compute metrics function to evaluate the model on the test set. During the training process, the accuracy and F1-score of our model will be calculated using the given compute function.

4.4 Hyper Parameter Search for Distillation Parameter

In the case of hyper parameter search for Distillation parameter alpha and temperature, Optuna is utilized. The specified parameter and temperature in the Distillation trainer are utilized as a hyper parameter search in order to maximize our "Knowledge Extraction". Hyper parameter optimization frameworks use Optuna, which is also accessible via trainer API to facilitate integration. Distillation Trainer is a subclass of the Trainer API, therefore we can directly use this without any code changes. When employing Optuna for hyper parameter optimization, hyper parameter space is necessary. In this example, we are attempting to optimize and maximize the number of train epochs, learning rate, alpha, and temperature parameters of our student model. To initiate a hyper parameter search, we need to invoke a hyper parameter search, which will provide hp-space and the number of trials to execute. Since we are using Optuna to determine the optimal hyper parameter, we must re-tune using the optimal hyper parameter from the optimal run. We have overwritten the default hyper parameter from the best run and trained them again.

5 Results

Table 1 depicts the results that are evaluated on the test set of the GLUE official benchmark. The best results for each group of student models are mentioned in bold. The architecture of TinyBERT4 and BERTTINY is (M = 2, H = 128, di = 1200), BERTSMALL is (M = 6, H = 512, di = 2048), and BERTSMALL is (M = 4, H = 512). All models are learned in a single-task manner. The inference speedup is

Table 1 Results obtained on the BERT models

Model	Parameter	Speedup	Accuracy on SST2 (%)
BERT-Base	109 M	1×	93
Tiny-BERT	4 M	47.5×	84.5
BERT-Medium (L = 8, H = 512)	59 M	–	92
BERT-Small (L = 4, H = 512)	29 M	–	90.89
BERT-Small (L = 6, H = 128)	11 M	–	86.4
BERT-Mini (L = 4, H = 256)	15 M	–	88.7

evaluated on a single NVIDIA K80 GPU. BERTBASE (N = 12, d = 768, di = 3072, and h = 12) is essentially used as the teacher model that contains the 109 M parameter. We have used g (m) = 3*m as the layer mapping function so that each Distillation model used as the student will learn every 3 of the BERT base model that are used for the Teacher model. The learning rate lambda for each layer is set to 1. To compare the Distillation student model, we also consider the same architecture for other models (DistllBERT6, TINYBERT6, etc.). Here, TinyBERT6 (M = 6, d 0 = 768, d 0 i = 3072, and h = 12) is the architecture followed. The Student model (Tiny BERT, etc.) includes the learning of General Distillation and Task-Specific Distillation. We have set the maximum sequence length as 128 for General Distillation and by utilizing the English Wikipedia (2500 M words) as the text corpus dataset and performing the intermediate layer distillation from pre-trained BERTBASE while also keeping other hyper parameters same as the BERT pre-training [1]. For Task-Specific Distillation, we consider fine-tuned BERT as supervision, where we first perform intermedia layer distillation on augmented data for 20 epochs with some specific batch size and a learning rate of 6e−5, after which we perform the prediction distillation on the augmented data 5 for 3 epoch for the batch size from 16, 32, and learning rate from 1e−5, 2e−5, 3e−5 on the development set. We consider the maximum length as 64 for single sentences and 128 for sequence pair sentence tasks in the case of Task-Specific Distillation. We can also boost the performance by a large amount by increasing the m-number of layers in the student model. Here for Tiny BERT, we use the google/bert-uncased-L-2-H-128-A-2 model which has layer 2, which means when we change our student model to other BERT distillation models, e.g. Distill BERT-base-uncased, it will perform better in terms of accuracy.

6 Conclusion

Although deep neural networks (DNNs) show their extraordinary power in various recognition, detection, translation, and tracking tasks, it is very challenging to deploy DNN models on resource-constrained IoT devices (networked embedded systems). The compute capability, network, and storage capacities of edge devices are relatively on the lower side. Therefore, there are research and practical efforts such as model partition, pruning, and quantization at the expense of small accuracy loss. Recently proposed knowledge distillation (KD) aims at transferring model knowledge from a well-trained model (teacher) to a smaller and faster model (student), which can significantly reduce the computational cost and memory usage and prolong the battery lifetime. In short, knowledge distillation is being touted as the way forward for optimizing NLP-centric models. 24 distinct BERT models can be considered for the distillation process as Teacher and Student models.

We have utilized some of them to test our proposed work and have demonstrated an improvement in performance metrics and total redundancy, which will allow us to become acquainted with the vast array of BERT-based experiments. The Tiny BERT student model achieves an accuracy of 84.5%, which is excellent for our model. Tiny BERT has over 96% fewer parameters than BERT-Base, and it runs approximately 47.5 times faster while retaining over 90% of BERT's performance as tested on the SST-2 dataset. After modifying the number of layers and adjusting the parameters, the performance of the medium and small BERT models increased by 92 and 91%, respectively, with a significant increase in speed. BERT Small with six layers and BERT Mini are fresh additions that perform better than previously adopted concepts and strategies. Compared to the BERT base model, the number of parameters in all of the student models used in the distillation process is reduced by several times, with minimal loss in precision.

References

1. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805v2 [cs.CL]
2. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV XLNet: generalized autoregressive pretraining for language understanding. https://doi.org/10.48550/arXiv.1906.08237
3. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V RoBERTa: a robustly optimized BERT pretraining approach. arXiv:1907.11692v1
4. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R ALBERT: a lite BERT for self-supervised learning of language representations. In: ICLR 2020. https://github.com/google-research/ALBERT
5. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv:1910.10683v3 [cs.LG]
6. Clark K, Luong M-T, Le QV, Manning CD (2020) ELECTRA: pre-training text encoders as discriminators rather than generators. arXiv:2003.10555v1 [cs.CL]
7. Wang A, Singh A, Michael J, Hill F, Levy O, Bowman SR (2019) GLUE: a multi-task benchmark and analysis platform for natural language understanding. arXiv:1804.07461v3 [cs.CL]
8. Kovaleva O, Romanov A, Rogers A, Rumshisky A (2019) Revealing the dark secrets of BERT. https://doi.org/10.18653/v1/D19-1445
9. Voita E, Talbot D, Moiseev F (2019) Analyzing multi-head self-attention: specialized heads do the heavy lifting, the rest can be pruned. https://doi.org/10.18653/v1/P19-1580
10. Han S, Mao H, Dally WJ (2016) Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. arXiv:1510.00149v5 [cs.CV]

11. Gong Y, Liu L, Yang M, Bourdev L (2014) Compressing deep convolutional networks using vector quantization. https://doi.org/10.48550/arXiv.1412.6115
12. Paraphrasing complex network: network compression via factor transfer—scientific figure on ResearchGate. https://www.researchgate.net/figure/The-structure-of-a-KD-Hinton-et-al-2015-b-FitNets-Romero-et-al-2014-c-AT_fig1_323184386. Accessed 22 July 2022
13. Learning cross-lingual phonological and orthagraphic adaptations: a case study in improving neural machine translation between low-resource, languages. https://www.researchgate.net/figure/The-transformer-architecture-as-described-in-Vaswani-et-al-2017-and-adapted-from-Li-et_fig4_335917224. Accessed 22 July 2022
14. Kim Y, Petrov P, Petrushkov P, Khadivi S, Ney H (2019) Pivot-based transfer learning for neural machine translation between non-English languages. arXiv:1909.09524v1 [cs.CL]
15. Sanh V, Debut L, Chaumond J, Wolf T, Face H (2020) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108v4 [cs.CL]
16. Pang B, Lee L Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales

AuthBlock: Authentication Framework Using Ethereum Blockchain

Sumedh Kamble and B. R. Chandavarkar

Abstract When employing authentication mechanisms to store user credentials, a subtle point to note is that they are easily vulnerable to cyber attacks like sharing of user data without their consent, password stealing on a large scale, etc. By decentralizing ownership of credentials and providing a framework for confirming one's record in an unalterable chain of data, i.e., Distributed Ledger Technology (DLT) in general and blockchain can provide a solution. Blockchain technology can help reduce the risk of attacks and user data leaks through backdoors by establishing a secure platform for online service providers to authenticate users without a single point of failure. Blockchain is being utilized increasingly for trusted, decentralized, secure registration, authentication, and valuation of digital assets (assets, real estate, etc.) and transactions, governing interactions, recording data, and managing identity among numerous parties. Smart contracts are used to do transactions on the blockchain. This work aims to analyze the shortcomings of traditional authentication systems and hence provide a blockchain-based authentication solution to address them. In this paper, we suggest AuthBlock, a robust, lightweight, and secure blockchain-based authentication system. It can be used by multiple parties as an authentication framework in parallel without any interference. The proposed approach leverages the Ethereum blockchain along with its provision of smart contracts. The proposed method is tested on the Ethereum localnet created using Go Ethereum (Geth) and evaluated to analyze user authentication, verification, and cost.

Keywords Blockchain · Authentication · Go Ethereum · Security · Smart contracts

S. Kamble (✉) · B. R. Chandavarkar
Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore 575025, India
e-mail: kamblesumedharvind.212is012@nitk.edu.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_9

1 Introduction

The basic idea behind authenticating someone or something is to verify whether they are what they claim to be. This is an online system or service's main procedure for allowing users to log in and establish their identities. Therefore, it is crucial to safeguard the authentication process to stop identity theft and spoofing attempts. Personal privacy concerns are addressed insufficiently or are outright ignored, even though there are several legal concerns related to the interchange of sensitive data elements.

Blockchain [14] is a decentralized, unchangeable database that organizes asset tracking and transaction recording in a business network. An asset can be anything from material to intangible things (like an idea, a patent, copyright, or a brand). Practically anything of value can be stored and transacted on a blockchain network, minimizing risk and working efficiently for all parties. Due to its decentralized feature, it helps to provide security and encryption [12]. Following are the steps for a transaction to get accepted into the blockchain:

- A participant inputs a transaction, which the miners should authenticate and validate the transaction using digital signatures.
- A block is created using these multiple transactions.
- This newly created block is broadcasted to all the participant nodes in the blockchain network.
- A block is appended to the latest copy of the blockchain after authorized nodes have verified the transaction. (Miners are considered to be special nodes in blockchain networks, which are often compensated for this computational labor, aka Proof of Work, or PoW [10]; for which they receive rewards in the form of cryptocurrencies.
- The transaction is completed when the update is delivered throughout the network.

In order to avoid possible cyber attacks like impersonation attacks, password spraying, credential stuffing, etc., it is crucial that the users must verify themselves using a strong authentication system [4]. User identification and authentication are made possible via blockchain-based authentication. The immutable blockchain ledger validates and guarantees the legitimacy of the users, transactions, and messages. Smart contracts that are created and deployed on blockchains are used to authenticate blockchains [11]. There is no longer a requirement for a third party to verify transactions. While security and privacy are considerably improved, costs can be decreased. In a distributed setting, much more effort would be put into hijacking the authentication process. Blockchain-based authentication allows for the storage of verification and encryption keys on the blockchain while signature and decryption keys remain on the device. It offers some defenses against serious cyberattacks, including phishing, man-in-the-middle, and replay attacks.

Traditional authentication systems work centrally, where an authentication server stores all the user credentials. Whenever a user is authenticated, its credentials are compared with those available with an authentication server. The Kerberos [7] protocol allows a user to authenticate themselves depending on the type of service the user intends to get. This process may/may not happen on a continuous basis depending on the requirement of the user. It is also dependent on the fact that there is a prolonged secret key shared between the user and the authentication servers. But being a central system, it has its drawbacks. Password guessing, synchronization of clocks, continuous availability of Key Distribution Center (KDC), etc., are a few among them. The use of blockchain helps in mitigating these issues and provides a more secure, available, and flexible solution.

In this paper, AuthBlock, a decentralized authentication framework, is introduced. The proposed approach is developed using the Ethereum blockchain platform and its smart contract functionalities. The proposed method is evaluated on Ethereum localnet created using Go Ethereum (Geth) and analyzed user authentication, verification, and cost. It can be used by multiple parties as an authentication framework in parallel without any interference. The main contributions of the paper are stated as follows:

- A brief overview of shortcomings of traditional authentication system.
- Create a local Ethereum blockchain and design a decentralized blockchain-based authentication framework on top of it.
- Use Ethereum and its smart contracts developed in Solidity programming language to implement the system.
- Rigorous testing of the proposed system on the local Ethereum blockchain to verify user authentication, cost (ETH or any currency).

The rest of the paper is organized as follows. Section 2 briefly describes the components and the working of blockchain. Section 3 describes the architecture of AuthBlock, using blockchain. Section 4 describes the implementation of AuthBlock using the Ethereum blockchain. Section 5 explains the results of the system. Section 6 concludes the work with the scope for future work.

2 Working of Blockchain

Figure 1 shows the overall architecture of Blockchain. It consists of three important entities: blocks, nodes, and miners.

2.1 Blocks

Each block in a chain is made up of three basic components:

- The information to be stored in a block.
- A 32-bit whole number called the nonce. When a block is made, the nonce is randomly generated, which produces the hash of the header. Hash is a 256-bit alphanumeric string that is attached to the nonce. It has to start with a preset amount of zeroes.

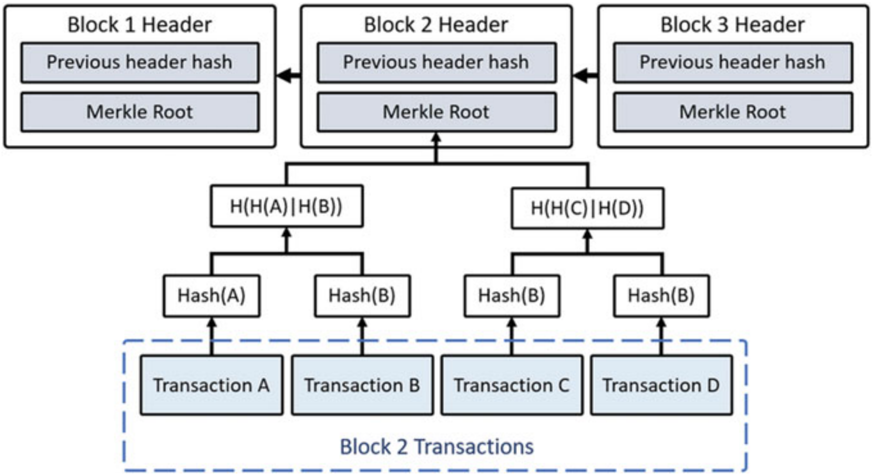


Fig. 1 Blockchain architecture [14]

- The cryptographic hash is produced by a nonce. Before mining, the data in the block is regarded as signed and permanently tied to the nonce and hash.
- **The Merkle tree** [5] inscribes the data on the blockchain in an orderly and secure manner. It enables us to quickly verify blockchain data, as well as the fast transfer of large amounts of data between nodes on the blockchain network.

2.2 Miners

- Mining is the task by which special nodes add new blocks to the blockchain [2].
- Each block has its unique nonce and hash; however, it also refers to the hash of the block just before it is in the chain, making it difficult to mine a block. Longer the chain more difficult it is to add a new block. The complexity of this work is set to keep the rate of new block generation (in the Bitcoin blockchain) at approximately one every ten minutes. However, it may differ from other blockchain architectures [13].
- Miners use specialized software to solve the computationally expensive arithmetic problem of finding a nonce that generates a good hash. It takes approximately 4 billion nonce-to-hash combinations to find the correct one because the hash is 256 bits; however, the nonce is only 32 bits. Once the miners declare that they have found the “golden nonce,” their block is included in the blockchain.

2.3 Nodes

- One of the essential concepts of blockchain technology is decentralization. No computers or businesses are allowed to own chains. Instead, it functions as a distributed ledger among the chain's linked nodes. Any electrical device that maintains a copy of the blockchain is called a node [8].
- Each node has its own local copy of the blockchain, and the network must algorithmically approve each newly added block to update, trust, and validate the chain. The transparency of the blockchain makes it easy to see and view all the actions in your ledger. Each participant receives a unique alphanumerical identification number that indicates the transaction.

In a blockchain network, two nodes can exchange data or information (transaction), and the blockchain network verifies the reliability of the transaction. Next, multiple validated transactions create the block. Now the node tries to insert this newly created block into the main blockchain. Adding new blocks to the blockchain is done through a process called Proof of Work. This is an incentive-based system in which nodes (also known as miners) solve mathematical puzzles of a certain difficulty level. Participating nodes need high computing power to run this process. Therefore, mining is a costly issue.

Once the node successfully resolves this mathematical problem, a new hash value is created for the block in question. As more and more people get the correct answer for the same block, it will be permanently added to the main blockchain. Each node receives a reward (in the form of a cryptocurrency such as Bitcoin) to solve this puzzle. Each block in the blockchain has a block number and a timestamp in the order in which it was added to the blockchain [2]. Also, each block is added to the previous block by hashing. The hash gives each block a unique number that acts as its digital signature. This makes the blockchain very secure.

There are primarily two methods to evaluate the accuracy of any blockchain application. The first step is to completely rewrite the blockchain code, execute it on a local system, and check for accuracy. The second is to test your results by running your application on any blockchain simulator. The latter way makes our work simple, whereas the former is a more complex method. Because of this, just as with conventional networks, we have blockchain simulators where we can test our application before launching it on the mainnet.

3 Architecture Design of AuthBlock

As shown in Fig. 2 the user can register to use the application. During registration, the user has to provide details like username, phone number, Adhaar ID number, and password. The role is assigned as $USER$, and the application ID for the respective application is automatically added. Basic validation of these details is done before forwarding. These details are then forwarded to the smart contract using

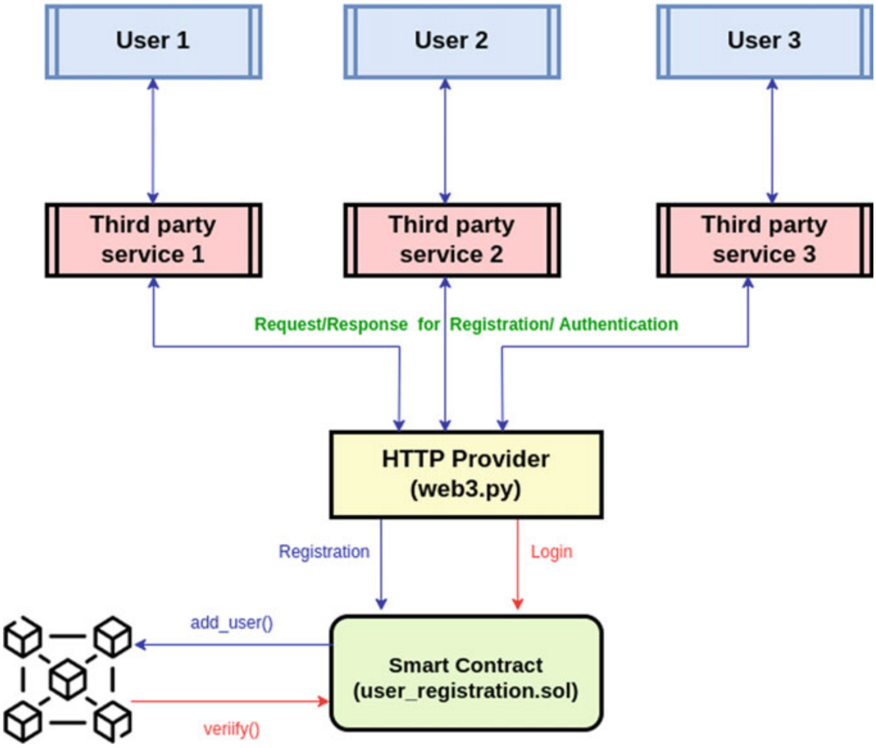


Fig. 2 Proposed system

the HTTP Provider of the blockchain. This HTTP Provider is an interface between blockchain/smart contract and user interface. A transaction is then built using these details, signed with the default account's private key, and sent to the blockchain. In smart contract, there is function $add\_user()$ which takes the user details as parameters. When the transaction reaches the blockchain nodes, it is mined by nodes as a block, and user details are stored on the blockchain. Once these details are stored on the blockchain, the user is registered. After registration, only he can use the application.

When the user attempts to Login into the system, the username is used to retrieve its details. In the smart contract, there is a function $verify()$, which takes the username as a parameter and verifies the user. Once the hash of entered password matches the stored hash, the user is verified, and only then can he use the system.

Once the transaction is built and sent to the blockchain, it is added to the transaction pool. The transaction pool is the group of all transactions that have been submitted but not yet included in a block. When you confirm this transaction, it is not immediately performed but instead becomes a pending transaction and is added to it. Sending a transaction on a blockchain requires a small fee. The application can choose how much it wants to pay—even 0 ETH (gas price) in our system. It depends on whether

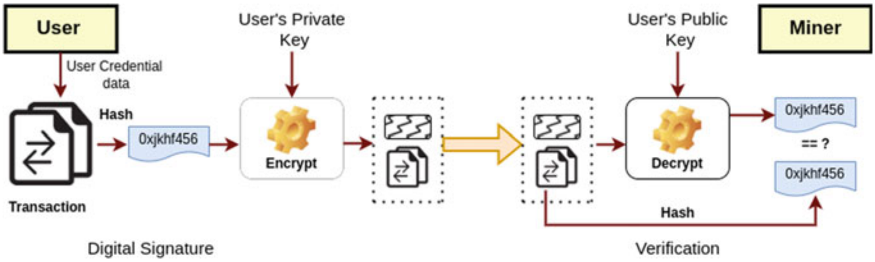


Fig. 3 Transaction workflow

or not a transaction gets included in a block and processed depending on supply and demand. Gas price is directly proportional to the time required for confirmation of transaction [9] i.e., the higher the gas price higher the priority of the transaction, and more quickly it will be executed, i.e., mined on blockchain. One of the crucial concepts of blockchain is decentralization. Any computer can act as a node. Hence nodes can be any number of machines from a service provider. Whenever a new service provider joins the network, it must add some blockchain nodes. This increases the distributed nature of blockchain, making it more secure and hard to tamper with.

3.1 Working of AuthBlock

Due to the provision of the PoW process, the need to include a third party in a two-party transaction is completely removed, thereby establishing a whole new level of trust between the unknown and untrusted stakeholders. A third-party intermediary is required otherwise to produce trustworthy records and transactions. Blockchain may serve as a complete replacement to the existing centralized entities by building trust between stakeholders through cryptographic security measures and cooperation.

In this system, during registration, the user enters their data (username, password, Aadhaar ID, etc.). A transaction is created for this data. Public key cryptography proves that a specific organization creates the transaction. Each organization will be assigned $< PU_o, PR_o >$, where PU_o is the public key, and PR_o is the private key of the organization. The private key is kept in a digital wallet in the blockchain framework. For registration of users, i.e., to complete the transaction, it is signed using the organization's private key, i.e., a digital signature is created. This digital signature will be transmitted along with the transaction to the blockchain. The public key is used to validate that the message is received from the same user. As shown in Fig. 3, the user data is hashed in hash value H_1 and then signs H_1 using the PR_o to generate the digital signature. The digital signature and transaction together are broadcasted to the blockchain network. The miner makes use of the organization's PU_o to decrypt the received digital signature to get the hash value say H_2, and the hash of the transaction is also calculated of the transaction, say, H_3. Then the miner verifies if H_2 is equal to H_3 or not. The miner verifies the transaction and starts the mining procedure if they are the same. $user\_registration.sol$ smart contract is used to store the data on the blockchain. This way, the user is registered onto the blockchain for the specific organization.

For authentication, when a user enters their username and password, the smart contracts have a function $nameToPassword()$, which will retrieve the hash of the password for that username. This hash will be matched with the entered password's hash to authenticate. After this authentication, only the user will be logged on to the subsequent system of that organization.

4 Implementation

To implement the user authentication system, first, we need a blockchain that can be a local blockchain or any publicly available test nests. Go Ethereum is used in this implementation, which is an official implementation of the Ethereum blockchain.

4.1 Go Ethereum: Geth

Along with C++ and Python, Go Ethereum [6] is one of the three original implementations of the Ethereum protocol. It is entirely open source and developed in Go. Go Ethereum is available as a library that you can embed in Go, Android, or iOS projects or as a standalone client called Geth that can be installed on just about any operating system. A machine that is running Geth becomes an Ethereum node. In the peer-to-peer Ethereum network, data is shared directly between nodes instead of controlled by a central server. Because they are compensated in ether, Ethereum's native token, nodes compete to create new blocks of transactions to deliver to their peers (ETH). Each node verifies a new block upon getting it. Chain is the term used to describe the arrangement of distinct blocks. Geth uses the data in each block to update its "state," or the ether balance of each Ethereum account.

After installing Geth, a *genesis.json* has to be created, which would act as our first block in the blockchain. It defines the data in the first block of a blockchain, as well as rules for the blockchain itself, viz., difficulty, mining algorithm, pre-funded accounts, etc. Blockchain is then initialized using this genesis file using the command: *geth init—datadir data genesis.json*. It will create a directory called "data," which will contain the blockchain data for this node. Also, it will have a Keystore that will save all the addresses and private keys for the accounts created in this node. After initialization nodes can be created using command: *geth—datadir data—networkid 12345*. The same process is followed to create multiple nodes on the same or different. We have to use the same genesis file for initialization for each node to ensure that the new node gets added to the same blockchain. After creating multiple nodes, these nodes have to be connected as peers of each other. This can be done using

Geth command: *admin.addPeer(enodeid)*. This has to be done for each node using peers' enode ID, which is a unique id of a node in the blockchain. Once the nodes as added as peers, the blockchain starts to sync. Mining can be started using Geth command: *miner.start(noOfThreads)*. Once mining is started, the etherbase accounts on the nodes will be funded with ETH as a reward for mining. Once blocks are mined on a particular node, they will be propagated to other nodes in the blockchain. Geth also provides a JavaScript console that interacts with the blockchain.

4.2 Smart Contracts

Smart contract(s) is/are a self-executing code in which the conditions of the settlements between the two parties are programmed. The settlements and their programmed code are made available throughout a decentralized blockchain network. Transactions are transparent and peremptory, and the code governs their execution.

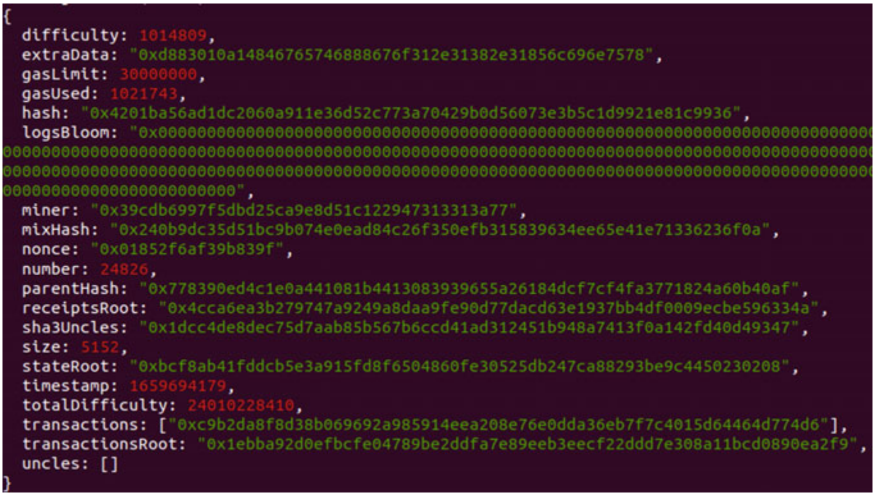
Smart contract is written in Solidity [3] programming language. To deploy a smart contract, first, it has to be compiled. For interacting with smart contracts, the $py - solc - x$ library is available in Python. Smart contract was compiled using this library, which is a Python wrapper for the $solc$ Solidity compiler. To interact with the blockchain, the $Web3.py$ library is used. It is a Python library for interacting with the Ethereum blockchain. Using this program, the smart contract was deployed on the Geth blockchain. Smart contracts are deployed on blockchain as a normal transaction. After the compilation of the contract, the transaction has to be built and signed using the user's PR_o, and then the transaction is broadcasted to the blockchain network. This transaction is then mined, and the smart contract is stored on the blockchain.

User details can be stored on the blockchain using this smart contract. The blockchain stores details like username, user ID, password, Aadhaar number, and role. This smart contract has functionalities like adding users and verifying user passwords through blockchain. While registering user has to enter the details, which are encrypted and then sent to the blockchain where this smart contract is executed, adding the user details to the blockchain.

The same smart contract can be used to verify the user. User ID or username can be used to verify the user using its password. Other functionalities like password change and retrieving all the user details can be executed.

5 Results

The smart contract has to be deployed on the blockchain. For transactions on the blockchain, we require gas. The same contract should use the same amount of gas deployed on a blockchain [1]. Gas is the price incurred by using one unit of gas. To calculate the cost of conducting an operation, we multiply the gas price by the amount used.



```
{
  difficulty: 1014809,
  extraData: "0xd883010a1484676574688867 6f312e31382e31856c696e7578",
  gasLimit: 30000000,
  gasUsed: 1021743,
  hash: "0x4201ba56ad1dc2060a911e36d52c773a70429b0d56073e3b5c1d9921e81c9936",
  logsBloom: "0x0000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000",
  miner: "0x39cdb6997f5dbd25ca9e8d51c122947313313a77",
  mixHash: "0x240b9dc35d51bc9b074e0ead84c26f350efb315839634ee65e41e71336236f0a",
  nonce: "0x01852f6af39b839f",
  number: 24826,
  parentHash: "0x778390ed4c1e0a441081b4413083939655a26184dcf7cf4fa3771824a60b40af",
  receiptsRoot: "0x4cca6ea3b279747a9249a8daa9fe90d77dacd63e1937bb4df0009ecbe596334a",
  sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
  size: 5152,
  stateRoot: "0xbcf8ab41fddcb5e3a915fd8f6504860fe30525db247ca88293be9c4450230208",
  timestamp: 1659694179,
  totalDifficulty: 24010228410,
  transactions: ["0xc9b2da8f8d38b069692a985914eea208e76e0dda36eb7f7c4015d64464d774d6"],
  transactionsRoot: "0x1ebba92d0efbcfe04789be2ddfa7e89eeb3eecf22ddd7e308a11bcd0890ea2f9",
  uncles: []
}
```

Fig. 4 Deployed smart contract on blockchain

$$\text{Total Cost} = \text{Gas Price} * \text{Amount of Gas Consumed}$$

Figure 4 shows that to deploy the contract, we require 1021743 units of gas. Transaction costs are based on the cost of sending data to the blockchain. Execution costs are based on the cost of computational operations, which are executed as a result of the transaction. So its total cost has to be calculated in terms of Ether (ETH) or any currency. The gas price at the time of writing this paper was 6 $gwei$. Table 1 shows the total cost to deploy our smart contract and the cost of important functions in it. This cost will be incurred each time we do the operation on the blockchain.

In traditional authentication systems, authentication credentials are stored on authentication servers, i.e., authentication is dependent on third-party/trusted authority. Also, each organization or each app has to use its authentication system, which leads to increased requirements of resources and cost. So there was a need for a framework that would work as a single system for multiple organizations. Along with this, the system should be scalable, secure, and flexible. As the proposed system is implemented using blockchain, it inherits the advantages of blockchain. There is no need for centralized servers. No third-party/trusted authority is required as the blockchain will ensure that the trust lies in the system. Even though the nodes in the blockchain will be distributed over multiple organizations, the data on the blockchain will be secured and cannot be tampered with. The user credentials stored on the blockchain will be encrypted and available only to its respective organization.

With the help of blockchain technologies, the system can be protected against issues with authentication, data integrity, and non-repudiation. First, specific modifiers are available in the solidity programming language that can authenticate the

Table 1 Smart contract deployment and execution costs

Functions	Gas price	Ethers	INR
contract creation	6	0.00701	926.93
adduser()	6	0.00143	189.16
retrieve()	6	0.00024	31.8
nameToPassword()	6	0.000155	20.48
userIdToPassword()	6	0.000154	20.44

user. Second, blockchain is unchangeable since no data stored there can be altered. After execution, the details of the transaction and function cannot be replaced. Hence though multiple service providers are using the identical blockchain, the user data is secured. They can access only their data using *app_id*. Also, in the proposed framework, the smart contract can be customized according to the need of user data. Each of them can use their smart contract deployed on the identical blockchain, thus making it flexible for multiple service providers.

6 Conclusion

Because of the centralized authentication system, administrators in businesses have problems with authentication and security. Being a centralized system, it is prone to severe cyberattacks, including phishing, man-in-the-middle, and replay attacks. In this paper, AuthBlock, an architecture that uses the Ethereum blockchain to efficiently manage and authenticate the users, has been proposed. The proposed method creates a local blockchain and an authentication framework on top of it. It uses Ethereum smart contracts to construct the intended system logic, i.e., to verify the user and give or restrict access based on the user's roles. Furthermore, we have evaluated the performance of the proposed solution. AuthBlock enables user identity verification, security, and flexibility thanks to the distributed nature of blockchain technology.

References

1. Ali N (2022) Smart contract and transaction fee on Ethereum
2. Aljabr AA, Sharma A, Kumar K (2019) Mining process in cryptocurrency using blockchain technology: bitcoin as a case study. J Comput Theor Nanosci 16(10):4293–4298
3. Bhattacharya D, Canul M, Knight S, Azhar MQ, Malkan R (2019) Programming smart contracts in ethereum blockchain using solidity. In: Proceedings of the 50th ACM technical symposium on computer science education, p 1236
4. Chen Y, Bellavitis C (2020) Blockchain disruption and decentralized finance: the rise of decentralized business models. J Bus Ventur Insights 13(e00):151

5. Chen YC, Chou YP, Chou YC (2019) An image authentication scheme using Merkle tree mechanisms. Futur Internet 11(7):149
6. Dhulavvagol PM, Bhajantri VH, Totad S (2020) Blockchain Ethereum clients performance analysis considering e-voting application. Procedia Comput Sci 167:2506–2515
7. El-Emam E, Koutb M, Kelash H, Allah OF (2009) An optimized Kerberos authentication protocol. In: 2009 international conference on computer engineering & systems. IEEE, pp 508–513
8. Elrom E (2019) Blockchain nodes. In: The blockchain developer. Springer, pp 31–72
9. Garreau M (2022) How does a transaction get into blockchain? Accessed 22 Jul 2022
10. Gemeliarana IGAK, Sari RF (2018) Evaluation of proof of work (pow) blockchains security network on selfish mining. In: 2018 international seminar on research of information technology and intelligent systems (ISRITI). IEEE, pp 126–130
11. Ismail R (2017) Enhancement of online identity authentication though blockchain technology
12. Kosba A, Miller A, Shi E, Wen Z, Papamanthou C (2016) Hawk: the blockchain model of cryptography and privacy-preserving smart contracts. In: 2016 IEEE symposium on security and privacy (SP). IEEE, pp 839–858
13. Vukolić M (2015) The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In: International workshop on open problems in network security. Springer, pp 112–125
14. Zheng Z, Xie S, Dai H, Chen X, Wang H (2017) An overview of blockchain technology: architecture, consensus, and future trends. In: 2017 IEEE international congress on big data (BigData congress). Ieee, pp 557–564

Vision-Based Driver Assistance System to Detect Mountainous Roads Using GLCM and Haralick Texture Features

Jyoti Madake, Akshay Vyas, Vaibhav Pawale, Vibha Chaudhary, Shripad Bhatlawande, and Swati Shilaskar

Abstract Steep roads with potholes, irregular slopes, and other barriers affect vehicle safety and accident prevention in mountainous areas. The paper proposes a computationally efficient computer vision and machine learning-based approach to detect these roads. The dataset includes damaged, steep, and narrow mountainous roadways. The Field of View (FoV) creation step in preprocessing increased the algorithm's accuracy. The five statistical texture parameters—Entropy, Correlation, Homogeneity, Contrast, and dissimilarity—are extracted using the Gray-Level Co-Occurrence Matrix (GLCM) technique. The accuracy of various combinations of features, orientations, and distances varied. Haralick's five horizontal, diagonal, and vertical GLCM-based features provided 90.95% accuracy. Experimentally, the Light Gradient Boosted Machine (LGBM) classifier predicted mountain roads most accurately.

Keywords Road detection · FoV · GLCM · Haralick · LGBM

J. Madake (✉) · A. Vyas · V. Pawale · V. Chaudhary · S. Bhatlawande · S. Shilaskar
Vishwakarma Institute of Technology, Pune 411037, India
e-mail: jyoti.madake@vit.edu

A. Vyas
e-mail: akshay.vyas19@vit.edu

V. Pawale
e-mail: vaibhav.pawale19@vit.edu

V. Chaudhary
e-mail: vibha.choudhary19@vit.edu

S. Bhatlawande
e-mail: shripad.bhatlawande@vit.eud

S. Shilaskar
e-mail: swati.shilaskar@vit.edu

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_10

1 Introduction

Nowadays, tourism in hill stations, canyons, and mountainous areas has increased. Regular transportation for food, medicine, and daily needs is required in the mountainous region. In such hilly areas, the construction and development of good-quality roads are quite difficult, which automatically affects transportation. New drivers passing through such areas face fatal and threatening routes, which can sometimes cause accidents. Hence, reducing the worst effects of congestion on slender roads and in mountainous areas is important. There has been rapid progress in autonomous vehicles and driverless Advanced Driver Assistance Systems (ADAS). Detection of mountainous roads featured by the ADAS system will be helpful for autonomous vehicles traveling in mountainous regions. Identification of mountainous roads can be of great value for transportation systems, tourism, and automatic vehicles, and it will also ensure the safety of drivers as well as tourists. This paper describes a GLCM-based model capable of detecting mountainous roads, which can be integrated with the driver assistance system to make the vehicle driver aware of road conditions. Analyzing the texture features of such mountainous roads can be helpful for better prediction. This paper presents a model that extracts the minimum features and requires less computational time than other descriptors.

2 Literature Review

Several tries have been made to expand a technique for reading street properties by the usage of a combination of images from automobile cameras and image processing techniques to better effectively investigate a mountainous surface. Many computer vision researchers are now concerned about drivers' safety and help to improve it by identifying different road characteristics.

The damaged roads [1] database is created manually by collecting smartphone images, and a comprehensive street damage dataset is composed, consisting of 9053 damaged road images captured using a mobile launched on a vehicle. The images are classified into 8 classes: wheel marks, longitudinal construction joint, lateral cracks, alligator cracks, potholes, crosswalk blur, and white line blur. This model is based on the modern method of Convolution Neural Network (CNN). The model is trained using the MobileNet Single Shot Detector (SSD) and SSD Inception V2 frameworks. Results obtained from SSD Inception V2 and SSD MobileNet are also compared. Cheng et al. [2] introduce an effective method of measuring the width of the route using clouds available through LiDAR-based Mobile Mapping System (MMS). They used a MMS platform equipped with four laser scanners, three cameras, and Global Navigation Satellite System (GNSS)/INS to collect Light Detection and Ranging (LiDAR) clouds and RGB images across two Interstate highways in the United States. Vehicles use a web camera, laser telemeter, Inertial Moment Unit (IMU), Differential Global Positioning System (DGPS), and other sensors to detect potholes and do road quality assessments [3]. Both functions are performed using the above-mentioned components, and each has a different precision than a standard road and an access road. Goyal et al. suggest the design of an invalid road acquisition algorithm using Arduino [4]. Other components include the accelerometer ADXL335, ESP8266-01, and NEO-6M GPS. The system uses Euler viewing angles to stabilize the accelerometer figures obtained from the Arduino device by establishing the allowed angle. Usage of a CNN-based single-dimensional image measurement technique [5] to measure the depth of an inclined RGB image clicked by a Kinect sensor mounted on the robot, and the picture element coordinates of the top, as well as bottom sides of the slope, are achieved by Soble edge extraction. Feature extraction is done using CNN after segmentation of the image into multiple pixels by Simple Linear Iterative Clustering (SLIC). Two HC-SR04 Ultrasonic Modules (USM) were used at different heights for the use of slope detection novels and computational methods. The model helps to detect positive and negative slopes and calculates the inclination and declination angles from a proper distance. Two slope sensing techniques, named Static Platform and Rotatable Platform, are used, and slopes are calculated using trigonometric calculations [6].

Extraction of road surfaces is carried out with the combination of Linear Binary Pattern (LBP) and GLCM [7] to upgrade the robustness of features. The process includes extracting ROI, converting RGB to Grayscale, applying the feature extractors which extract the minimum features, and the classification is carried out using KNN. The features of the GLCM used are ‘Energy’ and ‘Contrast’. Different land types such as arable land, forest land, residential land, and water are classified using the SVM classifier after the texture features of each region have been extracted using various GLCM features [8]. They have carried out SVM classification on multiple datasets like GaoFen-2 Data, QuickBird Data, and GeoEye-1 Data and analyzed the accuracy for all. Classification accuracy for those 3 types of satellite images is 92.43%, 93.26%, and 96.75%, respectively.

A system for the recognition of large vehicles [9] on minor roads is proposed, which includes background and shadow removal, detection of moving objects, etc. The purpose of the plan is to identify road signs [10] to ensure safety as accidents have been increasing due to the ignorance of drivers. When using it, consider the Region of Interest (ROI) and the Gaussian smoothness used for noise reduction.

Cai et al. estimated the slope value using a geometric algorithm and the edges are detected with Canny Edge detection. After major photo editing, a Support Vector Machine (SVM) and Gaussian Mixture Model (GMM) are used to spot and draw out traffic details, and the outcomes are fully tested [11]. A model for finding routes in hilly areas [12] has been developed, and in the preprocessing, they removed the noise and other external objects such that the FoV can be selected. The edges are detected with a canny edge detector and the Hough Transform is used to detect parameter curves. A vision-based dip detection method [13] helps to decide the inclination to use the Canny Edge Detection method in the short term. This paper focuses on slope measurement by detecting the bending angle of lanes. The mathematical model is developed using route information and a canny algorithm with a Hough transform to convert a curve into a parameter space [14]. An Unmanned Aerial Vehicle

(UAV)-based dip surveillance system [15] is introduced with two major functions: border identification and swarm-intelligence-based route designing. Canny and U-Net see boundaries, ruptures, and rocks. In the paper [16], the authors, Sivaraman and Trivedi, provide research conducted on vision-based vehicle detection using features such as Histogram of Oriented Gradient (HOG) and Haar. These properties are extremely well described in the car recognition literature. The Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm [17] is used to improve the image quality for mountain outlines and low-value images. It focuses on high-quality feature extraction for mountainous regions, as these regions are mostly cloudy throughout the year.

A monocular camera was obtained in the study, and the slope measurement was performed in three ways, namely Geometric-Based (GB), Covariance-Based (CB), and location-based factors (LFB). The GB approach works exceptionally well on upright lanes and curves with large radii on rough roads, whereas the CB method works for complex roads [18]. The longitudinal velocity is measured to confirm the inclination of the road. And while in comparison to the automobile kingdom in veDYNA [19], which may be concluded at a longitudinal pace because of the layout of the viewer, one layout has excessive accuracy, and the other has quick monitoring speed.

3 Methodology

This paper proposes a mountainous road detection system that uses GLCM for texture feature extraction. A combination of 18 features has been used with different directions, distances, and features. The classification process in the proposed model uses LGBM which is a Decision tree-based gradient-boosting framework. It helps in increasing efficiency and minimizing memory usage. All the steps which are carried out are depicted in Fig. 1.

3.1 Dataset

Collecting meaningful data and building the dataset is the most challenging task while developing a machine learning model. A whole dataset of about 8,000 images is prepared by capturing the images of mountainous roads using a smartphone and extracting some images from the mountainous road trip videos. To our knowledge, there is no dataset available for the mountainous roads. For the first time, we have collected such data from various routes in rural and hilly areas and extracted it from the videos available on the Internet.

Some of the road images from the Internet that we used for this dataset are the Leh-Manali Highway, Rohtang Pass, Karakorum Highway, and Skippers Canyon. Other images are collected manually from some mountainous regions of Maharashtra

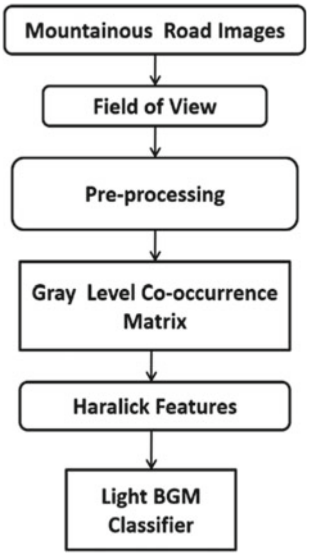


Fig. 1 Flowchart of the proposed algorithm

(India). Those images were captured by the smartphone on the hilly roads around Pune and Nanded (Maharashtra).

3.2 Preprocessing

The main step of preprocessing consists of extracting the Field of View (FoV) of each image so that the features obtained are fewer and the computational time required becomes less, resulting in better accuracy of the model.

The resolution of the image in Fig. 2 is 3149 × 2362. All the images of such different sizes are cropped as shown in Fig. 3 to get the region of interest and then resized to the same size of 300 × 300. The full process can be seen in Fig. 4 to understand how preprocessing is carried out to get the required image.

The next step is the grayscale conversion, in which a color image three-dimensional array is converted into a two-dimensional array. In this process, 24-bit RGB values get converted into 8-bit grayscale values. The mean of red, green, and blue pixel values is taken for each pixel to get the grayscale value, so it has only one layer from 0 to 255, whereas an RGB image has 3 layers. Another reason for this conversion is to gather the data to be used for the GLCM algorithm. The formula for the same has been given in (1):

$$G(i, j) = \frac{R(i, j) + G(i, j) + B(i, j)}{3} \tag{1}$$

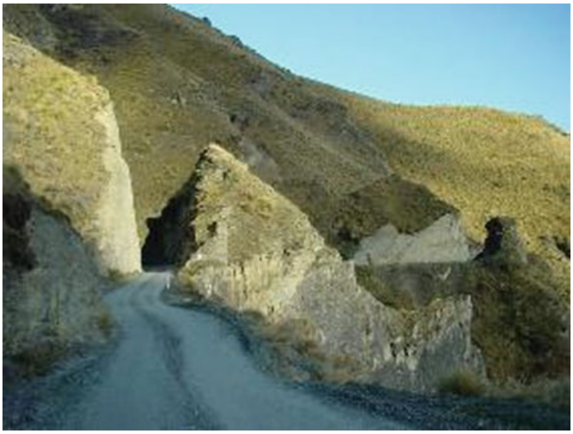


Fig. 2 Original mountainous road image (sample)

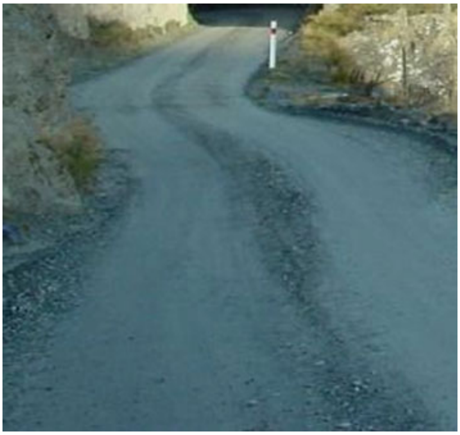


Fig. 3 Field of View (FoV) extraction

In this case, *R*, *G*, and *B,* all refer to the colors red, green, and blue. *i* is the row and *j* is the column.

3.3 Feature Extraction Using GLCM

GLCM returns a square matrix after calculating the Haralick features of the image. It is a second order statistical texture analysis method that looks at two pixels (i, j) and the relationships among them, then it defines how often those pairs of pixels are present in an image in the given direction (Θ) and distance (d). It returns the matrix which contains the gray value relation given by a kernel mask. When the calculation of pairs of pixels is carried out, the texture of an image is characterized by functions of GLCM, and the matrix gives the statistical measures.

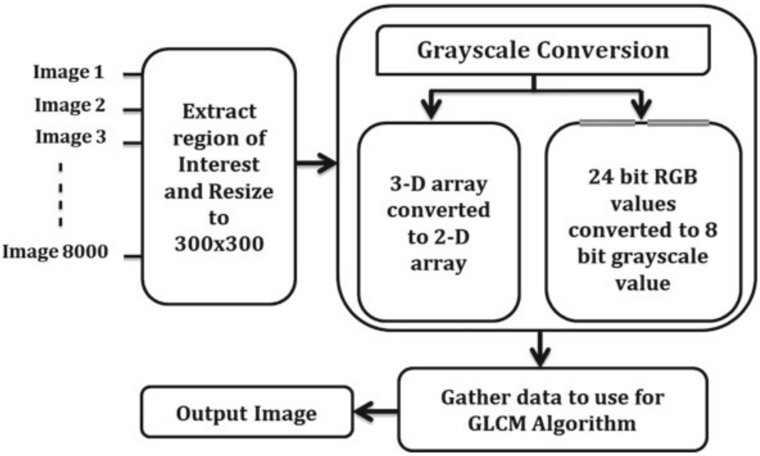


Image 1
Image 2
Image 3
Image 8000
Extract region of Interest and Resize to 300x300
Grayscale Conversion
3-D array converted to 2-D array
24 bit RGB values converted to 8 bit grayscale value
Gather data to use for GLCM Algorithm
Output Image

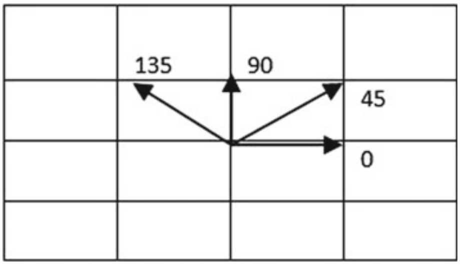
Fig. 4 Preprocessing

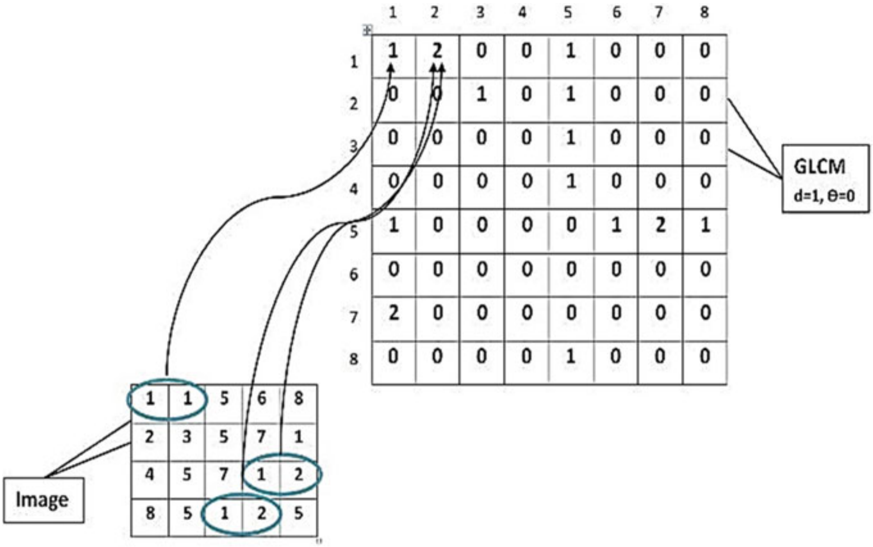
In this model, the different combinations of features, angles, and distances are implemented to extract the features. Energy, Correlation, Entropy, Homogeneity, and contrast are some of the Haralick features extracted from co-occurrence matrices. Angles are 0, 45, 90, 135, and 180 as in Fig. 5. The direction of the spatial relationship is horizontal for the $\Theta = 0$, vertical for $\Theta = 90$, and diagonal for the case of 45 and 135. The value of $d = 1$ represents that the pair of pixels is 1 pixel away likewise. The work of the distance parameter is shown in Fig. 6.

Energy Feature: The Sum of squared elements is returned by the Energy in GLCM, and the range is [0,1] where 1 is for the constant image. It is also known as uniformity or angular second movement. The formula for the Energy feature is given below in Eq. (2):

$$\text{Energy} = \sum_{i,j=0}^{N-1} (p_{ij})^2 \tag{2}$$

Fig. 5 Working of directions in GLCM





	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	1
6	0	0	0	0	0	0	0	0
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

Fig. 6 Working distance in GLCM

Contrast Feature: The measure of the Intensity contrast between two neighboring pixels over the image is returned by Eq. (3):

$$\text{Contrast} = \sum_{i,j=0}^{N-1} p_{ij(i-j)^2} \tag{3}$$

Homogeneity Feature: Eq. (4) measures the closeness of distribution of elements in the GLCM to diagonal:

$$\text{Homogeneity} = \sum_{i,j=0}^{N-1} \frac{p_{ij}}{1 + (i - j)^2} \tag{4}$$

Correlation Feature: The probability occurrence of the pairs of pixels is returned by the feature using the below Eq. (5):

$$\text{Correlation} = \sum_{i,j=0}^{N-1} \frac{(i - \mu)(j - \mu)}{\sigma^2} \tag{5}$$

Entropy Feature: Formula (6) for entropy is used for characterizing the texture of the input image:

Table 1 Sample of extracted features

GLCM features	Image 1	Image 2	Image 3	Image 4
Energy	0.2295	0.0306	0.0166	0.0479
Correlation	0.8804	0.9485	0.9198	0.9271
Dissimilarity	14.136	5.9891	13.272	14.537
Homogeneity	0.1536	0.2935	0.1533	0.1926
Contrast	696.95	129.02	529.58	806.24
Entropy	7.3246	7.0834	7.4653	7.1506

$$\text{Entropy} = \sum_{i,j=0}^{N-1} -\ln\ln(p_{ij})p_{ij} \tag{6}$$

where

p_{ij} Element *i,j* of the normalized histogram counts or matrix.
N Number of levels.
μ mean.
σ^2 variance.

The performance of the model is measured multiple times using the Haralick texture features of GLCM. In the beginning, energy, correlation, homogeneity, dissimilarity, and contrast are calculated with neighboring angles of 0, 45, and 90 and the distance as 0, 1, 2, and 5. After computing the features in the first method, an accuracy of 89.57% is achieved. Using the same features, the angle is changed from 0 to 135 to find the relation between pixels diagonally. In that case, 90.95% accuracy was obtained. Then, after reducing some features, an accuracy of 88.65% and an accuracy of 87.45% is seen for 4 and 3 features, respectively. Feature values of some of the images are shown in Table 1.

The next and most important step is using the Light GBM classifier for training the data over 100 epochs. The combinations formed with different GLCM features, directions, and distances are used. Light GBM used 18 features and gave a training score of 0.8957. After training the model on 6000 images with LGBM, that model is used for the prediction of mountainous and non-mountainous roads. The steps in the prediction are shown in Fig. 7.

3.4 Classification

Light GBM is one of the Gradient Boosted Decision Tree (GBDT) algorithms, which is well known for its higher efficiency, low memory usage, faster training speed, better accuracy, and capability of handling large-scale data. LGBM's performance is improved by leaf-wise tree growth. It converts the image features extracted using

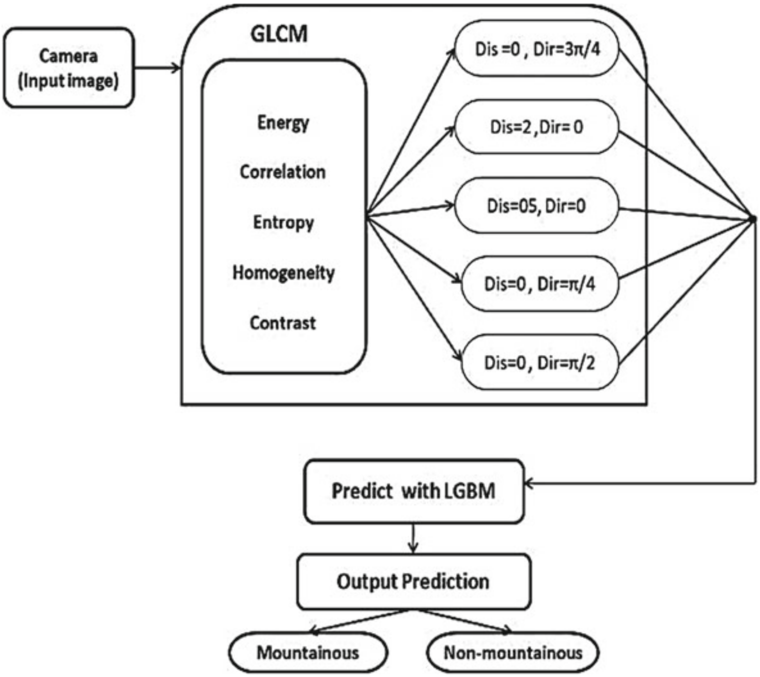


Fig. 7 Feature extraction using combinations of distances and directions

GLCM to an integer and trains the data over 100 epochs using the parameters to boost the computational time and accuracy. Parameter tuning is shown in Fig. 8. The boosting type is the main parameter to decide the boosting technique to be used in the model.

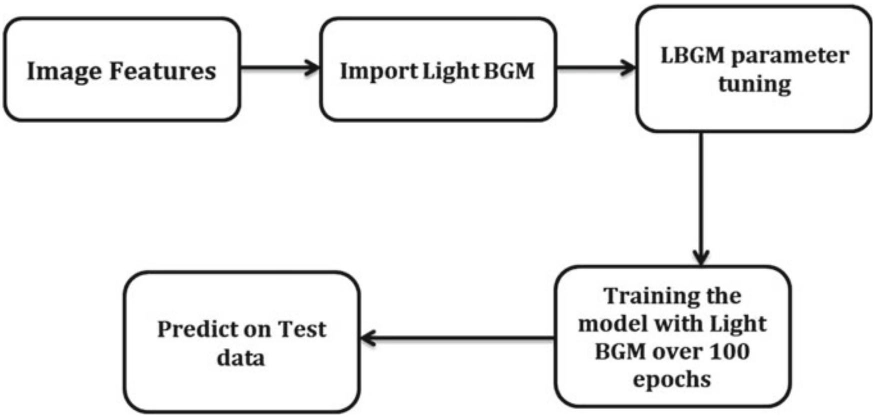


Fig. 8 LGBM implementation

Smart feature engineering and smart sampling are some of the techniques by which better models can be designed. Here are some of the smart feature engineering and smart sampling techniques used by Light GBM. The Bin way of splitting is the technique where a group of data is clubbed together, known as a bin, making the algorithm run faster. Exclusive Feature Bundling is the technique that creates a new feature known as a bundled feature, which can be used for further processing of the data, reducing the number of features and resulting in running the model faster and more efficiently. Gradient-based One-Side Sampling (GOSS) is another technique that arranges the gradients in descending order, and two subparts of gradients are created such that sampling is done from only one side of the subpart with lower gradient values, resulting in better accuracy of the model. The boosting technique we have used in this model is Dropouts × Multiple Additive Regression Trees (DART). It gives high accuracy for diverse tasks and uses dropouts to increase the model efficiency and deal with problems. It is used in ranking, regression, and classification tasks. Other parameters include the learning rate, which decides how fast the model runs. So, it should be low for less computational time. Then num_leaves is the controlling parameter that controls the complexity of the tree model, where max_depth is the depth of the tree. Assigning more value to num_leaves can cause overfitting of the model.

Algorithm 1: Mountainous Road detection

```
Data: RGB Images
Result: Prediction of the mountainous road
1.    For images in the directory
2.        Resize, grayscale
3.        Append the labels to array
4.    End for
5.    Encode the labels
6.    Extract the features of train data using GLCM
7.    Convert features into an integer vector
8.    Train the model over 100 epochs using LGBM
9.    Predict on test data
10.   Print confusion matrix and overall accuracy
11.   Import random image
12.   Predict the label for the image
```

Algorithm 1 gives the idea of a working code for the detection of mountainous roads. In the first four steps, a loop is shown for resizing, grayscaling, and labeling the images. Next comes the encoding of labels and then extracting the features of the image using GLCM. Those features are converted to integer vectors in step 7. Steps 8 and 9 are about training the model and then predicting the test data. The last steps are printing the confusion matrix and finding the model's accuracy.

4 Results and Discussion

The study area for this project includes the various mountainous roads across the northeast region of India and some mountain ranges present in Maharashtra. Also, images of the world's most dangerous routes, such as the Karakorum highway and Skippers Canyon, are collected by web scraping. All the collected images are divided manually into the train and test folders. These images are labeled separately as positive and negative, and afterward, those labels are encoded in numerical forms from 0 to $N - 1$, where N denotes the total number of labels.

A comparison of the proposed technique with existing methods is depicted in Table 2, and the research used different feature descriptors as discussed below.

For extracting the features of the image, various feature extractors have been used to compare the accuracy. Initially, Scale-Invariant Feature Transform (SIFT) was used for feature extraction. It helps to detect local features in an image and locate the key points. It converts those key points in the numerical data which are known as descriptors. As it uses 128 dimensions for the feature vector, it gives more features, but the time required for the computation is very high and not found to be very effective. Next is the Oriented FAST and Rotated BRIEF (ORB), which is a good alternative to SIFT in the case of finding descriptors. But it only uses 32 dimensions for the feature vector, so it computes very fewer features, resulting in lower accuracy than that of SIFT. Then, at last, we came up with GLCM for texture feature extraction and got the best accuracy in less computational time. All the results obtained for descriptors and classifiers are compared in Table 3.

The quality of the model is measured with evaluation metrics. Accuracy, Precision, Recall, and F1-Score are the metrics used in this paper. These metrics are compared by considering the different number of GLCM features in Table 4. And the same can be seen in graphical format in Fig. 9.

Random images are imported from test data and reshaped to the required dimensions after extracting the features from the image. Those features are used by the LGBM classifier to predict the label of the image. From the confusion matrix, 1770 images were true positive and true negative, whereas 229 images were false positive and false negative. One of the true predictions for the mountainous road is shown in Fig. 10.

Table 2 Comparison with existing methods

S. No.	Authors	Method and descriptors used	Results
1	Maeda et al. [1]	CNN, SSD using Inception V2 and SSD using MobileNet	Recall—71% Precision—77%
2	Zhou et al. [11]	CNN + OCSVM	Completeness—79.53% Quality—78.68%
3	Ustunel et al.	SIFT + PCNSA	Accuracy—83%
4	Proposed paper	Proposed method (GLCM)	Accuracy—90.95%

Table 3 Comparative analysis of descriptors

Descriptor	Classifier	Accuracy (%)
SIFT	Decision tree	77.26
	Random Forest	77.09
	KNN	75.99
	SVM	78.59
	GNB	75.99
ORB	Decision tree	70.77
	Random forest	72.31
	KNN	69.73
	SVM	70.63
GLCM	LGBM	90.95

Table 4 Performance metrics

	GLCM (3 features)	GLCM (4 features)	GLCM (5 features)
Accuracy	89.65	88.65	90.95
Precision	96.28	96.09	96.70
Recall	84.95	83.47	85.84
F1	90.27	89.33	90.94

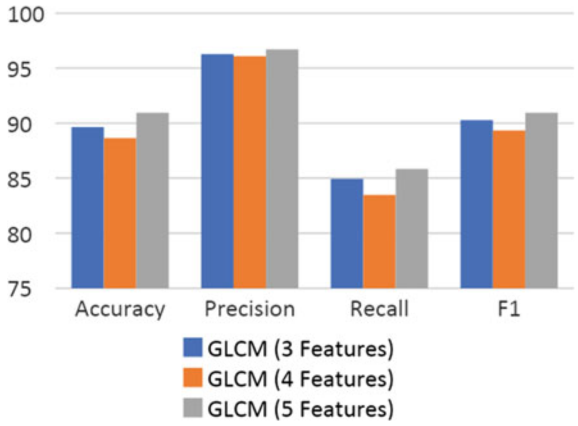
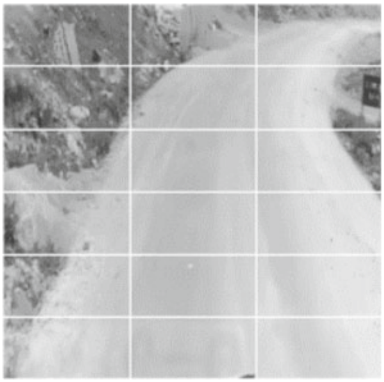


Fig. 9 Comparative analysis of Haralick features

Fig. 10 Sample result on the test dataset



Prediction: Mountainous
Ground Truth: Mountainous

5 Conclusion

This work compiles 8,000 images into a database and calculates the minimum number of features required for categorization and model training. GLCM is used to extract these features, delivering superior outcomes with less computing effort compared to other feature extractors. Numerous combinations of GLCM attributes have been employed in various directions for image texture extraction. Using a Light Gradient Boosting Machine (LGBM) as a classifier was the most advantageous component of the research. It expedites model training and increases memory utilization efficiency. LGBM employs a set of parameters with specific values that greatly contribute to the model's 90.95% accuracy. The model accurately predicts the kind of road and produces an audible warning for driving safety.

In the future, the focus will be on the fusion of other texture features with GLCM, which will further increase the accuracy and efficiency of the model. A further step is to study the robustness of direction change before feature extraction, so that satisfactory results can be achieved.

References

1. Maeda H, Sekimoto Y, Seto T, Kashiyama T, Omata H (2018) Road damage detection and classification using deep neural networks with smartphone images. Comput-Aided Civ Infrastruct Eng 33(12):1127–1141
2. Cheng Y-T, Lin Y-C, Ravi R, Habib A Detection and visualization of narrow lane regions in work zones using LiDAR-based mobile mapping systems
3. Hsu Y-W, Perng J-W, Wu Z-H Design and implementation of an intelligent road detection system with multisensor integration. In: 2016 international conference on machine learning and cybernetics, vol 1, pp 219–225

4. Gupta N, Ahmed Z, Vishnoi C, Agarwal AK, Ather D (2020) Smart road management system with rough road detection. TechRxiv. Preprint 12757979.v1
5. Du K, Xin J, Shi Y, Liu D, Zhang Y A high-precision vision-based mobile robot slope detection method in an unknown environment. In: 2018 Chinese automation congress, pp 3192–3197
6. Tareen SAK, Khan HM Novel slope detection and calculation techniques for mobile robots. In: 2016 2nd international conference on robotics and artificial intelligence, pp 158–163
7. Fauzi AA, Utaminingrum F, Ramdani F (2020) Road surface classification based on LBP and GLCM features using KNN classifier. Bull Electr Eng Inform 9(4):1446–1453
8. Zhang X, Cui J, Wang W, Lin C (2017) A study for texture feature extraction of high-resolution satellite images based on a direction measure and gray level co-occurrence matrix fusion algorithm. Sensors 17(7):1474
9. Triantafyllou D, Kotoulas N, Krinidis S, Ioannidis D, Tzovaras D (2017) Large vehicle recognition and classification for traffic management and flow optimization in narrow roads. In: IEEE smart world, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation 2017, pp 1–4
10. Goyal A, Singh M, Srivastava A (2019) Lane detection on roads using computer vision. Int J Eng Adv Technol
11. Zhou X, Chen X, Zhang Y (2018) Narrow road extraction from remote sensing images based on super-resolution convolutional neural network. In: IGARSS 2018—2018 IEEE international geoscience and remote sensing symposium, pp 685–688
12. Manoharan K, Daniel P (2019) Autonomous lane detection on hilly terrain for perception-based navigation applications. Imaging Sci J 67(8):453–463
13. Cai K, Chi W, Qing-Hu Meng M A vision-based road surface slope estimation algorithm for mobile service robots in indoor environments. In: 2018 IEEE international conference on information and automation (ICIA), pp 621–626
14. Guo Y, Chen G, Zhang J (2021) Measurement of road slope based on computer vision. J Phys: Conf Ser 1802(3):032063
15. Li Q, Min G, Chen P, Liu Y, Tian S, Zhang D, Zhang W (2020) Computer vision-based techniques and path planning strategy in a slope monitoring system using the unmanned aerial vehicle. Int J Adv Rob Syst 17(2):1729881420904303
16. Sivaraman S, Trivedi MM A review of recent developments in vision-based vehicle detection. In: 2013 IEEE intelligent vehicles symposium (IV), pp 310–315
17. Xu Z, Shen Z, Li Y, Xia L, Wang H, Li S, Jiao S, Lei Y (2020) Road extraction in mountainous regions from high-resolution images based on DSDNet and terrain optimization. Remote Sens 13(1):90
18. Song J, Song H, Wang S (2021) PTZ camera calibration based on improved DLT transformation model and vanishing point constraints. Optik 225:165875
19. Yin Z, Dai Q, Guo H, Chen H, Chao L (2018) Estimation road slope and longitudinal velocity for four-wheel drive vehicle. IFAC-Papers OnLine 51(31):572–577

Vision-Based Weather Condition Recognition for Driver Assistance

Jyoti Madake, Vedant Yawalkar, Yash Sambare, Rohan Waykos, Shripad Bhatlawande, and Swati Shilaskar

Abstract Many camera and sensor-based technologies have been launched in recent years to aid drivers in assuring their safety in a variety of driving situations. The decreased scene visibility in unfavorable weather conditions (fog/rain) is one of the issues that drivers experience while driving. The contrast and color of acquired photographs are poor in hazy weather circumstances, making it difficult to discern the image. A technique to separate images or categorize them from a dataset classification into the following categories, cloudy, sunny, foggy, rainy, and snowy, is done. To fulfill the aim, the Gray-Level Co-occurrence Matrix technique was used to collect weather-related attributes, and the necessary data processing was done. The system achieves a weather forecast accuracy of roughly 89.86%.

Keywords Gray-level co-occurrence matrix · Light gradient boosting machine · Haralick features · Driver assistance

J. Madake (✉) · V. Yawalkar · Y. Sambare · R. Waykos · S. Bhatlawande · S. Shilaskar
Vishwakarma Institute of Technology, Pune 411037, India
e-mail: jyoti.madake@vit.edu

V. Yawalkar
e-mail: vedant.yawalkar19@vit.edu

Y. Sambare
e-mail: yash.sambare19@vit.edu

R. Waykos
e-mail: rohan.waykos19@vit.edu

S. Bhatlawande
e-mail: shripad.bhatlawande@vit.edu

S. Shilaskar
e-mail: swati.shilaskar@vit.edu

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_11

1 Introduction

Thousands of people die in road accidents each year, as a result of meteorological conditions like fog, rain, and snow. Also, road accidents are increasing at a very high rate. In 2016, over 9,000 people died in fog-related car accidents, [1] with the count of mishaps rising to over 11,000 in late 2017. This represents an almost 20% increase in just one year. Worse, fatalities on roads due to fog-related collisions increased by nearly 100% between 2014 and 2017 (in absolute terms). According to research from around the world, increased visibility of the next vehicle [1] reduces crash risk by more than 30%. Fog detection is a difficult topic to solve since it relies on unknown variables such as depth, weather, and illumination conditions. The following conditions must be met by our algorithm: (1) The results must be precise, i.e., all foggy scenarios must be correctly recognized, and the chance of false detection (i.e., detecting fog when the image is not foggy) must be zero. (2) The algorithm must be quick and able to operate under real-time conditions. This is a significant constraint for a diagnosis function, as they will be integrated into more complicated intelligent vehicle systems that demand a significant amount of computer resources. (3) The model should be able to tolerate a wide range of daytime settings, including bright, gloomy, and wet conditions. The likelihood of a crash is expected to drop significantly. Consider how a self-driving car will react in bad weather: Reduced or missing sight: non-visible traffic signs and lane markings, lack of view due to sun glare, objects or cars in the route, unidentified persons, and so on. This document highlights the approaches and systems that have been discovered and deemed important for estimating bad weather. We discovered that the fog phenomenon is the most damaging after searching the scientific literature in recent years. This paper focuses on implementing appropriate computer vision techniques to interpret fog, rain, and snow from the image and alert/warn the driver. This will help in reducing road accidents significantly.

2 Literature Review

Fog classification is a preliminary activity that aids in the development of road vision-enhancing systems. To distinguish between homogeneous and heterogeneous camera hazy photos, synthetic images with various depth maps are used [2]. A fog-aware statistical feature-based referenceless perceptual picture defogging model is used. By using a multiscale Laplacian refinement, the fog warns weight maps successfully filter the most visible portions of three pre-processed images and combine them easily. The suggested model [3] performs better for darker, denser hazy images and normal defog test images, according to the results. Interpretation of the algorithm [4] for improving local visibility to account for the fact that road pictures, a third limitation is added to the inference of the atmospheric veil, assuming a fog with a visibility distance higher than 50 m. On a collection of 18 synthetic photographs where a uniform fog is merged according to Koschmieder's rule, the generated visibility enhancement technique [5] outperforms the original approach on road images. The semantic investigation of an object identification and classification issue is primarily concerned with low-level features, whereas this challenge was concerned with high-level picture data. As a result, increasing the number of convolution layers will have no effect on the total accuracy of weather image categorization [6]. It has been discovered through numerous trials that while present algorithms generate high visibility in fog-impaired photos, they also produce noise as well as color distortion in the sky area. Due to the inaccuracy of light transmission, the color of the sky region has been found to be significantly different from the original image [6]. A new approach is used [7] for predicting visibility in poor viewable weather settings, notably foggy air conditions, based on deep convolutional neural networks and the Laplacian of Gaussian filter. As can be seen from the outcomes, the technique obtained high precision during training with minimum loss. It can be proven that the accuracy of a 10-neighbor classifier is superior for categorizing an image as foggy or clear based on classification results provided by applying 19 algorithms [8]. An accuracy of 92.8% [9] was obtained using a confusion matrix and ROC curve, which is an excellent result. A perspective was devised to divide fog occurrences into three categories, no fog, fog, and intense fog, in order to expand the uses of outdoor computer vision systems. Feature extraction as well as unsupervised learning are the foundations of the technique. A five-component feature vector is created based on the contrast and detail characteristics of hazy photos. The Gaussian Mixture Describe is used to model the probability distribution of several classes in the training set in this study, and the model parameters are calculated using the expectation–maximization approach [10]. Spectral features combined with frequency information are useful for distinguishing between clear and foggy driving environments. In the study of night driving scenarios, images with high and low beam switching were examined [11]. Bronte and Miclea have focused [12, 13] on the fact that images deteriorate in fog and haze, and deterioration depends on distance, atmospheric particle density, and wavelength. The author put many frame-cleaning methods to the test and evaluated them using two different strategies. The first is concerned with the study of current measurements, while the second is based on psychophysical trials. The greater the visible wavelength, the better the clarity of the unblurred image. This research could have been used in outside surveillance, navigation, undersea adventures, and image-based rendering. The approach is based [14] on the dichromatic model, which is both simple and useful. Narasimhan was able to extract a number of important limits on various meteorological factors that generate color differences in the landscape. The authors proposed simple approaches to reconstruct the three-dimensional framework and real colors of sceneries from photographs captured in bad weather. The technique taken in [15] resulted in two important algorithms: (a) a post-processing rain detection and removal method for videos, and (b) an algorithm for determining the best during image collection, adjusting camera settings to reduce rain views. A rain-detecting camera lens system that adjusts to rain-reducing camera parameters automatically. This work analyzed known [16] atmospheric optics models and developed new ones while taking into account the limits that most vision applications experience. Nayar also discovered that [17] image analysis performed after acquiring polarization-filtered photos can remove hazy visual impacts. The method produces information on scene structure as well as the size distribution and density of particles in the air in addition to dehaze images. This publication's methodology is based on the partial polarization of air light. As a result, as the degree of polarization diminishes, its stability falls. Simple interactive tools [18] have also been observed for removing and adding weather effects to a single photograph. The three methods offered are simple to apply and may efficiently restore clear day colors and contrasts from photographs that have been damaged by rough weather. The restoration of contrast in atmospherically [19] damaged photos and video is one of the issues addressed. From two photos captured under various weather circumstances, the study described ways to find depth discontinuities and compute scene structure, and then author demonstrated how to restore contrast to any shot of a scene captured in poor lighting. The [20] technique works in a range of meteorological and viewing settings, even when polarization is minor because light dispersed by atmospheric particles (air light) is generally moderately polarized. It may be used with as little as two polarized photographs taken in opposite directions. We show experimental findings of comprehensive dehazing of outdoor sceneries, which show a significant improvement in scene contrast and color correction. An algorithm for image fog removal is done, which handles images with color channels as well as gray channels. Suggested techniques include a fog removal scheme based on Dark Channel Prior (DCP), Weighted Least Square (WLS), and High Dynamic Range (HDR). The suggested methodology's defogged photos are assessed using qualitative and quantitative analysis, and it is compared to other fog removal algorithms to determine its advantage [21]. To differentiate the foreground from the backdrop in the video stream, a background modeling approach is used. The image that is present in the background detects and measures fog because it is a constant meteorological condition. The foreground is used to identify rain because it is a dynamic occurrence. The proposed algorithms in [22] can then be integrated into current surveillance platforms in this way. This research considers two synthetic rainy picture models: the linear additive composite model (LACM model) and the screen blend model (SCM model). The fundamental notion is that finding a mapping between a wet image and rain streaks is simpler for the CNN network than finding a map between a clean and rainy image. The rain streaks have set characteristics, whereas clean photos contain a variety of characteristics. Experiments in this paper [23] show that constructed CNN networks perform better than state-of-the-art techniques in the real world and synthetic images, indicating that the suggested framework in this paper is effective. In this contribution, a neural network approach for calculating visibility in fog is presented. This solution uses a camera that could be mounted roadside or mounted in a car. This viewpoint provides estimates for visibility that are near the expected values for a huge range of hazy scenes. The main use of the method is that it is common in nature and neither requires any special camera calculations nor knowledge of depth map distances. The method used in [24] can be used in current traffic monitoring systems that use cameras. This acts as a support for driving that warns the driver and asks him to adjust his speed according to the estimated visibility. The goal is to create methods for validating computer vision systems under extreme weather. Chaabani and Hazar begin by digitally creating a rain picture simulator that is based on physical laws. It allows for the creation of images of rain that are physically accurate. After that, the simulator was tested using data from the Rain platform and Cerema R&D Fog. A procedure for evaluating the robustness of picture characteristics under wet settings is given [25]. Choi et al. have depicted a forecast model for perceptual haze thickness which is termed as "Fade" and picture defogging calculation that is termed as "De-fade" [26], in light of picture NSS and haze-mindful measurable highlights. Blur judges the level of perceivability of a foggy scene from a picture, while "De-fade" upgrades the perceivability of a foggy scene with no reference data, for example, many fog images of a similar scene, various levels of polarization, notable articles in the hazy scene, helper geological data, a profundity subordinate transferal map, arranged presumptions, and without preparing on human evaluated decisions. "Defade" accomplishes improved results on hazier, denser hazy pictures and on defog test pictures than best-in-class defogging calculations. The model that is proposed in [27] detects fog/haze with accurate results and gives maps specifying the regions in an image that set off its conclusion. Steps used to classify the image include Pre-processing, Feature Extraction, Windowing, FFT, Gabor Sampling, Feature Reduction, Scaling, and Classification. It first normalizes the input image in a pre-filtering procedure. Then it performs extraction of a feature that is contingent on the power spectrum of the photo, and then two-stage feature reduction is done with regard to Gabor filter sampling [28] and Principal Components Analysis (PCA). At last, SVM is used for the work of classification. Hussain [29] provides a solution to faded scene visibility problems faced by drivers by deep neural networks. Somavarapu et al. assumed the fog that is visible in a photo can be modeled by a function that is complex in nature and the author utilizes the neural network to model the fog using a mathematical model. The benefits of this method are (i) real-time operation and (ii) uses the least possible input, a single image, and works better with unrevealed images. Observations done on synthetic images tell that the proposed method has the potential to remove fog for better vision and security. Human location estimation is done utilizing SVM joining WLS (Weighted Least Squares), histograms of situated angles (HOG) is used. For the elimination of fog from pictures, weighted least square (WLS) channels are used, and, afterward, HOG and SVM calculations are used for the detection of humans.

3 Methodology

This study describes a method for detecting and classifying meteorological situations during driving on roads. The algorithm accepts the images and classifies the image in one of the five categories, cloudy, foggy, rainy, snowy, and sunny, and predicts which of these five categories or classes the image belongs to. The camera records the surroundings continuously and sends them to the processor-based system. The processor-based system categorizes and forecasts the weather condition. A novel approach to aid vehicle drivers through the use of computer vision has been proposed.

The approach involves various steps; the block diagram in Fig. 1 briefly illustrates them all. Over the term of the development of this work, multiple datasets, preprocessing steps, and algorithms were tested and analyzed. The system warns the vehicle drivers in case of adverse weather conditions like foggy, rainy, or snowy. Along with this, the system displays the weather condition in the image.

There are 20,000 images in the dataset which are further divided into five categories or five classes, namely foggy, cloudy, snowy, rainy, and sunny. The images were found on the Internet. Table 1 shows how the images in the collection are distributed.

The images were of different sizes, so each image is resized to 300 × 300 dimensions. Grayscale images were created from the scaled images using GLCM. Sample images from the dataset are given in Fig. 2. It includes one image of each class to give an overview of the images used during the training and testing of the GLCM model.

For characteristics that may be used to infer the degree of correlation between pairs of pixels, GLCM employs second-order statistics. It employs pairs of pixels, with the user able to specify the distance and angle between them. Correlation between nearby pixels is inferred via distance 1. Longer distances imply a larger scale of

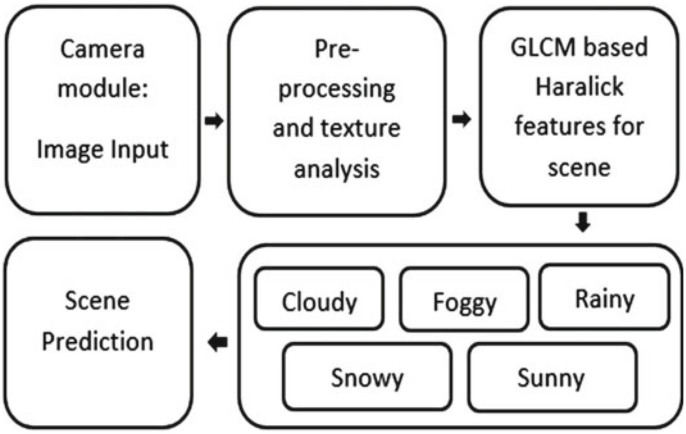

Camera module:
Image Input
Pre-processing and texture analysis
GLCM based Haralick features for scene
Cloudy
Foggy
Rainy
Snowy
Sunny
Scene Prediction

Fig. 1 Block diagram of system

Table 1 Details of the dataset

S. No.	Class	Number of images
1	Cloudy	4100
2	Foggy	3900
3	Rainy	4000
4	Snowy	3800
5	Sunny	4200
	Total	20,000

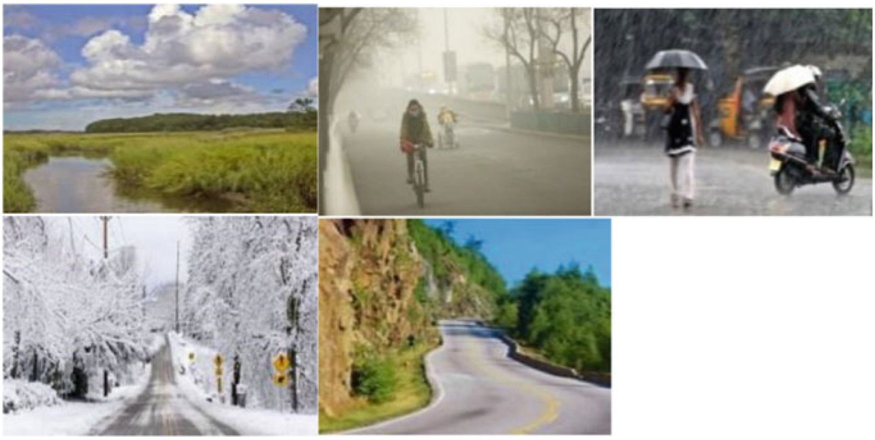


Fig. 2 Sample images from the dataset

association. Extraction of GLCM for various distances and angles between pixels is recommended for practical uses.

Importing the appropriate libraries was the initial step in creating the GLCM model; glob, greycomatrix, and greycoprops were imported in addition to the standard libraries needed to create any model. The glob command is used to find all file paths that match a given pattern. GLCM is computed using greycomatrix, and texture attributes for GLCM are calculated using greycoprops. Texture analysis helps in analyzing various different textures present in foggy, snowy, and rainy images. The texture in images used may feel nearly similar to human vision, but there is a huge variety in textures that can be analyzed by using greycoprops.

The next stage was to collect all the images and labels into arrays after loading the relevant libraries. Creating empty lists, then using the glob module to read the images, and using the for loop to resize images to 300 × 300 pixels was the next step. The images and labels are then appended to train images and train labels, respectively, and afterward converted to arrays. For train images, all of the above stages are completed. Now all the steps for the test/validation images as well as appended images (test) and labels to test images and test labels are repeated and converted to arrays. Figure 3 depicts the overall flow of the system proposed in this paper. It includes various blocks which helps in understanding the steps that were required while building this model.

The next step is pre-processing, which involves using a Label Encoder to encode classes: 0 for class cloudy, 1 for class foggy, 2 for rainy class, 3 for snowy class, and 4 for sunny class. After that, I divided the dataset (images) into two categories: training and testing. This paper utilized 80% of the images for training purposes and the rest of the images (20%) for testing. Later, a feature extractor was designed and the gray-level co-occurrence matrix (GLCM) for various distance values and orientations (angles) was computed. Assume we utilize a GLCM with a zero angle,

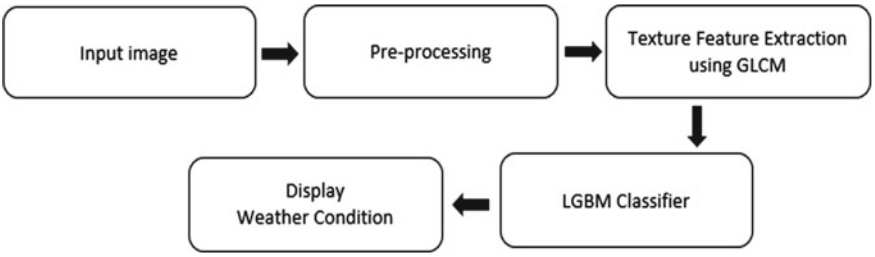

Input image
Pre-processing
Texture Feature Extraction using GLCM
LGBM Classifier
Display Weather Condition

Fig. 3 Overall flow of proposed system

which indicates the GLCM's orientation is horizontal. If the GLCM distance is 1, we just look horizontally at the next pixel to the current pixel.

Each element (i, j) in GLCM is the sum of the number of times that the pixel with a value i occurred in the specified spatial relationship to a pixel with value j in the input image. GLCM is a matrix where the row number and column number correspond to the number of gray levels in the image. Multiple GLCMs are computed for different offsets. Pixel relationships of varying directions and distances are defined by these offsets. Figure 4 gives offsets for various angles and distances used in the GLCM model. Distance is any fixed integer between 1 to the size of the image, and angles compare the diagonal pixels according to the value given to it.

The five texture features are extracted from GLCM, i.e., Dissimilarity, Contrast, Energy, Correlation, and Homogeneity, and along with that we also used Entropy, and all these are defined in the equations below where $P(i, j)$ is the GLCM value on (i, j) element, levels is the gray-level number used in the process of quantization, μ is mean of GLCM, σ^2 is the intensity variance of every pixel in a relationship that has contributed to GLCM, and b is logarithm function base. In addition to the five features mentioned above, Entropy is also used. The entropy is not the properties

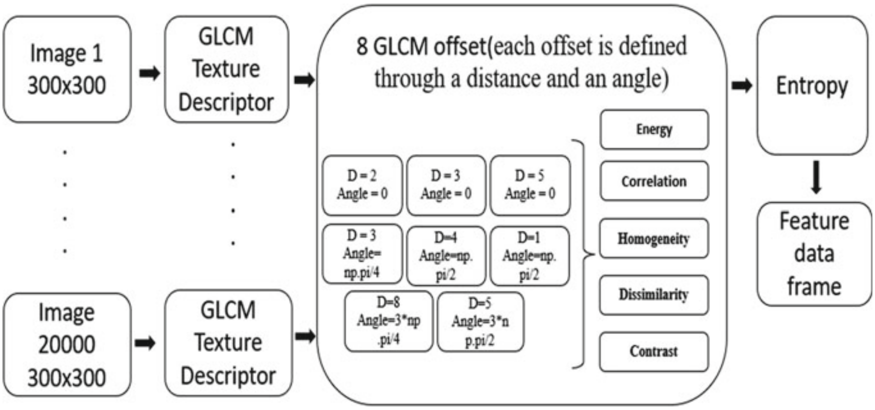


Fig. 4 Flow diagram for feature extraction

of a GLCM that can be calculated by scikit-image so this feature is calculated by using Shannon's entropy which was imported during the initial step of importing the necessary libraries.

Contrast: The result of Eq. (1) is a measurement of the intensity contrast that exists over the entire image between a pixel and its neighboring pixel:

$$\sum_{ij=0}^{\text{levels}-1} Pij(i-j)^2 \tag{1}$$

Dissimilarity: Eq. (2) returns how a pixel is correlated to its neighbor

$$\sum_{ij=0}^{\text{levels}-1} Pij|i-j| \tag{2}$$

Homogeneity: Eq. (3) returns a value that measures the proximity of the distribution of items in the GLCM-to-GLCM diagonal. Its value is between 0 and 1:

$$\sum_{ij=0}^{\text{levels}-1} Pij\Big/1+(i-j)^2 \tag{3}$$

Energy: Uniformity (textural) of an image, i.e., pixel pair repetitions, is measured by Energy Eq. (4). Its value ranges from 0 to 1. Energy also helps in deciding disorders present in the texture:

$$\begin{gathered}\text{ASM}: \sum_{ij=0}^{\text{levels}-1} Pij^2 \\ \text{Energy}: \sqrt{\text{ASM}}\end{gathered} \tag{4}$$

Correlation: Eq. (5) reveals the degree of correlation between a pixel and its nearby pixels. The value ranges from 1 to −1, with 1 indicating perfect positive correlation, −1 perfect negative correlation, and 0 indicating no correlation:

$$\sum_{ij=0}^{\text{levels}-1} Pij[(i-\mu i)(j-\mu j)]\Big/\sqrt{(\sigma i^2)(\sigma j^2)} \tag{5}$$

Entropy: Entropy is given by Eq. (6):

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} Pij\ \log b\ p(i,j) \tag{6}$$

There are a variety of features to describe image texture, for example, Gabor filters, local binary patterns, wavelets, and many more. Haralick's GLCM is one of the most popular texture descriptors. Describing image texture through GLCM features calculating GLCM for different offsets (each offset is defined through a distance and an angle), and extracting different properties like Contrast, Dissimilarity, Homogeneity, Energy, and Correlation from each GLCM offset are possible. Figure 4 flow diagram for feature extraction explains all 8 offsets used on each image in the dataset with different properties along with Entropy.

Total 8 different offsets (each offset is defined through distance and angle) are created. Mainly, angles are changed because this helps in extracting the texture for rainy and snowy images. graycoprops is used to calculate the statistics that are specified in properties from GLCM. After this, entropy is added as one more filter. Later, extraction of the features from the train images using the feature extractor that was defined is done, and then importing LightGBM. The Class names for LGBM start from 0, so reassign the labels from 1, 2, 3, 4, 5 to 0, 1, 2, 3, 4.

The number of data points in the train set is 1201 after using the LGBM model and the number of features used is 26. LightGBM is a gradient boosting decision tree algorithm launched by Microsoft in 2017. Like XGBoost, it doesn't use level-wise generation strategy, rather LightGBM uses leaf-wise generation strategy. It's fast, efficient, and has support for parallel and GPU learning. The dataset is trained with the aid of the LightGBM Classifier. In LightGBM, learning rate was set to 0.5 and the dart boosting type was implemented. A total of 500 iterations are used with 200 leaves, and 800 as maximum depth. Afterward, feature extracted data was trained under LightGBM under these parameters. As the features are less in number, so feature reduction is not done. The extracted features from GLCM for each image are given in Table 2.

The prediction on test data or test images using the model proposed is done. Extracting features from test data and reshaping, just like we did for training data was the next step. Also, inverse transform is done to get the original label back. Then metrics from sklearn are imported, and we print the overall accuracy, precision, recall, and F1-score of the model. Accuracy gives the correctness of the model proposed in this paper. Afterward, the confusion matrix is also plotted by importing the confusion matrix from sklearn. The confusion matrix shows where the model proposed predicted true values and where the model failed to do so. At last, check

Table 2 Extracted features

Texture image features	Image1	Image2	Image3	Image4
Energy	0.0214	0.0198	0.0180	0.0204
Correlation	0.9821	0.9937	0.9848	0.9663
Dissimilarity	5.8587	4.1404	5.4517	6.0619
Homogeneity	0.3014	0.2616	0.2525	0.2282
Contrast	142.98	38.227	82.806	161.31
Entropy	7.7712	7.6432	7.6923	7.5317

the results on a few random images to test the model and compare the original class and the class predicted by our model. This step also reversed the label encoder to the original name; this helps in getting the actual name of the class like cloudy, foggy, rainy, snowy, and sunny instead of assigned labels, i.e., 0, 1, 2, 3, 4, respectively.

4 Results and Discussion

The proposed system was tested on Jupyter Notebook software installed on a Lenovo Ideapad 3 Laptop, having 8 GB RAM with an Intel i5 11th Generation processor, an AMD Radeon graphics card, and a storage of 512 GB SSD. The data is categorized into five different types in this vision-based fog/rain interpretation system: foggy, cloudy, rainy, snowy, and sunny. The Gray-Level Co-occurrence Matrix (GLCM) classification system was applied, and it had a prediction accuracy of 89.86%. Our classification's confusion matrix is depicted in Fig. 3. The confusion matrix clearly shows that the model is able to identify the images properly the vast majority of the time, but that the model occasionally gets confused between rainy and cloudy images due to the presence of clouds in rainy weather images. Cloudy weather images and foggy weather images both have a similar level of ambiguity. Figures 5 and 6 show that the model can properly forecast the weather image, however Figs. 7 and 8 show that the prediction has gone wrong for the reasons indicated above.

The values of precision, recall, F1-score, and accuracy can be depicted in Table 3. Also, the equations for precision, recall, F1-score, and accuracy are given in Table 3.

Precision: In Eq. (7), the fraction of projected positive situations that were right is known as precision:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

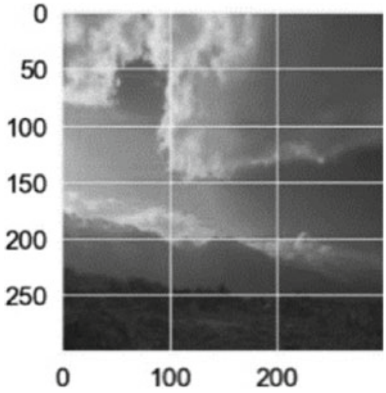


Fig. 5 Predicted: cloudy. Actual: cloudy

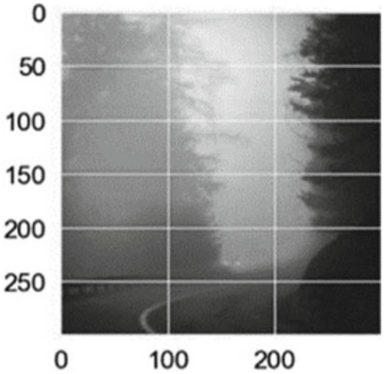


Fig. 6 Predicted: cloudy. Actual: cloudy

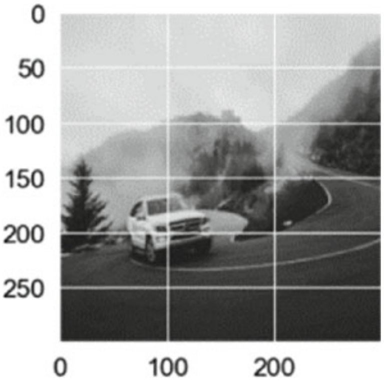


Fig. 7 Predicted: cloudy. Actual: foggy

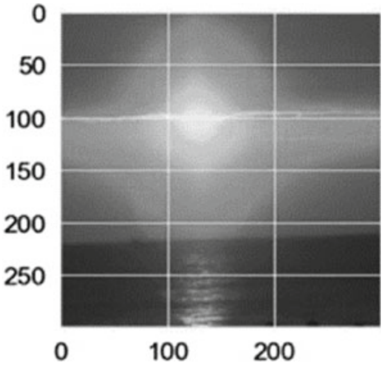


Fig. 8 Predicted: rainy. Actual: sunny

Table 3 Evaluation parameters of proposed model

Classification algorithm	Accuracy	F1	Precision	Recall
Model	0.8986	0.8986	0.9071	0.8986

Table 4 Comparison between existing system

Paper No.	Method used	Accuracy (%)
Goswami [5]	Convolutional neural network	63
Palvanov and Im Cho [7]	Deep hybrid convolutional neural network	95
Wan et al. [10]	Gaussian mixture model	89.8
Schechner et al. [18]	Gabor sampling	94
Galdren [27]	Weakly supervised learning and multiple instance learning	88
Proposed technique	GLCM-based Haralick features	89.86

Recall: In Eq. (8), the fraction of positive instances that were accurately detected, also known as True Positive Rate:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

F1-Score: In Eq. (9), the harmonic mean of accuracy and recall are used to calculate the F1-score:

$$\text{F1 Score} = \frac{2 * (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \tag{9}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{10}$$

TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative predictions, respectively. From Eq. (10), the overall accuracy for the model used in this paper is 89.86% which is further compared with other existing systems.

Table 4 shows the comparisons between the existing system and the model used in this paper.

5 Conclusion

An effective computer vision-based system to detect unfavorable weather conditions while driving and inform the driver is required, which can greatly reduce the risk of a fatality. This research proposed a computer vision-based system that detects weather from images and notifies the driver of adverse weather conditions such as foggy or wet weather. The Gray-Level Co-occurrence Matrix approach was used to capture the best characteristics related to weather conditions, and the necessary data processing was completed to meet the goal. The algorithm finally reaches a weather prediction accuracy of around 89.86%. In this research, a standard computer vision technique is employed for weather prediction, allowing this model to be implemented on devices with limited processing and storage. One of the system's shortcomings is that it is ineffective at night. This model will not be able to predict the weather while driving at night. Rainy and overcast weather have a lot of similarities, therefore the model might occasionally misread the weather when detecting it. The extraction of features will take a long time and a lot of resources if the dataset is too large. As a result, high-performance computing equipment will be necessary.

In the future, this computer vision-based technique, along with actually defogging and clearing the driver's view in severe weather circumstances, might aid the driver in driving comfortably and without the risk of a collision. This technique might potentially be beneficial for self-driving automobiles.

References

1. Over 1000 lives lost in fog-related road crashes—Priya Kapoor. https://timesofindia.indiatimes.com/india/over-10000-lives-lost-in-fog-related-road-crashes/articleshow/67391588.cms
2. Anwar MI, Khosla A (2015) Classification of foggy images for vision enhancement. In: International conference on signal processing and communication, pp 233–237
3. Choi LK, You J, Bovik AC (2014) Referenceless perceptual image defogging. In: Southwest symposium on image analysis and interpretation, pp 165–168
4. Tarel J, Hautière N, Cord A, Gruyer D, Halmaoui H (2010) Improved visibility of road scene images under heterogeneous fog. In: IEEE intelligent vehicles symposium, pp 478–485
5. Goswami S (2020) Towards effective categorization of weather images using deep convolutional architecture. In: 2020 international conference on Industry 4.0 technology (I4Tech), pp 76–79
6. Pal T, Bhowmik MK (2018) Quality enhancement of foggy images comprising of large sky region on SAMEER TU dataset. In: 2018 9th international conference on computing, communication and networking technologies (ICCCNT), pp 1–7
7. Palvanov A, Im Cho Y (2018) DHCNN for visibility estimation in foggy weather conditions. In: 2018 joint 10th international conference on soft computing and intelligent systems (SCIS) and 19th international symposium on advanced intelligent systems (ISIS), pp 240–243
8. Shrivastava S, Thakur RK, Tokas P (2017) Classification of hazy and non-hazy images. In: 2017 international conference on recent innovations in signal processing and embedded systems (RISE), pp 148–152
9. Anwar MI, Khosla A (2018) Fog classification and accuracy measurement using SVM. In: 2018 first international conference on secure cyber computing and communication (ICSCCC), pp 198–202
10. Wan J, Qiu Z, Gao H, Jie F, Peng Q (2017) Classification of fog situations based on Gaussian mixture model. In: 2017 36th Chinese control conference (CCC), pp 10902–10906
11. Pavlic M, Rigoll G, Ilic S (2013) Classification of images in fog and fog-free scenes for use in vehicles. In: IEEE intelligent vehicles symposium (IV), pp 481–486
12. Bronte S, Bergasa L, Alcantarilla PF (2009) Fog detection system based on computer vision techniques. pp 1–6
13. Miclea R-C, Ungureanu V-I, Sandru F-D, Silea I (2021) Visibility enhancement and fog detection, recent scientific papers with potential for application to mobile systems. Sensors 21:3370
14. Narasimhan SG, Nayar SK (2004) Vision and the atmosphere. Int J Comput Vis 48:233–254
15. Berman D, Treibitz T, Avidan S (2016) Non-local image dehazing. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 1674–1682
16. Garg K, Nayar SK (2007) Vision and rain. Int J Comput Vis 75:3–27

17. Nayar SK, Narasimhan SG (1999) Vision in bad weather. In: Proceedings of the seventh IEEE international conference on computer vision, vol 2, pp 820–827
18. Schechner YY, Narasimhan SG, Nayar SK (2001) Instant dehazing of images using polarization. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition, CVPR 2001, pp I–I
19. Narasimhan SG, Nayar SK (2003) Contrast restoration of weather degraded images. IEEE Trans Pattern Anal Mach Intell 25(6):713–724
20. Schechner YY, Narasimhan SG, Nayar SK (2003) Polarization-based vision through haze. 42:511–525
21. Anwar MI, Khosla A (2017) Vision enhancement through single image fog removal. Eng Sci Technol, Int J 20(3):1075–1083
22. Hautière N, Bigorgne E, Bossu J, Aubert D (2008) Meteorological conditions processing for vision-based traffic monitoring
23. Chen T, Fu C (2018) Single-image-based rain detection and removal via CNN. J Phys Conf Ser 1004(1)
24. Chaabani H et al (2017) A neural network approach to visibility range estimation under foggy weather conditions. Procedia Comput Sci 113:466–471
25. Duthon P, Bernardin F, Chausse F, Colomb M Methodology used to evaluate computer vision algorithms in adverse weather conditions. Transp Res Procedia 14:2178–2187
26. Choi LK, You J, Bovik AC Referenceless prediction of perceptual fog density and perceptual image defogging. IEEE Trans Image Process 24(11):3888–3901
27. Galdran A, Costa P, Vazquez-Corral J, Campilho A Weakly supervised fog detection. In: 25th IEEE international conference on image processing (ICIP), pp 2875–2879
28. Pavlić M, Belzner H, Rigoll G, Ilić S (2012) Image-based fog detection in vehicles. In: 2012 IEEE intelligent vehicles symposium, pp 1132–1137
29. Hussain F, Jeong J (2016) Visibility enhancement of scene images degraded by foggy weather conditions with deep neural networks. J Sens 2016:1–9

Vision-Based Distracted Driver Detection Using a Fusion of SIFT and ORB Feature Extraction

Jyoti Madake, Aditya Raje, Sarang Rajurkar, Rajas Rakhe, Shripad Bhatlawande, and Swati Shilaskar

Abstract Road accidents have increased in recent years for a variety of causes. Distracted driving is a leading cause of car crashes, and unfortunately, many of these crashes end in fatalities. In this study, we are using computer vision with machine learning, to provide a unique technique to determine whether a car driver is driving the car safely or not. In this study, two feature extraction approaches SIFT and ORB were implemented, with an emphasis on the driver's location, expression, and behavior. These two techniques effectively extract the features from the images in the dataset. This was followed by PCA to reduce dimensions. The implementation work further deployed several models including Decision Tree, Random Forest, KNN, and SVM to identify distracted driving behaviors. When using the ORB feature extractor, more accuracy was achieved in comparison with SIFT. The highest accuracy achieved was 90.75%, using KNN classifier. In the case of SIFT, the SVM RBF classifier gave the highest accuracy of 78.00%. It appears from the experiments that the suggested system has the ability to aid drivers in practicing safe driving behaviors.

Keywords Classification · Distracted driving · SIFT · ORB

J. Madake (✉) · A. Raje · S. Rajurkar · R. Rakhe · S. Bhatlawande · S. Shilaskar
Vishwakarma Institute of Technology, Pune 411037, India
e-mail: jyoti.madake@vit.edu

A. Raje
e-mail: aditya.raje19@vit.edu

S. Rajurkar
e-mail: sarang.rajurkar19@vit.edu

R. Rakhe
e-mail: rajas.rakhe19@vit.edu

S. Bhatlawande
e-mail: shripad.bhatlawande@vit.eud

S. Shilaskar
e-mail: swati.shilaskar@vit.edu

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_12

1 Introduction

Every year, around 1.3 million people are injured or killed in road traffic-related accidents. “Distracted Driving” is a term that means that the vehicle driver is not keeping an eye on the road, and is involved in some other activity, not related to driving and potentially harmful at that time. Possible distracted driving behaviors include texting and/or handheld cell phone use or talking to a co-passenger or drinking a beverage [1, 2].

Distracted driving can also put the lives of passengers of the vehicle as well as people around the vehicle at risk because ultimately everything revolves around the driver of the vehicle. With the rise in the number of vehicles on the road, the detection of such drivers is necessary in order to prevent accidents. Some papers have discussed the detection of such drivers using Convolutional Neural Networks (CNN), but this is a slow technique [6]. It takes so much time to work and hence is not worth using, as detection must be fast in order to alert the driver. Even though using these techniques gives higher accuracy, the time taken to implement and data size rules out using this. The approach discussed in this paper includes using traditional machine learning models to classify the distraction. The process starts with dataset pre-processing, then feature extraction, followed by clustering and dimensionality reduction and then classification. The advantage of using this approach is that the model is tested with different techniques and hence the best one can be used which gives higher accuracy. Also, the time taken is considerably less and makes it a good choice to use.

The topics in this paper are presented in the following order: The literature review is included in Sect. 2 of this document. In the second section, we will do a comparative analysis of the many approaches that have been taken to address the problem of distracted driving. In part III, both the methodology and the specifics of the execution are explained. The discussion of the results can be found in part IV, and then the conclusion and potential future applications can be found in Sect. 5.

2 Literature Survey

This section contains a summary of some of the most important and well-known studies in the subject of driver distraction detection systems that have been published.

A CNN module, feature synthesis module, and feature categorization module make up a three-module hybrid CNN framework (HCF) for accurate distracted driving detection [1]. The hybrid CNN has 96.74% classification accuracy, according to the data. Torres et al. [5] present a CNN module to detect and classify cell phone use in driver video surveillance. The proposed model can detect distracted drivers with 99 percent accuracy. The developers of [2] analyze stereo and lane tracking data to evaluate a driver’s visual and cognitive effort. The SVM algorithm is used to determine whether a person has cognitive impairment. When combined with eye movement tracking, this creates a powerful model for detecting distracted drivers. Driver safety was the study's objective [4]. By examining the driver's eyes, eye blinking, lips, and head position, an effective automated method for detecting driver fatigue was established. VJ, HOG, and GLCM feature extraction are employed in this framework. KNN and FFNN were used for proper recognition and performance evaluation in this study. 96% of alarms are accurate. In a similar way, the next study presents a new D-HCNN (deep convolutional neural networks) model with HOG feature extraction, L2 weight regularization, then dropout, and finally batch normalization. On AUCD2 and SFD3, accuracies are 95.59% and 99.87%, respectively [5, 6].

The next study presents a CNN-BiLSTM driver distraction posture detection approach. Spectral-spatial image properties were acquired and processed using this method. First, pre-trained CNNs gather spatial posture characteristics, then BiLSTM architecture extracts spectral properties from the stacked feature maps [7]. The major goal of the article [8] was to create a system using SURF key points on the state farm distracted driving detection dataset. The HOG feature extraction method is used to extract the driver behaviors in automobiles, and KNN is used to classify them. A two-stage distracted driving detection system was created to detect nine distracted driving behaviors. First, transfer learning and fine-tuning developed vision-based CNN models and then LSTM-RNN [9]. Next research [10] uses a Raspberry Pi-based computer vision system to continuously monitor driver inattentive behavior. The tracking eye movement, hand movement, and stance estimate to detect distracted drivers using CNN were evaluated to determine if the motorist is distracted [11]. Facial landmarks monitor eye blinking in real time and detect eyes and faces. Finally, data is evaluated to determine if the motorist is distracted.

Semi-supervised approaches for detecting driver attentiveness in real-world driving conditions were investigated in [12]. The cost of training data labeling was lowered because of this. This system employs the SS-ELM (semi-supervised extreme learning machine) and the Laplacian SVM. The model was built with the driver's eye and head motions in mind to discern between two different driver states. The two states of the driver are alert and intellectual. Implementing the SS-ELM method yielded a maximum accuracy of 97.2% and a G-mean of 0.959.

IVIS (an ML technique for reducing the dimensionality of very big datasets using Siamese neural networks) and PADAS (partially autonomous driving assistance systems) were employed by the authors in [13]. The state of the driver was modeled using SVM and neural networks. The tests were carried out on simulators, and the findings were then analyzed and classified. The Support Vector Machine (SVM) technique was utilized by the authors in their study [14]. The raw data gathered from the participants in the study were categorized subsequently after collecting eye and driving data from them. For each participant in the study, SVM models were trained, and the findings were projected as a binary outcome: distracted or not.

The architecture of the Convolutional Neural Network (CNN) is used and briefly defined in this research, and data analysis is also performed. ResNet-50 and transfer learning were used to develop the deep learning system, which had an accuracy of 89.71%. To analyze the model, a video of a real-world driving scenario was used, and important findings were extracted to validate the research. The next paper's research is built on a solid deep learning-based solution technique. The implementation of a genetically weighted ensemble of CNNs is the major topic of this paper. It is used to provide good classification performance using weighted ensembles of classifiers and a genetic algorithm. This paper also discusses the model that is constructed, which is mostly executed utilizing face and hand localizations, as well as skin segmentation. An accuracy rate of up to 92% was achieved. Also linked to this, the [17] shows a one-of-a-kind system that uses convolutional neural networks to automatically learn and guess pre-defined driving destinations (CNN). The major purpose was to predict which driving postures were safe and unsafe and to do so, discriminatory data was acquired from the driver's hand position. The dataset contains recordings of four different driving postures, including normal driving, answering a phone call while driving, eating, smoking, and chatting. This method, when compared to other popular procedures, yields the best results, with an overall accuracy of 99.78%. The next study discusses the use of CNN to identify inattentive drivers. Convolutional neural networks are trained using raw photographs, segmentation images, facial images, and hand images. The information is gathered in video format before being sliced into individual images. This system achieves a classification accuracy of 90% using CNN [15–18].

The study [19] uses human action recognition to detect inattentive driver posture. The residual block inception module and hierarchical RNN employed to predict pre-defined driving conditions. Yan et al. [17] identify the driver's hand position and predict safe/unsafe driving posture. Classification followed a sparse filtering pre-trained CNN model. Southeast University's driving posture dataset tested the method [17].

3 Methodology

The proposed system detects whether the driver is distracted while driving the vehicle. Input images (from the dataset) are captured from a camera fixed on the roof of the car from the inside and are used for processing. Before beginning the process of feature extraction on the image, they are first scaled, and then the Scale-Invariant Feature Transform (SIFT) and the Oriented Fast and Rotated Brief algorithms are utilized (ORB). This is followed by dimensionality reduction and then classification using traditional machine learning models. The proposed system is described in the block diagram in Fig. 1.

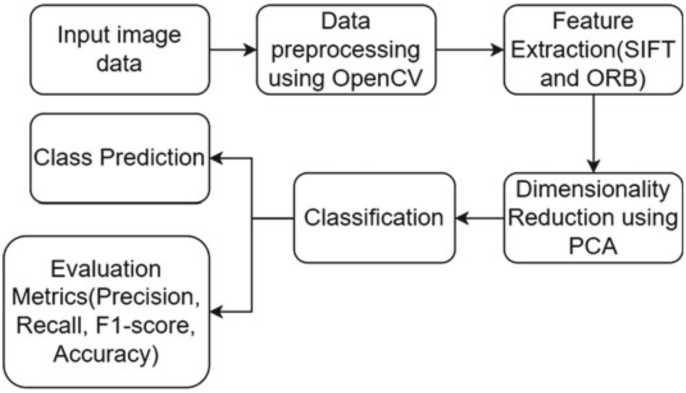

Input image data
Data preprocessing using OpenCV
Feature Extraction(SIFT and ORB)
Dimensionality Reduction using PCA
Classification
Class Prediction
Evaluation Metrics(Precision, Recall, F1-score, Accuracy)

Fig. 1 System block diagram

3.1 Dataset Collection

A dataset having around 4000 images was used. This dataset consists of some images of normal driving and the remaining mixed with images of drivers doing some distractive activity. Some images from the dataset are given in Fig. 2. The activities of the driver include operating on the phone, talking on the phone, drinking a beverage, and normal driving.

3.2 Feature Vector Extraction

In this particular implementation, two different approaches of feature extraction were tried out, and then the one that proved to be the most successful was selected to continue. ORB, which stands for Oriented Fast and Rotated Brief, was compared to SIFT, which stands for scale-invariant feature transform; however, ORB performed more effectively.

Approach 1: ORB Feature Extractor

ORB is an improved version of two existing methods, the FAST key point detector and the BRIEF descriptor. It leverages FAST to locate critical nodes and then employs the Harris corner measure to determine which N nodes are the most important. For the purpose of making a multiscale element, a pyramid is also employed. The extracted key points in an image from the dataset are visible in Fig. 3(a). The ORB descriptor was applied to extract features from 4000 grayscale images. There were 32 dimensions and the number of descriptors was 8,59,866. So, the ORB feature vector was (859,866 × 32).

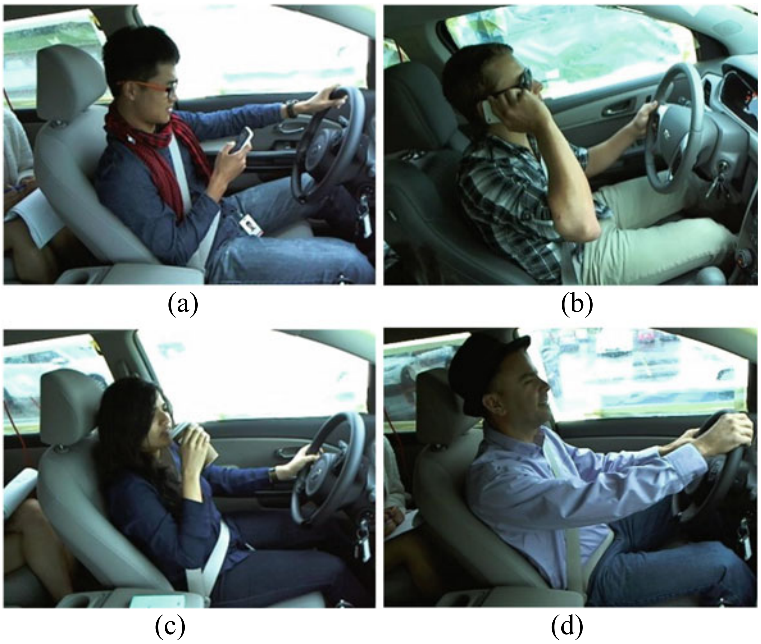


Fig. 2 Sample images from the dataset: operating the phone (**a**), speaking on phone (**b**), drinking a beverage (**c**), and normal driving (safe driving) (**d**)

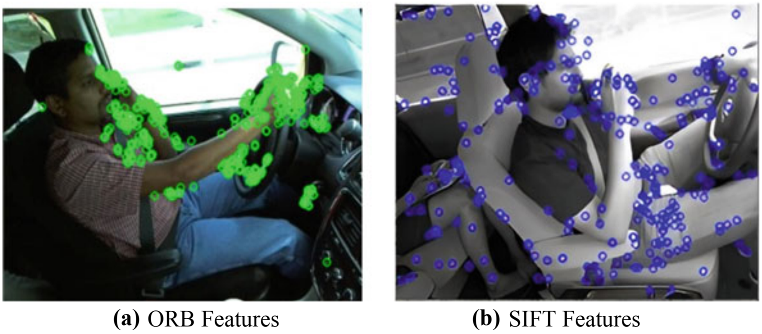

(a) ORB Features **(b)** SIFT Features

Fig. 3 **a** ORB features. **b** SIFT features

Approach 2: SIFT Feature Extractor

A local feature can be located, seen, and aligned using the SIFT, which is a feature descriptor algorithm based on images. SIFT decreases the possibility of errors by detecting and describing many locally important spots in the image. The extracted key points in an image from the dataset are visible in Fig. 3b. In the case of SIFT, the descriptor was applied to extract features from the same 5000 grayscale images.

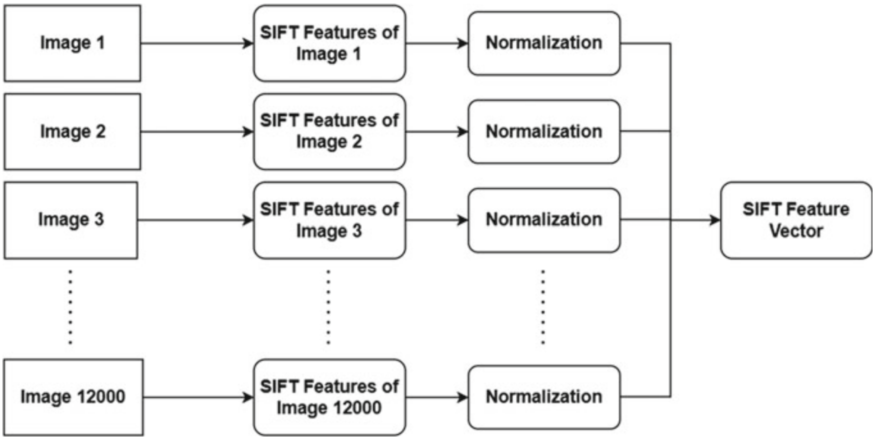


Fig. 4 Flow diagram of feature extraction using SIFT

There were 128 dimensions, and the number of descriptors obtained was 5,19,541. So, the size of the feature vector in this case for all images was (519,541 × 128).

Figure 4 represents the flow of an implementation of feature extraction using SIFT.

3.3 Dimensionality Reduction

Initially, the K-Means clustering technique is used to group similar data points together, and then principal component analysis is applied to further minimize the number of dimensions (PCA). Using the K-Means clustering algorithm, the input feature vector was divided into 19 groups ($k = 19$). Through the use of the elbow curve, the optimal value of k is determined.

Approach 1: Using SIFT

SIFT feature extractor was used first, and features of every image were predicted. Original feature vector was (519,541 × 128), which was reduced. A final CSV file with all k-means predicted data was compiled, of which the vector shape was (3999 × 20). After that, principal component analysis was applied taking 18 components, as this contained 97% of the information. The feature vector after this step was (3999 × 18). Then this feature vector was given for classification for detection. Figure 5 describes the dimensionality reduction process.

Approach 2: Using ORB

In case of ORB, again K-Means and PCA were applied. The original feature vector was (859,866 × 32) which changed to (4000 × 20) on applying K-Means taking 19

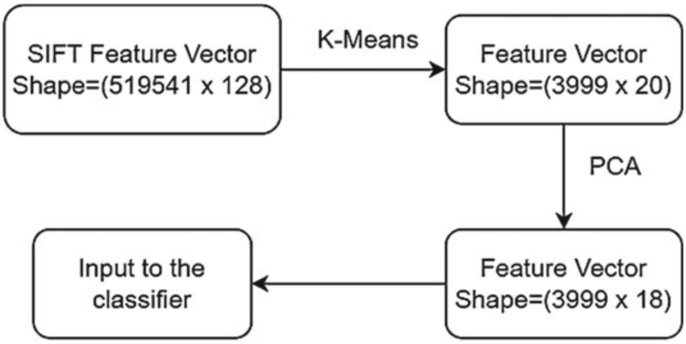

SIFT Feature Vector Shape=(519541 x 128)
K-Means
Feature Vector Shape=(3999 x 20)
PCA
Feature Vector Shape=(3999 x 18)
Input to the classifier

Fig. 5 Block diagram of dimensionality reduction when using SIFT

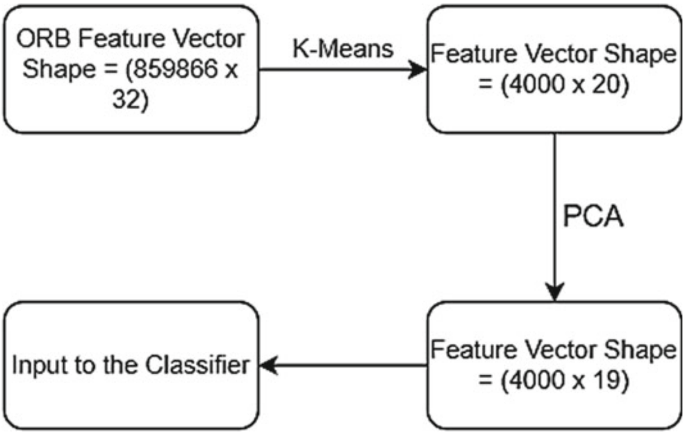

ORB Feature Vector Shape = (859866 x 32)
K-Means
Feature Vector Shape = (4000 x 20)
PCA
Feature Vector Shape = (4000 x 19)
Input to the Classifier

Fig. 6 Block diagram of dimensionality reduction when using ORB

clusters ($k = 19$). When PCA was applied taking 18 components, the feature vector changed to (4000×19). Figure 6 describes the dimensionality reduction when using ORB feature extractor.

3.4 Classification

The resulting feature vectors were fed into a series of classification algorithms. The system utilized a number of different classifiers, such as Decision Tree, KNN, Random Forest, SVM, and Voting Classifier, in order to identify which one provided the best fit for the model. Classifiers' predicted results were compared using a number of performance metrics. These metrics included accuracy, precision, recall, confusion matrix, specificity, sensitivity, and F1-score. The main goal of employing multiple classifiers was to simply understand how the training dataset behaved when fed into various models.

The first classifier was the decision tree. The decision tree algorithm is the most like how the human brain functions. Different attribute selection measures are used to determine the position of nodes in decision trees. It is a tree-structured supervised machine learning classifier. The parameter criterion = "entropy" is used in the model. It aids in identifying the splitting based on information gain entropy. Equation (1) determines the entropy in this algorithm. The decision tree algorithm's key benefit is that it aids in the generation of nearly all feasible solutions to a problem.

The equation for entropy is

$$E(S)\sum_{i=1}^{c} -p_i \log_2(p_i) \tag{1}$$

where p_i is the probability of class i. This entropy determines the information gain in each node of the decision tree.

KNN was the second classifier used. The KNN algorithm classifies data based on its proximity to other data classes. In this scenario, the neighbors to be evaluated are determined by the value of k ($k = 30$). When calculating the distance "d" in KNN, the Euclidean distance function is typically utilized. The formula in Eq. (2) is used to compute this distance:

$$\text{distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

(x_1, y_1) and (x_2, y_2) are the coordinates of the two points.

Random forest was the third classifier used. Multiple decision trees are combined in Random Forests. It predicts the categorization by estimating the maximum vote. Random Forests use a notion known as the Gini Index to determine the impurity of a given node. The Gini Index is calculated as follows: P+ and P− are the positive and negative class probabilities, respectively.

SVM was the fourth and final classifier employed. This classifier draws a line through the data, dividing it into classes. SVM employs a set of functions known as kernels. Linear, RBF, and Polynomial are the three SVM kernels used in the classification. The linear kernel is the most fundamental type of kernel and is often one dimensional. When there are several features and the data can be separated linearly, it proves to be the optimal function.

Non-linear model learning is made possible by a polynomial kernel, which maps the similarity between training samples' vectors to a feature space defined by polynomials of the original variables. The decision boundary calculation is determined with Eq. (3) given below:

$$F\left(x_i, x_j\right) = (x_i \cdot x_j + 1)^d \tag{3}$$

where x_i and x_j represent the data to be classified. A dot product of those values is taken to which 1 is added. d is the degree and $F(x_i, x_j)$ is the class-separation decision border.

The last kernel used in SVM was the RBF (radial basis function). It is typically chosen for non-linear data, and it facilitates accurate separations when no prior knowledge of the data is available. The decision boundary for the RBF kernel is calculated using Eq. (4).

$$F\left(x_i, x_j\right) = e^{\left(-\gamma \|x_i - x_j\|^2\right)} \tag{4}$$

where x_i and x_j represent the data to be classified. Value of γ varies between 0 and 1 and F(x_i, x_j) is the decision boundary to separate the classes.

```
Algorithm 1: Algorithm for distracted driver detection
Input: Driver images dataset (D)
Output: Distracted or not
1. Process every image from dataset
2. Perform image pre-processing
3. SIFT and ORB feature extraction and dimensionality reduction
4. Obtain PCA feature vector
5. Implementation of classifier's model
Classifiers, M = [KNN, Decision Tree, Random Forest, SVM(Poly), SVM(RBF), SVM(Linear)]
6. For each classifier, calculate accuracy of the model
7. Splitting a dataset and building a model
8. Calculating accuracy and evaluation metrics and saving the model
9. Return prediction-based model
```

4 Results

The dataset was split into training and testing datasets. 80% of images were taken for training the models and tested on the remaining 20%. The proposed system was tested on Jupyter Notebook software installed on an Asus TUF Gaming Laptop, having 8 GB RAM with an Intel i5 10th Generation processor, a 4 GB NVIDIA GTX 1650 graphics card and storage of 1000 GB SSD.

Using the ORB feature extractor, the following results were obtained. For the decision tree, a maximum depth of 13 was used for classifying the data. This algorithm provided an accuracy of 80.12%. The Random Forest classifier was created using an estimator of 100 and gave an accuracy of 88.00%. Next, the KNN algorithm was trained with 5 neighbors with value of p 2, and the accuracy obtained in this case was 90.75%. SVM algorithm was applied next, with linear kernel giving an accuracy of 77.62%, polynomial kernel accurate to 89.37%, and RBF (radial basis function) kernel giving an accuracy of 90.25%. The tables below summarize the results of the classifiers using both the feature extraction techniques. Table 1 describes the classifier performance when using SIFT and Table 2 describes the same when using ORB.

The following figures show the ROC-AUC curves and confusion matrices obtained after implementing the different algorithms after using ORB feature extraction, which can be used to summarize the performance of the classifiers. The first classifier is KNN, which gave the highest accuracy. Figure 7 shows the ROC-AUC curve and Table 3 shows the confusion matrix, respectively. Area under curve for KNN algorithm: 0.96.

The second classifier is SVM (RBF), which gave the second highest accuracy. Figure 8 shows the ROC-AUC curve and Table 4 shows the confusion matrix, respectively. Area under curve for SVM (RBF) algorithm: 0.96.

Next classifier is Random Forest. Figure 9 shows the ROC-AUC curve of Random Forest and Table 5 shows the confusion matrix, respectively.

Area under curve for Random Forest algorithm: 0.95.

Last classifier is Decision Tree. Figure 10 shows the ROC-AUC curve of Decision Tree classifier and Table 6 shows the confusion matrix, respectively.

Area under curve for Decision Tree algorithm: 0.78.

The chart in Fig. 11 is a bar chart, which depicts the accuracy values of the classifiers when using both feature extraction techniques. After implementing both the techniques of feature extraction (ORB and SIFT), it was observed that the KNN

Table 1 Performance of classifiers when using SIFT

Sr. no.	Classifier	Recall	Precision	F1-score	Accuracy (%)
1	SVM (RBF)	0.74	0.80	0.77	78.00
2	SVM (Linear)	0.80	0.76	0.78	77.37
3	KNN	0.66	0.83	0.73	76.25
4	SVM (Polynomial)	0.74	0.75	0.75	74.62
5	Random Forest	0.69	0.73	0.71	71.87
6	Decision Tree	0.66	0.67	0.66	66.25

Table 2 Performance of different classifiers when using ORB

Sr. no.	Classifier	Recall	Precision	F1-score	Accuracy (%)
1	KNN	0.96	0.89	0.92	90.75
2	SVM (RBF)	0.90	0.92	0.91	90.25
3	SVM (polynomial)	0.89	0.92	0.90	89.37
4	Random Forest	0.88	0.90	0.89	88.00
5	Decision Tree	0.84	0.81	0.83	80.1
6	SVM (linear)	0.79	0.80	0.79	77.62

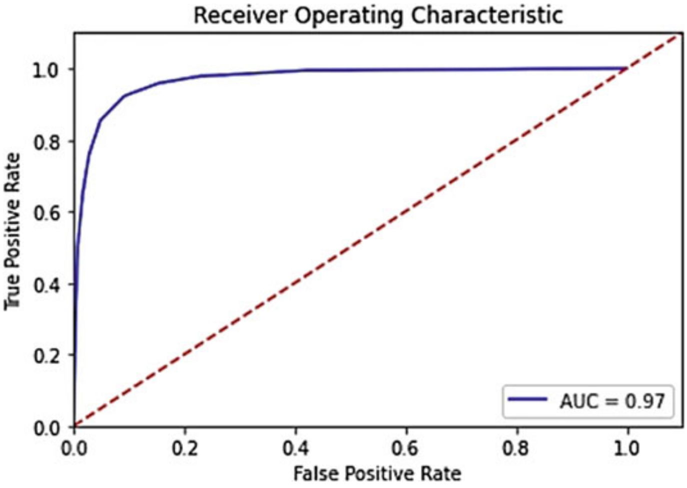


Fig. 7 ROC-AUC curve of KNN algorithm

Table 3 Confusion matrix of KNN algorithm

KNN	Actual positive	Actual negative
Predicted positive	414	21
Predicted negative	53	312

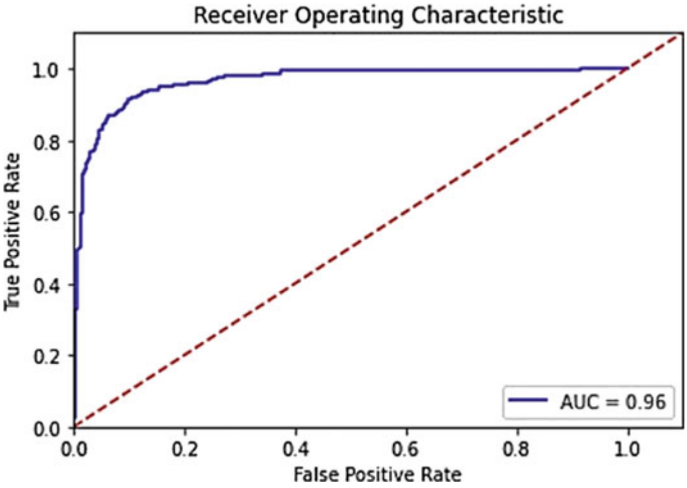


Fig. 8 ROC-AUC curve of SVM (RBF) algorithm

Table 4 Confusion matrix of SVM (RBF) algorithm

SVM (RBF)	Actual positive	Actual negative
Predicted positive	392	43
Predicted negative	35	330

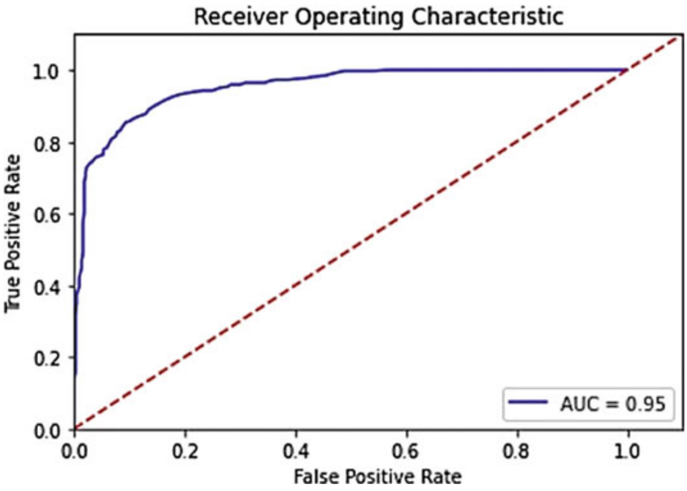


Fig. 9 ROC-AUC curve of Random Forest algorithm

Table 5 Confusion matrix of Random Forest algorithm

KNN	Actual positive	Actual negative
Predicted positive	389	46
Predicted negative	50	315

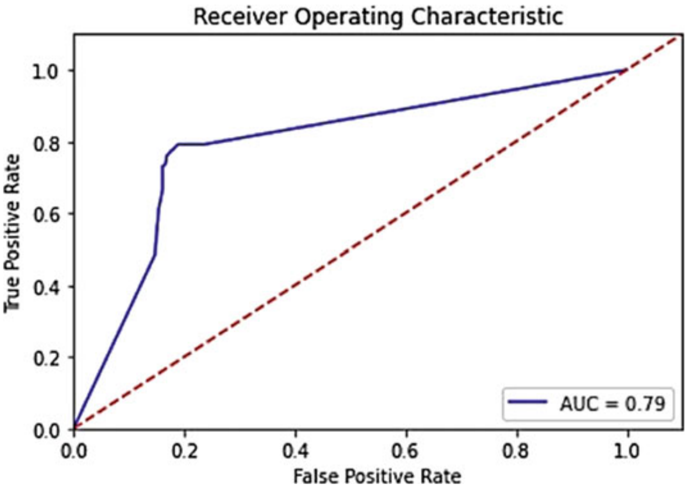


Fig. 10 ROC-AUC curve of decision tree algorithm

Table 6 Confusion matrix of decision tree algorithm

KNN	Actual positive	Actual negative
Predicted positive	358	77
Predicted negative	82	283

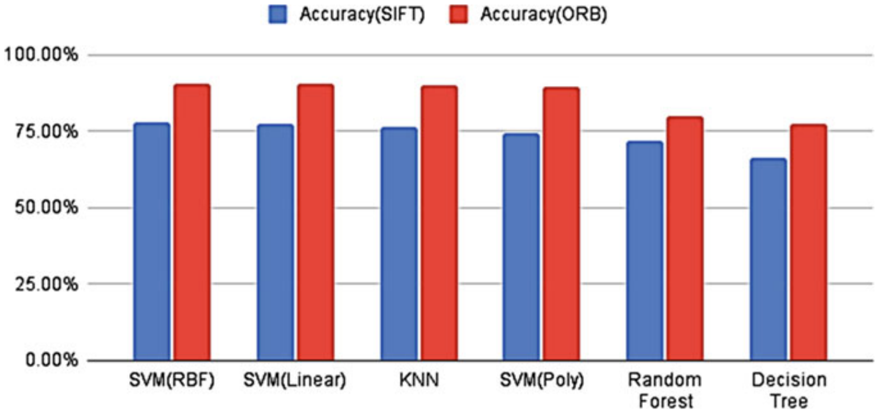


Fig. 11 Bar chart comparing accuracy of different classifiers

model has achieved an accuracy of up to 90.75% using the ORB feature extractor. As a result of this, for feature extraction, ORB is an efficient alternative to SIFT in the context of computation cost and matching performance.

In this study, the classifier techniques are introduced for the real-time detection of driver distraction, based on various life-threatening activities of the driver driving the vehicle. The proposed model based on supervised machine learning techniques has shown significant performance in constructing this detection model matching the performance level of CNN-based models. Deep learning methods are used by CNN frameworks like HCF to process the visual data and identify distracted driving behaviors. The majority of CNN models include cooperative pre-trained models that include ResNet-50, Inception V3, and Exception to extract behavioral information from the images in order to increase accuracy. Overfitting of the model occurs when the connected layers of the CNN framework are not trained to remove the filter anomalies from images. The entire system becomes more complex as a result. Various feature extraction strategies are used by the proposed model to extract features from the photos. In order to prevent the training model from overfitting the training data, dimensionality reduction methodology using PCA is used to implement the model. The results demonstrate that the suggested system recognizes distracted driving behaviors and performs well, performing similarly to CNN models with less complexity, with a classification accuracy of 90.75%.

The comparison between the proposed system and the systems that are now in place is laid out in the following table. The flow of the model, accuracy, limitations, and the outcomes of existing systems and the proposed system are described in Table 7.

The system that has been suggested is able to recognize and categorize the distracting activities that the driver is engaging in with precise prediction. Some of the currently in use systems have adopted a complex methodology, which not only makes the model more difficult to use, but also produces outcomes that are only marginally up to par.

Table 7 Comparison of proposed system with other existing studies

Name	Methodology	Accuracy (%)	Limitations	Outcome
Proposed System	SIFT and ORB feature extraction, machine learning classifiers	90.75	One side angle image	Classification of the driver's normal and distracted states
Hybrid CNN framework system	CNN techniques	96.74	Camera position different positions in vehicle	Real-time detection
Driver distracted Detection and Alerting System	Image processing algorithms and RGB-D sensor	96	Unable to detect distracted behavior at night	Complex implementation
Camera Vision System	Fusion of stereo vision and lane tracking data	80	not sufficient for the in-vehicle application	Needs to be robust for more accuracy

5 Conclusion and Future Scope

The author of this study suggests a technique for identifying distracted drivers. The image of the driver is taken as input, which is then processed and classified, and an output based on the classification is given out, which tells if the driver is distracted or not. When the SIFT feature extractor was used, the highest accuracy was 78.00% in case of the SVM RBF classifier. The highest accuracy of 90.75% was achieved using the KNN classifier with ORB feature extraction. The limitation of this system is that the dataset used here contains images taken from a single angle. To make it more efficient and better, images should be taken from different angles. Also, if the images are blurred or noisy, this becomes a problem, and the wrong prediction will be given out. The model can be deployed on simple hardware making it a handy and portable device. For this purpose, embedded system-on-modules (SOMs) like Jetson Nano or Raspberry Pi can be used.

References

1. Huang C, Wang X, Cao J, Wang S, Zhang Y (2020) HCF: a hybrid CNN framework for behavior detection of distracted drivers. IEEE Access 8:109335–109349
2. Kashevnik A, Shchedrin R, Kaiser C, Stocker A (2021) Driver distraction detection methods: a literature review and framework. IEEE Access 9:60063–60076
3. Kutila M, Jokela M, Markkula G, Rué MR (2007) Driver distraction detection with a camera vision system. In: 2007 IEEE International Conference on Image Processing, vol. 6, pp. VI-201. IEEE

4. Wathiq O, Ambudkar BD (2018) Driver safety approach using efficient image processing algorithms for driver distraction detection and alerting. In: Bhateja V, Coello Coello C, Satapathy S, Pattnaik P (eds.), Intelligent Engineering Informatics. Advances in Intelligent Systems and Computing, vol 695. Springer, Singapore
5. Torres, RH, Ohashi O, Garcia G, Rocha F, Azpúrua H, Pessin G (2019) Exploiting machine learning models to avoid texting while driving. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE
6. Qin B, Qian J, Xin Y, Liu B, Dong Y (2021) Distracted driver detection based on a CNN with decreasing filter size. IEEE Trans Intell Transport Syst
7. Mase JM, Chapman P, Figueredo GP, Torres MT (2020) A hybrid deep learning approach for driver distraction detection. In: 2020 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1–6. IEEE, 2020
8. Ai Y., Xia J, She K, Long Q (2019) Double attention convolutional neural network for driver action recognition. In: 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), pp. 1515–1519. IEEE
9. Omerustaoglu F, Sakar CO, Kar G (2020) Distracted driver detection by combining in-vehicle and image data using deep learning. Appl Soft Comput 96:106657
10. Kulkarni AS, Shinde SB (2017) A review paper on monitoring driver distraction in real time using a computer vision system. In: 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)
11. Poon Y-S, Kao C-Y, Wang Y-K, Hsiao C-C, Hung M-Y, Wang Y-C, Fan C-P (2021) Driver distracted behavior detection technology with YOLO-based deep learning networks. In: Product Compliance Engineering—Asia (ISPCE-ASIA) 2021 IEEE International Symposium on, pp. 01–05
12. Tianchi L, Yang Y, Huang GB, Yeo YK, Lin Z (2016) Driver distraction detection using semi-supervised machine learning. In: IEEE Transactions on Intelligent Transportation Systems 17(4):1108–1120
13. Tango F, Botta M (2013) Real-time detection system of driver distraction using machine learning. IEEE Trans Intell Transp Syst 14(2):894–905
14. Liang Y, Reyes ML, Lee JD (2007) Real-time detection of driver cognitive distraction using support vector machines. IEEE Trans Intell Transp Syst 8(2):340–350
15. Methuku J (2020) In-Car driver response classification using Deep Learning (CNN) based Computer Vision. IEEE Trans Intell Vehicles
16. Moslemi N, Soryani M, Azmi R (2021) Computer vision-based recognition of driver distraction: a review. Concurrency and Computation: Practice and Experience 33(24):e6475
17. Yan C, Coenen F, Zhang B (2016) Driving posture recognition by convolutional neural networks. IET Comput Vision 10(2):103–114
18. Eraqi HM, Abouelnaga Y, Saad MH, Moustafa MN (2019) Driver distraction identification with an ensemble of convolutional neural networks. J Adv Transport 1–12
19. Alotaibi M, Alotaibi B (2020) Distracted driver classification using deep learning. SIViP 14:617–624

Enhancing Security and Performance of Software Defect Prediction Models: A Literature Review

Ayushmaan Pandey and Jagdeep Kaur

Abstract There have recently been many advances in software defect prediction (SDP). Just-in-time defect prediction (JIT), heterogeneous defect prediction (HDP), and cross-project defect prediction (CPDP) are now used for better performance. Some issues remain a hurdle in the proper implementation of the SDP models. Security, class imbalance, heterogeneity, and redundant and irrelevant features are some of the significant issues. This research study provides a systematic literature review (SLR) on various approaches to solving the above-mentioned issues. Research papers from various well-reputed online libraries such as IEEE, Springer Link, and ScienceDirect have been reviewed critically in a systematic manner. Three well-defined research questions on the issues of security, class imbalance, and feature selection have been addressed. Recent papers relevant to the topics were reviewed systematically to answer the questions. The effectiveness of privacy-focused SDP model techniques has been examined in this paper. In this study, various techniques that can help developers manage the issue of class imbalance have been mentioned that can improve the predictive performance of the SDP models. Effective feature selection techniques are reviewed, providing a solid basis for future extension and application of these approaches on commercial datasets. The developers introduced promising frameworks, such as JIT with LIME, FILTER with SVM, LTCA, CDAA, FRLGC, and FTLKD. The datasets used for the research and application were from public repositories like NASA, PROMISE, and Re-Link.

Keywords Software defect prediction · Just-in-time models · Support vector machine · Class imbalance · Feature selection

A. Pandey (✉) · J. Kaur
Department of Computer Science and Engineering, Dr. B.R. Ambedkar National Institute of Technology, Jalandhar, Punjab, India
e-mail: ayushmaanpandey10@gmail.com

J. Kaur
e-mail: kaurj@nitj.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_13

1 Introduction

Software defect prediction is a crucial part of the SDLC. More accurately, it is part of the testing phase of the SDLC. Based on their training on some previous data, SDP models can predict whether a code change or any code addition will be defective or not. SDP is considered an essential tool, and if we effectively predict defects in software modules, then any organization can save a massive amount of its resources. Many updates and launches of products would have to be rolled back if defect predictions were not made correctly. The project cost could increase, and there would be delays and a waste of time. Therefore, machine learning is used to predict buggy software changes and save a lot of resources [19]. In software defect prediction, there are two classes in which any code change can be classified. One is a defect-inducing code change, and the other is a non-defect-inducing or clean code change class. About 80% of the defect-inducing code changes in most projects can be found in just 20% of the software module changes. Therefore, there is a class imbalance between the defect and non-defect classes.

Class imbalance is a very persistent and significant issue in software defect prediction. Almost every dataset available for SDP has a class imbalance. Class imbalance arises when there are more instances of one class than the other, resulting in the majority class being that of the first class (here, non-defect-inducing) and the other becoming the minority class (here, defect-inducing). Many techniques have been discussed to solve this issue and bring balance to the class. The most basic and popular solution for class imbalance is oversampling and undersampling techniques. Filtering techniques have also been proposed to solve the issue of class imbalance. It was observed that reducing class imbalance could enhance the performance of the SDP models. Another way to increase the predictive performance of models is by applying effective feature selection to datasets.

Feature selection is selecting a suitable subset of the original dataset for better efficiency and accuracy. The generated or commonly available datasets used for software defect prediction generally have a lot of features. Therefore, it is crucial to perform pre-processing on data before building a model for software defect prediction to get optimized and better results. Irrelevant and redundant features in a dataset can reduce the efficiency and predictive performance of many classifiers [17]. There are many techniques for feature selection, such as regularization techniques (L1 norm regularization, etc.), greedy search algorithms (K-nearest neighbors, etc.), and feature importance techniques (random forest algorithm, etc.). Feature selection can reduce overfitting, make data less redundant, increase accuracy, and reduce the training time of models. When redundancy is reduced, fewer decisions are made based on noise. Since misleading data is also reduced, the accuracy of the models is greatly improved since fewer training data points are available that make the model run faster.

Another major issue is the security of the datasets used to construct an SDP model. When there is a single project as a source and a target project, it is called With-in

Project Defect Prediction (WDPD). Generally, WDPD is used for SDP. When a software project is new and insufficient historical data is available to train an SDP classification model, a different method of defect prediction is used. In CDPD, the model is trained on different source projects and then applied to a target project to make defect predictions. Since data from various public and private projects is used, there is a need to map them in a single feature space to eliminate the heterogeneity problem. The dataset used for SDP contains vital and sensitive data about an organization's project. If an attacker gets access to this data, they can misuse it. Hence, there is a concern regarding the privacy of data used for software defect prediction. In the literature on the subject, various ways have been discussed to solve the issue of privacy protection. An approach-based on homomorphic encryption, differential privacy, etc. is followed. For their highly secure nature and directness, partial homomorphic encryption techniques are becoming very popular for secret data sharing [15].

2 Research Approach

A Systematic Literature Review (SLR) consists of predefined research questions that need to be answered through a well-defined and organized method to analyze more than one study or work. One of the initial steps in making an SLR includes defining a research protocol that helps us identify appropriate research questions to be answered. The research approach and criteria for the selection of papers, as well as which information to include, are also mentioned. Review planning, carrying out the review, and reporting the review are the three stages of the SLR process.

2.1 Research Question

This study aims to gather information on different techniques that can help improve the efficiency and security of models and approaches defined for SDP by emphasizing evaluation standards, simulation tools, and datasets from open repositories. An SLR's ultimate objective is to conduct a critical evaluation to discover the answers to the research questions that represent these objectives. The research questions that need to be answered in this research are mentioned below

RQ1. What are the security aspects of software defect prediction and what are the different techniques to solve data privacy issues?

RQ2. What are the most effective techniques for feature selection to get better results and remove heterogeneity in SDP models?

RQ3. Which techniques can solve the class imbalance problem, and what effect does it have on the performance of SDP models?

2.2 Data Sources

Different libraries and search spaces are data sources for accessing research papers for our studies. To conduct this research, several reputed data sources such as "IEEE Xplore," "IEEE Access," "Springer Link," and "ScienceDirect" have been visited to access the primary papers.

2.3 Paper Selection Criteria

Research papers published in English from 2018 to 2022 are prioritized to conduct this study. The focus is to select research papers that have developed approaches to make the predictive performance of SDP models more accurate and secure. Empirical studies that include experiments on public, open-source, and commercial datasets to efficiently classify software defects and then present the evaluation of these experiments are selected.

The latest published research papers are given more preference. The selected research papers can only belong to journals, conferences, or books. Discussion on security challenges in SDP and their solutions is one of the selection criteria for research papers. Papers that focused on solving the class imbalance issue and providing effective feature selection methods for improving the predictive performance of classifiers were selected for conducting this research.

2.4 Data Extraction

The following details have been extracted from the data obtained from each primary research paper: proposed/used class imbalance solving technique, proposed/used feature selection technique, proposed/used privacy protection technique, criteria of performance evaluation, datasets utilized for the experiments, and comparison of different methods for software defect prediction.

As shown in Fig. 1, recent research studies have been reviewed to better understand the current work in SDP.

3 Findings

RQ1. What are the security aspects of software defect prediction and the different techniques to solve data privacy issues?

In SDP, the defective modules of software are predicted, and this saves us some testing resources and costs. Privacy issues are rising in SDP as datasets and modules

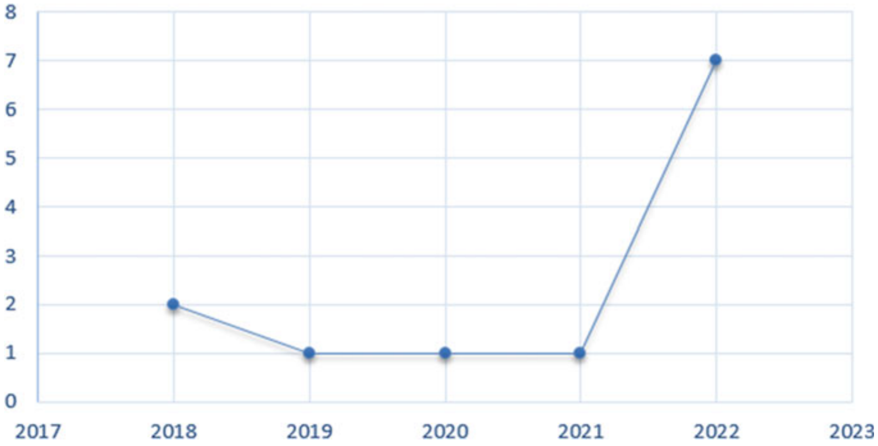


Fig. 1 Year-wise proportion of primary studies

can become vulnerable to attacks in a prediction model. The data used for training and testing SDP contains business privacy and protection. If sensitive information gets into the hands of an attacker, then the software will become vulnerable to various attacks. When a project is new and the proper amount of historical data is unavailable, then heterogeneous defect prediction is used. In heterogeneous defect prediction, defective software modules of the target project are predicted using a model prepared by training it on various source projects. In this scenario, two issues must be addressed: data islands and data privacy. The motive behind using HDP is that collecting information from multiple sources can be more helpful. Building a model based on information collected from various projects can improve the model's performance. The critical issues in HDP that need to be addressed are data privacy, security, efficiency, and accuracy. There are various major issues with heterogeneous defect prediction models. As the data has heterogeneity, there is a difference between the features of public and private projects, and the models or approaches can't be applied directly. Some data pre-processing needs to be performed, which leads to the exposure of data privacy for private projects and, due to the absence of laws and regulations, can prove illegal. Some projects can only share data among themselves and form data islands.

In [9], a homomorphic encryption-based novel HOPE method is proposed to solve privacy issues. Homomorphic encryption provides semantic security. It is not easy to get a relationship between plaintext and ciphertext. There is unidirectional encryption. HOPE is a method based on a homomorphic encryption approximation algorithm used for logistic regression. The sigmoid function in logistic regression is approximated. The client sends encrypted data using the Paillier algorithm to the server. Then, based on this information, the server calculates the weighted matrix using the HOPE method, and the encrypted weighted matrix is sent back to the client.

Then, at the client end, decryption of the weighted matrix is performed, and a software defect prediction model using logistic regression is created. The experiment is performed in the form of three control groups: Group 1 deals with unencrypted data at each step; in Group 2, the client sends encrypted data, and the server is unaware and deals with it as unencrypted data; in Group 3, the server is aware and uses the HOPE method for further processing. The experiments were conducted on the MORPH open-source real-world dataset. For purposes of testing the predictive performance of a given method, accuracy and AUC are used as performance measures. It was shown that the prediction model's accuracy is approximately 81% when applied to unencrypted data but drops to 50% when an attacker develops the model using encrypted data. Hence, the attacker cannot get any useful knowledge from the cipher texts, which means HOPE ensures security. When the server uses the HOPE method to encrypt the data and form a cipher text, the final prediction is about 79% accurate. As a result, the performance of the categorization is not greatly impacted by the method used. Security can be developed further by utilizing various brand-new, cutting-edge homomorphic encryption techniques to enhance HOPE's functionality.

Using datasets from prime repositories in SDP in most studies could put external legitimacy at risk. SDP models can be shared instead of SDP datasets to solve this issue. But SDP model sharing can bring up other privacy-related problems like model inversion attacks. This work used a differential privacy method for SDP model sharing. In [10], a novel method called A-DPRF was introduced as differential privacy can be very effective even when the privacy budget is selected carefully. The novel method proposed is completed in three steps. The first step is data pre-processing, then a novel sampling method is performed, and lastly, classification via the random forest is achieved. The first step performed in the experiment is data pre-processing, in which the minority class is oversampled and continuous features are discretized. Data pre-processing can help us solve the class imbalance problem, and the discretization of attributes will lead to better model performance. In the next step, the novel sampling method, i.e., A-Sampling, is used to generate a random forest using the Laplace and exponential methods and to produce training sets for each decision tree. The PROMISE repository was utilized to provide the datasets used to validate the results. This work uses B-DPRF and random forest as the benchmark approaches, and A-DPRF is compared to them. It was found that these approaches function better when SMOTE is used. Also, with the increase in privacy protection, the performance of the A-DPRF model will decrease. The baseline method RF performs the best, but A-DPRF performs better than B-DPRF. As shown in Fig. 2, it can be observed that NASA is the most frequently used dataset. Most of the papers use open-source public databases. PROMISE and AEEEM are also very frequently used.

In HDP, defects are predicted in one software module using data collected from different projects. In this scenario, two issues must be addressed: data islands and data privacy. The motive behind using HDP is that collecting information from multiple sources can be more helpful. Building a model based on information collected from various projects can improve the model's performance. The key issues in HDP that need to be addressed are data privacy, security, efficiency, and accuracy. Federated transfer learning is one such way to solve the problems mentioned above by improving safety. The three components of the FTLKD methodology described in [11] are private data encryption, private model creation, and communication. The first step is to perform encryption on private data. Data was homomorphically encrypted using Shamir sharing technology, and during subsequent processes, the data was only retained in its encrypted form. Since the information is encrypted or cipher text, all actions are carried out directly. Use public data to train a CNN model during the private model development step, and then fine-tune the model via model transfer learning. After that, a secret model is given the parameters obtained in this stage. They have updated the private model using knowledge distillation throughout the communication step. In each iteration of this stage, we update the secret model with a portion of the new public dataset. The update comes to an end after a certain number of rounds. They used three publicly accessible databases for software defect prediction in the experimental setting: NASA, AEEEM, and SOFT LAB. One database is chosen as the public data source, while six private datasets from other projects in other databases are used in three tests. It was found that when the communication rounds increase, all the measurements increase to varying degrees. The growth rate is rapid from around 0 to around 8, after which it stabilizes. NASA, AEEEM, and SOFT LAB are three publicly available databases we employ to anticipate software defects in the experimental context. Here, six private datasets from other projects in other databases are used in three additional tests, and one database is chosen as the public data source. The research used accuracy, AUC, and G-mean as performance measures to evaluate how well predictions performed.

It was discovered that all metrics increase to varying degrees as the communication rounds increase. The growth rate picks up speed from round zero through round eight before stabilizing.

In HDP, heterogeneous data collected from various projects is used to build a model for predicting a defect in the software modules prone to errors in a private project. HDP is very useful in the case of new projects, as they don't have much historical data within the project. In the absence of historical data, HDP can provide developers and testers with the required solution in the absence of heterogeneous data from various other sources. There are various significant issues with heterogeneous defect prediction models. As the data has heterogeneity, there is a difference between the features of private and public projects, and the models or approaches can't be applied directly. This needs some kind of data pre-processing to be performed which leads to the exposure of data to private projects and, due to the absence of laws and regulations, can prove illegal. Some projects can only share data among themselves and form data islands. Federated learning is a new machine learning approach that can solve the problems mentioned above [16]. Therefore, this paper proposes a novel Federated Reinforcement Learning via Gradient Clustering (FRLGC) approach. The FRLGC approach proposed in [12] k has several steps to be followed, which are carried out according to the outline provided in federal learning. We use Principal Components Analysis (PCA) for dimensionality reduction in the pre-processing data. The use of PCA in the pre-processing step removes issues like data redundancy and creates a common space of features among source projects. Then, all the clients' deep Q network (DQN) data is trained locally.

Then, for better privacy and security, Gaussian differential privacy is applied, and parameters belonging to each local model are encrypted. Then the researchers performed model aggregation by selecting some clients and forming their clusters using K-means. Then, at each cluster, local aggregation is performed, and the central collection is committed to building a global model. Each client goes through 4 stages: data pre-processing, local training, data encryption, and model aggregation before the global model is broadcast to all. Updates are stopped when the private model of each client converges to the maximum number of communication rounds. The following step involves conducting evaluations and comparing the outcomes. They employed three class-unbalanced datasets from the SDP fields of NASA, AEEEM, and Re-link to assess the performance of the FRLGC algorithm. AUC and G-mean are used as performance metrics for better evaluation of predictive performance. AUC and G-mean mean values when there are four clients are 0.65 and 0.55, respectively, and these values are maintained in subsequent rounds with five and six clients.

The proposed FRLGC method was compared against various non-federated and federated learning methods to judge its effectiveness. The two federated learning methods used for comparison were Fed-Avg and FTLKD, whereas the three non-federated learning methods were CCA+, KCAA+, and KSETE. The experiments were performed on all the combinations of the dataset used. In rounds with five and six clients, the times rise by 16.752 s and 129.407 s, respectively, compared to rounds with four clients. Compared with the federated and non-federated models, FRLGC performed better and gave an increased value of AUC and G-mean. In the future, more defect datasets can be involved to check the generalizability of FRLGC; since this algorithm needs multiple communication rounds, we can further reduce communication costs to get better results.

Summary SDP involves operations on data that is very crucial to any company. If data is not protected then any attacker would have access to it and data can be misused. Homomorphic encryption, differential privacy and federated learning are some of the techniques that can provide privacy protection. Various models based on these techniques, such as, HOPE, FTLKD, FRLGC and A-DPRF were introduced. These models performed operations on encrypted data and preserved privacy. AUC, G-mean and accuracy were used performance metrics. It was observed that these models provided security while preserving the predictive performance of the classifiers or learners.

RQ2. What are the most effective techniques for feature selection to get better results and remove heterogeneity in SDP models?

The process of choosing a suitable subset of features from the initial set of features is known as feature selection. Feature selection, when appropriately done, comes with a lot of advantages. It can effectively reduce model complexity, increase the model's performance, and reduce noise induced by irrelevant features, reducing the generalization error. There are many techniques for feature selection, such as regularization techniques (L1 norm regularization, etc.), greedy search algorithms (K-nearest neighbors, etc.), and feature important techniques (random forest algorithm, etc.). The generated or commonly available datasets used for software defect prediction generally have a lot of features. Therefore, it is essential to perform pre-processing on data before building a model for software defect prediction to get optimized and better results.

Irrelevant and redundant features in a dataset can reduce the efficiency and predictive performance of many classifiers. Feature selection can reduce overfitting, make data less redundant, increase accuracy, and reduce the training time of models. When redundancy is reduced, fewer decisions are made based on noise. Since misleading data is also reduced, the accuracy of the models is greatly improved since we have fewer training data points that make the model run faster.

In [5], nested stacking and heterogeneous feature selection were introduced to prove SDP models' efficiency and resource allocation. This procedure is divided into three steps: feature selection and dataset pre-processing, nested-stacking classifier, and model classification performance assessment. Class imbalance and redundant or unnecessary features will hurt the model's accuracy and are the main problems with SDP.

Nested Stacking (NS) uses techniques such as heterogeneous characteristic selection and normalization to enhance the features of the data at some point throughout the data pre-processing step, enabling the model to produce superior classification results. Nested Stacking Classifiers are built on the idea of stacking several different baseline modes together to improve classification performance overall. There are three layers altogether in the nested-stacking classifier. Three boosting algorithms, as well as a straightforward stack, combine to form the first layer. It is nested and includes MLP and Random Forest. The Gradient Boosting Decision Tree (GBDT) is the meta-classifier. The final layer is the meta-classifier of Logistic Regression, which makes the final classification. The performance metrics used in [5] are precision, recall, F1-score, and AUC. They have applied Within-Project Defect Prediction (WPDP) and Cross-Project Defect Prediction (CPDP) to both the Kamei and PROMISE datasets to get better results.

The experimental findings for WDPD Kamei demonstrate that the proposed stacking method has an increase in AUC and an improvement in the F1-score compared to other JIT models. According to the experimental findings based on the cross-validation conducted inside the WPDP PROMISE project, the F1-score of NS has grown. In the case of CPDP Kamei, the experiment results show an increase in AUC and F1-score, respectively. In the case of CPDP PROMISE, the experiment results on cross-project cross-validation show an average rise in F1-score, respectively. There are some datasets in which the other models outperform nested-stacking classifiers, but mostly they get better and more consistent results. In the future, they can work on an automated prediction system that can prepare the most optimized model combined with suitable parameters to improve efficiency.

Irrelevant and redundant features in a dataset can reduce the efficiency and predictive performance of many classifiers. We can perform dimensionality reduction to solve this problem, but data loss due to poor attributes can significantly affect the model's accuracy. In [6], they applied the Local Tangent Space Alignment method for feature extraction and SVM for prediction. LTSA requires very few parameters to be adjusted and is a very robust learning method. LTSA outperforms other famous learning algorithms like Laplacian Eigen-mapping (LE) in the ability of feature extraction and selection. Hence, it increases classification accuracy. SVM is a popular classifier that can work with higher dimension data and give us good predictive performance. The model has been designed sequentially. LTSA is first applied to the dataset with its parameters (d and k) determined by grid search, and the dimension of the dataset is reduced.

Then the reduced set is used as input for the SVM. The kernel function applied here is the radial basis function (RBF). Then the model is optimized and tested using TFCV. The datasets used to verify the experiment are CM1, KC3, PC1, PC4, MC2, and MW1. In ten-fold cross-validation (TFCV), the input dataset is divided into ten parts. One is taken as a test set, and the other is used for training. A total of ten experiments are conducted in this manner, and the average of their performances is taken. The model is applied to all the datasets, and its performance is noted separately. As compared to single SVM and LLE-SVM, LTSA-SVM performs better. It has better prediction precision and has an increased value of F-measure.

The Heterogeneous Defect Prediction (HDP) model will map public and private projects into the same feature space by considering metrics. This problem is termed the heterogeneous problem. HDP models apply different techniques to bring together public and private metrics into the same feature space. In [7], a novel Conditional Domain Adversarial Adaptation (CDAA) is proposed to deal with the heterogeneity mentioned above in SDP. The CDAA network comprises three layers or underlying networks: generator, discriminator, and classifier. The generator is the feature extractor; the discriminator is used to differentiate between source and target instances, and the classifier is there to classify the instances. This model was then implemented to find out whether label information is useful for heterogeneous problems in SDP or not. Several effective performance metrics were used to evaluate the model. 28 projects from five distinct public repositories were subjected to the model's application. The experiment's findings indicate a significant performance improvement between the CDAA without target training data and the CDAA with ten percent target training data. According to the statistical examination of the findings between CDAA and other approaches, CDAA could perform better than the CPDP and HDP methods. In the future, this model can be applied to more software projects and expanded to commercial projects to evaluate this method. The problem of class imbalance also needs to be addressed in HDP.

Earlier, SDP was mostly performed within a project, and this kind of software defect prediction was known as WPDP. As all the processes were performed within a project, the source and target data were from the same project. But in the case of new projects, an appropriate amount of historical data is not present. The training data is insufficient. Therefore, Cross-Project Defect Prediction (CDPD) was introduced.

Here, the training is done on the data from the source project, and predictions of defects in target projects are made.

The feature sets or space of the private and public projects should be the same for CPDP models to work, but sometimes we have to perform data pre-processing to make them the same.

In [8], a Two-stage Cost-sensitive Local Model (TCLM) was proposed to meet challenges like characteristic selection, class imbalance, and feature extraction. TCLM divides the prediction model into two parts: characteristic selection and training of the model, with cost information added to each stage independently. Cost information-based feature selection is carried out during the feature selection stage to identify the best feature subset for the source project. Using TCLM, issues with feature adoption, data adoption, and class imbalance are fixed.

Additionally, there is less overlap between features. To further eliminate heterogeneity, data from both private and public projects is translated to high-dimensional kernel space. The source project is then separated into k subsets using a clustering technique, and local predictive models are trained on each subset independently. Class imbalance is addressed by inducing cost-sensitive learning. The model training step gives different weights to defective and non-defective classes. For the evaluation of the performance of the proposed TCLM method, five projects were selected from AEEEM, eight projects from NASA, eight projects from PROMISE, and three projects from Re-Link. TCLM was compared with related methods in other literature to investigate its effectiveness. TCLM was compared with state-of-the-art methods, such as CCA+ and HDP-KS, for heterogeneous CDPD. AUC was chosen as the performance metric to evaluate the performance of the TCLM method. In 15 out of the 24 datasets used for comparing and evaluating performance, TCLM outperforms other state-of-the-art methods. The performance of the global and local models was also reached by changing the k (cluster) value between 1 and 10. One means that all the data available is used as the training data, and a global model is built. The model performed the best for $k = 3$. K-means replaced the k-medoids method. It was observed that the K-medoids outperformed K-means in every aspect. In future work, the most appropriate number of clusters for adaptively used datasets can be worked on. The proposed approach will be applied to more open-source and commercial projects for better evaluation.

Summary Feature selection is another effective way to enhance the performance of prediction models. Feature selection can also solve the problem of heterogeneity which makes CPDP models more efficient. Nested-Stacking classifier, LTSA-SVM, TCLM and CDAA were different techniques that were reviewed in this study. AUC, F-measure and accuracy were used as performance metrics for evaluation and comparison purpose. These feature selection techniques improved the performance of classifiers and outperformed other models such as, LLE-SVM, CCA+ and HDP-KS.

As shown in Fig. 2, it was observed that NASA is the most famous public repository for SDP. Other popular public repositories are AEEEM and PROMISE.

RQ3. Which techniques can solve the class imbalance problem, and what effect does it have on the performance of SDP?

Class imbalance is a very persistent problem in the process of SDP. Class imbalance provides a hurdle for software defect prediction models to better predict performance. In software defect prediction, any code change submitted can be classified as defect-inducing (faulty) or non-defect-inducing (non-faulty). In an ideal scenario, for the best results, the frequency of occurrence of a faulty class and the frequency of occurrence of a non-faulty class should be almost equal. But 80% of the defects are often found in only 20% of the software change modules. The faulty or defect-inducing class is very important but very scarce. In this case, the non-faulty class becomes the majority, and the defective class becomes the minority in the case of SDP, creating a class imbalance. This imbalance or unequal distribution of the classes in the training dataset can lead to the less efficient predictive performance of the SDP. Class imbalance can also be represented as the result of the division between the majority class and the minority class and is known as the imbalance ratio, or IR.

$$\text{IR} = \text{Majority} \div \text{Minority}$$

There are several ways to handle the class imbalance issue, and they are quite easy to understand and implement. Some of these methods are Random Undersampling (it removes the instances of the majority class randomly to reduce class imbalance), Random Oversampling (it adds the copies of instances from the minority class to

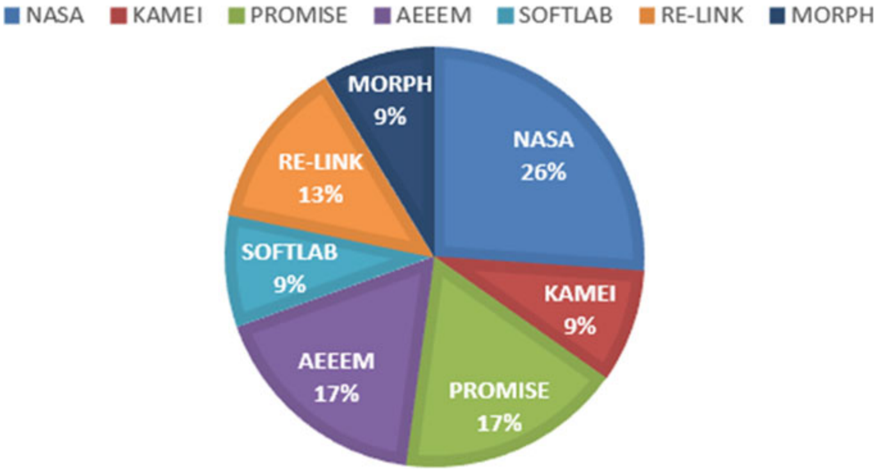


Fig. 2 Types of dataset repositories

reduce class imbalance), Undersampling using Tomek links (two instances of opposite classes that are very close are called Tomek links; here, the instances from the majority class in these links are removed to reduce class imbalance), and SMOTE (Synthetic Minority Oversampling Technique: we add synthetic points between an instance of k-nearest neighbors). In [2], the authors have discussed an effective SDP model using Support Vector Machines (SVMs) and a novel FILTER technique to restore the balance of the class.

SVM is used for classification as well as regression problems. The SVM algorithm draws or creates the best line among data points to segregate them into classes. The best line is often referred to as the decision boundary and is used to put new points into the correct category. The extreme points, or the border points that help define the decision boundary, are known as support vectors (hence the name of the classifier Support Vector Machine). SVM can be linear or non-linear depending on the data type; if the data can be separated linearly, linear SVM is used. If the data is not linearly separable, a straight line cannot be drawn to segregate the classes, and then non-linear support vector machines are used. So, to classify non-linear data, use kernel functions with support vector machines.

The Kernel functions transform the non-linear data surface in a single plane into a linear equation in a higher dimensional space. Kernel functions can work based on two mathematical facts: (i) Every kernel function can be expressed as dot products, and (ii) Many learning algorithms can be expressed in terms of dot products. Several kernel functions can be applied with SVM. In [1], three SVM kernels were used with and without the novel FILTER technique they have introduced for resolving class imbalance. The SVM Linear Kernel is the most basic kernel function that is one-dimensional, and it is used mostly for text classification problems, as they are linearly separable. When compared to other kernel functions, it is less efficient and less accurate. It represents the linear kernel more generally. SVM RBF is frequently used with non-linear data. When no prior knowledge of data exists, it helps to make the proper separation. Six models were tested to get a broader view of the results.

The FILTER technique is proposed to filter out instances that are not faulty within a certain radius of faulty instances, thereby restoring the class balance necessary for improved performance. The dataset is CM1, KC1, KC2, PC1, and JM1. The dataset is divided into 80% for training and 20% for testing. On average, the IR was reduced by 18.81%, and the false negatives (instances classified as non-faulty but faulty) were reduced accordingly. The SVM models with filtering have shown an increase in performance of 9.32% in SVM Linear, 16.74% in SVM-RBF, and 14.06% in SVM-Polynomial. SVM-RBF performed the best with 95.68%, 96.48%, and 94.88% for precision, AUC, and F-measure, respectively. In terms of precision, AUC, and F-measure, the suggested FILTER method enhances the overall performance of the SVM-RBF SDP.

Just-in-time prediction models are now used very often for SDP. These models perform analysis whenever a code change has been made and its submission is made, predicting the likelihood of a defect. Defects are predicted in every code change submitted by the developer [13]. The JIT models provide high speed, directness, fine granularity, and defect tracking. However, there are some major issues with the JIT model, and they provide a hurdle for the accuracy and performance of these models. Class imbalance, irrelevant features, and redundant features are some of the issues mentioned above.

In [2], a JIT model is applied for software defect prediction based on the Random Forest Classifier, and its performance is recorded. Then they used Local Interpretable Model-agnostic Explanations (LIME) to find relevant and important features. Then we again calculate the performance after removing irrelevant features. LIME adds disturbance to every sample input feature, and then they check the change in output. LIME is a very flexible and model-agnostic interpretability method that can be applied to any machine learning model. LIME tinkers with the feature values of a single data sample and monitors the results. In this way, they find the relevant features. As the dataset can be large, SP LIME is automatically used to search for a suitable sample set. The experiment was conducted on six open-source projects. Accuracy, recall, F1-measure, and AUC are the indicators that are employed in this. The JIT SDP model achieves a recall rate of 68.88% and an accuracy of 71.52%. When we applied LIME and SPLIME to find the most influential features and only those features were used for prediction, 96% of the original workability was attained with only 45% of the effort.

In [3], it was mentioned that concept drifts, or changes to defect-generating processes, are one of the most common problems with JIT-SDP. They observed different values for the percentage of examples of each class over time, leading to changes in the percentage of examples of each class. Another issue in JIT-SDP is verification latency. Software change labels can come at a later time than the commit time. These issues can hinder the predictive performance, stability, and reliability of JIT-SDP. They have used online JIT-SDP models, which constantly update classifiers with new training examples. Concept drifts can introduce class imbalances in the training dataset as, with time, the number of instances of clean and defect-inducing classes can change. Concept drift can result in a clean class being a majority class and defect-inducing as a minority class. Various methods have been discussed earlier to deal with this problem. Some of these methods are Sliding Window (we slide the window and use the most recent training examples), Oversampling Online Bagging (OOB), and Undersampling Online Bagging (UOB) [18]. To achieve balance, oversample the majority class and undersample the minority class. The state-of-the-art method before this work was Oversampling Rate Boosting (ORB). Depending on the degree of imbalance in the forecasts, it modifies the sampling rate. In the previous work, it was observed that ORB outperformed every other model. In this paper, they check its reliability. The performance metrics used here were as follows: 1. Rec (0): recalls on clean classes, 2. Rec (1): recalls on defect-inducing classes, 3. $|\text{rec}(0) - rec(1)|$: Absolute difference of both the recalls, and 4. G-mean of rec (0) and rec (1).

These measures were chosen because they can be used to evaluate prediction performance in the learning process for class imbalance. It was found that ORB is better than other methods, but it is not stable as it has high |rec(0) − rec(1)| as 20 units in fifty percent of the datasets, which means one recall is good and the other is 20 units bad. The standard deviation of the metrics used is pretty high for all the methods. Hence, they are not that stable either. To solve the problem of stability and reliability, they have introduced Prediction-Based Sampling Adjustment (PBSA), a novel online JIT model. In PBSA, keep an eye on the moving average of earlier predictions. The monitoring of the moving average can be equal to the monitoring of the class imbalance ratio. The concept recovery mechanism of PBSA can provide more stability in predictive performance. In this mechanism, a process of recovery training with examples from classes other than the biased class, which is the most recent, is performed. PBSA outperformed the state of the art in every performance metric used in most datasets. However, compared to the standard models, the proposed PBSA could deliver a more stable performance than ORB.

Summary It was observed that solving the issue of class imbalance can enhance the performance of prediction classifiers. If data is pre-processed and balance is maintained to the class then false negatives will reduce and precision of model will increase. Performance of famous classifying learners such as SVM can be improved through introducing a FILTER technique that reduces IR. LIME is an interpretability technique that can help us reduce our efforts and provide similar results as before. PBSA is a new sampling technique that gave better results when compared to the state of the art ORB sampling technique.

Classify the data into faulty or non-faulty classes based on a labeled dataset in software defect prediction.

Here, one of the major issues is class imbalance. The non-faulty class can become the majority class. As a result, the model can get overfitted and classify the defect-inducing changes as non-defective. There are various techniques to handle the issue of class imbalance and improve the performance of classifiers. In [4], oversampling-based techniques are used. They introduced a novel oversampling-based ensemble method and tested it on PROMISE datasets. They have combined random oversampling, Majority Weighted Minority Oversampling Technique, and Fuzzy-Based Feature Instance Recovery to build an ensemble classifier. These three will generate duplicate or pseudo instances of the defective class and bring down the class imbalance ratio. This way, the performance will eventually increase. Then the sample will be classified by each learning method, and then either the average of the results will be taken, or some voting strategy will be applied to get the final results. The metrics taken here to measure the predictive performance were true positive rate, false negative rate, AUC, and F-measure. The proposed method gives us the lowest average false positive rate.

Also, the proposed method scores the highest AUC in all datasets. The proposed method has a better recall rate and accuracy than the previous state-of-the-art methods. This ensemble method can be tested in the future using other classifiers as base classifiers, such as SVM, neural networks, and decision trees (Table 1).

Table 1 Different techniques and their respective issue solved

References	Techniques	Issues solved
[2]	Applied LIME interpretability to a JIT-SDP model to perform better feature selection	Feature selection, class imbalance
[1]	A novel FILTER technique was introduced to solve the class imbalance and improve the performance of the SVM SDP model	Class imbalance
[3]	A PBSA technique was implemented for better stability in the performance of SDP models through concept drift recovery	Class imbalance
[5]	For better performance, a nested-stacking classifier with heterogeneous feature selection was made	Feature selection
[6]	The LTSA technique was used for feature selection and reduction to improve the performance of the SVM classifier	Feature selection
[4]	They have combined random oversampling methods with fuzzy instance recovery to build a novel ensemble classifier	Class imbalance
[7]	A novel CDAA is proposed to deal with the problem of heterogeneity in SDP	Feature selection
[9]	A logistic regression-based novel HOPE method and a homomorphic encryption scheme are proposed to solve the privacy issues	Security
[10]	A novel method A-DPRF was introduced as differential privacy can be very effective even when the privacy budget is selected carefully	Security
[11]	The FTLKD privacy-focused approach was designed and executed to provide security and better performance	Security
[12]	The FRLGC privacy-focused approach was designed and executed to provide security and better performance	Security
[8]	This work proposed a TCLM technique to meet challenges like feature selection, class imbalance, and feature extraction	Class imbalance, feature selection

4 Future Challenges

This study covered recent research conducted to enhance existing and new SDP models. Various major issues, such as class imbalance, heterogeneity, redundant and irrelevant features, and data privacy, were addressed in the reviewed studies. Recent advancements have shown promising results in solving these issues. In the future, it would be a task to implement these models in real-world databases. More advanced feature selection techniques should be used for better feature selection. Advanced forms of homomorphic encryption should be used with SDP models, which will challenge researchers to provide better accuracy. In privacy protection, the next step would be to analyze the effect of secure SDP techniques on commercial and real-world data provided by the companies. Some limitations that have been found in the reviewed papers are listed below.

i. Some techniques that were discussed did not improve the accuracy of the models. When LIME was used with JIT models, it reduced the feature set, but the accuracy was not increased [2]. Therefore, appropriate techniques such as novel FILTER discussed in [1] and LTSA-SVM discussed in [6] should be used that improve the performance and accuracy of the existing classifiers greatly.

ii. Security techniques also affect the accuracy of the prediction models. Classifier accuracy and precision also need to be improved while working on security [11, 12]. Some major issues like class imbalance and redundant features need to be solved while providing security to improve the performance of a secure classifier and provide an overall improved SDP model.

iii. The security and enhancement techniques increase the complexity of SDP models. They include complex tasks such as selecting the appropriate base classifier for nested stacking, selecting the optimal feature selection algorithm, and selecting the secure and functional encryption technique. These complexities need to be dealt with efficiently [4, 5]. While preparing a nested or ensemble classifier, simple and resource-efficient base classifiers should be selected so that further complexities can be reduced. State-of-the-art feature selection and security mechanisms should be used to create more advanced novel techniques to get better efficiency.

iv. Some techniques' time complexities, such as LTSA-SVM and Ensemble classifiers, need to be reduced [6].

v. Privacy protection approaches need to be applied to commercial and real-world datasets. The proposed approaches are tested on open and public datasets as of now. Applying privacy protection methods to data that is commercial and produced in the real world will give the researchers a better idea about the effectiveness of these methods. The security level provided by these methods against real-world attacks can only then be tested. They can be truly tested in real-world scenarios only [11, 12].

vi. Advanced homomorphic encryption techniques should be applied and compared with each other so that there can be a reliable encryption technique for SDP [9]. Fully homomorphic encryption techniques are capable of doing complex operations on ciphertext which can provide better security and operability [14]. Therefore, partial homomorphic encryption-based techniques can be replaced by fully homomorphic encryption techniques which are more advanced (Table 2).

Table 2 Merits and demerits of different approaches

Reference	Year	Approach	Advantages	Disadvantages
[2]	2022	Applied LIME interpretability to a JIT-SDP model to perform better feature selection	Cut the redundant and irrelevant features	Accuracy is reduced a bit or remains the same
[1]	2022	A novel FILTER technique was introduced to solve the class imbalance and improve the performance of the SVM SDP model	The imbalance ratio is reduced, which results in increased performance	The filter technique needs to be improved for better results
[3]	2022	A PBSA technique was implemented for better stability in the performance of SDP models through concept drift recovery	PBSA models achieve more stable performance	Sampling can lead to some critical data loss
[5]	2022	For better performance, a nested-stacking classifier with heterogeneous feature selection was made	Nested stacking improves the performance in both within-project and cross-project validations	These nested classifiers increase the complexity of the models
[6]	2019	The LTSA technique was used for feature selection and reduction to improve the performance of the SVM classifier	LTSA-SVM improves the performance of the standard SVM. It is also better than LLE-SVM	This technique has high time complexity
[4]	2018	They have combined random oversampling methods with fuzzy instance recovery to build a novel ensemble classifier	The proposed technique effectively reduces the imbalance ratio	Selecting a base classifier is a long and complex task
[7]	2020	A novel CDAA is proposed to deal with the problem of heterogeneity in SDP	As compared to state-of-the-art techniques, it has increased performance	The techniques are not yet applied to commercial datasets
[9]	2022	A Logistic Regression-based novel HOPE method and a homomorphic encryption scheme are proposed to solve the privacy issues	Attackers cannot retrieve helpful information from the encrypted data	Accuracy is affected in the process of providing security
[10]	2018	A novel method A-DPRF was introduced as differential privacy can be very effective even when the privacy budget is selected carefully	The privacy protection level is increased with the use of A-DPRF	With the increase in the level of privacy protection, the performance of the A-DPRF model will decrease

(continued)

Table 2 (continued)

Reference	Year	Approach	Advantages	Disadvantages
[11]	2021	The FTLKD privacy-focused approach was designed and executed to provide security and better performance	This approach provides better security than non-federated learning methods	Only public datasets were used for testing. Commercial datasets need to be tested
[12]	2022	The FRLGC privacy-focused approach was designed and executed to provide security and better performance	This approach provides better security than federated as well as non-federated learning methods	Only public datasets were used for testing. Commercial datasets need to be tested
[8]	2022	This work proposed a TCLM technique to meet challenges like feature selection, class imbalance, and feature extraction	They are removing heterogeneity, resulting in increased performance	The technique was only applied to public datasets

5 Conclusion

Federated learning is one of the most popular approaches to making the SDP model more secure. An SLR was completed to follow the research studies done recently in the field of SDP to increase efficiency and create more secure models. The most relevant analyses were those published in reputable online libraries such as IEEE, Springer Link, and ScienceDirect. Three well-defined research questions were answered through a systematic study of recent papers covering the different aspects of growth in increasing the efficiency of more than one software defect prediction model. The results mentioned in our findings showed that solving the issue of class imbalance could get better predictive performance from the SDP models. Effective feature selection can remove the problem of heterogeneity in heterogeneous defect prediction models and increase their efficiency. This review can be extended in the future by including other methods to improve class balance, feature selection, and privacy protection.

References

1. Goyal S (2022) Effective software defect prediction using support vector machines (SVMs). Int J Syst Assur Eng Manag 13(April 2022):681–696
2. Zheng W, Shen T, Chen X, Deng P (2022) Interpretability application of the just in time software defect prediction model. J Syst Softw 188(111245):11
3. Cabral GG, Minku LL (2022) Towards reliable online just-in-time software defect prediction. IEEE Trans Softw Eng 1(8):1–1

4. Huda S et al (2018) An ensemble oversampling model for class imbalance problem in software defect prediction. IEEE Access 6(1):24184–24195
5. Chen LQ, Wang C, Song SL (2022) Software defect prediction based on nested-stacking and heterogeneous feature selection. Complex Intell Syst 8(4):3333–3348
6. Wei H, Hu C, Chen S, Xue Y, Zhang Q (2019)Establishing a software defect prediction model via effective dimension reduction. Inf Sci, 399–409
7. Gong L, Jiang S, Jiang L (2020) Conditional domain adversarial adaptation for heterogeneous defect prediction. IEEE Access, 150738–150749
8. Huang Y, Xu X (2022) Two-stage cost-sensitive local models for heterogeneous cross-project defect prediction. In: IEEE 46th annual computers, software, and applications conference, pp 819–828
9. Yu C, Ding Z, Chen X (2022) HOPE: software defect prediction model construction method via homomorphic encryption. IEEE Access, 69405–69417
10. Zhang D, Chen X, Cui Z, Ju X (2018) Software defect prediction model sharing under differential privacy. In: IEEE SmartWorld, ubiquitous intelligence & Computing, advanced & Trusted computing, scalable computing & Communications, cloud & Big data computing, Internet of people and smart city innovation, pp 1547–1554
11. Wang A, Zhang Y, Yan Y (2021) Heterogeneous defect prediction based on federated transfer learning via knowledge distillation. IEEE Access, 29530–29540
12. Wang A, Zhao Y, Li G, Zhang J, Wu H, Iwahori Y (2022) Heterogeneous defect prediction based on federated reinforcement learning via gradient clustering. IEEE Access, 1–1
13. Kamei Y et al (2013) A large-scale empirical study of just-in-time quality assurance. IEEE Trans Softw Eng 39(6):757–773. https://doi.org/10.1109/TSE.2012.70
14. Van Dijk M, Gentry C, Halevi S, Vaikuntanathan V (2010) Fully homomorphic encryption over the integers. In: Gilbert H (ed) Advances in cryptology—EUROCRYPT 2010. EUROCRYPT 2010. Lecture notes in computer science, vol 6110. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13190-5_2
15. Anggriane SM, Nasution SM, Azmi F (2016) Advanced e-voting system using Paillier homomorphic encryption algorithm. Int Conf Inform Comput (ICIC) 2016:338–342. https://doi.org/10.1109/IAC.2016.7905741
16. Wei K et al (2020) Federated learning with differential privacy: algorithms and performance analysis. IEEE Trans Inf Forens Secur 15:3454–3469. https://doi.org/10.1109/TIFS.2020.2988575
17. Jiarpakdee J, Tantithamthavorn C, Treude C (2020) The impact of automated feature selection techniques on the interpretation of defect models. Empir Softw Eng 25:3590–3638. https://doi.org/10.1007/s10664-020-09848-1
18. Wang S, Minku LL, Yao X (2015) Resampling-based ensemble methods for online class imbalance learning. IEEE Trans Knowl Data Eng 27(5):1356–1368. https://doi.org/10.1109/TKDE.2014.2345380
19. Shirabad JS, Menzies TJ (2005) The PROMISE repository of software engineering databases. In: School of information technology and engineering, University of Ottawa, ON, Canada, Tech. Rep. http://promise.site.uottawa.ca/SERepository/. Accessed 12 Apr. 2018

Malicious Sensor Node Extraction from Infested Acoustic Signals in Underwater Wireless Sensor Networks

Prateek and Rajeev Arya

Abstract Existing works in underwater localization depend on acoustic signals for echosounding services. However, echosounders suffer from positioning errors if the underwater nodes report incorrect location due to any reason. One such reason is due to malicious sensor nodes. The present work addresses the issue of malicious sensor nodes in an underwater wireless sensor network (UWSN). The concept of "Infested Acoustic Signal (*IAS*)" is defined to characterize echo chirp signals sent out by the malicious sensor nodes. A localization technique is proposed to extract malicious sensor nodes from the infested acoustic signals with the help of stating a Malicious Sensor Node (*MSN*) Extraction Algorithm. The localization performance of the proposed technique is evaluated with respect to parameters such as Malicious to Healthy sensor node Ratio (*MHR*) and Location Occupation State (L_{OS}). The results are compared to standard techniques. It is found that the proposed method performs superior to the state of the art by at least 24.4% in terms of delta L_{OS}, whereas it is 3% in the vicinity of the lower bound in terms of localization accuracy. The advantage of the proposed research is achieved in terms of the removal of malicious node effects and subsequent lowering of positioning error.

Keywords Malicious sensor node · Acoustic signal · Underwater localization

1 Introduction

ACOUSTIC signals are the sound waves which propagate transversely in the medium as pressure waves. They are faster in fluids than in air. They suffer lower attenuation than Electromagnetic waves in water. Therefore, underwater communication is mostly preferred using acoustic signaling. Wireless Sensor Networks which are deployed underwater utilize acoustic communication for sensing their surroundings. The correct coordinates of each sensor node deployed underwater are essential for

Prateek · R. Arya (✉)
Wireless Sensor Networks Lab, Department of Electronics and Communication Engineering, National Institute of Technology Patna, Ashok Rajpath, Patna, Bihar 800005, India
e-mail: rajeev.arya@nitp.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_14

the proper localization of the underwater wireless sensor networks (UWSNs). The proper functioning of each sensor node ensures accurate positioning estimation of the target. However, if any sensor node records wrong location data due to any reason, the estimation of the target incurs errors, which leads to the failure of localization in the UWSN. The present work focuses on the prevention of localization failures due to one such phenomenon called "malicious sensor nodes". The key idea behind this article is to extract malicious effects from certain specific acoustic signals emanating from the malicious sensor nodes. The key contributions of this paper are as follows:

- The concept of "Infested Acoustic Signal" is presented to indicate the presence of malicious sensor nodes.
- A formalism is presented to extract malicious sensor nodes from Infested Acoustic Signals in UWSNs.
- The proposed formalism is evaluated based on two new parameters, namely "Malicious to Healthy sensor node Ratio (*MHR*)" and "Location Occupation State (L_{OS})".

The rest of the paper is organized as follows: Sect. 1.1 focuses on the literature survey and the identification of research gaps. Section 2 proposes the method of malicious sensor node extraction to be evaluated and discussed with respect to different parameters in Sect. 3. The key findings are summarized in the final section.

1.1 Related Work

A variational message passing-based algorithm has been used to comment upon malicious attacks in Mobile IoT networks [1], but it is yet to be explored in terms of underwater networks [2]. The disturbance in the acoustic signature of sonar in underwater models due to scattering at the water–sediment interface has been attempted in [3]; however, malicious effects by the sensor nodes are yet to be explored. The compromise faced by the UWSN due to the malicious anchor nodes [4] has been solved using a convex programming approach [5]. But the parameters of evaluation are limited to robustness analysis and cooperative behavior. Based on the literature survey, some of the lacunas identified are as follows:

- Lack of sufficient works in the analysis of acoustic signal signature to identify malicious sensor nodes.
- The importance of underwater mathematical modeling to nab the culprit behind malicious sensor nodes in UWSNs.
- The need for diversification of evaluation parameters to comment upon the loss of localization accuracy from multiple perspectives.

The present work aims to address these issues in a systematic manner, as explained in Sect. 2.

2 Methods

The prime objective of this research is to facilitate a UWSN infested with a malicious sensor node by observing and reading the infested acoustic signals during communication for localization. Some of the preliminary assumptions to be taken are as follows:

- The sensor nodes and anchor nodes are distributed randomly throughout the underwater three-dimensional space.
- The UWSN is distributed in a non-sparse environment.
- The energy reserve per sensor node is sufficient that it does not interfere with the acoustic communication.

Infested acoustic signals are considered as one of the most important indications of malicious node presence. We have selected infested acoustic signals for detection since they have a high probability of appearance as compared to other acoustic signals. In this article, we acquire the infested signal by use of a hydrophone in an underwater wireless sensor system. The hydrophones pick up the infested acoustic signal in the form of a modified chirp signal echo. The hydrophones may be placed randomly or at strategic locations and the reference anchor node is placed on the four corners of the water body. The hydrophones work as transducers as they convert the acoustic signals from the malicious sensor node into electric current. These electric signals are then passed on through a suitable detection algorithm which decides the nature of the malicious node. The description of the malicious sensor node is described next.

2.1 Malicious Sensor Nodes

Sensor nodes whose working functions turn defective are termed as malicious sensor nodes. The defect in the function can be due to a multitude of reasons. The consequence of having one or more malicious sensor nodes in the system is that the localization of the target gets impaired considerably.

Definition 1. If H_i represents the *ith* Healthy Sensor Node for all $i \in N$, where N is the total number of healthy sensor nodes in the UWSN, then M is a Malicious Sensor Node of the UWSN iff M results in Infested Acoustic Signals.

One of the identifying features of malicious sensor node in the system is the presence of Infested Acoustic Signals, described in the next section.

2.2 Infested Acoustic Signals

Acoustic signals in a healthy underwater sensor network consist of chirp signals which are merely delayed in time and attenuated in amplitude, due to the acoustic losses of the media. However, the presence of malicious sensor node results in a dramatic change in the acoustic signature of the sensor nodes.

Definition 2. For a malicious sensor node M, an acoustic signal is Infested if the chirp signal echo is severely altered in both shape as well as in timbre. The Infested Acoustic Signal is henceforth abbreviated as IAS.

However, mere detection of IAS in the UWSN is not enough. It is essential that the node ID of the malicious sensor node be identified, followed by a suitable method to get rid of the malicious node or to essentially restore the usual localization of the target. This is explained in the extraction approach, mentioned in the next section.

2.3 Extraction Approach

Different devices are used for sensing hydroacoustic waveforms depending upon the depth, salinity, underwater pressure, the intensity of sound waves, etc. The most common device suitable for this purpose is a hydrophone. Once the hydrophone deployed underwater detects the infested acoustic signal, it is the job of the extraction approach to overcome the deficiency in localization accuracy due to the malicious sensor node and restore target positioning.

2.4 Proposed Malicious Sensor Node (MSN) Extraction Algorithm

The complete framework of the infested acoustic signal-based malicious node extraction is described in the current section.

Malicious Sensor Node Extraction Theorem

1. Malicious Sensor Node M identification is possible if IAS is detected.
2. Localization compensation is possible if M is extracted from the UWSN.

Proof.
The flow is as follows:

- The reference anchor nodes are placed at the eight corners of the controlled underwater environment.

$$L(A_i) = \left\{ \begin{array}{l} (0,0,0),\,(0,y_{\max},z_{\max}),\,(x_{\max},0,z_{\max}),\,(x_{\max},y_{\max},0), \\ (0,0,z_{\max}),\,(x_{\max},0,0),\,(0,y_{\max},0),\,(x_{\max},y_{\max},z_{\max}) \end{array} \right\} \tag{1}$$

- Hydrophones are deployed at strategic locations of the UWSN scenario.

$$L(H_i) = \begin{cases} 1, & \forall P\big(d_{A_iH_i}\big) > 0.5 \\ 0, & otherwise \end{cases} \tag{2}$$

- Hydrophones capture the infested acoustic signals at different instances of time.
- IAS is passed through the extraction algorithm.
- The statistical parameters of the UWSN are calculated before and after the extraction procedure.

3 Evaluation and Discussion

The performance of the extraction algorithm is evaluated here with respect to parameters such as Localization accuracy, localization ratio and malicious sensor node to healthy sensor node ratio.

3.1 Malicious Sensor Node to Healthy Sensor Node Ratio (MHR)

It represents the number of malicious nodes present in the UWSN, indicative of the deviation in localization from the normal behavior. It is abbreviated as *MHR*, given by

$$MHR = \frac{\text{\#Malicious Sensor Node}}{\text{\#Healthy Sensor Node}} \tag{2}$$

3.2 Location Occupation State (L_{OS})

The average two-dimensional separation between the estimated location of the malicious sensor node ($\hat{M}$) and the N number of healthy sensor nodes (H_i) gives us an idea of the estimation accuracy of the proposed method. Ideally, the L_{OS} should be as close to the true location occupation state as possible. L_{OS} is given by the expression

$$L_{OS} = \frac{1}{N}\sum_{i}^{N}\overline{\hat{M}H_i} \tag{3}$$

4 Discussion

Figure 1 shows the performance of the proposed MSN extraction method when the *MHR* value is varied from 0 to 1.00 in increments of 0.25. The result is compared to Cramer Rao Lower Bound (CRLB). In terms of localization error of the target with respect to the *MSN*, the proposed method closely trails behind CRLB, demonstrating near-ideal behavior for lower malicious effects. These errors increase in magnitude as the malicious effect increases when moving along the positive *x*-axis.

Figure 1 demonstrates the comparative analysis of various standard underwater localization techniques such as the Hop-Count-based ToA method [6, 7], 3-Dimensional Underwater Localization [8, 9], Gradient-based Localization [10, 11] and the TDoA method [12, 13]. The proposed MSN extraction Algorithm is observed to exhibit lower delta L_{OS} at UWSN sizes of 10, 20, 30 and 40 nodes. Here, delta

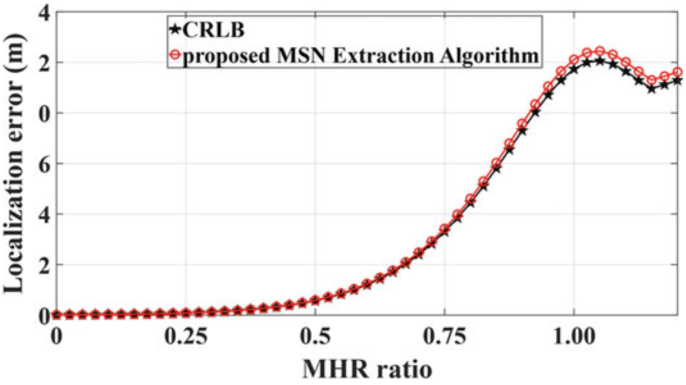


Fig. 1 Localization performance of the proposed method with varying malicious to healthy nodes ratio (MHR)

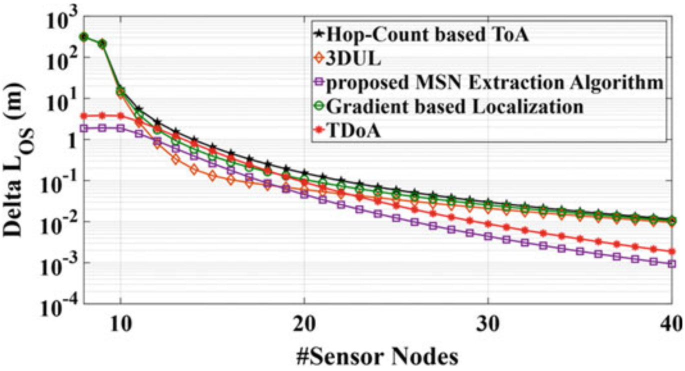


Fig. 2 Localization Occupation State (L_{OS}) performance with varying sizes of the UWSN

L_{OS} refers to the error due to the difference between true and estimated Location Occupation State. In general, the delta reduces with larger UWSNs offering a greater number of H_i than the smaller UWSN sizes.

5 Conclusion

The concept of infested acoustic signals was established as an integral part of the underwater wireless sensor networks. The major finding of the MSN extraction algorithm was a reduction in the delta L_{OS} by around 50% at smaller as well as moderate UWSN sizes, compared to the competing techniques. The proposed work was found to be in the 3% vicinity of CRLB in terms of the localization error. Further tests should involve a detailed mathematical analysis of the extraction of malicious sensor node effect under different environmental scenarios, to ascertain the strengths and weaknesses of this algorithm.

Acknowledgements This work was supported by the Ministry of Electronics and Information Technology, Government of India, File No. 13(29)/2020-CC&BT.

References

1. Li Y, Ma S, Yang G, Wong KK (2021) Secure localization and velocity estimation in mobile IoT networks with malicious attacks. IEEE Internet Things J 8:6878–6892. https://doi.org/10.1109/JIOT.2020.3036849
2. Kumar GH, G.P. R (2022) Node localization algorithm for detecting malicious nodes to prevent connection failures and improve end-to-end delay. Comput Commun 190:37–47. https://doi.org/10.1016/J.COMCOM.2022.04.001
3. Kargl SG, España AL, Williams KL, Kennedy JL, Lopes JL (2015) Scattering from objects at a water-sediment interface: experiment, high-speed and high-fidelity models, and physical insight. IEEE J Ocean Eng 40:632–642. https://doi.org/10.1109/JOE.2014.2356934
4. Mukhopadhyay B, Srirangarajan S, Kar S (2021) RSS-based localization in the presence of malicious nodes in sensor networks. IEEE Trans InstrumMeas 70. https://doi.org/10.1109/TIM.2021.3104385
5. Wang D, Yang L (2012) Cooperative robust localization against malicious anchors based on semi-definite programming. Proc IEEE Mil Commun Conf MILCOM 3–8. https://doi.org/10.1109/MILCOM.2012.6415656
6. Tu Q, Zhao Y, Liu X (2022) Recovery schemes of Hop Count Matrix via topology inference and applications in range-free localization. Expert Syst Appl 200:116906. https://doi.org/10.1016/j.eswa.2022.116906
7. Jo C, Lee C (2016) Multilateration method based on the variance of estimated distance in range-free localisation. Electron Lett 52:1078–1080. https://doi.org/10.1049/el.2016.0226
8. Zhang Y, He P, Pan F (2017) Three-dimension localization of wideband sources using sensor network. Chinese J Electron 26:1302–1307. https://doi.org/10.1049/cje.2017.07.003
9. Arafat MY, Moh S (2021) Bio-inspired approaches for energy-efficient localization and clustering in UAV networks for monitoring wildfires in remote areas. IEEE Access 9:18649–18669. https://doi.org/10.1109/ACCESS.2021.3053605

10. Feng S, Yang L, Qiu G, Luo J, Li R, Mao J (2019) An inductive debris sensor based on a high-gradient magnetic field. IEEE Sens J 19:2879–2886. https://doi.org/10.1109/JSEN.2018.2890687
11. Heydari A, Aghabozorgi MR (2020) Sensor placement for RSSD-based localization: optimal angular placement and sequential sensor placement. Phys Commun 42:101134. https://doi.org/10.1016/j.phycom.2020.101134
12. Salari S, Chan F, Chan YT, Read W (2018) TDOA estimation with compressive sensing measurements and Hadamard matrix. IEEE Trans Aerosp Electron Syst 54:3137–31342. https://doi.org/10.1109/TAES.2018.2826230
13. Ma F, Yang L, Zhang M, Guo FC (2019) TDOA source positioning in the presence of outliers. IET Signal Process 13:679–688. https://doi.org/10.1049/iet-spr.2019.0020

Stock Market Prediction and Analysis Using Different Machine Learning Algorithms

Naman Rohilla, Yogesh Goyal, Ajay Khasiya, Bhavesh N. Gohil, and A. K. Shukla

Abstract With around 21 major stock exchange groups worldwide, the stock market is one of the significant choices for investors to invest funds. Prediction of the stock price with high precision is challenging due to the high volume of investors and market volatility. The volatility of the market is due to non-linear time series data. To handle such data, various algorithms are available. Many works have been done to predict a stock price; however, the prediction with high accuracy is still a challenge to achieve. This paper deals with the prediction of the daily high price of a stock. The experiment is done using AI (Artificial Intelligence) and ML (Machine Learning) algorithms (LSTM, XGBoost and Regression specifically). The model is trained using previously available stock data, and the acquired knowledge (trained model) is used to predict the stock price precisely. The accuracy achieved in the proposed algorithm is around 97%. Different types of datasets are utilised to achieve favourable outcomes. It can be observed that building a proper model can help an investor to create a better stock portfolio, as better prediction improves the net profit for the investor.

Keywords Stock market prediction · Machine learning · Time series data · Algorithms · Volatility of the market

1 Introduction

In stock market, the purchase and selling of stocks, shares, currencies and other derivatives through tangible forums supported by traders (a flexible and interconnected system) take place. Investors can purchase and sell shares in public corporations (public corporations are the companies whose stocks are available on a stock exchange platform).

N. Rohilla · Y. Goyal (✉) · A. Khasiya · B. N. Gohil · A. K. Shukla
Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat, India
e-mail: yogeshgoyal1031@gmail.com

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_15

Due to market volatility, accurately foreseeing the stock price is a typical task [14]. The market is volatile because of two reasons:

1. Known variables (open and close prices, high and low prices, volume, etc.);
2. Unknown variables or sentiments (company reputation, inside information, election results, natural disaster, rumours in the market, country budget, etc.).

In stock market prediction, as the data is time series data, the strategy implied is also based on the same. This paper demonstrates the performance of multi-time guessing algorithms. For predicting high stock prices, the project uses Long Short-term Memory (LSTM), XGBoost and Regression.

Time series forecasting modelling is essential in predicting the data like the stock market. Time series analysis is a statistical subset commonly used in econometrics and operations research. Stock prices, for example, high and low prices, are highly volatile and several variables influence their worth.

Predictions are made solely based on known variables. Data from a public corporation is used as the training dataset for model training. However, risk strategies and safety measures have been implemented.

Individual observations are required to evaluate the prediction during real-life implications. Many factors are integrated and examined, such as the trading strategy to be used, sophisticated mathematical functions to show the condition of a particular stock, machine learning algorithms to anticipate future stock values and unique stock-related concerns. Work done here is to anticipate future stock value using machine learning algorithms.

Further, in Sect. 2, related works have been discussed to understand the past work on the topic. Section 3 discusses the mathematical model of the proposed algorithms and gives a comparative analysis of methodologies based on precision. Section 4 discuss some challenges, open issues and promising future directives for development based on this analysis.

2 Related Work

Different approaches had been proposed earlier to predict stock prices; a few are discussed here, along with their drawbacks.

Authors [7] proposed a model based on the SVM algorithm for the prediction of the stock price of IBM Inc. SVM requires a considerable dataset value for the training of the model, and overfitting is also not an issue. The model worked well for the selected corporation, but if considered a new corporation with a relatively low dataset, the prediction of such stock price would not be that accurate using the SVM model.

Authors [12] proposed a work with the LSTM algorithm to predict stock prices. In the model, they predicted the stock's closing price and achieved around 95% accuracy. The datasets used were Google, Nifty50, TCS, Infosys and Reliance. The performance of this model is high due to consistent change in the dataset, if a dataset provided to the model has rapid change, the model will fail, This paper discusses how the LSTM model performed with the dataset of Tesla, and the model achieved an accuracy of around 30% due to extreme changes in the dataset.

Authors [14] proposed a model based on CNN. The dataset used were Infosys, TCS and CIPLA. The accuracy achieved in the model is excellent. With every type of dataset, the model will perform outstanding until the prediction is based on sentiment analysis. CNN algorithm has a fixed input and output ratio, which doesn't allow prediction based on sentiments. This factor of CNN limits the future goal in stock price prediction.

Authors [2] compared models based on Decision Boosted Tree, SVM and LR. They concluded that Decision Boosted Tree performs better than SVM and LR while predicting a curve motion up/down. The accuracy achieved is good, but in the practical world, prediction of the motion of the curve is not sufficient to invest with low risk. Along with the motion of the curve, one needs to predict the value of the stock, which may lead to a change in conclusion.

Authors [8] proposed a framework based on ANN and RF and concluded that ANN performs better. They predicted the next day's closing price using the framework. The dataset used were Nike, Goldman Sachs, Johnson and Johnson, Pfizer and JP Morgan Chase and Co. The analyses done are comparative rather than result-oriented. The accuracy achieved is not sufficient for actual investment.

All of this related work has different drawbacks, such as low accuracy for new companies, only predicting motion curves, etc. Here, a modal approach is proposed to overcome these drawbacks.

3 Different ML Models and Experimental Results

The algorithms used to predict high stock prices are Long Short-Term Memory (LSTM), XGBoost and Regression.

Figure 1 shows the structure of the program to its controllable levels. For this demonstration exercise, high prices of ONGC, Tata Steel and Tesla stocks (dataset considered is of past seven years (2015–2022)) have been predicted. All the analysis is done in Jupyter Notebook, an open-source web application that allows one to create and share documents that contain live code, equations, visualisations and narrative text.

NumPy and Pandas are used for data cleaning, which provides high-performance, easy-to-use data structures and data analytic tools for manipulating numeric and time series data.

For data pre-processing and modelling, the data was analysed with Sklearn, which includes tools for machine learning and statistical modelling, including classification, regression, clustering and dimensionality reduction. For visualisation, matplotlib is used to create static, animated and interactive visualisations.

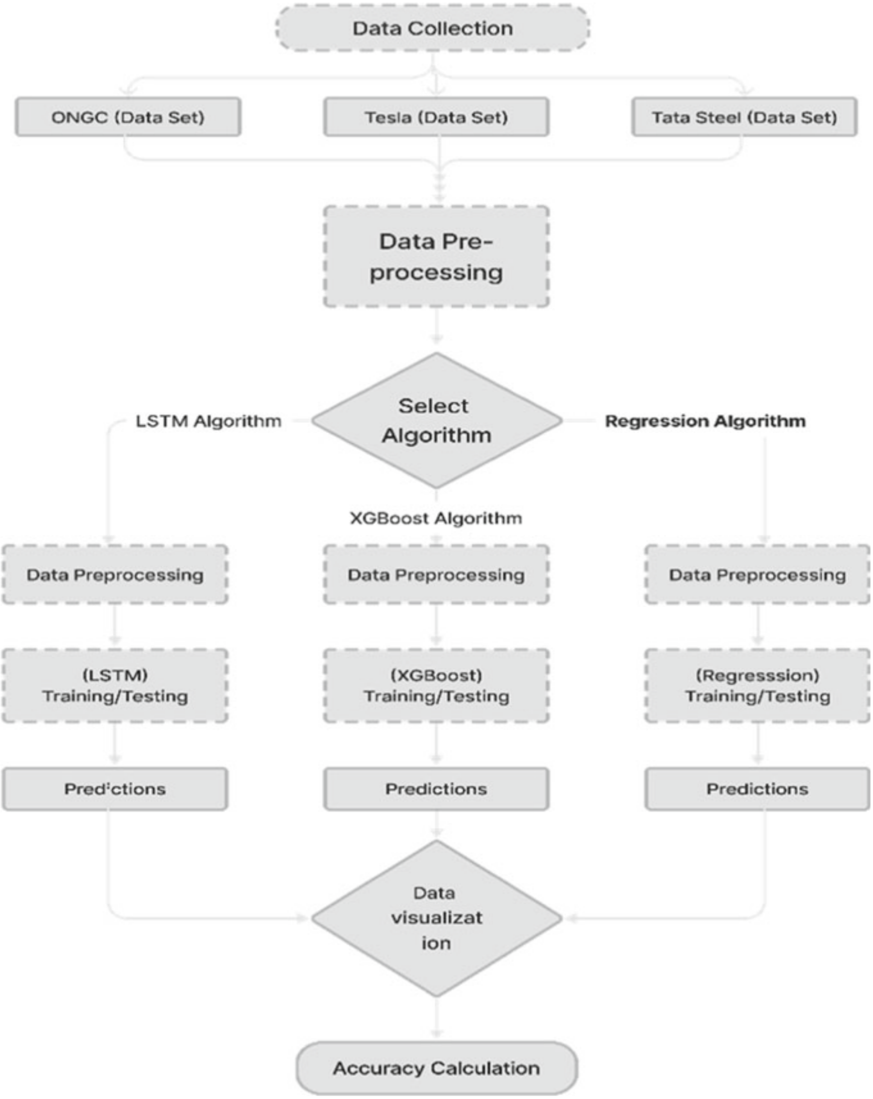


Fig. 1 Experimental structure of the model

(i) LSTM: Long Short-Term Memory[5, 13] network is a particular type of RNN that can learn long-term dependencies. They are currently widely utilised and function wonderfully in a variety of environments. It was specially designed to solve the issue of long-term dependencies. It is created so that it remembers data for a long time and doesn't have to work hard to access that data. The initial stage starts with the decision made by 'forget the gate layer'. It decides which information cells can ignore (Fig. 2).

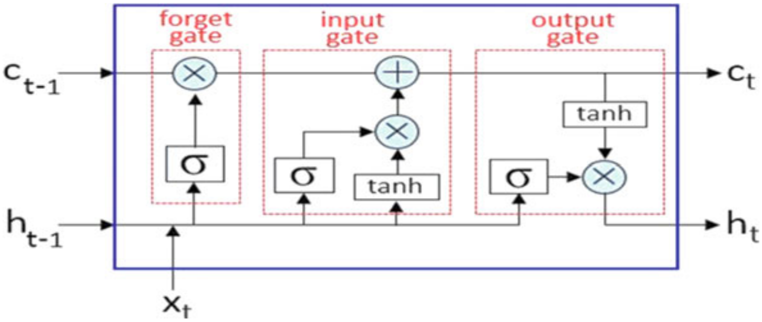


Fig. 2 Structure of LSTM

This is a sigmoid layer. h_{t-1} and x_t are two dependent variables which return values from 0 to 1. This value is stated to C_{t-1} cell. '1' means the information will be kept, while '0' means to reject the information (FIg. 3).

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$

$$c_t = c_{t-1} \otimes f_t \tag{2}$$

It further moves to the 'input gate layer' [5]. It decides what other information needs to be kept. Again, it's a sigmoid layer which is updated by the *tan(h)* layer, it creates a C_{t-1} vector for new cell value (Fig. 4).

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{3}$$

$$a_t = \tanh(W_i x_t + U_a h_{t-1} + b_a) \tag{4}$$

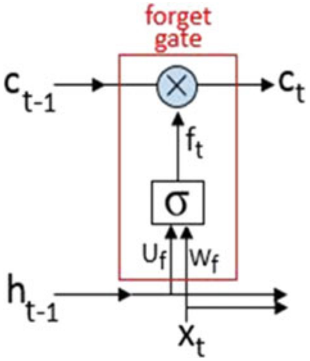


Fig. 3 Inside the forget gate

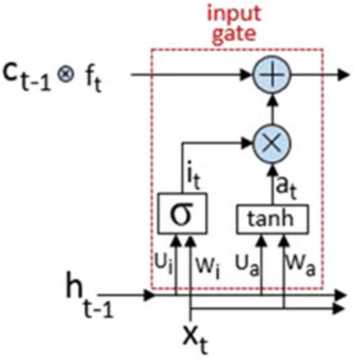


Fig. 4 Inside the input gate

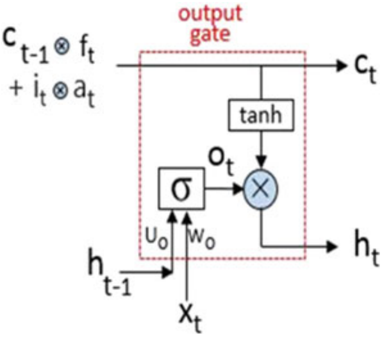
$$c_t = (c_{i-1} \otimes f_t + i_t \otimes a_t) \tag{5}$$

Finally, both the above states are joined together. The transition is made from the previous cell state C_{t-1} to the new cell state C_t. As it's the preceding phase, All the remaining data needs to be collected. Let's now move on to the items that are decided to forget earlier. A value i_t C_t is appended to each state value. This states by what value each state value has been updated. The output which needs to get filtered depends on the cell condition. A sigmoid layer is used to decide which cell state should be output. Further, it is multiplied by the sigmoid gate output and is processed through *tan(h)* function. It is done to get values ranging from -1 to 1. The value generated is the required result (Fig. 5).

$$O_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6}$$

$$h_t = \tan h(c_t x_t \otimes O_t) \tag{7}$$

Fig. 5 Inside the output gate



(ii) XGBoost: Extreme Gradient Boosting [10] is one of the methods for ensemble learning. Relying on a few methods of machine learning models is not always been enough. It is a technique for methodically integrating the predictive abilities of several learners. Because of this, a single model is created that includes the outcomes of several models (Fig. 6).

(iii) Regression: Regression [9] is a machine learning algorithm which comes under supervised learning. It is entirely based on independent variables and dependent variables. It is mainly based on predicting and creating the relationship between variables (Fig. 7).

The regression model used (y) as a dependent variable and ($\hat{x}$) as an independent vector. Thus, the linear relationship between $\hat{x}$ and y is identified as a result of this regression approach. The above figure $\hat{x}$ indicates the person's experience gain, and

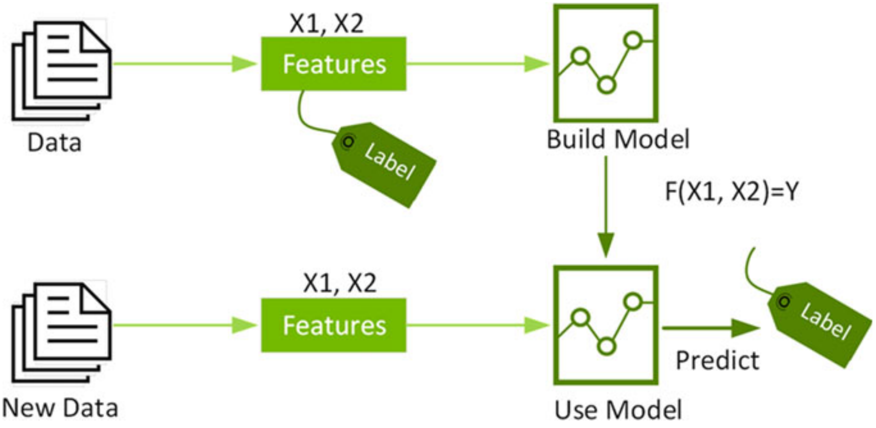


Fig. 6 XGBoost model

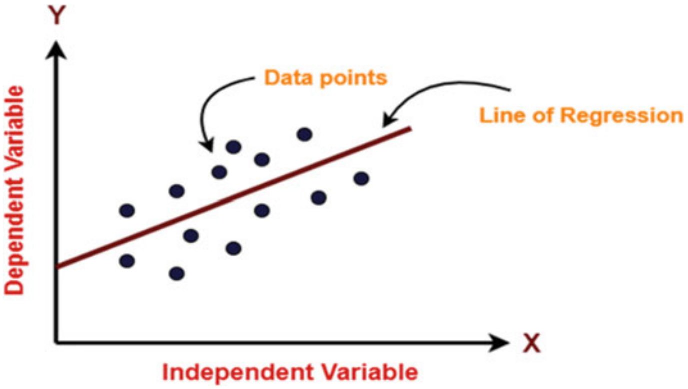


Fig. 7 Relationship between Dependent and Independent variables

y indicates that person's salary. The line which fits the best for the model is called the regression line.

Linear regression hypothesis function*:*

$$y = \theta_1 x_1 + \theta_2 x_2 \ldots\ldots + \theta_n x_n + c \tag{8}$$

To train, there are some essential variables to define [15]

- Parameter
- Univariate

x: independent vector or input data.

y: the labels of the data (supervised learning) or dependent variable.

Now, it will find the best line for the model to predict the value of y and $\widehat{x}$ after training the model. The fit line is determined by the coefficient of $\widehat{x}$.

c: intervene of the data.

θ_i: coefficient of x_i.

After getting the best θ_i value, we will get the best-fit line. So, when we use the model to predict the value of y for the input value of x_i, it will indicate the value of y. The model tries to forecast the value of y using the best-fit regression line such that the error difference between anticipated and projected values are as small as possible. As a result, updating the θ_i values is crucial to discover the optimal value that minimises the discrepancy between the predicted and actual values of y.

$$minimize \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2 \tag{9}$$

$$J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2 \tag{10}$$

J: Cost function, also known as Root Mean Squared Error (RMSE) between the predicted y value (prediction) and real y value (actual (y)). Gradient Descent: Gradient Descent is used by the model to update θ_i values to minimise the Cost function (RMSE value) and get the best-fit line for the model. Starting with random θ_i integers, the aim is to iteratively update the values until the cost is as low as feasible.

3.1 Dataset

Dataset of Tata Steel, ONGC and Tesla are considered in the project. The data compromise of closing, opening, high, low, volume and percentage of change of the stock price on a particular day of the last 7 years (2015–2022). Stock prices of

ONGC and Tata Steel are in INR, while that of Tesla is in USD. Different types of datasets are considered in order to understand the circumstances of algorithms.

3.2 Experimental Results

The company's historical stock price has been examined to determine the pattern for predicting future stock prices. Stock price prediction is based on the companies' historical data without sentimental value. The accuracy achieved is shown in below Table 1 (Figs. 8 and 9).

LSTM performed well for the dataset with fewer variations (ONGC) but, for the dataset with moderate variation (Tata Steel), it performed well with one specific epoch while failed at every other epoch value, this result probably occurred due to the memory-based methodology of LSTM (Fig. 10).

XGBoost failed to predict all the cases. The algorithm cannot extrapolate target values beyond the limits of the training dataset when making the prediction, and the input space of any given problem is limited (Figs. 11 and 12).

Regression performed the best in the considered case, as the algorithm creates the optimal value of slopes in a given linear expression, considering high price as the dependent variable and other values as an independent variable, linear expressions are generated to obtain optimal values of sloped to predict high price accurately.

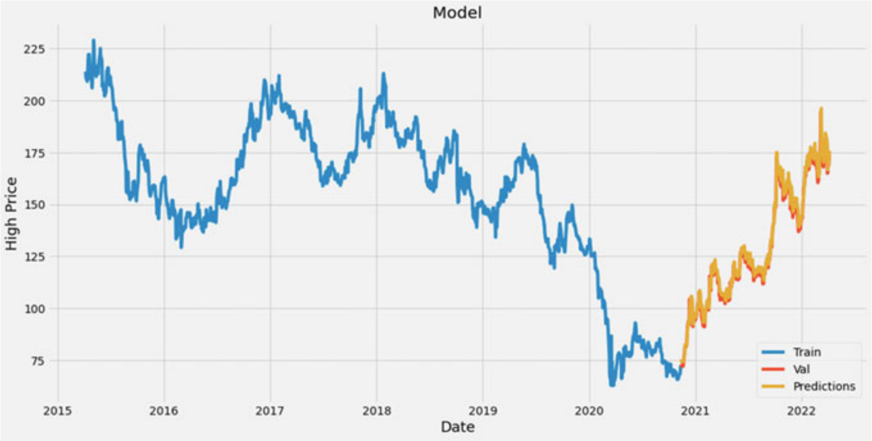


Fig. 8 Line graph: Prediction graph of high price. Algorithm: LSTM. Dataset: ONGC

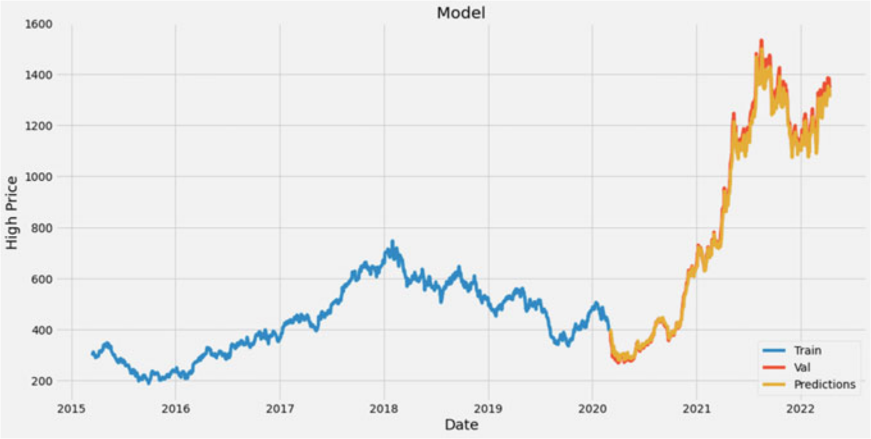


Fig. 9 Line graph: Prediction graph of high price. Algorithm: LSTM. Dataset: Tata Steel

Table 1 Algorithms accuracy table

Algorithms accuracy on multiple stocks (in %)			
Algorithms	ONGC stock	Tata Steel stock	Tesla
LSTM	97.55%	89.19%	<30%
XGBoost	Failed	Failed	Failed
Regression	99.95%	96.53%	97.41%

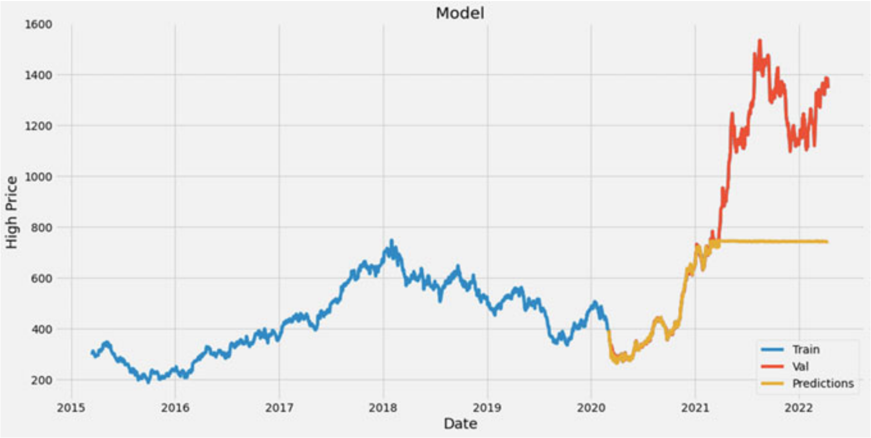


Fig. 10 Line graph: Prediction graph of high price. Algorithm: XGBoost. Dataset: Tata Steel

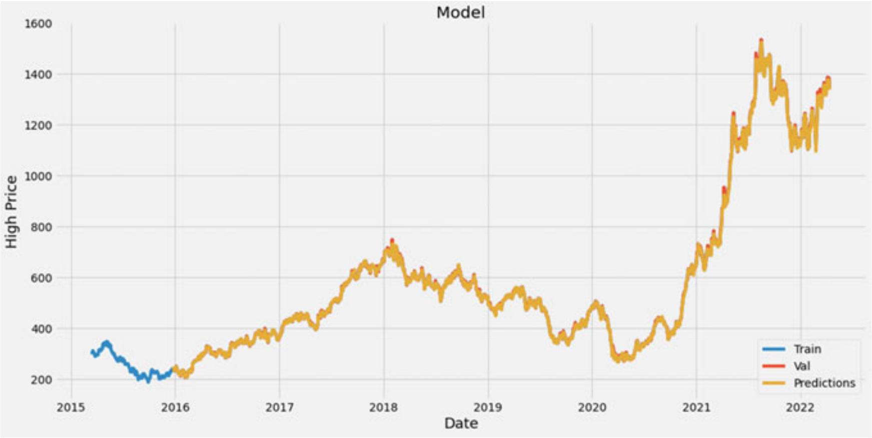


Fig. 11 Line graph: Prediction graph of high price. Algorithm: Regression. Dataset: Tata Steel

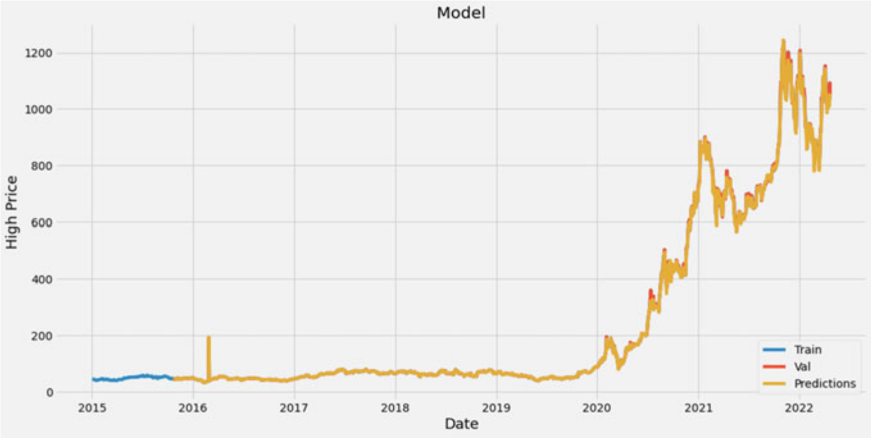


Fig. 12 Line graph: Prediction graph of high price. Algorithm: Regression. Dataset: Tesla

4 Conclusions and Future Work

The datasets from Tata Steel, ONGC and Tesla were examined to determine the pattern for predicting the future high price of the stick. The models used are LR, LSTM and XGBoost. The dataset collected is readily available on investing.com [16]. Seven years of data from these companies have been pre-processed. The training–testing ratio of the model is 80:20 and the accuracy achieved was very different for each type of dataset and model. The best accuracy achieved is in the LR model.

The proposed model differs significantly from previous works. Different algorithms were compared with different types of datasets to understand which algorithm performed better under which circumstances. LR is the best algorithm among the models that were performed with each type of dataset, but there are still some stones left to be unturned.

The aim is to create a model which, along with the available dataset, considers sentiments and human reactions for predicting stock price. Also, integrating this model with one's Demat account for automatic trading to achieve the best profit will be the goal to achieve.

References

1. Das AK, Chandramouli S, Dutt S (2018) Machine Learning. Pearson Education India
2. Nayak A, Manohara Pai MM, Pai RM (2016) Prediction models for the indian stock market. Proc Comput Sci 89:441–449, ISSN 1877-0509
3. Ong CS, Deisenroth MP, Faisal AA (2020) Mathematics for machine learning. Cambridge University Press, illustrated edition
4. Wang C-F (2019) The vanishing gradient problem. https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484. Medium American Publisher
5. Blog CC (2015) Understanding LSTM Networks. https://colah.github.io/posts/2015–08-Understanding-LSTMs/. San Francisco, CA
6. Brownlee J (2021) Ensemble learning algorithms with Python, volume 450 pages. Machine Learning Mastery Vermont, Victoria
7. Vanukuru K (2018) Stock market prediction using machine learning. Research paper published by Sreenidhi Institute of Science Technology India
8. Vijh M, Chandola D, Tikkiwal VA, Kumar A (2020) Stock closing price prediction using machine learning techniques. Proc Comput Sci 167:599–606. ISSN 1877-0509
9. Gupta M (2017) ML, Linear Regression. https://www.geeksforgeeks.org/ml-linear-regression/ . Blog published by Geeks-For-Geeks, India
10. Sundaram RB (2018) An end-to-end guide to understand the math behind XGBoost. https://www.analyticsvidhya.com/blog/2018/09/ an-end-to-end-guide-to-understand-the-math-behind-xgboost/. Blog Published by Analytics Vidhya, India
11. Mitchell R (2017) Gradient Boosting, Decision Trees and XGBoost with CUDA. https://developer.nvidia.com/blog/gradient-boosting-decision-trees-xgboost-cuda/. Published by NVIDIA Developer, Santa Clara, California
12. Rahul S, Sandeep ON, Dinesh S, Maruthi A, Raju SR (2021) Stock price prediction. Report published by Anil Neerukonda Institute of technology and sciences, India
13. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Computation, MIT Press Direct, U.S, 9(8):1735–1780
14. Selvin S, Ravi V, Gopalakrishnan EA, Menon V, Soman Kp (2017) Stock price prediction using LSTM, RNN and CNN-sliding window model. International Conference on Advances in Computing, Communications and Informatics, MIT Manipal India, Pages 1643–1647, 09
15. Gawali S. Linear Regression in Machine Learning. https://www.analyticsvidhya.com/ blog/2021/06/linear-regression-in-machine-learning/. Blog Published by Analytics Vidhya, India, 2021
16. **Dataset**: https://in.investing.com/equities/

A Novel Fake Job Posting Detection: An Empirical Study and Performance Evaluation Using ML and Ensemble Techniques

Cheekati Srikanth, M. Rashmi, S. Ramu, and Ram Mohana Reddy Guddeti

Abstract Recently, everything can be accomplished online, including education, shopping, banking, etc. This technological advancement makes it easy for fraudsters to scam people online and acquire easy money. Numerous cyber crimes worldwide exist, including identity theft and fake job postings. Nowadays, many companies post job openings online, making recruitment simple. Consequently, fraudsters also post job openings online to obtain money and personal information from job seekers. In the proposed work, we aimed to decrease the frequency of such scams by using ensemble techniques such as AdaBoost, Gradient Boost, Stacking classifier, XgBoost, Bagging, and Random Forest to identify fake job postings from genuine ones. This paper proposes various featurization techniques such as Response coding with Laplace smoothing, Average Word2vec, and term frequency-inverse document frequency weighted Word2vec. We compared the performance of ensemble techniques with machine learning (ML) algorithms on publicly available EMSCAD dataset using accuracy and F1-score. Bagging classifier outperformed all the models with an accuracy of 98.85% and an F1-score of 0.88 on imbalanced dataset. On balanced dataset, XgBoost achieved 97.89% accuracy and 0.98 F1-score. From the experimental results, it is observed that a combination of ensemble and featurization techniques using Laplace smoothed Response coding and BoW stood superior to most of the state-of-the-art works on fake job posting detection.

Keywords Ensemble techniques · Fake job posting · Featurization · Identity theft · Machine learning

C. Srikanth (✉) · M. Rashmi · S. Ramu · R. M. R. Guddeti
Department of Information Technology, National Institute of Technology Karnataka, Surathkal, Mangalore 575025, India
e-mail: cheekati.srikanth298@gmail.com

M. Rashmi
e-mail: rashmi.187it003@nitk.edu.in

S. Ramu
e-mail: ramu.197it006@nitk.edu.in

R. M. R. Guddeti
e-mail: profgrmreddy@nitk.edu.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_16

1 Introduction

In the past few years, job hunting has been arduous work, nearly requiring a full-time commitment. Going from one job posting to the next and completing countless forms do require concentration and patience. Due to technological advancement, our working methods are also evolving [1]. Many businesses today prefer to post job openings online [2]. It is very simple for recruiters and candidates too [3]. Numerous online recruitment portals such as Monster, LinkedIn, Shine, and Naukri [2], aim to simplify job seekers' most challenging tasks and assist them in obtaining employment with their dream company. Recruiters can easily find the most qualified candidates, among other advantages [3].

Due to the increasing unemployment, many people are desperate to find jobs and earn livelihoods. It is likely that unaware youth, out of their desperation, might come into contact with several websites claiming to offer employment. Anyone using online job search websites should be careful to avoid potential scams. The number of fraudulent websites asking job seekers for money or bank account details is increasing rapidly. This online recruitment process has also gathered much attention from scammers [4]. These scammers have leveraged new ways to reach people, such as fake job postings.

A fraudulent job advertisement (job ad) targets the employment industry [4]. Occasionally, scammers also post bogus job listings under the name of a reputable organization, tarnishing their credibility. Scammers post these jobs to obtain bank account information, national insurance numbers, and other personal information. As the fake advertisement can be part of a more comprehensive operation targeting such as identity fraud, people may be asked to send money directly as a fee [5] to secure an interview or as part of the onboarding process.

When searching for a job online, it can be difficult to distinguish between legitimate and fraudulent job postings. Scammers frequently post fake advertising with attractive offers that grab people's attention. These advertisements are smartly designed and posted in the exact locations where legitimate employers post job openings [1]. This makes it more difficult to detect fake jobs, and our growing online presence also makes us more vulnerable.

Our objective is to present a method that will provide reliable results when determining the authenticity of job postings. Using machine learning, we can detect such fake job postings by giving various attributes of a job ad as input to the ML model, and it predicts if the job is fake or not. Hence, people won't fall into those fraudulent job traps. As a result, the financial and privacy issues brought on by these bogus job ads will be resolved.

This paper proposes a novel method for identifying fake job postings and distinguishing them from genuine ones using ensemble techniques with featurization strategies: Response Coding with Laplace Smoothing, Average Word2vec, and TFIDF-weighted Word2vec. The primary contributions of our work are as follows:

- Featurization of categorical attributes using Response Coding with Laplace Smoothing and text attributes using BoW, TFIDF, Avg-W2V, and TFIDF-W2V.

– Balanced the EMSCAD dataset with ADASYN and built the models on both balanced and imbalanced datasets to detect fake job postings.
– Evaluated the performance of ML and ensemble models using both the proposed featurization strategies and the current techniques.

The paper is structured as follows: Sect. 2 discusses the literature survey. Section 3 describes the proposed methodology. Section 4 focuses on the results and observations of this work. Section 5 concludes and outlines the path of future work.

2 Literature Survey

Existing works on online recruitment fraud are discussed in this section. Diverse studies have revealed that review classification, false information detection, fake news detection, and spam classification have garnered considerable interest in the fraud detection domain. Numerous works utilizing NLP, ML, and Deep Learning (DL) have been proposed.

Vidros et al. [4] mentioned that online job portals have been widely used as platforms for fraud, and fake job postings damage an organization's reputation and cause applicants' privacy to be compromised, but it is difficult to identify such jobs. Authors used "Employment Scam Aegean Dataset" (EMSCAD) [6] which contains real-life job ads. The "Bag of Words" (BoW) approach is used for featurization. Random Forest (RF) has better performance with 90% of accuracy.

Ranparia et al. [7] proposed a DL model to predict fake job predictions. This study revealed that fraudulent job postings contain fewer words than real jobs. Authors used Sequential Network for modeling, and Glove [8] for word embedding, and this model performed well with an accuracy of 97%. The authors web scrapped the data from LinkedIn to test the model's performance.

Keerthana et al. [9] presented a Machine Learning (ML) model. The authors used TFIDF and BoW feature engineering techniques to improve the model's performance. They experimented with ML models such as MLP, RF, K-Nearest Neighbours (KNN), SVM, and LoR algorithms are used. The MLP outperformed all the models with an accuracy of 71%.

Tabassum et al. [1] curated 4000 job ads. They developed ML model using algorithms such as Decision Tree (DT), RF, Voting Classifier (VC), and Gradient Boosting (GB) with TFIDF featurization method. Lal et al. [10] used LoR, RF, and VC for fake job detection. They extracted 21 binary features from the EMSACD dataset and classified them into three categories: Linguistic, Contextual, and Metadata. This study considered real jobs as a positive class.

Mehaboob and Malik [11] presented a framework for fraud job detection and used the EMSCAD dataset. Due to class disparity, a balanced dataset of 940 job postings was designed. In this, 470 of 940 job postings are genuine, while 470 are fraudulent. They investigated the individual and combined effects of organization characteristics, job description, and compensation structure on fraud. Nasser et al. [3]

built an Artificial Neural Network (ANN)-based model. Additionally, these authors undersampled the data, resulting in information loss.

Habiba et al. [5] compared various ML algorithms such as KNN, RF, DT, SVM, Naive Bayes (NB), and MLP for fake job post prediction. However, they considered only categorical and binary attributes, such as Telecommuting, Employment_type, Required_experience, etc., on imbalanced dataset.

Nindyati et al. [2] investigated the detection of employment scams. The authors created the IESD dataset for fake job posting detection. The authors proposed context-based behavioral features, and their study revealed that behavioral features could enhance the model's performance.

Amaar et al. [12] proposed a method using BoW and TFIDF as feature extraction methods. The authors considered 10 independent attributes for experimentation. MLP outperformed all models with an accuracy of 97% and F1-score of 0.85 on imbalanced dataset. Naudé et al. [13] designed an ML system to distinguish fake jobs into three categories: multi-level marketing, identity theft, and corporate identity theft. Rule set features and four attributes from EMSCAD: Description, Benefits, Requirements, and Company_profile are taken into account. To mitigate class imbalance, undersampling was done and their research showed that a combination of rule set features with part-of-speech tags and BoW on text data gives better performance.

Literature survey revealed that most of the works in fake job posting detection, considered EMSACD dataset and only a few attributes from the dataset have been taken into account. In the proposed work, we examined all the features to gain insightful knowledge on fraudulent job ads. In the featurization step, all text attributes are either disregarded or supplemented with categorical attributes. When text features are combined with categorical, if any category has more than two words, it is misunderstood. Hence, we have taken the necessary steps to featurize different types of attributes by proposing various methods. All the studies considered TFIDF and BoW techniques for word embedding, which do not capture semantic meaning. Therefore, to address this issue, using a pretrained Word2vec model, we proposed two featurization techniques, such as Avg-W2V and TFIDF-W2V. EMSACD dataset is severely imbalanced and causes overfitting on the majority class data. To balance the dataset, some authors did undersampling, which leads to information loss. Hence, we used ADASYN [17] method to balance the data. Most of the studies used only classical ML techniques. The identification of fraudulent job postings must be accurate. Otherwise, job seekers can lose a great opportunity or be ripped off. Therefore, instead of relying on a single model prediction, we combined various base models by utilizing ensemble strategy to build a cutting-edge model with high accuracy and F1-score.

3 Proposed Methodology

The methodology we employed to detect fake job postings is described in this section. The first and foremost step is data collection, and we apply various preprocessing steps to the dataset. Featurization is performed on the preprocessed dataset. The data

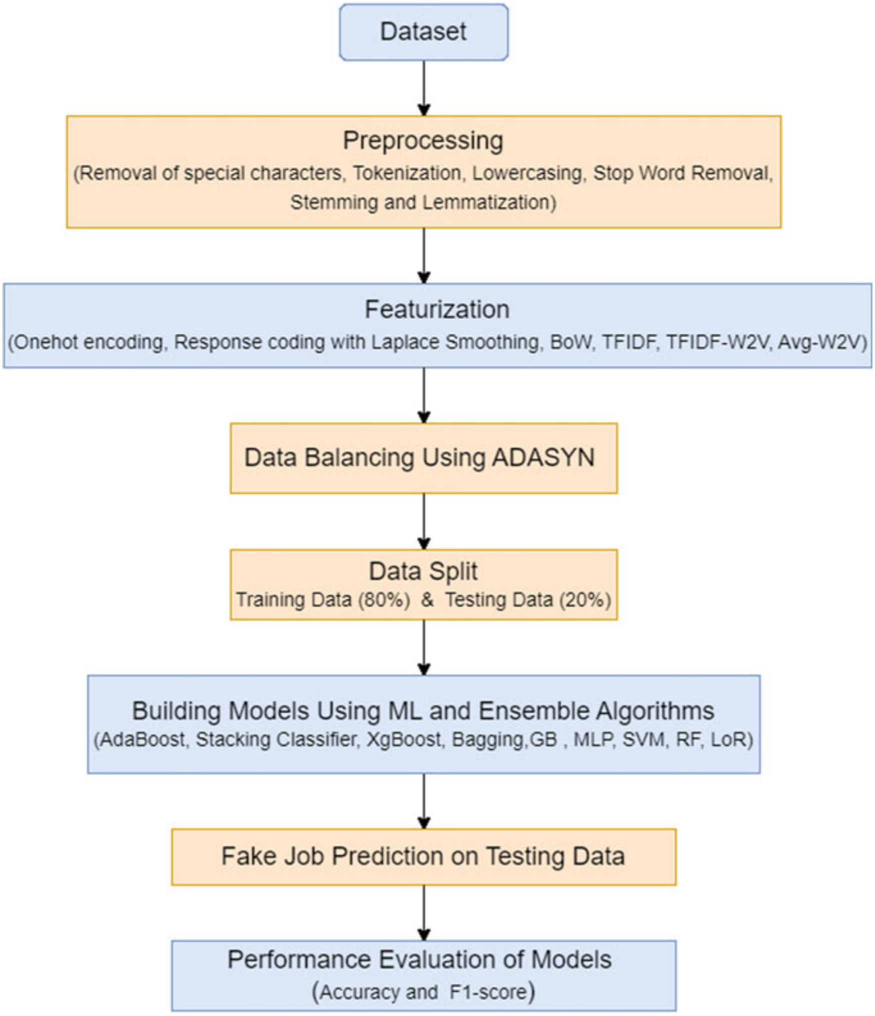

Dataset
Preprocessing
(Removal of special characters, Tokenization, Lowercasing, Stop Word Removal, Stemming and Lemmatization)
Featurization
(Onehot encoding, Response coding with Laplace Smoothing, BoW, TFIDF, TFIDF-W2V, Avg-W2V)
Data Balancing Using ADASYN
Data Split
Training Data (80%) & Testing Data (20%)
Building Models Using ML and Ensemble Algorithms
(AdaBoost, Stacking Classifier, XgBoost, Bagging,GB , MLP, SVM, RF, LoR)
Fake Job Prediction on Testing Data
Performance Evaluation of Models
(Accuracy and F1-score)

Fig. 1 Architecture of the proposed methodology

is balanced using the ADASYN method to reduce the impact of data imbalance problems. Next, the dataset is divided into training data and testing data. The classifiers are built using various ML and ensemble techniques on the training data, and finally, we evaluate the models on the testing data using accuracy and F1-score. Figure 1 illustrates the architecture of the proposed methodology.

3.1 Data Collection

A benchmark dataset EMSCAD is used in the proposed work for fake job news detection. The EMSCAD is publicly available on Kaggle [14], which is provided by the University of the Aegean Laboratory of Information and Communication Systems Security. The dataset consists of 17 job posting attributes and 17,880 Real-life Job advertisements, of which 17,014 are Real, and 866 are Fake job advertisements. In this work, we considered 15 of the 16 independent features, including Title, Location, Department, Company_profile, Required_experience, Required_education, Industry, Description, Requirements, Benefits, Telecommuting, Has_company_logo, Has_questions, Employment_type, and Function. "Fraudulent" is our target variable.

3.2 Preprocessing

In the proposed work, text data is used to detect fake job postings. Generally, various types of noise can be found in job postings extracted from various sources. Noise can be special characters, digits or HTTP links. It has no meaningful information that the classifier can use, so we need to remove them. To do this, we used regular expressions. All the patterns that do not belong to the regular expression "[a–z A–Z]" and HTTP links were eliminated from the job descriptions. Now the data contains only letters. Some of the job postings have missing values; keeping them will be preferable to deleting them due to the high likelihood that they are fraudulent. Preprocessing consists of the steps such as tokenization, lower casing, removal of stopwords, stemming, and lemmatisation.

3.2.1 Tokenization

In this step, all the text data is converted into tokens. Tokens can be sentences, words, subwords, or alphabets. This work transformed job postings into words using the Natural Language Toolkit (NLTK) tokenizer.

3.2.2 Lower Casing

One of the most straightforward steps of preprocessing is lower casing. A good technique to prevent various spellings of the same term from being submitted to NLP algorithms due to mixed cases is to lowercase all text. There are various circumstances in which certain words should not be written in lowercase. Choosing lowercase for all words depends on the use case and domain.

3.2.3 Stop Word Removal

Stop words are terms that are frequently employed in a certain language. Few stop words in the English language are “by”, “a”, “my”, and “on”. Actually, stop words do not give important information, so they are typically filtered out. It is crucial to ensure that all the non-informative words are included in the stop word list. In the proposed work, we considered stop words defined in the NLTK library. However, we can always create a customized stop word list based on the use case.

3.2.4 Stemming

In this step, the text is stripped off any affixes. Inflected words, such as various verb forms, are reduced to their corresponding base form. In most cases, a word derived from stemming is not a meaningful word. In the proposed work, Porter Stemmer is used from the NLTK library.

3.2.5 Lemmatisation

This approach removes any affixes from words, but the returned stem or root is always a dictionary word. To get at the root form, lemmatisation does not merely remove the endings of inflected words. Before lemmatisation takes place using a Word Net dictionary, part-of-speech tags are added to each word, which leads to a root that is a genuine dictionary word.

Once the preprocessing is finished, we will combine all tokens into sentences. We can perform featurization on these sentences. Figure 2 shows the word cloud of fraudulent and legitimate job ads, the importance or frequnecy of a word is represented by size.

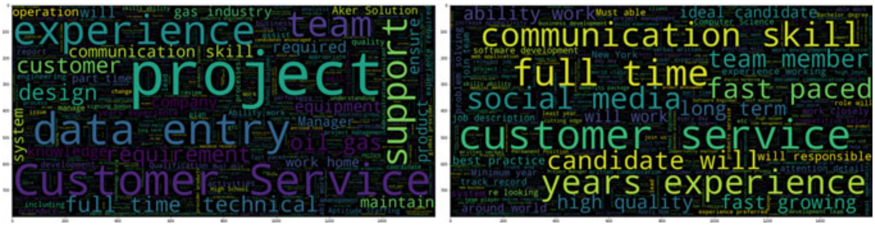


Fig. 2 Word cloud of job ads. Left: Fraudulent job ads, Right: Legitimate job ads

3.3 Featurization

ML models cannot work on non-numeric data. Hence, we must convert non-numeric data into vectors of integers or floats. Featurization can be carried out in a variety of methods. In this work, categorical and text data are used. Hence, for text data, we used BoW, TFIDF, Avg-Word2vec, and TFIDF Word2vec. One-Hot Encoding and Response Coding with Laplace Smoothing is used for categorical data.

3.3.1 One-Hot Encoding

In the proposed work, for few categorical features: Employment_type, Required_experience, and Required_education feature extraction is done using One-Hot Encoding. This approach creates a new binary feature for each possible category. The feature of each sample that originally belonged to that category is given a value of 1. For instance, the Employment_type has four categories: contract, full-time, part-time, and other. So, we create four binary variables for this. If the employment type of a particular job description is full time, then we will make the full-time variable as 1, and the remaining all are 0s.

3.3.2 Response Coding with Laplace Smoothing

When the number of categories is more in a categorical variable, then one hot encoding increases the dimensionality, creating sparsity in the numerical vector. Importantly, if a particular category never occurs in the training data, it leads to an error during run time. To avoid that, we used Laplace smoothing for the features industry, function, and department because there are high chances of new categories for these features occurring. As part of this technique, in a K-class classification problem, we need to create K new features that embed the likelihood that a particular data point belongs to a specific class, given a categorical feature value. Probability of a fake class for the department D is calculated using the Eq. (1).

$$Pr(Y = \text{Fake} \mid \text{dept} = D) = \frac{n(\text{dept} \;=\; D \quad \cap \quad Y = \text{fake}) + 1}{N + 1 * K} \tag{1}$$

where K represents the number of classes in the target variable; N represents the number of postings whose department is D.

3.3.3 Average Word2vec and TFIDF-Word2vec

Word2vec is an algorithm that creates word embeddings, merely vector representations of words [15]. In this work, pre-trained Word2vec model is used. It gives a

300-dimensional vector W for every word in the text corpus. The EMSCAD dataset has five text features: Title, Company_profile, Description, Requirements, and Benefits. All these text features are combined, creating a new feature called Text. For every word of a Text feature, a numerical vector is created. We created a numerical vector for a sentence by taking mean of all the word vectors in it. This approach is called Average Word2vec (Avg-W2V). We proposed one more approach called TFIDF-Word2vec (TFIDF-W2V).

TFIDF, defined as the term frequency-inverse document frequency [16], shown in Eq. (2), is used to determine how significant the words are throughout the entire document based on each word's frequency. Every word in the job description that is unique is given a score in the TFIDF.

$$\mathrm{TFIDF}(t, d, D) = \mathrm{TF}(t, d) \times \mathrm{IDF}(t, D) \tag{2}$$

where t, d, and D represent the word, job posting, and set of all job postings, respectively.

$\mathrm{TF}(t, d)$ calculates how often a term t appears in one job posting d. Short sentences have fewer instances of a single term, whereas large sentences have more instances of that same word. However, the word in both statements has the same significance, which is why normalization is required. So, word frequency is divided by the overall word count in that particular job posting.

$\mathrm{IDF}(t, D)$ quantifies a word's significance, and it is defined as shown in Eq. (3). Most occurring words in the job postings will have smaller IDF values. But the least occurring words will have greater IDF value as these are important to the job postings.

$$\mathrm{IDF}(t, D) = \log \frac{|D|}{1 + |\{d \in D : t \in d\}|} \tag{3}$$

TFIDF-W2V of a job posting is calculated using Eq. (4).

$$\text{TFIDF-W2V}(d) = \frac{\sum_{t \in d} W_t \cdot \mathrm{TFIDF}(t, d, D)}{\sum_{t \in d} \mathrm{TFIDF}(t, d, D)} \tag{4}$$

3.4 Data Balancing Using ADASYN

For data balancing, we can do undersampling or oversampling. Undersampling leads to loss of information. If we go for oversampling, information present in the minority class is repeated. So, the model overfits on minority class. We employed an adaptive synthetic (ADASYN) method to avoid all these issues. The fundamental idea of ADASYN is to use a weighted distribution for various minority class samples based on how challenging it is for them to learn. More synthetic data is generated for difficult-to-learn minority class examples than for easier-to-learn examples [17].

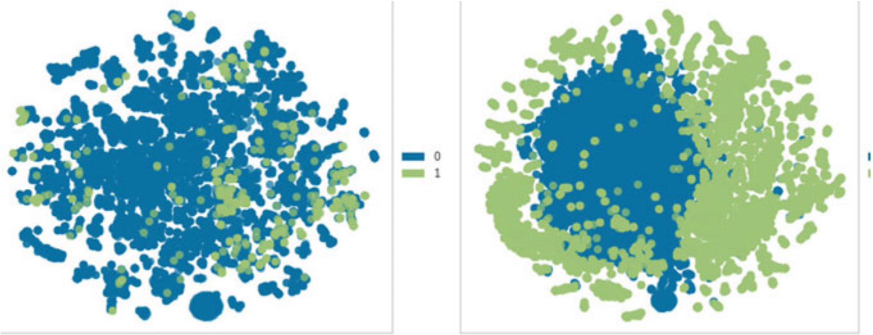


Fig. 3 Distribution of job ads. Left: Before balancing, Right: After balancing

Figure 3 shows the 2D-visualization of job ads before and after data balancing; 0 represents the Legitimate class, and 1 represents Fraudulent.

3.5 Machine Learning and Ensemble Algorithms

In order to determine whether a job posting is genuine or fake, this work uses nine supervised learning algorithms.

Logistic Regression (LoR): It is an ML classification algorithm and separates data points by finding a hyperplane. LoR assumes that all the data points are linearly separable. LoR uses a sigmoid function, which gives a probability value that can be mapped to two discrete classes.

Support Vector Machine (SVM): It is one category of ML technique that works on the concept of the hyperplane. It classifies the data points by creating a maximum marginal hyperplane between them. The maximum marginal hyperplane is the furthest away from the closest training samples in any class.

Multi-Layer Perceptron (MLP): It is an ANN, which has at least three layers: an input, a hidden, and an output layer. Except for input layer, every other layer has activation functions, which are non-linear in our task. It uses backpropagation for training and can classify non-linear separable data also.

Random Forest (RF): This classifier uses several decision trees (DTs) on different subsets of training data and majority voting is used to increase the performance. The RF uses the predictions from each tree rather than depending solely on one decision tree. Increased DT density in the forest results in improved accuracy and prevents overfitting.

Bagging: Bagging, also called Bootstrap aggregation, is the ensemble technique that is used to reduce variance of the model. In Bagging, training points are sampled at random with replacement, and Base learners are independently trained on them. In this task, we used SVM as the base learner, and majority voting is considered for more accurate predictions.

AdaBoost: AdaBoost, short for Adaptive Boosting. It uses an iterative strategy to enhance weak classifiers by learning from their mistakes. Each instance receives a new set of weights, with instances that were misclassified receiving heavier weights. It uses "sequential ensembling" and used to reduce both bias and variance. It combines several poor classifiers to produce a strong classifier with a good performance. AdaBoost is most effective when the DT is used as a base estimator.

Gradient Boost: In Gradient Boosting, each tree corrects the mistake made by its predecessor. Instead of modifying the weights of the training instances like AdaBoost, each tree is trained using the residual mistakes of the predecessor as labels. In the proposed work, 200 estimators are used.

Stacking Classifier: In this method, a meta-classifier is used to integrate various classification models. In this task, the individual classifiers, RF and SVM algorithms, are trained using training data; then, predictions of these individual learners are used as features of meta-classifier. In our case, MLP is a meta-classifier and the final prediction comes from it.

XgBoost: It is a Gradient-Boosted DT implementation. In this method, DTs are created in a sequential manner. We will give a weight to each feature before being fed into the DT. Before fed into the next tree, more weight is given to the features that are incorrectly predicted by the previous tree. A robust and accurate model is created by combining many classifiers.

4 Results and Analysis

In this section, we describe evaluation metrics and experimental results observed. After preprocessing and featurization on the dataset, we obtained 17,880 samples, out of which 17014 are real job ads, and 866 are fake job ads. Clearly, the data is heavily imbalanced. Hence, we balanced the dataset using ADASYN and two experiments are performed: one is on the imbalanced dataset, and another is on balanced data. In both cases, we split the entire dataset into training data and testing data in the ratio of 80:20, respectively.

4.1 Evaluation Metrics

In the evaluation process, we measure the trained model performance on the testing data. This model will compare ground truth with its own predictions. When our model makes predictions, four different possibilities exist:

- True Positives (TP): Indicates the total number of job ads correctly predicted as Fraudulent.
- True Negatives (TN): Indicates the total number of job ads correctly predicted as Legitimate.
- False Positives (FP): Indicates the total number of job ads that are predicted as a fraudulent but they belong to the Legitimate class.
- False Negatives (FN): Indicates the total number of job ads that are predicted as a legitimate but they belong to the Fraudulent class.

We evaluate our model based on the accuracy score and F1-score.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

$$\text{F1-score} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}} \tag{6}$$

4.2 Results on Imbalanced Dataset

Tables 1 and 2 show the accuracy and F1-score of ML and ensemble models with various featurization strategies on the imbalanced dataset. From Table 1, accuracy of all models, except for the GB, is relatively same when BoW and TFIDF are employed. With BoW, the Bagging classifier outperformed with an F1-score of 0.88 and an accuracy of 98.85%, however, with TFIDF features, Stacking classifier performance is adequate. The Avg-W2V and TFIDF-W2V featurizations resulted in the Bagging model surpassed all other models, with 0.78 F1-score and 98% accuracy. These findings suggest that ensemble-based approaches can help the baseline classifiers perform better. When compared to Avg-W2V and TFIDF-W2V, BoW and TFIDF features stood superior.

4.3 Results on Balanced Dataset

EMSCAD dataset is imbalanced as the majority of the samples belong to the Legitimate class. For data balancing, We used ADASYN, which generates synthetic points. Tables 3 and 4 show the accuracy and F1-score comparison of various models on balanced dataset, respectively. For the majority of models, TFIDF features outperform

Table 1 ML and Ensemble models accuracy (in %) comparison with BoW, TFIDF, Avg-W2V, and TFIDF-W2V on imbalanced dataset

Algorithms	BoW	TFIDF	Avg-W2V	TFIDF-W2V
LoR	98.55	98.6	96.7	96.45
RF	98.13	98.04	97.57	97.2
SVM	98.38	98.63	96.64	96.34
MLP	98.8	98.74	97.82	97.82
Bagging	98.85	98.49	98.01	98.15
AdaBoost	96.9	97.04	95.3	95.39
GB	94.21	93.88	93.34	92.9
Stacking	98.63	98.8	98.01	96.2
XgBoost	97.51	97.68	97.06	97.23

Table 2 ML and Ensemble models F1-score comparison with BoW, TFIDF, Avg-W2V, and TFIDF-W2V on imbalanced dataset

Algorithms	BoW	TFIDF	Avg-W2V	TFIDF-W2V
LoR	0.83	0.83	0.65	0.61
RF	0.76	0.74	0.67	0.66
SVM	0.82	0.84	0.65	0.63
MLP	0.87	0.86	0.77	0.76
Bagging	0.88	0.82	0.78	0.78
AdaBoost	0.68	0.68	0.55	0.54
GB	0.55	0.52	0.51	0.46
Stacking	0.85	0.87	0.73	0.62
XgBoost	0.69	0.72	0.65	0.66

BoW, Avg-W2V, and TFIDF-W2V. XgBoost excelled all other models on the BoW and TFIDF features, with the highest accuracy of 97.89 and an F1-score of 0.98 using BoW strategy. GB performed well on both Avg-W2V and TFIDF-W2V features and achieved 94% accuracy and 0.95 F1-score. From the experimental results it is observed that on balanced data sets as well, ensembles clearly outperform traditional ML techniques.

Figures 4 and 5 show the performance comparison of various models on imbalanced and balanced datasets with featurization techniques: BoW, TFIDF, Avg-W2V, and TFIDF-W2V.

Table 5 highlights the performance of the proposed method (consists of ensemble algorithms with featurization techniques using Laplace smoothed Response coding and BoW) in comparison with the existing works for fake job posting detection. From Table 5, it is evident that by using more features from the dataset and appropriate featurization techniques for categorical and text variables, the performance of the

Table 3 ML models accuracy (in %) comparison with BoW, TFIDF, Avg-W2V, and TFIDF-W2V on balanced dataset

Algorithms	BoW	TFIDF	Avg-W2V	TFIDF-W2V
LoR	85.94	89.33	86.53	83.58
RF	95.36	90.8	89.88	88.67
SVM	80.64	88.74	85.08	78.02
MLP	86.0	88.74	87.6	85.08
Bagging	88.54	89.82	86.65	83.45
AdaBoost	97.28	96.73	92.72	92.41
GB	94.69	95.39	94.91	94.63
Stacking	97.44	96.63	91.67	91.77
XgBoost	97.89	97.14	93.69	94.2

Table 4 ML models F1-score comparison with BoW, TFIDF, Avg-W2V, and TFIDF-W2V on balanced dataset

Algorithms	BoW	TFIDF	Avg-W2V	TFIDF-W2V
LoR	0.84	0.88	0.86	0.82
RF	0.95	0.90	0.89	0.87
SVM	0.76	0.87	0.84	0.74
MLP	0.84	0.87	0.86	0.83
Bagging	0.87	0.89	0.86	0.82
AdaBoost	0.97	0.97	0.93	0.92
GB	0.94	0.95	0.95	0.95
Stacking	0.97	0.97	0.91	0.92
XgBoost	0.98	0.97	0.94	0.94

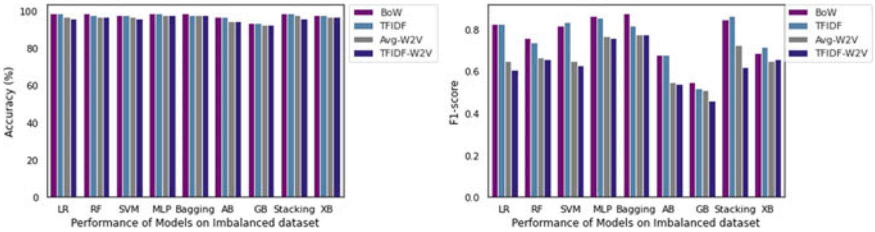


Fig. 4 Performance of ML and Ensemble models on imbalanced dataset, Left: Accuracy, Right: F1-score

model can be improved. Additionally, we proposed two featurization techniques: Avg-W2V and TFIDF-W2V, whose performance is comparable to BoW and TFIDF on the balanced dataset.

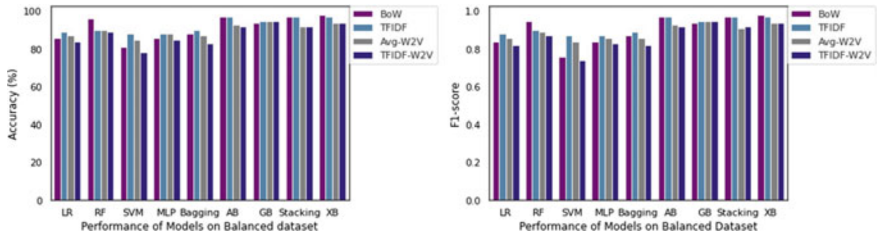


Fig. 5 Performance of ML and Ensemble models on balanced dataset, Left: Accuracy, Right: F1-score

Table 5 Comparison of the proposed method with existing works

References	Dataset	Accuracy (%)
Keerthana et al. [9]	EMSCAD	71
Ranparia et al. [7]	EMSCAD	97
Proposed method	EMSCAD	98.85

5 Conclusion and Future Works

This work classified job postings into fake or real based on job descriptions. We performed the experiments with various featurization techniques: BoW, TFIDF, Avg-W2V, and TFIDF-W2V on classical machine learning and ensemble models. We obtained good results, and it was observed that on the imbalanced dataset, the Bagging classifier performed well with BoW vectors and the Stacking classifier with TFIDF vectors. Ensemble approaches like AdaBoost, Stacking classifier, and XgBoost are equally effective with BoW and TFIDF features on the balanced dataset. Proposed featurization techniques: Avg-W2V and TFIDF-W2V are comparable to the existing methods on the balanced dataset. From the findings, we conclude that ensemble techniques perform better than classical ML models. In future work, we will consider deep learning models with various feature vector generation techniques such as Glove and FastText [18] to detect fake job ads.

References

1. Tabassum H, Ghosh G, Atika A, Chakrabarty A (2021) Detecting online recruitment fraud using machine learning. In: 2021 9th international conference on information and communication technology (ICoICT), pp 472–477. https://doi.org/10.1109/ICoICT52021.2021.9527477
2. Nindyati O, Bagus Baskara Nugraha IG (2019) Detecting scam in online job vacancy using behavioral features extraction. In: 2019 international conference on ICT for smart society (ICISS), vol 7, pp 1–4. https://doi.org/10.1109/ICISS48059.2019.8969842
3. Nasser IM, Alzaanin AH, Maghari AY (2021) Online recruitment fraud detection using ann. In: 2021 palestinian international conference on information and communication technology

(PICICT), pp 13–17. https://doi.org/10.1109/PICICT53635.2021.00015

4. Vidros S, Kolias C, Kambourakis G, Akoglu L (2017) Automatic detection of online recruitment frauds: characteristics, methods, and a public dataset. Futur Internet 9(1). https://doi.org/10.3390/fi9010006
5. Habiba SU, Islam MK, Tasnim F (2021) A comparative study on fake job post prediction using different data mining techniques. In: 2021 2nd international conference on robotics, electrical and signal processing techniques (ICREST), pp 543–546. https://doi.org/10.1109/ICREST51555.2021.9331230
6. Kambourakis G (2017) Employment scam Aegean dataset. http://emscad.samos.aegean.g. Accessed 29 Aug 2022
7. Ranparia D, Kumari S, Sahani A (2020) Fake job prediction using sequential network, pp 339–343. https://doi.org/10.1109/ICIIS51140.2020.9342738
8. Pennington J, Socher R, Manning C (2014) GloVe: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1532–1543. https://doi.org/10.3115/v1/D14-1162
9. Keerthana B, Reddy AR, Tiwari A (2021) Accurate prediction of fake job offers using machine learning. In: Bhattacharyya D, Thirupathi Rao N (eds) Machine intelligence and soft computing, vol 1280. Springer, Singapore, pp 101–112. https://doi.org/10.1007/978-981-15-9516-5_9
10. Lal S, Jiaswal R, Sardana N, Verma A, Kaur A, Mourya R (2019) Orfdetector: ensemble learning based online recruitment fraud detection. In: 2019 twelfth international conference on contemporary computing (IC3), pp 1–5. https://doi.org/10.1109/IC3.2019.8844879
11. Mehboob A, Malik MS (2020) Smart fraud detection framework for job recruitments. Arab J Sci Eng 46. https://doi.org/10.1007/s13369-020-04998-2
12. Amaar A, Aljedaani W, Rustam F, Ullah DS, Rupapara V, Ludi S (2022) Detection of fake job postings by utilizing machine learning and natural language processing approaches. Neural Process Lett 54:1–29. https://doi.org/10.1007/s11063-021-10727-z
13. Naudé M, Adebayo K, Nanda R (2022) A machine learning approach to detecting fraudulent job types. AI SOCIETY. https://doi.org/10.1007/s00146-022-01469-0
14. Real/fake job posting prediction. https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction. Accessed 29 Aug 2022
15. Mikolov T, Corrado GS, Chen K, Dean J (2013) Efficient estimation of word representations in vector space, pp 1–12. https://doi.org/10.48550/arXiv.1301.3781
16. Qaiser S, Ali R (2018) Text mining: use of TF-IDF to examine the relevance of words to documents. Int J Comput Appl 181. https://doi.org/10.5120/ijca2018917395
17. He H, Bai Y, Garcia E, Li S (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning, pp 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969
18. Joulin A, Grave E, Bojanowski P, Mikolov T (2016) Bag of tricks for efficient text classification. https://doi.org/10.48550/arXiv.1607.01759

A Conceptual Model for Click Fraud Detection and Prevention in Online Advertising Using Blockchain

Rohitkumar Jigalur and Chirag Modi

Abstract Click frauds in online advertising are increasing intentionally or unintentionally which is unnecessarily raising the cost to the advertisers. As per our observation from the literature, the existing machine learning and statistical analysis-based solutions offer insufficient accuracy in click fraud detection. In addition, a user's privacy, click data immutability and trust between the advertiser and publisher need to be achieved. To address these issues, we propose a conceptual model for click fraud detection and prevention using blockchain. It verifies ad clicks by mapping the user's mail id with their phone number and corresponding national identity value (NIV) which is unique per user in order to distinguish genuine and fraud clicks. Based on this mapping, it filters the genuine and fraud clicks by considering one click per user, which are recorded as transactions over the blockchain. The privacy of the users' information is preserved by exchanging hashed information using the SHA256 hash function. For the users, a reward mechanism is introduced to encourage the participation of genuine users. In addition, the genuine rate per advertisement is derived to represent the reputation of the publisher and website owner. The functionality of the proposed model is tested using the testbed at NIT Goa.

Keywords Online advertising · Click fraud detection and prevention · Hash function · Blockchain

1 Introduction

Online advertising has been growing rapidly for the promotion of various businesses and services. In the online advertising system, an advertiser deals with the publisher to publish his/her advertisement on various websites. The publisher selects the number of websites based on the number of visitors/users and publishes the advertisement on the selected websites. The visitors/users click on the advertisement as per their interest/unknowingly, and such clicks are forwarded to the publisher. The publisher

R. Jigalur · C. Modi (✉)
National Institute of Technology Goa, Farmagudi, Ponda, Goa 403401, India
e-mail: cnmodi@nitgoa.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_17

sends such click information to the advertiser who pays the publisher based on a deal. In addition, website owners are paid by a publisher as per their deal. The payment in such deals is based on mainly three different models viz. cost-per-click (CPC) [1], cost-per-action (CPA) [1] and cost-per-thousand-impression (CPTI) [1]. In the CPC model, an advertiser has to pay the publisher for every click made on the advertisement over the website by the users. In the CPTI model, an advertiser pays the publisher based on different actions performed on the advertisement over the websites. Different actions such as advertisement clicking, advertisement link sharing and copying the image/contents of the advertisement are considered in this model. In the CPTI model, the publisher is paid by the advertiser for every thousand actions such as advertisement clicking, browsing and sharing.

In online advertising, click frauds are rapidly increasing, and thus unnecessarily increasing the cost to the advertisers [2]. Click frauds are caused due to the several clicks performed by a user intentionally or unintentionally using manual processes or scripts. There have been several types of click frauds reported yet, which include system-generated clicks, robotic clicks, manual clicks, crowdsourcing, hit inflation attacks, incentivized traffic, click farms, impression fraud, etc. [3]. A publisher can also increase the number of clicks by making users click without their interest. Here, some rewards are given to the users for maximizing the number of clicks. Automatic redirection to the advertisement causes click fraud. In addition, botnets are created to increase the number of clicks.

In the literature, various traffic analysis, machine learning and blockchain-enabled solutions have been explored to detect click frauds in online advertising. In traffic analysis based solutions, the statistics about the received traffic at ad network are generated and based on which decision is made about fraud clicks. For example, if there is a high amount of click traffic from a particular IP address during a short interval, it is considered as fraud click traffic. This solution can detect the fraud clicks which are generated in short intervals. However, it fails in detecting the fraud clicks, if a botnet is created and multiple click traffic is generated in long intervals. Machine learning-based solutions analyze the click data patterns and distinguish between fraud and genuine clicks. In the literature, various machine learning techniques such as KNN and Artificial Neural Network (ANN) have been explored to detect click frauds. However, these solutions need a high number of click fraud evidence. In addition, the mixing of genuine and fraud clicks hinders the detection accuracy. Still, there is a scope for further improvements in terms of achieving trust among the entities involved in online advertising. Thus, a blockchain [4] can be integrated with an online advertising system to securely store the click records and to prevent any data update. Blockchain-enabled solutions attempt to achieve the user's privacy, data immutability, data transparency and trust among the participants [5].

In this paper, we design a conceptual model to detect and prevent click frauds in online advertising. The proposed model uses a blockchain to record the involved entities' data and ad click information. It verifies whether a click is genuine or not by mapping the user's mail id with the phone number and corresponding NIV which is unique per user. Based on this mapping, it filters the genuine and fraud clicks by considering the click per user. The use of blockchain helps to prevent click data modification. The identified genuine and fraud clicks are recorded on the blockchain which is later accessed for payment settlement. The cost to an advertiser is considered based on cost-per-click per user only. For users, a reward mechanism is introduced for encouraging the participation of genuine users. In addition, a genuine rate is introduced to derive the reputation of the publisher and website owner. The functionality of the proposed model is tested using the testbed at NIT Goa.

The rest of the paper is organized as follows: Sect. 2 presents the existing models of fraud click detection with the research gaps. Section 3 presents the proposed model in detail. The implementation screenshots of the proposed model are discussed in Sect. 4. Finally, Sect. 5 concludes our research work with the references at the end.

2 Related Work

In the literature, various traffic analysis, machine learning and blockchain-enabled solutions have been explored to detect click frauds in online advertising.

Traffic analysis-based solutions generate statistics about network traffic over ad networks and based on any deviation, click frauds are detected. Lee et al. [6] have performed the click fraud detection by analyzing the ad network traffic and applying the combination of different anomaly scores viz. inter-arrival time difference, diurnal activity difference and Eigenscore difference on the received traffic. Gabryel et al. [7] have applied a dynamically modified Bag-Of-Words (BoW) algorithm on ad network traffic to identify the multiple clicks from the same source or bots. Pearce et al. [8] have demonstrated ZeroAccess, a click fraud botnet, and presented the complexity of detecting click frauds. Clicktok [9] applies two types of click fraud detection such as mimicry and bait-click. Mimicry detects click spam by analyzing the repeated click traffic. In the bait-click method, a pattern of bait clicks is injected into the user's device and the mimicking of the bait clicks are considered as fraud clicks. Wiatr et al. [10] have applied a set of interoperable flags to click traffic. These flags are tagging the click events to detect any fraud clicks. AdSherlock [11], a click fraud detection for mobile apps, performs online and offline procedures. In the offline procedure, it generates the URL patterns which are later used for recognizing the click request and detecting the click frauds in the online procedure.

In machine learning technique-based solutions, the detection model is learned using the previously observed click data patterns. The learned model is later used to distinguish between fraud and genuine clicks. In the literature, various machine learning techniques such as KNN and Artificial Neural Network (ANN) have been explored to detect click frauds. Jianyu et al. [12] have proposed a coding method to convert the nominal attributes of the ad network traffic into numerical attributes. They have considered different attributes viz. IP address, cookie, user agent, etc. to train different machine learning models such as LR, DT, GBDT and XGBOOST for click fraud detection. It is observed that DT achieves higher detection accuracy. Zhang et al. [13] have combined Cost-sensitive Back Propagation Neural Network (CSBPNN) with the Artificial Bee Colony (ABC) algorithm for click fraud detection in mobile advertising. Here, ABC is used to optimize the feature selection in click data, whereas the weights of the CSBPNN are updated to reduce the classification error. For ad click fraud detection, Thejas et al. [14] have applied a deep learning model consisting of ANN, auto-encoder and generative adversarial network to click data and differentiated the genuine and fraud clicks. Minastireanu et al. [15] have used a gradient boosting decision tree algorithm to detect click frauds. Mouawi et al. [1] have implemented different classifiers viz. KNN, SVM, and ANN for detecting click fraud, and it is concluded that KNN offers higher detection accuracy. Thejas et al. [16] have proposed a Cascaded Forest and XGBoost (CFXGB) model for click fraud detection. This model performs transformation of the features and then the classification of click data to improve the detection accuracy. Choi et al. [17] have investigated different machine learning techniques for click fraud detection and concluded the importance of machine learning techniques in click fraud detection with the related challenges for further optimization.

As per our observation, the traffic analysis and machine learning-based solutions to detect click frauds are unable to achieve sufficient accuracy. The machine learning-based solutions require a high number of evidence of fraud clicks to improve the detection accuracy. In addition, there is a need of protecting the user's privacy, improving the trust among the entities involved in online advertising and offering data security, transparency and immutability. Thus, a blockchain [4] can be integrated with an online advertising system to securely store the click records and to prevent any data update. Kshetri et al. [2] have shown the potential of blockchain in online advertising and detecting/reducing click frauds. Lyu et al. [18] have proposed a blockchain-enabled click fraud detection and prevention scheme. It uses the concept of bilinear pairing to append the user's signature with the click and thus, it is able to differentiate the clicks between a user and a machine. In addition, the privacy of the users is protected using the CP-ABE crypto algorithm. Here, a consortium blockchain helps to improve the transparency of clicks. Still, there is a scope for further improvements in terms of reducing the click data processing overhead, preventing the privacy of the users' information, securing click data and improving the trust among the entities involved in online advertising.

3 Proposed Model

The objective of the proposed model is to detect and prevent click frauds in online advertising, while preventing the privacy of the users' information, securing the click data and improving the trust among the entities.

4 Notations and Assumptions

In the proposed model, we use different notations which are detailed in Table 1.

Table 1 Notations and their detail

Notation	Detail
A, P, U, WO	Advertiser, Publisher, User and Website Owner respectively
$E_i(id)$	Id of the *ith* entity (Advertiser, Publisher, User or Website Owner)
$Email(U_i), Phone(U_i)$	Email Id and phone number of *ith* user
SHA-256	Secure Hash Algorithm-256
MSP	Mail service provider
NIA	National identity agency
NIV	National identity value

In the proposed model, the following assumptions are considered, which are true in real-time implementation.

1. One advertiser can have multiple ads.
2. One advertiser can send the ads to multiple publishers.
3. One publisher can publish the obtained ads on different websites.
4. Advertising deal is made between P-A and P-WO.
5. A user can have multiple email ids and multiple mobile numbers.
6. A user has one unique national identity value. Ex: Aadhaar number in India.
7. If the mail id is not linked with the mobile number as its backup number, then that mail is treated as fraud/anonymous mail-id.

4.1 Design of the Proposed Model

A design of the proposed model for click fraud detection and prevention is shown in Fig. 1. It involves the advertiser, publisher, website owner, users, advertisement portal, central buffer, MSP, NIA and blockchain. The working of the proposed model in steps is as follows:

Step1. Registration of entities (A, P, WO): Initially, all the advertisers, publishers and website owners register with the advertisement portal (script of the model) by providing their personal identifiable information (PII) details such as name, address, email and phone. This information is stored only in a hash format using the SHA256 hash function.

Step 2. Verification of entities (A, P, WO): After the submission of PII, the advertisement portal verifies the phone number and email-id using OTP-based verification. On successful verification, it generates a unique ID $E_i(id)$ for the registered entities, which is stored in the central buffer. The central buffer is assumed as a secured database with a backup database of the advertisement portal. It stores each entity's information. In addition, a hash of the same "$SHA_{256}(E_i(id))$" is sent as a transaction to the entity blockchain. Thus, the hash of each entity's id is recorded in the

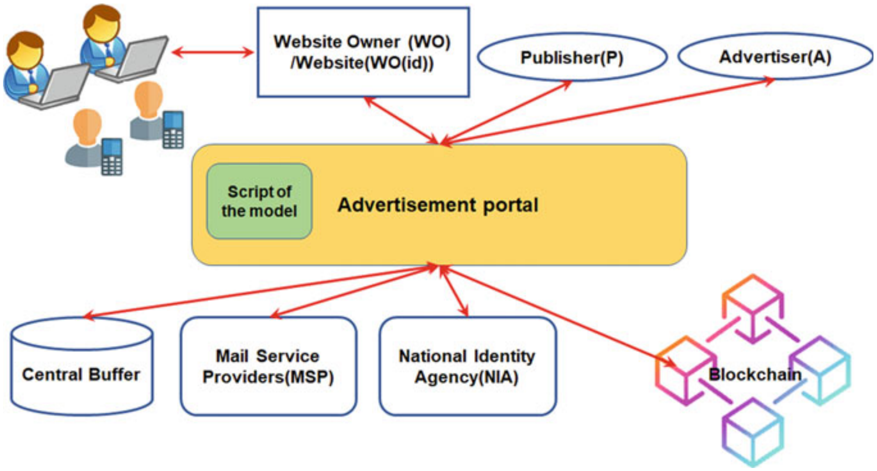

Website Owner (WO) /Website(WO(id))
Publisher(P)
Advertiser(A)
Script of the model
Advertisement portal
Central Buffer
Mail Service Providers(MSP)
National Identity Agency(NIA)
Blockchain

Fig. 1 Design of the proposed model for click fraud detection and prevention

blockchain in append-only manner. On failure of verification, entities have to follow step 1.

Step 3. Agreements ($A_i \rightarrow \exists P_i, P_i \rightarrow \exists WO_i$): Each advertiser makes a cost-per-click (CPC) model-based agreement with the selected publishers through the advertisement portal. Similarly, each publisher makes an agreement with the selected website owners through the advertisement portal. For getting more rewards, the publisher chooses website owners based on their number of users. In this paper, we have considered a web service-based centralized agreement with the assumption that the advertisement portal is trusted and secured. In future, distributed smart contracts can be considered for making agreements between advertisers and publishers; and publishers and website owners to improve trust. Then, the advertiser sends an advertisement to the selected publishers.

Step 4. Advertisement Publishing: A publisher who receives an advertisement from the advertiser sends that advertisement to the selected website owners by appending hashed id of the advertiser and hashed email id of the publisher and advertiser. Thus, with the advertisement, the following information is appended.

$$SHA_{256}(A_i(id)), SHA_{256}(email(P_i)), SHA_{256}(email(A_i)) \rightarrow \text{Website owners}$$

The website owner publishes the received advertisement along with the above information on his/her website.

Step 5. User Clicks and Rewards: The users of the website open the website as per their requirements and may click the displayed advertisement with their interest or by mistake. To encourage the participation of more number of users, we have considered a reward mechanism for the users. However, a user will get a reward (% of a service charge of service provider fee) per advertisement rather than per click. To get a reward, a user has to register on the advertisement portal as discussed in step

1. Upon successful verification using step 2, a user is entitled to a reward. If a user disagrees to give his email id, upon clicking on the advertisement, the advertisement portal fetches the email id from the user's browser where he is logged in. If a user is not logged in and he/she clicks the advertisement, the email id field of the user is considered as void. The void clicks are not considered for the payment in cost-per-click model. Finally, the advertisement portal gets the following data upon clicking on an advertisement:

> $SHA_{256}(email(U_i)/void)$, $SHA_{256}(A_i(id))$, $SHA_{256}(email(P_i))$, $SHA_{256}(email(A_i))$, $WO_i(id)$ and $A_i(id)$

Step 6. Click Verification: The central buffer matches the $SHA_{256}(email(U_i))$ and $A_i(id)$ with the previously recorded genuine and fraud clicks. If any match is found, the current click is considered as fraud click and stored on the blockchain. If it is a new click, the received mail id of the user $SHA_{256}(email(U_i))$ is sent to the mail service provider (MSP) to check whether a phone number is linked with the mail id or not. MSP matches the received $SHA_{256}(email(U_i))$ with the hash of the user mail id stored in its database. If any match is found, it finds the associated phone number, and the hash of the phone number $SHA_{256}(phone\ number)$ is returned to the central buffer. In case the phone number is not associated with the received hash of the mail id, that mail id is considered as fake/anonymous and, thus, the corresponding click is considered as fraud click. After receiving the hash of the phone number $SHA_{256}(phone\ number)$, the central buffer sends it to the national identity agency (NIA) to receive the user's unique national identity value (e.g. Aadhar number in India). The NIA matches the received $SHA_{256}(phone\ number)$ with the hash of NIV stored in its database. If any match is found, it finds the associated NIV and the hash of the NIV $SHA_{256}(NIV)$ is returned to the central buffer. The corresponding click is considered as genuine click. In case, the NIV is not associated with the received hash of the phone number, the corresponding click is considered as fraud click. Thus, the advertisement portal has a list of genuine and fraud clicks with the associated hashed information. In the proposed model, we have considered NIV (which is unique per user) to identify genuine and fraud clicks. Thus, even though a user performs multiple clicks using different mail ids or phone numbers, his NIV/her value is unique, and, thus, redundancy in the list of genuine clicks is removed and fraud clicks are prevented. The list of genuine and fraud clicks is recorded on the blockchain so that later entities (advertiser, publisher and website owner) can verify the click information during payment settlement.

Step 7. Genuine Rate Monitoring: The advertisement portal continuously monitors the number of genuine and fraud click transactions on the blockchain and displays the genuine rate per advertisement using the following equation:

$$\text{Genuine Rate} = \frac{x}{x+y} \times 100 \tag{1}$$

Where $x =$ number of genuine click transactions recorded on blockchain and $y =$ number of fraud click transactions recorded on the blockchain. The genuine rate acts as the reputation of the publisher and website owner, and thus it helps an advertiser to choose genuine publishers for the advertisement.

Step 8. Payment Settlement: The payment is settled between the advertiser and publisher by considering the cost-per-click per user. Between the publisher and website owner, the payment is settled as per the agreement. The registered users get a reward from the website owner after the consideration of their clicks as genuine. In future, payment settlement can be carried out over blockchain.

4.2 Security Analysis of the Proposed Model

In the proposed model, the information of entities is exchanged in the form of hash value derived using SHA-256 instead of revealing the actual data. The information privacy of the entities is preserved due to the one-wayness property of the hash function. It is highly impractical to derive the actual data from the given hash value. The SHA-256 has 128-bit collusion resistance, which indicates that the success probability of breaking the SHA-256 is 50%, if an adversary randomly generates 2^{128} messages. However, it is highly impractical in real time due to the limited computing capabilities of an adversary. If a user performs multiple clicks using the same mail id, only the first click is considered as genuine upon retrieval of the hash of his/her phone number and NIV from MSP and NIA, respectively. Thus, a publisher is paid only if a user clicks an advertisement using mail which is associated with a phone number, and that phone number is associated with his/her NIV. Otherwise, it causes a loss to the publisher even though the user is genuine. However, it helps to detect and prevent fraud clicks. If a user performs multiple clicks using different mail ids and each mail id is associated with different phone numbers, only one click is considered as genuine, in which the phone number is linked to his/her unique NIV. Thus, the proposed model detects and prevents fraud clicks with the privacy preservation of users' information.

The security of the proposed model can be further improved by securing an advertisement portal, central buffer and network where data-in-transit and data-at-rest can be modified by an adversary. Such modification can affect the working of the proposed model. To overcome this problem, secured network communication through SSL/TLS can be considered. For securing the data at rest, strong access control and authentication mechanisms need to be incorporated.

5 Implementation

For the implementation of the proposed model, we have used different tools such as Node.js, Pyautogui library and Postgres. For front-end development and blockchain implementation in the proposed model, we have used Node.js (HTML, CSS, JS).

PostgreSQL is used as a central data buffer. Python is used for generating the automated clicks, and it is integrated with JavaScript. In addition, Pyautogui library is used to fetch the mail id of the users from the browser. The implementation screenshots of the proposed model are shown in Figs. 2, 3, 4 and5. Figure 2 shows the screenshot of the advertisement portal through which advertisers, publishers and website owners can register for publishing the advertisement.

Figure 3 shows the sample database of the central buffer, where all the data about advertisers, publishers and website owners are stored in a hash format using the SHA256 algorithm. Thus, the privacy of the entity's information is preserved.

Figure 4 shows the sample database of NIA, created for hash matching of the received phone number and its associated NIV. Thus, the privacy of the entity's information is preserved.

Figure 5 shows the sample structure of the implemented blockchain with proof of work consensus for the proposed model. Here, the transaction related to the genuine and fraud clicks are recorded. In future, the scalability and throughput of the blockchain can be further improved by considering feasible consensus. In this paper, we have focused only on detecting and preventing fraud clicks rather than improving the performance of blockchain. The proposed system successfully detects and prevents the fraud clicks in online advertising with privacy preservation of the users' information.

In contrast to the existing traffic analysis [6–11] and machine learning- [12–17] based click fraud detection solutions, the proposed model does not require to

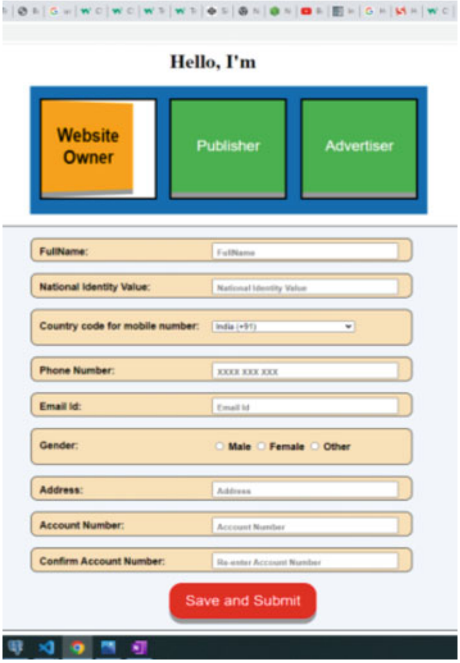
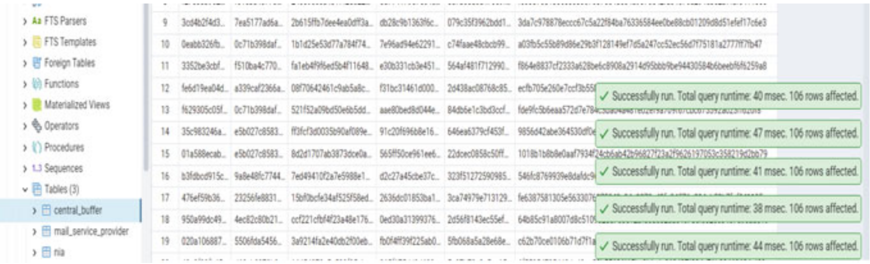


Fig. 2 Front-end of the proposed model

Hello, I'm

FullName:	FullName
National Identity Value:	National Identity Value
Country code for mobile number:	India (+91)
Phone Number:	XXXX XXX XXX
Email Id:	Email Id
Gender:	○ Male ○ Female ○ Other
Address:	Address
Account Number:	Account Number
Confirm Account Number:	Re-enter Account Number



9	3cd4b2f4d3...	7ea5177ad6a...	2b615ffb7dee4ea0dff3a...	db28c9b1363f6c...	079c35f3962bdd1...	3da7c978878eccc67c5a22f84ba76336584ee0be88cb01209d8d51efef17c6e3
10	0eabb326fb...	0c71b398daf...	1b1d25e53d77a784f74...	7e96ad94e62291...	c74faae48cbcb99...	a03fb5c55b89d86e29b3f128149ef7d5a247cc52ec56d7f75181a2777ff7fb47
11	3352be3cbf...	f510ba4c770...	fa1eb4f9f6ed5b4f11648...	e30b331cb3e451...	564af481f712990...	f864e8837cf2333a628be6c8908a2914d95bbb9be94430584b6beebf6f6259a8
12	fe6d19ea04d...	a339caf2366a...	08f70642461c9ab5a8c...	f31bc31461d000...	2d438ac08768c85...	ecfb705e260e7ccf3b55
13	f629305c05f...	0c71b398daf...	521f52a09bd50e6b5dd...	aae80bed8d044e...	84db6e1c3bd3ccf...	fde9fc5b6eaa572d7e78
14	35c983246a...	e5b027c8583...	ff3fcf3d0035b90af089e...	91c20f696b8e16...	646ea6379cf453f...	9856d42abe364530df0e
15	01a588ecab...	e5b027c8583...	8d2d1707ab3873dce0a...	565ff50ce961ee6...	22dcec0858c50ff...	1018b1b8b8e0aaf7934f24cb6ab42b96827f23a2f9626197053c358219d2bb79
16	b3fdbcd915c...	9a8e48fc7744...	7ed49410f2a7e5988e1...	d2c27a45cbe37c...	323f51272590985...	546fc8769939e8dafdc9
17	476ef59b36...	23256fe8831...	15bf0bcfe34af525f58ed...	2636dc01853ba1...	3ca74979e713129...	fe6387581305e563307
18	950a99dc49...	4ec82c80b21...	ccf221cfbf4f23a48e176...	0ed30a31399376...	2d56f8143ec55ef...	64b85c91a8007d8c510
19	020a106887...	5506fda5456...	3a9214fa2e40db2f00eb...	fb0f4ff39f225ab0...	5fb068a5a28e68e...	c62b70ce0106b71d7f1a

✓ Successfully run. Total query runtime: 40 msec. 106 rows affected.

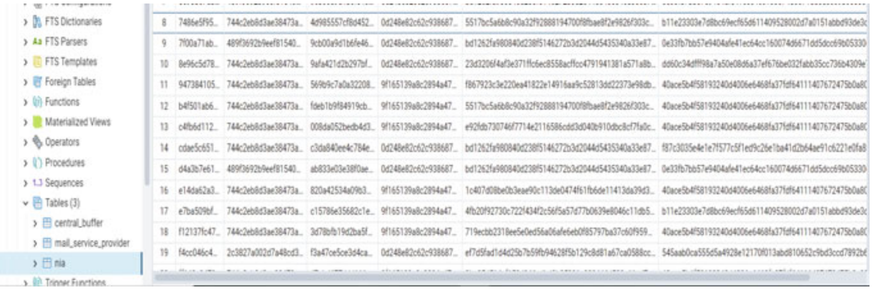
✓ Successfully run. Total query runtime: 47 msec. 106 rows affected.

✓ Successfully run. Total query runtime: 41 msec. 106 rows affected.

✓ Successfully run. Total query runtime: 38 msec. 106 rows affected.

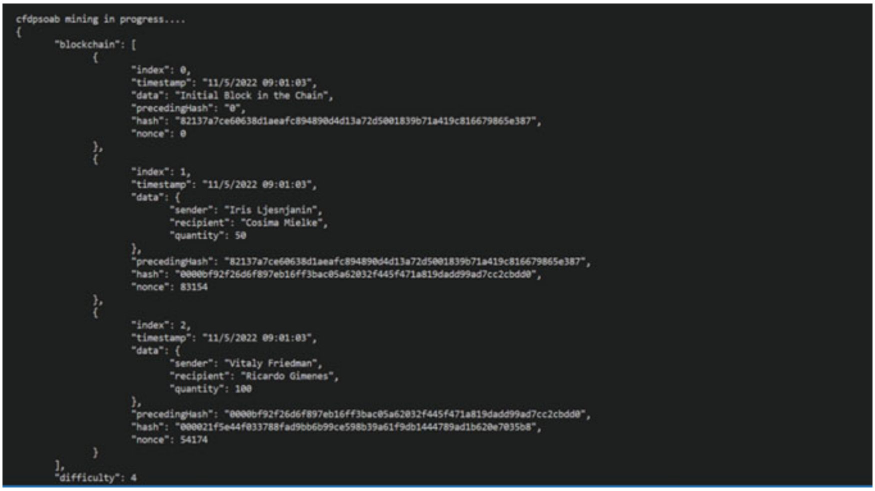
✓ Successfully run. Total query runtime: 44 msec. 106 rows affected.

Fig. 3 Central buffer in the proposed model



8	7486e5f95...	744c2eb8d3ae38473a...	4d985557cf8d452...	0d248e82c62c938687...	5517bc5a6b8c90a32f92888194700f8fbae8f2e9826f303c...	b11e23303e7d8bc69ecf65d611409528002d7a0151abbd93de3c
9	7f00a71ab...	489f3692b9eef81540...	9cb00a9d1b6fe46...	0d248e82c62c938687...	bd1262fa980840d238f5146272b3d2044d5435340a33e87...	0e33fb7bb57e9404afe41ec64cc160074d6671dd5dcc69b05330
10	8e96c5d78...	744c2eb8d3ae38473a...	9afa421d2b297bf...	0d248e82c62c938687...	23d3206f4af3e371ffc6ec8558acffcc4791941381a571a8b...	dd60c34dfff98a7a50e08d6a37ef676be032fabb35cc736b4309e
11	947384105...	744c2eb8d3ae38473a...	569b9c7a0a32208...	9f165139a8c2894a47...	f867923c3e220ea41822e14916aa9c52813dd22373e98db...	40ace5b4f58193240d4006e6468fa37fdf64111407672475b0a8c
12	b4f501ab6...	744c2eb8d3ae38473a...	fdeb1b9f84919cb...	9f165139a8c2894a47...	5517bc5a6b8c90a32f92888194700f8fbae8f2e9826f303c...	40ace5b4f58193240d4006e6468fa37fdf64111407672475b0a8c
13	c4fb6d112...	744c2eb8d3ae38473a...	008da052bedb4d3...	9f165139a8c2894a47...	e92fdb730746f7714e2116586cdd3d040b910dbc8cf7fa0c...	40ace5b4f58193240d4006e6468fa37fdf64111407672475b0a8c
14	cdae5c651...	744c2eb8d3ae38473a...	c3da840ee4c784e...	0d248e82c62c938687...	bd1262fa980840d238f5146272b3d2044d5435340a33e87...	f87c3035e4e1e7f577c5f1ed9c26e1ba41d2b64ae91c6221e0fa8
15	d4a3b7e61...	489f3692b9eef81540...	ab833e03e38f0ae...	0d248e82c62c938687...	bd1262fa980840d238f5146272b3d2044d5435340a33e87...	0e33fb7bb57e9404afe41ec64cc160074d6671dd5dcc69b05330
16	e14da62a3...	744c2eb8d3ae38473a...	820a42534a09b3...	9f165139a8c2894a47...	1c407d08be0b3eae90c113de0474f61fb6de11413da39d3...	40ace5b4f58193240d4006e6468fa37fdf64111407672475b0a8c
17	e7ba509bf...	744c2eb8d3ae38473a...	c15786e35682c1e...	9f165139a8c2894a47...	4fb20f92730c722f434f2c56f5a57d77b0639e8046c11db5...	b11e23303e7d8bc69ecf65d611409528002d7a0151abbd93de3c
18	f12137fc47...	744c2eb8d3ae38473a...	3d78bfb19d2ba5f...	9f165139a8c2894a47...	719ecbb2318ee5e0ed56a06afe6eb0f85797ba37c60f959...	40ace5b4f58193240d4006e6468fa37fdf64111407672475b0a8c
19	f4cc046c4...	2c3827a002d7a48cd3...	f3a47ce5ce3d4ca...	0d248e82c62c938687...	ef7d5fad1d4d25b7b59fb94628f5b129c8d81a67ca0588cc...	545aab0ca555d5a4928e12170f013abd810652c9bd3ccd7892b6

Fig. 4 Sample NIA database in the proposed model



```
cfdpsoab mining in progress....
{
        "blockchain": [
                {
                        "index": 0,
                        "timestamp": "11/5/2022 09:01:03",
                        "data": "Initial Block in the Chain",
                        "precedingHash": "0",
                        "hash": "82137a7ce60638d1aeafc894890d4d13a72d5001839b71a419c816679865e387",
                        "nonce": 0
                },
                {
                        "index": 1,
                        "timestamp": "11/5/2022 09:01:03",
                        "data": {
                                "sender": "Iris Ljesnjanin",
                                "recipient": "Cosima Mielke",
                                "quantity": 50
                        },
                        "precedingHash": "82137a7ce60638d1aeafc894890d4d13a72d5001839b71a419c816679865e387",
                        "hash": "0000bf92f26d6f897eb16ff3bac05a62032f445f471a819dadd99ad7cc2cbdd0",
                        "nonce": 83154
                },
                {
                        "index": 2,
                        "timestamp": "11/5/2022 09:01:03",
                        "data": {
                                "sender": "Vitaly Friedman",
                                "recipient": "Ricardo Gimenes",
                                "quantity": 100
                        },
                        "precedingHash": "0000bf92f26d6f897eb16ff3bac05a62032f445f471a819dadd99ad7cc2cbdd0",
                        "hash": "000021f5e44f033788fad9bb6b99ce598b39a61f9db1444789ad1b620e7035b8",
                        "nonce": 54174
                }
        ],
        "difficulty": 4
```

Fig. 5 Sample blockchain structure in the proposed model

generate the network traffic statistics to detect the click frauds. It detects click frauds in real time by matching users' attributes to the unique value. Thus, irrespective of traffic interval, it accurately detects and prevents the click frauds. Unlike, machine learning-based solutions, the proposed system does not require the past evidence of click frauds to train the model. In addition, the mixing of genuine and fraud clicks does not affect the detection accuracy. Thus, the proposed model has reduced computation with higher detection accuracy. The existing blockchain-based click fraud detection solution [18] suffers from high computational costs for media sites to identify the real clicks as it involves public key cryptography-based encryption for each click. In the proposed model, the detection and prevention of click frauds are performed through hash matching of the user's click attributes, and thus reducing the overall computational cost.

6 Conclusion and Future Work

In online advertising, fraud clicks are unnecessarily increasing the cost to the advertisers without any promotional benefits. To detect and prevent fraud clicks in online advertising, we have proposed a conceptual model using blockchain. It detects and prevents the fraud clicks by considering the cost-per-click per user. It maps the email id of the users with their phone number using MSP, who are clicking an advertisement. The mapped phone number is further mapped with the NIV to determine genuine or fraud clicks. Thus, multiple clicks on an advertisement from a user with different mail ids and phone numbers can be detected and prevented due to the unique NIV. Here, the privacy of the users' data is protected with the help of the SHA256 hash function. The blockchain helps to prevent any modification to genuine and fraud click transactions and thus, it improves trust. In the proposed model, central buffers act as primary storage for blockchain. Thus, it prevents data from dropping in the blockchain. In addition, it implements a reward mechanism for the genuine users. This encourages more users to participate in online advertising. The genuine rate in the proposed model indicates the reputation score of the publishers and website owners. This helps an advertiser to choose genuine publishers and website owners for the advertisement.

In the future, the security of the advertisement portal can be further improved by considering the multiple servers with a backup database for a central server. The scalability and throughput of the blockchain can be further improved by considering feasible consensus for faster confirmation of the transactions. Interfaces with the advertisement portal, NIA, MSP and blockchain need to be secured to prevent data modification during data-in-transit.

References

1. Mouawi R, Awad M, Chehab A, Hajj E, Imad H, Kayssi A (2018). Towards a machine learning approach for detecting click fraud in mobile advertizing. 2018 International Conference on Innovations in Information Technology (IIT), IEEE, pp 88–92
2. Kshetri N, Voas J (2019) Online advertising fraud. Computer 52(1):58–61
3. Gohil N, Meniya A D (2020) A Survey on Online Advertising and Click fraud detection. In: 2nd National Conference On Research Trends in Information and Communication Technology, pp 1–5
4. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf
5. Muzumdar A, Modi C, Madhu G, Vyjayanthi C (2021) A trustworthy and incentivized smart grid energy trading framework using distributed ledger and smart contracts. J Netw Comput Appl 103(074):183–184
6. LEE S-C, Faloutsos C, Chae D-K, Kim S-W (2017) Fraud Detection in Comparison-Shopping Services: Patterns and Anomalies in User Click Behaviors. IEICE Transactions on Information and Systems E100.D(10): 2659–2663
7. Gabryel M, Przybyszewski K (2019) The dynamically modified BoW algorithm used in assessing clicks in online ads. Artificial Intelligence and Soft Computing, Springer, in Proceedings of the International Conference on Artificial Intelligence and Soft Computing, pp 350–360
8. Pearce P, Dave V, Grier C, Levchenko K, Guha S, McCoy D, Paxson V, Savage S, Voelker G M (2014) Characterizing Large-Scale Click Fraud in ZeroAccess. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14). Association for Computing Machinery, pp 141–152
9. Nagaraja S, Shah R (2019) Clicktok: Click Fraud Detection using Traffic Analysis. Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks - WiSec 19, pp 1–11
11. Wiatr R, Vladyslav L, Miłosz D, Renata S, Jacek K (2020) Click-Fraud Detection for Online Advertising. In book: Parallel Processing and Applied Mathematics, 13th International Conference, PPAM 2019, pp 261–271
11. Cao C et al (2021) AdSherlock: efficient and deployable click fraud detection for mobile applications. IEEE Trans Mob Comput 20(4):1285–1297
12. Jianyu W, Chunming W, Shouling J, Qinchen G, Zhao L (2017) Fraud detection via coding nominal attributes. In: Proceedings of the 2017 2nd International Conference on Multimedia Systems and Signal Processing—ICMSSP, 2017, pp 42–45
13. Zhang X, Liu X, Guo H (2018) A Click Fraud Detection Scheme based on Cost sensitive BPNN and ABC in Mobile Advertising. In: 2018 IEEE 4th International Conference on Computer and Communications (ICCC), pp 1360–1365
14. Thejas G. S., Boroojeni K. G., Chandna K, Bhatia I, Iyengar S. S., Sunitha N R (2019) Deep Learning-based Model to Fight Against Ad Click Fraud. Proceedings of the 2019 ACM Southeast Conference - ACM SE, pp 176–181
15. Minastireanu E A, Mesnita G (2019) Light GBM Machine Learning Algorithm to Online Click Fraud Detection. IBIMA Publishing, https://ibimapublishing.com/articles/JIACS/2019/263928/, last accessed 14/9/2022
16. Thejas GS, Dheeshjith S, Iyengar SS, Sunitha NR, Badrinath P (2021) A hybrid and effective learning approach for Click Fraud detection. Machine Learn Appl. https://doi.org/10.1016/j.mlwa.2020.100016
17. Choi J-A, Lim K (2020) Identifying machine learning techniques for classification of target advertising. ICT Express 6(3):175–180
18. Lyu Q, Li H, Zhou R, Zhang J, Zhao N, Liu Y, Leng J (2022) BCFDPS: A blockchain-based click fraud detection and prevention scheme for online advertising. Sec Commun Netw. https://doi.org/10.1155/2022/3043489

Machine Learning-Based Malware Detection and Classification in Encrypted TLS Traffic

Himanshu Kashyap, Alwyn Roshan Pais, and Cheemaladinne Kondaiah

Abstract Malware has become a significant threat to Internet users in the modern digital era. Malware spreads quickly and poses a significant threat to cyber security. As a result, network security measures play an important role in countering these cyber threats. Existing malware detection techniques are unable to detect them effectively. A novel Ensemble Machine Learning (ML)-based malware detection technique from Transport Layer Security (TLS)-encrypted traffic without decryption is proposed in this paper. The features are extracted from TLS traffic. Based on the extracted features, malware detection is performed using Ensemble ML algorithms. The benign and malware file datasets are created using features extracted from TLS traffic. According to the experimental results, the 65 new extracted features perform well in detecting malware from encrypted traffic. The proposed method achieves an accuracy of 99.85% for random forest and 97.43% for multiclass classification for identifying malware families. The ensemble model achieved an accuracy of 99.74% for binary classification and 97.45% for multiclass classification.

Keywords Malware · TLS · Ensemble · Machine learning

1 Introduction

The Internet has become a vital, valuable resource in our lives. We usually use the Internet for numerous functions like doing transactions, social community interactions, etc. We generally rely on the fact that our private information remains private on the Internet. Maintaining the privacy of our Internet activities and the information we share regularly has become essential. This worry prompted the creation of the Secure Socket Layer (SSL) protocol in early 1996, which was subsequently surpassed

H. Kashyap · A. R. Pais · C. Kondaiah (✉)
Information Security Research Lab, Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal 575025, Karnataka, India
e-mail: chkondaiah.217cs001@nitk.edu.in

A. R. Pais
e-mail: alwyn@nitk.edu.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_18

by the TLS protocol in 1999 [2]. The TLS 1.2/1.3 protocol performs encryption at the transport layer. According to Request for Comments (RFC) [2], TLS is a cryptographic protocol that ensures confidentiality, the integrity of information, and privacy between communicating users.

TLS lies above the transport layer and below the application layer and primarily delivers TLS over UDP [21]. TLS is primarily used to encrypt HTTP traffic and transform it into HTTPS traffic. It can also be used with any application layer packet, for example, to increase email security by encrypting a Simple Mail Transfer Protocol(SMTP) packet using TLS and transforming it into SMTPS packets. TLS 1.3 includes multiple significant improvements and outperforms the previous version. Almost all major browsers support it, including Chrome, Firefox, and Opera.

In this paper, we examined encrypted TLS handshake interactions between users. We discovered how malicious TLS flow differs from enterprise flow in cipher suites, cryptographic parameters, and communication extensions. Malicious flows often employ old encryption parameters, whereas enterprise flow employs current ones. These differences between the parameters are utilized to classify the malware traffic. Network data has a statistical nature, which ML could use to solve the encrypted traffic classification problem. ML algorithms can detect network irregularities by evaluating metadata like protocol usage, packet density and size, and communication direction. Intrusion detection systems are inefficient because they decrypt network packets, compromising the integrity and confidentiality of the data. To address this problem, an ML model inspired by the CISCO research team [3–5] is proposed in this work that focuses on network metadata rather than network packet contents.

The rest of the paper is structured as follows: In Sect. 2, the related literature is reviewed. Section 3 discusses the proposed work. In Sect. 4, Experimentation and Results are presented. Finally, the conclusion is presented in Sect. 5.

2 Review on Literature

A lot of research is now being done on analyzing encrypted TLS traffic. This study does not aim to regurgitate all of the existing knowledge on the subject but rather to review some of the proposed ML-based TLS analysis approaches and highlight the study's distinctive aspects.

In several methodologies, flow-based metadata is used for detecting the flaws in the encrypted network traffic. These methods generate significant network flow. This data is known as IPIFIX [1] or NetFlow [6].

There have been many works [7, 9, 17, 19, 24] that use the flow-based features to detect the malware in encrypted network traffic. Some papers used inter-arrival time and size of the network traffic flow to extract the information, while some have used data-dependent features on the packet size to analyze the site fingerprinting within the encrypted network traffic.

In [23], a Hidden Markov model was employed to predict the types of IP traffic protocols based on the size and duration of the network traffic data. Similarly, [12] investigates a model for TLS flow classification based on a weighted ensemble of a second-order Hidden Markov model. As a feature, TLS records the amount of the DNS request.

The majority of the available information originates from a CISCO team that has investigated [3–5, 8, 14, 15, 25] the detection of encrypted malware. They have chosen to concentrate on encrypted TLS communication in this particular work [5]. It focuses on distinguishing regular network flow from malware based on NetFlow and TLS handshake metadata combinations. It has examined both malware family identification and normal traffic distribution without decrypting the communication. In their research, they use a variety of features like Flow MetaData, the sequence of packet lengths, the distribution of bytes, and the information included in TLS headers. The literature survey is summarized in Table 1.

Reference [11] have worked on supervised learning with the Weka tool but they did not address catching effects at the client browser. Reference [17] have worked packet-related features but this work did not address other related features. Reference [3] used only application layer. Reference [8] used only TLS header features and could not avoid false positives. Reference [13] used only two features. Reference [22] has worked unsupervised features to detect the malware but did not address the lack of a comprehensive and class imbalance. To overcome these problems, we are proposing an Ensemble ML technique that classifies the malware based on transport layer traffic without decrypting the data at the application layer. Hence, the proposed technique is computationally efficient.

3 Proposed Work

The architecture of the proposed ML-based malware detection and classification model is given in Fig. 1.

3.1 Dataset

A publicly available dataset [10] of malicious and legitimate TLS network traffic was used to carry out the experiments. Between November 2021 and March 2022, benign packet files were recorded while browsing the website based on Alexa's top million listings. This study chooses a set of malicious TLS traffic data. It is common for packet capture files from the same malware family to be concatenated into a single file containing all the flows that were captured. Malware families with less than 50 flows are discarded after preprocessing. Table 2 lists the malware families based on the number of flows after preprocessing.

Table 1 Literature survey summary

Author	Features	Technique	Dataset	Purpose	Weakness
[11]	Flow-based metadata [such as packets and network flows]	Supervised learning, preprocessing done with WEKA tool, Decision Tree, IBk, AdaBoost classifier	NSL-KDD	Detection of anomalies in network traffic dependent on big data	Catching effects at the client browser are not considered
[17]	Packet inter-arrival time, payload size, entropy-based effective bandwidth	Naive Bayes	Utilizing the network monitor, they created their dataset	Classification of encrypted traffic using commonly available information alone	The independence of flows
[3]	TLS handshake meta-data,DNS flows and HTTP headers	DataOmnia approach	CISCO threat grid	Detect threats in encrypted traffic with high accuracy	This is applicable only application layer
[8]	TLS encrypted headers information	Logistic regression classifier	Enterprise network	Detect threats in encrypted traffic without decryption	We still cannot avoid false postives
[13]	The sequence of TLS record size, type, and direction are extracted as features from the reassembled TCP session	SVM and CNN	Stratosphereips IPS	detect the presence of malware in a network flow	This is applicable only two features
[22]	Unsupervised feature adaptive learning for feature extraction	Logistic regression with neural network	Stratosphere IPS	Classification of eight different types of malicious SSL/TLS traffic with unsupervised learning	The lack of a comprehensive, class imbalanced, realistic and convincing public dataset in the area of encrypted malicious traffic detection

Table 2 Malware families

Malware family	Number of flows	Unique destination port	Unique source port	Unique client keys
Trickbot	6191	3	1580	7
Dridex	3482	16	544	3
Artemis	2119	17	533	3
Zeus	1459	1	331	3
Trojan	465	54	266	2
Unclassified	428	13	252	6
Dynamer	395	2	296	3
Vawtrak	294	1	283	3
Emotet	243	2	217	5
Miuref	191	1	85	2
Yakes	175	22	159	3
Ursnif	137	14	132	2
HTBot	80	7	73	2
Geodo	72	1	72	2
Normal	5673	2	1656	7

3.2 Preprocessing

The different steps involved in data preprocessing are as follows:

- The initial stage in developing an ML model is to collect a large amount of malicious and benign data under various scenarios. Packet sniffers can collect data, but TLS flows must be present. Cisco Joy tool [15] is an open-source analysis application that extracts data features from packet capture files (.pcap) and then parses the output into a JSON file using a flow-oriented technique similar to IPPIX or Netflow. TLS flows, packets, and bytes distribution are all extracted using it.
- Unnecessary flows in JSON files are removed prior to extracting the features [25].
  - TLS flow with an incomplete handshake or less than three packets in each direction were deleted.
  - TLS flow with an insufficient number of incoming and outgoing packets.
  - Finally, flows lacking critical elements such as cipher suites and server certificates were excluded. A benign flow may be present in a malware flow. Such flows are deleted by verifying their destination host to see if it is in Alexa's top million lists.
- To improve ML model efficiency, normalization is performed. After normalization, each numerical feature is assigned a value between 0.0 and 1.0. The following Eq. 1 denotes x_i as normalized value

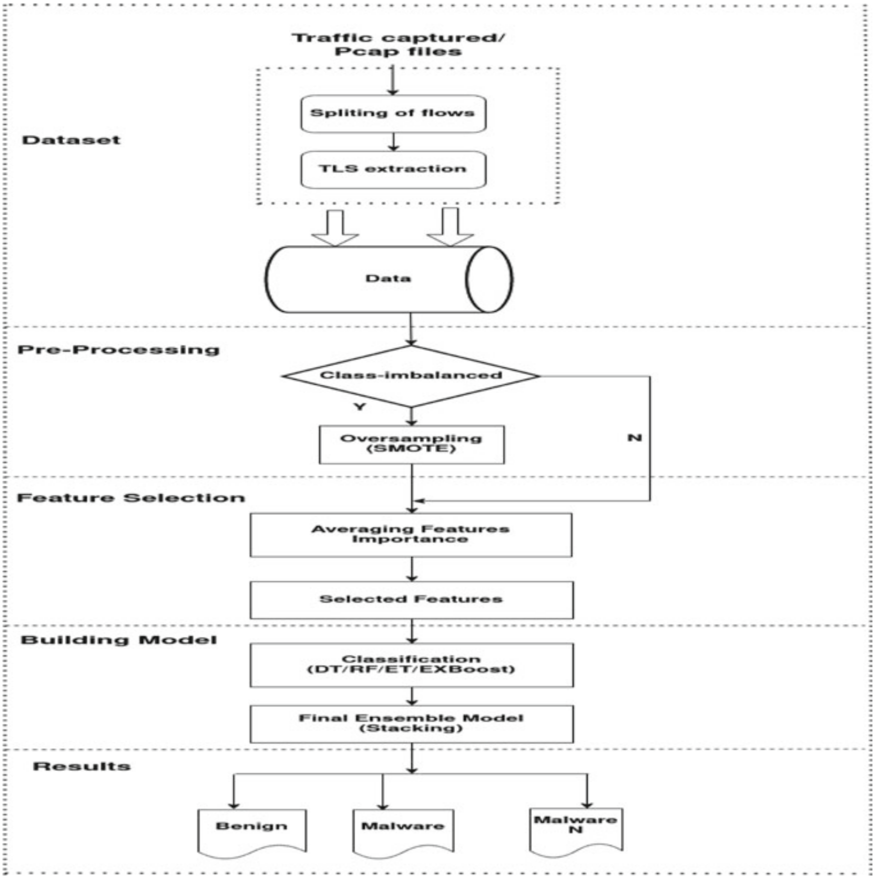


Fig. 1 The proposed stacking framework

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

where x is the initial value, max and min are the maximum and minimum values, respectively, for each feature.

3.3 Feature Extraction

To extract features, we use the CISCO Joy tool [15]. The data is exported from the open-source project using an appropriate JSON format. Constructing ML classifiers requires using conventional flow features, conventional "side-channel" characteristics, and additional information obtained from an unencrypted TLS handshake communication. CISCO has selected the following characteristics based on their observations and study on TLS parameters [16].

- **Flow Metadata**: Non-TLS-related data properties are observable in any Netflow traversing the network.
- **Distributions**: When it comes to packet flows, frequency analysis is necessary to determine the terms for data characteristics that cannot be retrieved entirely from network packets.
  - **Packet length sequences(SPL)**: The lengths of the packets are commonly categorized into different bins so that the processing is efficient. For example, a packet with a size between 0 and 150 bytes will be put in the first bin, while a packet between 150 and 300 bytes will fall in the second bin
  - **Time Sequence of Packets (SPT)**: We can use the identical reasoning as in SPL, however, in place of the packet length, we will utilize the interarrival packet durations.
  - **Byte Distribution(BD)**: We use an array with a length of 256, enabling us to easily maintain track of each byte that appears in the packet payload.
- **TLS Metadata**: All data components are gathered from TLS handshake packets to combat TLS malware operations in encrypted flows. These packets include the serverhello packet, the clienthello packet, client key exchange, and the change cipher suite.

The clienthello message is in charge of compiling a list of cipher suites and a list of supported TLS extensions. On the other hand, the serverhello message will compile the cipher suite and TLS extension chosen before establishing the connection. The certificate message is used to retrieve the server's certificates. By analyzing the client key exchange message, one may determine the size of the client's public key. TLS sessions are analyzed to gather sequences of packet lengths, sequences of record lengths, and timings. The selected features used in the experiments are listed in Table 3. We have proposed new features of TLS handshake presented in Table 4. In addation, we extracted new statistical features that are presented in Table 5

3.4 Ensemble Learning and Feature Selection

To improve accuracy, we use a stacking mechanism as shown in Fig. 1. The stacking [20] uses two layers. The first layer uses trained base predictions, which are fed as input to meta-layer. The meta-layer performs the robust classification. As part of the stacking ensemble approach, the first layer comprises of four trained algorithms. A meta-classifier is used for the second layer.

This technique uses an ensemble feature selection(FS) technique that averages the feature importance list generated by four independent tree-based ML models. The tree-based ML algorithms compute the importance of each feature based on each tree, the average of the output of tree is used for FS. Each feature's importance is

Table 3 Selected features

Feature name	Number of features	Data type
SourcePort	1	Boolean
DestinationPort	1	Boolean
Number of incoming bytes	1	Integer
Number of outgoing bytes	1	Integer
Number of incoming packets	1	Integer
Number of incoming packets	1	Integer
Sequences of inter-arrival time	100	Stochastic matrix
Sequences of packet length	100	Stochastic matrix
Byte distribution Standard Deviation	1	Float
Byte distribution Mean	1	Float
Byte Entropy	1	Float
Client CipherSuites	348	Binary vector
Client Extensions	52	Binary vector
Ec_point_format	2	Binary vector
Number of client extension	1	Integer
Supported groups	23	Integer
Client Key length	1	Integer
Number of server alternative name (SAN)	1	Integer
SelfSigned certificate	1	Boolean
Validity	1	Integer
isMalware	1	Boolean

Table 4 New proposed TLS handshaking features

Feature name	Description	Number of features	Data type
Client hello packet version	What version of the hello packet it is	1	Integer
Selected cipher suite	Which cipher suite is selected	1	Integer
Server name	If its ranked in the top alexa 1 million site	1	Boolean
Server hello packet version	Which version of hello packet it belongs	1	Integer
Number of server extension	Number of extension exchange by server	1	Integer
Server extensions	Extensions which server exchange	52	Stochastic matrix

Table 5 New proposed statistical features

Feature name	Description	Number of features	Data type
average_arrival	The average amount of time between when the first 15 packets arrive after the TLS handshake	1	Integer
standard_arrival	After the TLS handshake, the standard deviation of the arrival timings of the first 15 packets is computed	1	Integer
maximum_arrival	Maximum amount of delay between the first 15 packets' arrivals	1	Integer
minimum_arrival	Minimum amount of delay between the first 15 packets' arrivals	1	Integer
average_byte	Following the TLS handshake, the average number of bytes contained in the first 15 packets	1	Integer
standard_byte	After the TLS handshake, the number of bytes in the first 15 packets was used to calculate the Standard Deviation	1	Integer
maximum_byte	Maximum amount of bytes between the first 15 packets' arrivals	1	Integer
minimum_byte	Maximum amount of bytes between the first 15 packets' arrivals	1	Integer

summed up to 1.0. To select features from the importance lists, we sort them from most important to least important and then pick them till the sum of their importance equals 0.9. Other features with a sum of less than 0.1 are neglected, which helps us reduce the computation costs.

3.5 Machine Learning (ML) Algorithms

To evaluate the performance of TLS handshaking features and statistical features, we applied various ML classifiers such as K-Nearest Neighbors, Logistic Regression, Bayes Classifier, Decision Tree, Random Forest, XGBost, Gaussian Naive Bayes Classifier, Bernoulli Naive Bayes, and Ensemble Learning Model (Stacking) to train the proposed model. The main intention of comparing various classifiers is to choose the best classifier suitable for our feature set. From the experimental results, we observed that RF and Ensemble Learning models outperformed other classifiers.

4 Experimentation and Results

4.1 Classifying Encrypted Traffic

We used distinct subsets of data features to train different ML models for the initial binary-class classification findings. In experiment 1, we extracted features like Flow MetaData (Meta), packet length sequences, and inter-arrival periods (SPLT) and applied various classifiers. In experiment 2, we extracted features only from TLS data and applied various classifiers. In experiment 3, the classifier was trained using all the data features as well as an extra feature, whether or not the server certificate was self-signed (SS). In experiment 4, the classifier was trained using only the Source Port and Destination Port from the metadata (Meta) features, leaving out the distribution features entirely and taking all the TLS features while leaving out the client key length.

4.1.1 Malware Versus Benign

All malicious TLS flows extracted from the pcap file were cross-validated. Following the filtering step, the studies yielded 15731 malicious and 5673 benign flows. Tables 6 and 7 display the result of a tenfold cross-validation for the state's problem. We observed that using all of the data perspectives produced better results.

We removed the SPLT features and added 65 new TLS handshake and statistical features, which increased the accuracy of the random forest classifier. Table 6 shows that every dataset outperforms the dataset that does not include new features in terms of random forest classifier accuracy.

Finally, we trained the proposed stacking approach with all features, we observed it produces less accuracy than random forest. Table 9 shows the most relevant features based on the weights employed by the stacking model, moreover, we trained the suggested stacking approach with only 30 features, yielding an accuracy of 99.74% which can be seen in Table 8

Table 6 Classification results without addition of new features

Experiment	Features	Classifier	Acc. %	Pre. %	Recall %	F1-score %
Experiment1	META+SPLT+TLS+SS	Gau.NB	96.78	96.49	95.26	95.85
		Ber.NB	97.14	95.50	95.22	96.28
		KNN	97.51	96.50	97.24	96.86
		LR	92.73	91.92	89.23	90.45
		DT	99.61	99.63	99.37	99.50
		RF	99.81	99.77	99.75	99.76
Experiment2	TLS	Gau.NB	96.80	96.84	94.96	95.85
		Ber.NB	97.19	97.60	95.22	96.33
		KNN	98.28	97.02	98.72	97.83
		LR	97.25	97.36	95.60	96.43
		DT	98.96	98.23	99.17	98.69
		RF	99.00	98.31	99.19	98.74
Experiment3	META+SPLT+SS	Gau.NB	91.06	87.70	91.13	89.14
		Ber.NB	89.18	86.61	85.51	86.04
		KNN	93.77	92.04	92.12	92.08
		LR	94.89	94.75	92.07	93.30
		DT	99.36	99.23	99.16	99.19
		RF	99.47	99.21	99.44	99.32
Experiment4	Reduced features	Gau.NB	97.06	97.28	95.21	96.18
		Ber.NB	97.19	97.60	95.22	96.33
		KNN	96.41	95.36	95.53	95.45
		LR	96.78	96.73	95.02	95.83
		DT	98.04	97.36	97.68	97.52
		RF	98.12	97.57	97.66	97.61

4.1.2 Malware Family

We utilized an ensemble model to determine how effectively a trained classifier can detect TLS traffic generated by various malware families. The results of this model and a random forest are in Table 10. The classification report generated by the stacking model is also listed in Table 10. After feature selection, we trained our ensemble model with 48 features. Top features of the feature-selected ensemble model are listed in Table 11.

According to this blog [8], you can only achieve maximum accuracy if you use all the available features. However, this is no longer true. According to our findings, we do not need all the features (META+SPLT+TLS+SS). META+SPLT+TLS+SS has 640 features and uses different classifiers on the dataset. Random forest achieves the highest accuracy of 99.81%, as shown in Table 6. Moreover, we perform the experimentation on TLS+new (431+65) features. We observed random forest achieves the highest accuracy of 99.85% as shown in Table 7. In the case of binary classification for encrypted traffic, the accuracy and size are significantly improved.

Table 7 Classification results after addition of new features

Experiment	Features	Classifier	Acc. %	Pre. %	Recall %	F1-score %
Experiment1	META+NewFeatures+TLS+SS	Gau.NB	39.67	65.40	58.75	38.64
		Ber.NB	97.21	97.62	95.26	96.36
		KNN	97.43	96.51	97.00	96.75
		LR	91.26	89.91	87.39	88.53
		DT	99.83	99.84	99.74	99.79
		RF	99.85	99.81	99.76	99.79
Experiment2	META+NewFeatures+SS	Gau.NB	39.67	65.40	58.75	38.46
		Ber.NB	97.21	97.62	95.26	96.36
		KNN	97.43	96.51	97.00	96.75
		LR	91.26	89.91	87.39	88.53
		DT	99.69	99.57	99.64	99.61
		RF	99.83	99.86	99.82	99.84
Experiment3	TLS+NewFeatures	Gau.NB	39.67	65.40	58.75	38.46
		Ber.NB	97.21	97.62	95.26	96.36
		KNN	97.43	96.51	97.00	96.75
		LR	91.26	90.59	88.11	89.24
		DT	99.83	99.57	99.64	99.61
		RF	99.85	99.81	99.76	99.79
Experiment4	Reduced+NewFeatures	Gau.NB	39.67	65.40	58.75	38.46
		Ber.NB	97.21	97.62	95.26	96.36
		KNN	97.55	96.59	97.25	96.91
		LR	90.30	89.35	85.29	87.02
		DT	99.79	99.81	99.66	99.74
		RF	99.80	97.83	99.83	99.81

Table 8 Classification results using ensemble model

Classifier	Acc. %	Pre. %	Recall %	F1-score %
Stacking	99.7485	99.7318	99.6188	99.6751
FS stacking	99.7485	99.7807	99.5702	99.6748

In this study [18], the accuracy of binary classification was 92%, however, the random forest achieved 99.85% accuracy and the stacking model achieved 99.74% accuracy. They did not perform a multiclass classification of malware families, as we did in our research.

The accuracy of the neural networks used in this study was 89.25% [26]. In comparison to our previous research, the accuracy of our stacking-based model increased to 99.74%, while random forest produced 99.85%. Similarly, they have not addressed malware family multiclassification, whereas our work does.

They discovered several features in this [22] study that differ from ours based on their highest accuracy of 94.75%. The proposed method was able to achieve maximum accuracy of 99.85%, which is significantly higher than the state of the art.

Table 9 The most important 20 features classification model by stacking model for binary classification

Importance	Feature description
0.4436	num_of_client_exts
0.0357	Client Extension:user_mapping”
0.0323	nb_san
0.0317	Client Extension:status_request
0.0289	Client Extension: client_authz
0.0279	Validity
0.0264	Client Extension: server_authz
0.0258	byte_dist_mn
0.0245	Server Extension:trusted_ca_keys
0.0224	Bytes_in
0.0149	Server Extension:client_certificate_url
0.0137	’num_of_server_exts
0.0136	avg_byte
0.0133	Number of certificate
0.0124	Packet length transition probability matrix: [200,300)
0.0117	Client Extension:truncated_hmac
0.0099	min_byte
0.0099	CipherSuite:TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
0.0098	CipherSuite:TLS_DH_DSS_WITH_DES_CBC_SHA
0.0095	Server Extension:max_fragment_length

Table 10 Classification results for multiclass classification

Classifier	Acc. %	Pre. %	Recall %	F1-score %
RF	97.4332	94.1012	86.7509	88.9634
Stacking	97.4506	89.4247	90.1578	89.7607
FS Stacking	97.4105	89.5859	88.5233	88.9892

5 Conclusion

This paper proposes an efficient Ensemble ML algorithm for detecting and classifying malware in TLS-encrypted data. The Random Forest Classifier and Ensemble ML Classifier (stacking) have an advantage over other approaches when distinguishing between malicious and benign data. It achieves 99.85% accuracy for the random forest classifier and 97.43% accuracy for the ensemble classifier on multiclass classification for identifying malware families. The Ensemble model, on the other hand,

Table 11 The most important 20 features classification model by stacking model for multiclass classification

Importance	Feature description
0.0819	ciphersuites:TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA
0.0544	Validity
0.0535	Dst_Port
0.052	num_of_client_exts
0.0503	Client Cipher Suite:TLS_DH_RSA_WITH_DES_CBC_SHA
0.0252	Client_keylength
0.0239	Client Extension:trusted_ca_keys
0.0222	Packet length transition probability matrix: [500,600)
0.02	Bytes_out
0.0186	Bytes_in
0.0184	Packet length transition probability matrix: [1100,1200)
0.0183	Server_version
0.0146	Client Extension:truncated_hmac
0.0133	supported group 0
0.0132	num_of_server_exts
0.0127	client Extension:user_mapping
0.0122	min_byte
0.0116	Client Extension:client_authz
0.0116	Cipher Suite:TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
0.0115	std_byte

achieves 99.74% accuracy for binary classification and 97.45% accuracy for multi-class classification while being computationally more efficient than Random Forest. In future, we propose to develop a malware detection mechanism based on deep learning techniques.

References

1. Aitken P, Claise B, Trammell B (2013) Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information. RFC 7011. https://doi.org/10.17487/RFC7011, https://www.rfc-editor.org/info/rfc7011
2. Allen C, Dierks T (1999) The TLS protocol version 1.0. RFC 2246. https://doi.org/10.17487/RFC2246, https://www.rfc-editor.org/info/rfc2246
3. Anderson B, McGrew D (2016) Identifying encrypted malware traffic with contextual flow data. In: Proceedings of the 2016 ACM workshop on artificial intelligence and security, association for computing machinery, New York, NY, USA, AISec '16, pp 35–46. https://doi.org/10.1145/2996758.2996768
4. Anderson B, McGrew D (2017) Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity. In: Proceedings of the 23rd ACM SIGKDD

international conference on knowledge discovery and data mining, association for computing machinery, New York, NY, USA, KDD '17, pp 1723–1732. https://doi.org/10.1145/3097983.3098163

5. Anderson B, Paul S, McGrew DA (2016) Deciphering malware's use of TLS (without decryption). http://arxiv.org/abs/1607.01639
6. Claise B (2004) Cisco systems NetFlow services export version 9. RFC 3954. https://doi.org/10.17487/RFC3954, https://www.rfc-editor.org/info/rfc3954
7. De Lucia M, Cotton C (2018) Identifying and detecting applications within TLS traffic. p 31. https://doi.org/10.1117/12.2305256
8. Decryption BADEMTW (2017) blog. https://blogs.cisco.com/security/detecting-encryptedmalware-traffic-without-decryption
9. Dietrich C, Rossow C, Pohlmann N (2013) CoCoSpot: clustering and recognizing botnet command and control channels using traffic analysis. Comput Netw 57:475–486. https://doi.org/10.1016/j.comnet.2012.06.019
10. Garcia S, Uhlir V (2013) Malware capture facility project. Cvut
11. Hajirahimova M, Aliguliyev R (2019) Classification ensemble based anomaly detection in network traffic 6:12–23. https://doi.org/10.18488/journal.76.2019.61.12.23
12. Houser R, Li Z, Cotton C, Wang H (2019) An investigation on information leakage of DNS over TLS. In: Proceedings of the 15th international conference on emerging networking experiments and technologies. Association for Computing Machinery, New York, NY, USA, CoNEXT '19, pp 123–137. https://doi.org/10.1145/3359989.3365429
13. de Lucia MJ, Cotton C (2019) Detection of encrypted malicious network traffic using machine learning. In: MILCOM 2019-2019 IEEE military communications conference (MILCOM), pp 1–6. https://doi.org/10.1109/MILCOM47813.2019.9020856
14. McGrew D, Anderson B (2016a) Enhanced telemetry for encrypted threat analytics. In: 2016 IEEE 24th international conference on network protocols (ICNP), pp 1–6. https://doi.org/10.1109/ICNP.2016.7785325
15. McGrew D, Joy BA (2016) Cisco joy. https://github.com/cisco/joy
16. McGrew DA, Anderson B (2016) Enhanced telemetry for encrypted threat analytics. In: 2016 IEEE 24th international conference on network protocols (ICNP), pp 1–6
17. Moore AW, Zuev D (2005) Internet traffic classification using Bayesian analysis techniques. In: Proceedings of the 2005 ACM SIGMETRICS international conference on measurement and modeling of computer systems. Association for Computing Machinery, New York, NY, USA, SIGMETRICS '05, pp 50–60. https://doi.org/10.1145/1064212.1064220
18. Pai KC, Mitra S, Chari S M (2020) Novel TLS signature extraction for malware detection. In: 2020 IEEE international conference on electronics, computing and communication technologies (CONECCT), pp 1–3. https://doi.org/10.1109/CONECCT50063.2020.9198590
19. Panchenko A, Lanze F, Pennekamp J, Engel T, Zinnen A, Henze M, Wehrle K (2016) Website fingerprinting at internet scale. In: NDSS
20. Pavlyshenko B (2018) Using stacking approaches for machine learning models. In: 2018 IEEE second international conference on data stream mining processing (DSMP), pp 255–258. https://doi.org/10.1109/DSMP.2018.8478522
21. Rescorla E, Modadugu N (2012) Datagram transport layer security version 1.2. RFC 6347. https://doi.org/10.17487/RFC6347, https://www.rfc-editor.org/info/rfc6347
22. Wang Z, Fok KW, Thing VL (2022) Machine learning for encrypted malicious traffic detection: approaches, datasets and comparative study. Comput Secur 113(102):542. https://doi.org/10.1016/j.cose.2021.102542
23. Wright CV, Monrose F, Masson GM (2006) On inferring application protocol behaviors in encrypted network traffic. J Mach Learn Res 7:2745–2769. https://doi.org/10.5555/1248547.1248647
24. Wurzinger P, Bilge L, Holz T, Goebel J, Kruegel C, Kirda E (2009) Automatically generating models for botnet detection. In: Backes M, Ning P (eds) Computer security-ESORICS 2009. Springer, Berlin, Heidelberg, pp 232–249

25. Zander S, Nguyen T, Armitage G (2005) Automated traffic classification and application identification using machine learning. In: Proceedings of the The IEEE conference on local computer networks 30th anniversary. IEEE Computer Society, USA, LCN '05, pp 250–257. https://doi.org/10.1109/LCN.2005.35
26. Zhou Z, Bin H, Li J, Yin Y, Chen X, Ma J, Yao L (2016) Malicious encrypted traffic features extraction model based on unsupervised feature adaptive learning. J Comput Virol Hacking Tech, 1–6. https://doi.org/10.1007/s11416-022-00429-y

Public Data Auditing Scheme Using RSA and Blockchain for Cloud Storage

A. Vamshi, R. Eswari, and Shivam Dubey

Abstract With the widespread cloud storage devices, most users store their data over cloud storage. A dishonest user might raise a false claim about the integrity of their data to get compensation. On the other hand, a corrupt Cloud Service Provider (CSP) might hide the fact from the user that their data has been compromised. So, ensuring the integrity of the data is of utmost importance in such cases. The integrity of user data is ensured through Third Party Auditor (TPA), who is supposed to be trusted and possess the computation resources to perform the auditing. Although this TPA reduces the burden of verifying the data, there's still a risk of TPA, who may not be honest. The TPA might collude with the CSP or not perform its task correctly. This paper proposes a Public Data Auditing Scheme using RSA and Blockchain (PDASRSAB) for cloud storage to bring security and transparency into the auditing process, which increases confidence in cloud storage, along with the experimental analysis for the proposed scheme.

Keywords Blockchain · RSA · Data auditing · Cloud storage

1 Introduction

Cloud computing is widely gaining importance from users and researchers. It has been one of the fastest adoptions into mainstream life compared to other technologies in this domain. Cloud computing is an efficient way to provide/manage information and communication resources to remote users. The importance of cloud computing in today's world stems from this process of solving several problems faced by the average user and large organizations. With such an exponential rate of development of cloud computing, we can now access its facilities much more quickly and efficiently. Cloud computing has different applications in today's world, like Big Data Analytics, SaaS (i.e., Software as a Service), SD-WAN (i.e., Software-Defined Wide Area Networking), and so on. However, there are a lot of challenges, like lack of

A. Vamshi (✉) · R. Eswari · S. Dubey
Department of Computer Applications, National Institute of Technology, Tiruchirappalli, India
e-mail: vamshiadouth47@gmail.com

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_19

privacy, quality, and safety protocols in existing methods. Cloud Service Providers (CSPs) around the world have faced many issues, such as the LinkedIn data breach in 2021, Amazon cloud's storage outage in 2021, Facebook data breach in 2019, Twitter data breach in 2018, Tencent cloud user's silent error in 2018, and Intuit's power failure in 2010. So, it becomes essential in such a scenario to ensure the integrity of the data to increase the confidence of users and organizations in cloud services.

Now the question arises, "How can we verify the integrity of the data?". Neither the user, i.e., the Data owner (DO), nor the Cloud Service Provider (CSP) can blindly trust each other. A malicious CSP might hide an incident of data failing in integrity verification, or a malicious DO might claim false data integrity. Therefore, in such a case, we need a Third Party Auditor (TPA) who can verify the integrity of the data. There are many existing Data Auditing schemes where a TPA with sufficient computing power checks the data integrity when the DO suspects it or can't verify it on its own. The user or DO sends a request for the audit to a Third Party Auditor or TPA. The TPA then generates a challenge and sends it to the CSP (it asks the CSP for proof to verify the data integrity). In response, the CSP generates the proof by performing cryptographic operations on the data blocks and then provides the proof to the TPA. Once TPA receives proof generated by CSP, it verifies the data integrity and sends the Audit report to the DO.

The problem with such Auditing schemes is that the DO has to trust the TPA completely. It might be possible that the TPA might not be performing its task correctly and efficiently or that the TPA could collude with the malicious CSP to conceal that DO's data. Another possibility is that the TPA's system is hacked by some hacker, which again poses a considerable security risk. We can replace the TPA with a blockchain network to avoid the above risks. Introducing blockchain into the auditing scheme can eliminate the possibility of a malicious TPA. Since tampering with a blockchain is not a feasible task and also due to its transparency, a blockchain network is an ideal candidate for the job of a trusted TPA. The DO can send the auditing request to the Public Auditor (i.e., blockchain network), which can verify the integrity of the data by generating a challenge for the CSP. Based on the proof given by the CSP, it can tell whether the data's integrity is compromised.

1.1 Contribution

We propose a Public Data Auditing Scheme for Cloud Storage in this paper. The following are the contributions made in this paper:

1. We propose a Public Data Auditing Scheme using RSA and Blockchain for Cloud Storage, which is not dependent on a traditional Third Party Auditor (TPA). This prevents the auditing process from malicious auditors and makes it more secure.
2. We used an RSA-based key generation center (KGC) to sign the data and store it in the cloud.

3. In experimental analysis, we test the proposed scheme for various file sizes, which shows that the proposed scheme is suitable for cloud-based devices.

2 Related Work

Juels et al. [1] first proposed PoRs—Proofs of Retrievability for large files, which checks the correctness of data stored [1]. Later, Ateniese et al. [2] proposed PDP—Provable Data Possession at Untrusted Stores. They used RSA-based homomorphic linear authenticators for data-auditing. In 2011, Wang et al. [3] came up with Privacy-Preserving Public Auditing for Secure Cloud Storage. In this approach, TPA verifies the correctness of the cloud data. Zhu et al. [19] proposed a model of cooperative provable data possession for verifying data integrity in multi-cloud storage. Yang et al. (2013) [4] proposed a dynamic auditing protocol in which a TPA verifies the correctness of the cloud data by using verifiable homomorphic tags and data fragmentation [5]. Later, Yang et al. proposed a model that provides auditing for data over cloud storage which supports traceability and privacy. Its drawback is the burden of computational burden on the Group Manager (GM), and dependency on the TPA [6]. Debiao He et al. proposed a Public Auditing Scheme which utilizes certificateless cryptography. The scheme has four entities, a third-party auditor, a KGC, a user, and a cloud server and uses certificateless public auditing (CLPA) [7]. Singh et al. proposed a Secure Data Dynamics and Public Auditing Scheme for Cloud Storage. This model consists of three entities: D.O, CS, and TPA. It uses AES256, RSA-15360, and SHA512 algorithms. The drawback is its dependency on TPA [8]. Although the various models and schemes proposed by different researchers [9–17] have some advantages, one of the major drawbacks of these schemes is that they make the assumption that CS or TPA has no motivations whatsoever, to collude in the auditing process.

3 Preliminaries

3.1 RSA Algorithm

The RSA (Rivest, Shamir, and Adleman) algorithm is an asymmetric cryptography algorithm [18]. Ron Rivest, Adi Shamir, and Leonard Adleman proposed this algorithm in the year 1977. Asymmetric cryptography works with two different keys known as Public Key and Private Key. In the proposed scheme, we use the RSA algorithm to generate the keys and sign the data. As of now, there are no published works that claim to break the system as long as a very large key is used. The algorithms involved in RSA are Public-Private key generation, encryption, and decryption.

Key Generation:

1. Select any two different prime numbers p and q (preferably large prime numbers for stronger encryption).
2. Calculate the RSA modulo N, computed as $N = p * q$.
3. Calculate a derived number Z, such that $Z = (p - 1) * (q - 1)$.
4. Randomly choose an integer e in such a way that the following conditions are met:

$$e < Z. \tag{1}$$

$$\gcd(e, Z) = 1. \tag{2}$$

5. Return the public-private key pair. Here, d is a private key (e, N) and d is a public key.

Encryption:

1. Input the public key (e, N) and message m.
2. Generate the cipher-text c such that: $c = m^e \bmod N$.

Decryption:

1. Input the private key d and cipher-text c.
2. Generate the corresponding message m by computing $m = c^d \bmod N$.

3.2 Merkle Hash Tree (MHT)

Merkle Hash Tree (MHT) is a tree data structure where the leaf nodes contain the data blocks. The non-leaf node contains the combined hash of its child node. The Merkle Hash Tree was patented by Ralph Merkle in 1979. The structure of the MHT is such that it allows a very competent mapping of large data. Even a small change in the data blocks can cause the hash value in the root to change drastically. The hash in the root acts as the fingerprint for the whole data (Fig. 1).

3.3 Blockchain

Blockchain is a distributed database which is append-only by nature and is shared among the different nodes. It was first developed in 2008 by an anonymous person(s) by the name Satoshi Nakamoto. As the name suggests, it's a chain of blocks that can be defined as a kind of Digital Ledger Technology (DLT) that consists of a growing list of records called blocks. These blocks contain a hash of transaction data, previous block, timestamp, etc. Blockchain works on a P2P (peer-to-peer) network, in which all the nodes follow a consensus protocol for verifying/adding new transactions.

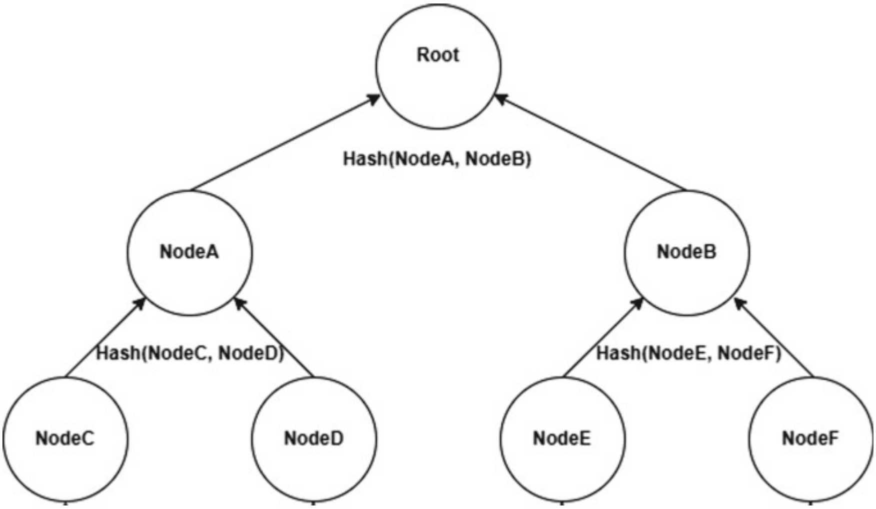


Fig. 1 Overview of MHT

The transactions cannot be altered in a blockchain once recorded. Blockchain was recently popularized because of its use in bitcoin (Cryptocurrency). The following are the 3 fundamental properties of a secure blockchain system:

1. X-chain consistency: During the mining process at a given time, the blockchains of two miners can only differ in the last X blocks.
2. (l, X)-chain quality: The probability of any X consecutive blocks over a blockchain are produced by an opposing miner having hash-rate $<51\%$ of entire network's mining hash-rate which is negligible.
3. Chain growth: The height of a blockchain steadily increases w.r.t. long term as well as short term.

4 Proposed Scheme

4.1 System Model

In the proposed scheme, the user (Data owner) divides the file in blocks. These blocks are then encrypted by using the public key provided by the RSA-based Key Generation Center. Once the file is divided into blocks, then for each block, there is a hashtag generated by performing a hash operation on each one of the data blocks. Then the user broadcasts these hashtags on the blockchain network. After that, the user uploads the encrypted file on the cloud server. Figure 2 shows the system model.

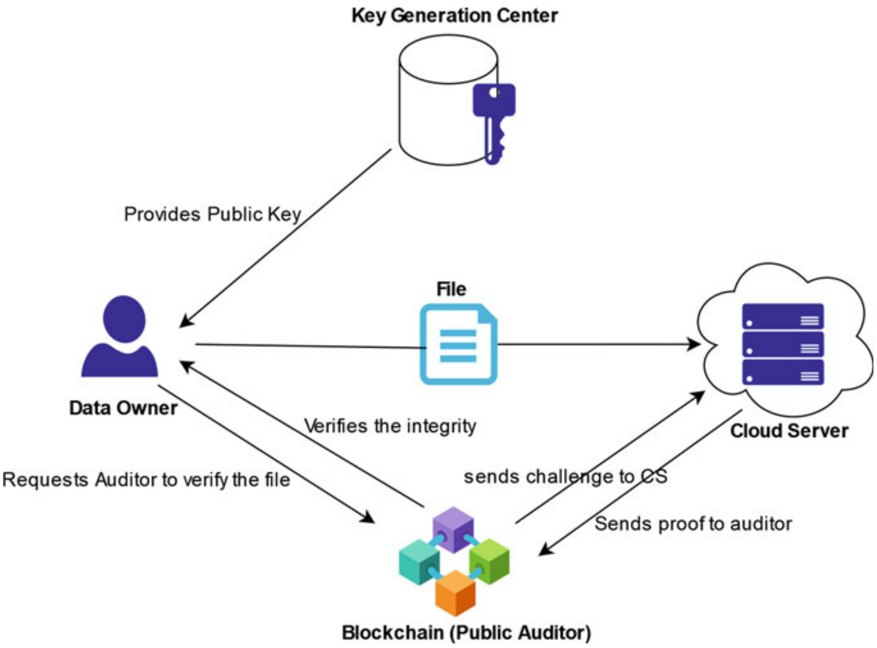


Fig. 2 System model

Whenever a Data Owner (DO) needs to verify the integrity of its file, it sends an auditing request to the public auditor (blockchain). The public auditor then generates a challenge for the CSP. On receiving the challenge, the cloud server calculates the MHT root for the blocks of the file stored. Then the cloud server sends the MHT root the proof to the public auditor. When the public auditor receives the proof, it then compares this proof with the MHT root calculated by itself. If both match, it means the file's integrity is not compromised.

4.2 PDASRSAB-Auditing Framework

The following algorithms are used in the system model:

1. GenerateKey(): Used by the KGC to generate public-private keys for encryption-decryption.
2. Encrypt(): Used by the Data Owner for encryption of file using public-key provided by KGC.
3. Decrypt(): Used by the CSP to decryption of file using the private key provided by KGC.

Table 1 Notations

Description	Symbols
Key generation center	KGC
Data owner	DO
Cloud service provider	CSP
Derived number	Z
Public key	(e, N)
ith data block	Bi
Hash of ith block	Hi
ith encrypted block	E_{Bi}
Message	m
Cipher text	c
Root of Merkle Hash tree	MHT_{root}

4. DivideAndGenHashTags(): Used by the Data Owner and CSP to divide the encrypted files into blocks and then generate hashtags for each corresponding block.
5. MHT_{root}(): Used by the Data Auditor (Blockchain) to calculate the MHT root for the blocks of the file.
6. GenerateProof(): Used by the CSP proof generation (MHT root) from the blocks of the files.
7. VerifyProof(): Used by the Data Auditor (Blockchain) to compare the MHT root that it has calculated with the one provided by the CSP as proof.

4.3 Notations

Table 1 shows the frequently used notations in the proposed scheme.

4.4 PDASRSAB Construction

Setup: On the input of two parameters p and q (both prime numbers, preferably large prime numbers for stronger encryption), the algorithm returns the public-private key pair $(e, N), d$. Here (e, N) is a public-key and d is a private key. First, the algorithm calculates the RSA modulo N, which is computed as $N = p * q$. Then a derived number Z is calculated as $Z = (p - 1) * (q - 1)$. Once Z is computed, the algorithm randomly chooses an integer e such that the following conditions are met:

1. $e < Z$.
2. gcd $(e, Z) = 1$, i.e., both should be co-prime.

Now, for the corresponding value of e, another integer d is calculated in such a way that $e * d = 1 \bmod Z$.

Sign Generation On the input of public key, i.e., (e, N) and message, i.e., m, the Data Owner generates the cipher-text c such that $c = m^e \bmod N$.

Divide and Generate Hashtag On input of the cipher-text c (obtained after encryption), the algorithm first divides it into n blocks $(E_{B1}, E_{B2}, E_{B3}, ...E_{Bi})$. For each block, a corresponding hashtag value is calculated as $(H1, H2, H3, ..Hi)$ where Hi = Hash(E_{Bi}). With the help of these hashtags, the MHT_{root} is calculated, which later helps in verifying the integrity of our data.

Calculate MHT root On input of the list of hashtags $(H1, H2, H3, ...Hi, ...)$ for the corresponding blocks $(E_{B1}, E_{B2}, E_{B3}, ...E_{Bi}, ...)$ of message m broadcasted by the user over the blockchain network, this algorithm computes the $\mathrm{MHT}_{\mathrm{root}}$ as

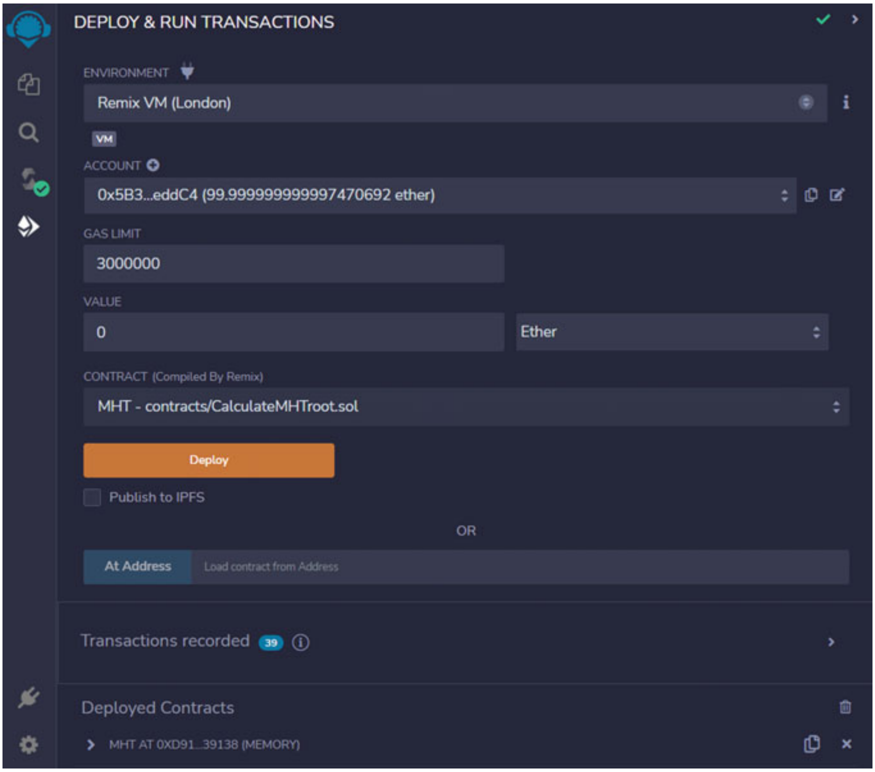
$\mathrm{MHT}_{\mathrm{root}} = Hash(NodeA, NodeB)$
$NodeA = Hash(NodeC, NodeD)$
$NodeB = Hash(NodeE, NodeF)$... till the leaf node is reached.

Here, each non-leaf node is a hash of its two child nodes. So in this way, if there's a change even in a single block Hashtag Hi of the message m, the MHT_{root} will be completely different, hence making it very easy to verify the integrity of the message by simply comparing the MHT_{root}.

Proof Generation On input of the cipher text c received from the DO, CSP divides it into n blocks $(E_{B1}, E_{B2}, E_{B3}, ...E_{Bi}, ...)$. For each block, the value of a corresponding hashtag is calculated as $(H1, H2, H3,$ âĿ¦$Hi,$ âĿ¦$)$ where $Hi = Hash(EBi)$. The CSP then receives a challenge from the Public Auditor (Blockchain). In response to the challenge, CSP computes the MHT_{root} as

$MHT_{root} = Hash(NodeA, NodeB)$
$NodeA = Hash(NodeC, NodeD)$
$NodeB = Hash(NodeE, NodeF)$... till the leaf node is reached. Once the MHT_{root} is computed, the CSP sends it as the proof to the Public Auditor.

Verify On input of the proof generated by the CSP (which is nothing but the MHT_{root} computed by the CSP from the corresponding Hashtags of the message m), the algorithm compares the proof and the MHT_{root} to verify the integrity of the data. If the value of the proof and MHT_{root} are found to be equal, then the integrity of the message m is conserved; otherwise, it is compromised (Fig. 3).
if $ProofCSP = MHT_{root} => Integrity$ conserved
if $ProofCSP \neq MHT_{root} => Integrity$ compromised


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Remix VM (London)

ACCOUNT: 0x5B3...eddC4 (99.999999999997470692 ether)

GAS LIMIT: 3000000

VALUE: 0 Ether

CONTRACT (Compiled By Remix): MHT - contracts/CalculateMHTroot.sol

Transactions recorded 39

Deployed Contracts

MHT AT 0XD91...39138 (MEMORY)

Fig. 3 Smarts contracts deployment

5 Experimental Analysis

5.1 Cryptographic Setup

For the Cryptographic setup, we have used the RSA algorithm. This part was run on AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx 2.30 GHz with 8.00 GB RAM, using C++ language and STL library.

5.2 Blockchain Setup

For the blockchain setup, we have used Remix—Ethereum IDE, as shown in Fig. 1, which allows the development, deployment, and administration of smart contracts for Ethereum-like blockchains. The protocol for verifying the integrity of the message is implemented on the blockchain by using smart contracts. A Smart contract is basically a program that runs over an Ethereum blockchain. This program resides at a particular address on the blockchain. For this setup, smart contracts are written in solidity 0.8.3. The smart contract is implemented on Environment: Remix VM (London), used to connect to the sandbox blockchain in the browser. The Remix VM is its own "blockchain". On each and every reload a new blockchain is started, and the old one is deleted. Figure 1 shows the deployment of the smart contract. The details of the smart contract deployed locally are as follows:

Account: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
Gas Limit: 3000000
Memory: 0xd9145CCE52D386f254917e481eB44e9943F39138
Unit: Ether

The following are the transaction and execution costs of various tasks/operations performed by the smart contract:

1. Broadcasting hashtag corresponding to each data blocks on the blockchain network.
   transaction cost: 48834 gas
   execution cost: 48834 gas
2. Calculating MHT root.
   transaction cost: 157223 gas
   execution cost: 157223 gas
3. Checking the integrity of data.
   transaction cost: 0 gas
   execution cost: 23624 gas.

5.3 Experimental Results

We evaluate the performance of our scheme by measuring the time taken in various operations like Sign generation, proof generation, and verification for different file sizes. The results are given in Table 2. We have taken an average of readings with the same file size.

Table 2 Experimental results

S. No.	File size (bytes)	Sign Gen (s)	ProofGen (s)	Proof verify (s)
1	2	0.5118	0.0031	0.0108
2	4	0.5221	0.0047	0.0129
3	8	0.5332	0.0073	0.0147
4	16	0.569	0.0094	0.0203

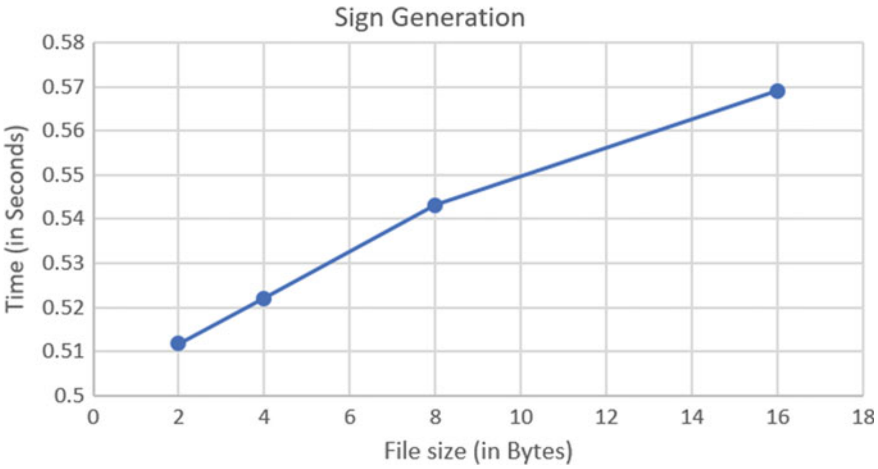


Fig. 4 Sign generation

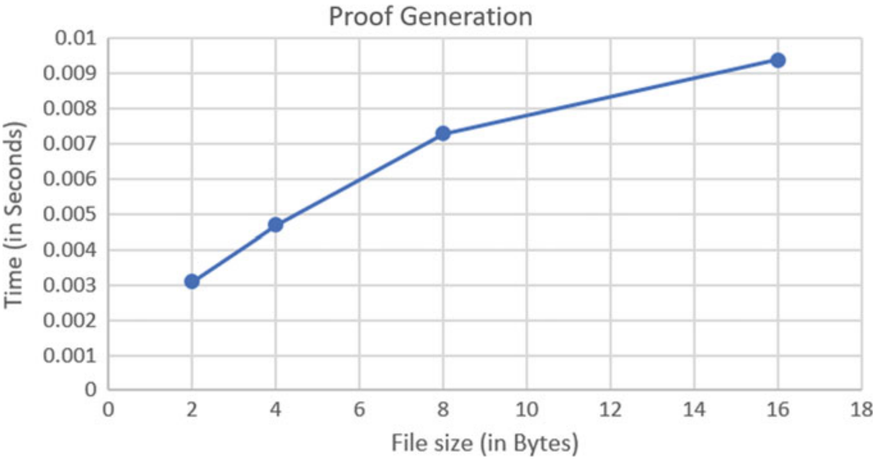


Fig. 5 Proof generation

The graphs in Figs. 4, 5, and 6 show the relation between the size of the file and the time taken in Sign Generation, Proof Generation, and Proof Verification, respectively.

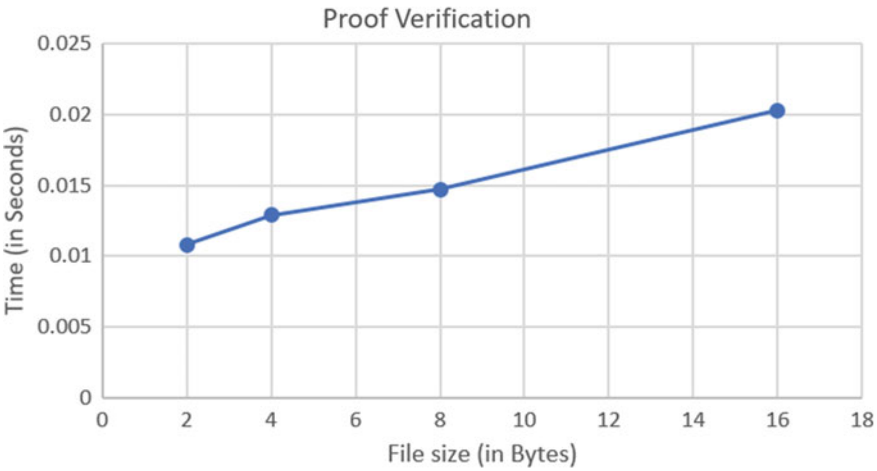


Fig. 6 Proof verification

6 Conclusion

We discussed the drawbacks and risks involved in the available public data auditing schemes that rely on a TPA. To overcome the drawbacks of TPA-based auditing schemes, we proposed a Public Data Auditing Scheme using the RSA and Blockchain (PDASRSAB)-Auditing framework, which uses the RSA algorithm for key generations, and Blockchain as a public auditor. We explained the proposed model in detail, followed by the experimental analysis.

References

1. Juels A, Kaliski Jr BS (2007) PORs: proofs of retrievability for large files. In: Proceedings of the 14th ACM conference on Computer and communications security, pp 584–597
2. Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, Song D (2007) Provable data possession at untrusted stores. In: Proceedings of the 14th ACM conference on computer and communications security, pp 598–609
3. Wang Q, Wang C, Ren K, Lou W, Li J (2010) Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans Parallel Distrib Syst 22(5):847–859
4. Begam OR, Manjula T, Manohar TB, Susrutha B (2012) Cooperative schedule data possession for integrity verification in multi-cloud storage. Int J Modern Eng Res (IJMER) 3:2726–2741
5. Yang K, Jia X (2012) An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE Trans Parallel Distrib Syst 24(9):1717–1726
6. Yang G, Jia Y, Shen W, Qianqian S, Zhangjie F, Hao R (2016) Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability. J Syst Softw 113:130–139
7. He D, Zeadally S, Libing W (2015) Certificateless public auditing scheme for cloud-assisted wireless body area networks. IEEE Syst J 12(1):64–73

8. Singh P, Saroj SK (2020) A secure data dynamics and public auditing scheme for cloud storage. In: 2020 6th international conference on advanced computing and communication systems (ICACCS). IEEE, pp 695–700
9. Deng L, Yang Y, Gao R, Chen Y (2018) Certificateless short signature scheme from pairing in the standard model. Int J Commun Syst 31(17):e3796
10. Deng L, Yang Y, Chen Y (2019) Certificateless short aggregate signature scheme for mobile devices. IEEE Access 7:87162–87168
11. Islam SKH (2014) A provably secure id-based mutual authentication and key agreement scheme for mobile multi-server environment without ESL attack. Wirel Pers Commun 79(3):1975–1991
12. Scott M, Costigan N, Abdulwahab W (2006) Implementing cryptographic pairings on smartcards. In: International workshop on cryptographic hardware and embedded systems. Springer, pp 134–147
13. Karati A, Islam SH, Biswas GP (2018) A pairing-free and provably secure certificateless signature scheme. Inf Sci 450:378–391
14. He D, Huang B, Chen J (2013) New certificateless short signature scheme. IET Inf Secur 7(2):113–117
15. Gao G, Fei H, Qin Z (2020) An efficient certificateless public auditing scheme in cloud storage. Concurr Comput: Pract Exp 32(24):e5924
16. Li J, Yan H, Zhang Y (2018) Certificateless public integrity checking of group shared data on cloud storage. IEEE Trans Serv Comput 14(1):71–81
17. Zhou R, He M, Chen Z (2021) Certificateless public auditing scheme with data privacy preserving for cloud storage. In: 2021 IEEE 6th international conference on cloud computing and big data analytics (ICCCBDA). IEEE, pp 675–682
18. Rivest RL, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21(2):120–126
19. Zhu Y et al (2012) Cooperative provable data possession for integrity verification in multicloud storage. IEEE transactions on parallel and distributed systems 23(12): 2231–2244

Neuroimaging Data Analysis of an Artificially Intelligent Human Mind

Ajay Singh, Namrata Dhanda, and Rajat Verma

Abstract It was depicted by the specialists, how four decades of study on cognitive architectures has brought about a critical union of different strategies towards a bunch of essential suppositions, they name it, the Standard Model of Mind. The SMM initially made to address a consensual viewpoint of "human-like mind," either from artificial intelligence or cognitive exploration, which, if valid, should be valid for the human mind. We give a prompt examination of this speculation in light of a re-examination of fMRI data that span across many cognitive cycles and intricacy. The SMM was separated to two unmistakable potential decisions that entered either helpful or fundamental assumptions of the SMM using a spread-out approach (Dynamic Causal Showing) to focus on utilitarian relationship between mind locales. The results reveal that the SMM beats various models generally through each dataset, showing that the SMM best tends to the utilitarian prerequisites of cerebrum activity in fMRI data.

Keywords Dynamic causal modelling · Cognitive architectures · fMRI · Effective connectivity

1 Introduction

A basic issue of discussion in the field of cognitive architecture is the connection between the parts of design and their relationship to the mind. A few plans, like SPAUN and LISA, are intended to copy the organic circuits of the mind and depend on counterfeit neurons as building blocks. These frameworks address cognitive modelling at the circuit level, in light of the possibility that capability rises up out

A. Singh · N. Dhanda (✉)
Department of Computer Science and Engineering, Amity University Uttar Pradesh, Lucknow, India
e-mail: ndhanda@lko.amity.edu

R. Verma
Department of Computer Science and Engineering, Pranveer Singh Institute of Technology, Kanpur, India

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_20

of structure. One more class of plans, like Soar or ACT-R, depend on another option, practical methodology, with building blocks that are more unique and undeniable level mental parts like perceptual frameworks and memory that are along these lines planned post-hoc on unambiguous cerebrum districts. Nonetheless, throughout the course of recent many years, mental structures have bit by bit combined around a bunch of shared basic suspicions, temporarily named the "Standard Model of the Mind" (hereafter, the SMM) [1].

1.1 The Standard Model of Mind

A few late explorations have endeavored to connect building structures to functional mind movement. These works have focused on the Standard Model of Mind (SMM), a theoretical clarification of the standards shared by various designs. As indicated by the SMM, cognition creates at the most elevated level from the exchange of five cognitive elements: perception, action, long-term memory, procedural memory, and working memory. These parts might be connected to five enormous scope mind circuits, and a network of coordinated associations can be laid out between them [2]. Figure 1 outwardly portrays the fundamental components used in the model and their structure [3].

The model is communicated as a series of expectations or assumptions about the idea of specific parts and calculations shared by "humanlike minds," which are fundamentally and practically similar to human brains. These suppositions are isolated into four classifications: Structure and Processing; Memory and Data; Learning; Perception and Motor Control [1]. Table 1 features the significant assumptions hidden the customary idea of the SMM.

Despite the fact that the convergence was particularly shown to apply to the three architectures—Soar, ACT-R, and Sigma—the same argument may be made for other comparable methods. The design of intelligent systems in artificial intelligence and robotics, continuous study in cognitive psychology, and headways in neuroscience are only a couple of the elements that have without a doubt added to this union. Because of the variety of causes that have led to this convergence, the authors propose that

Fig. 1 The SMM in graphic form [3]

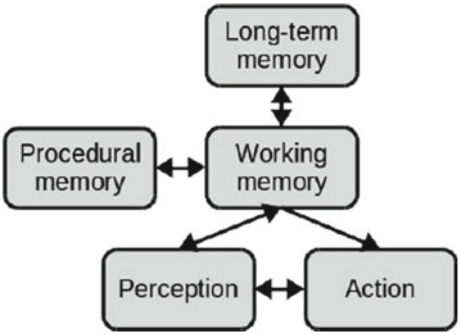


Table 1 Architectural assumptions of the SMM

S. no.	Categories	Assumptions
1	Structure and processing	The main goal of architectural processing is to promote limited rationality rather than optimality. It depends upon task-independent components
		Architectural processing causes a high degree of parallelism. Soar and ACT-R cause asynchronous parallel processing across modules while Sigma causes synchronous parallel processing across modules
		In human cognition, behaviour is governed by sequential action selection that occurs at a rate of roughly 50 ms every cycle
		Complex behaviour results from a series of distinct cognitive cycles that work in their environment, with no different structural module for worldwide advancement
2	Memory and data	Declarative and procedural long-term memories incorporate image structures and quantitative data connected with them
		Momentary working memory gives worldwide correspondence across all cognitive modules, perceptual modules as well as motor modules
		Procedural long-term memory provides global control. It consists of rule-like circumstances and activities, whereas it practices control by changing the items in working memory
		ACT-R provides single declarative memory. Soar handles episodic and semantic memories, whereas Sigma governs procedural memory. Collectively, it can give factual information
3	Learning	Learnable long-term memory material includes both image structures and quantitative data
		Learning occurs because of performance, online and progressively, and is habitually founded on a reversal of the progression of data from the rendition
		In any event, procedural learning involves reinforcement learning and procedural organization. Reinforcement learning produces loads over activity choice while procedural organization produces social mechanization
		Declarative learning entails acquiring facts and fine-tuning their metadata. More complicated kinds of learning are made up of combinations of simpler forms of learning
4	Perception and motor control	Perception results in the storage of symbol structures and related metadata in distinct working memory buffers. Perceptual learning creates new patterns and fine-tunes old ones
		There may be several such perceptual modules, each with its buffer and input from a distinct modality. Top-down information from working memory can alter the perception
		Motor control uses its buffers to translate symbolic relationship structures into outward actions. It acquires new action patterns and fine-tunes old ones

the SMM addresses a typical design for intelligent, individual behaviour that may be deployed in a biological or artificial system.

First, human data is the SMM's main non-functional testbed because, despite the paper's compelling theoretical arguments, humans are still the only species known to be capable of intelligent, general-purpose behaviour. Furthermore, the human brain must adhere to the same principles as non-biological intelligent systems. This suggests that the SMM should be somewhat or completely reflected in the engineering of the human mind [4].

By looking at examples of successful association across cortical and its subcomponents locales in the neuroimaging datasets from various regions, we show in this study that the SMM can measure up to different alternative structures. We likewise show that the SMM reliably gives a preferred clarification of the information over different structures of similar intricacy.

2 Designing a Test for the Cognition Model

The SMM is comprised of various forecasts, including those in regard to the number and reason for the center modules, their relative associations, and insights about the data stream between modules. It likewise contains forecasts in regard to the sort of data traded among modules and its related metadata, as well as the calculations administering learning in the framework.

The SMM's most general concepts—those involving individual components and their functional interactions—are the focus of this investigation. The basic idea is that the various elements shown in Fig. 1 may be connected to particular brain circuits, and that the functional connectivity between these circuits, represented by the arrows in the figure, corresponds to the preferred method of information transfer between various brain regions.

3 Connectivity Testing Using Dynamic Causal Modelling

The association of the model components is likely the least contentious with brain circuits or areas. Distinct modules, for example, have been mapped onto specific brain areas in numerous cognitive architectures. The investigation of functional connectivity across areas, on the other hand, necessitates great caution. We used DCM, or Dynamic Causal Modeling, to investigate the functional connectivity across areas [5]. DCM is a numerical methodology for assessing practical availability between network "hubs" addressing explicit brain regions. A unique framework state condition is utilized to fit the time series of information in a hub in DCM:

$$\frac{dy}{dt} = Ay + \sum_i x_i B(i)y + \sum_j y_j D(j)y + Cx$$

where the four matrices A, B, C, and D indicate the natural associations between different locales (A), the areas that are straightforwardly influenced by task inputs, like specific stimuli or conditions (C), the modulatory impacts of task conditions on association among areas (B), and the modulatory impacts that a 'hub' or 'node' can have on connecting among two different hubs in the network (D) [3].

DCM is the primary tool in our inquiry due to its many characteristics. The first is that it is a top-down analytical strategy that is more appropriate for comparing and evaluating a priori assumptions than for exploratory information examination or we can say data analysis. Another advantage of DCM is that it emphasises connections that are effective and directed rather than just correlations between different time series. [3]. This qualifies it for analysing the arrow directionality in Fig. 1. Due to its improved ability to control temporal distortions brought on by the delayed neurovascular coupling response DCM beats rival approaches in this area. It creates the BOLD signal recorded in the functional MRI and may vary greatly between regions. Another benefit of DCM is that parameterization is entirely performed using a layered Bayesian method, giving each of the models the best chance of accurately representing the data and eliminating physiological parameters that are improbable.

The capacity to directly compare models, which is what this study aims to do, relies heavily on the use of Bayesian estimation. Finally, network dynamics are initially described in terms of condensed neuronal activity, which is subsequently translated into the proper modality, making DCM modality independent. This suggests that fMRI and EEG data may be fitted using the same model (with high spatial resolution, 3D spatial dissemination of perceptions, and poor temporal resolution). Consequently, DCM offers the perfect place to begin for a number of different studies.

4 Implementing the Cognition Model in DCM

It calls for further caution to examine the functional connection across areas. Here, we have opted to use Dynamic Causal Modeling to investigate the functional connectivity across areas [6]. A mathematical technique called DCM is used to calculate the functional connectivity between network 'nodes' that represent certain brain areas. We must first transform the SMM into a network of linked zones in order to start the test. This strategy next calls for figuring out which of the locales are associated with the model elements and how the relationship exemplifies the model principles between regions.

4.1 Modules and Brain Regions

We know, the Standard model is consisting up of five unique components. Given that each test in this part requires manual reactions, the action module may be unmistakably linked to the brain's motor regions (Brodmann Area 1). We decide to interface the perception module with the primary visual cortices due to the intricacy of the image data in the four tasks examined here (Brodmann Area 18) [7]. This process resulted in a placement that was very uniform and constant across tasks. After significant investigation, it was discovered that the working memory module is housed inside the dorsal prefrontal cortex. It is a part of the brain that is also responsible for maintaining and revising short-term memories. The medial temporal lobe and the hippocampus both have a role in creating and maintaining stable representations in the long-term memory module [8]. The procedural module of the model can be recognized as the basal ganglia, an assortment of cores engaged with the securing of procedural abilities and action choice [9].

4.2 Intrinsic Connectivity

In the SMM's DCM implementation, the bulk of arrows in Fig. 1 may be simply transformed into interconnections among the relating brain regions. The arrows between the model's procedural memory module of the model and working memory module of the model is an exception, and it needs to be interpreted in light of the procedural module's specific function in the SMM. For a successful connection, two procedural module assumptions have major ramifications. A certain assumption holds that procedural knowledge offers global control in the form of situations and behaviours that follow rules, whereas assumption contends that the firing of the procedural rules triggers the cognitive cycle of the Standard Model by changing the working memory's contents. In other words, wireless signals to the global working memory region are handled by the procedural module.

By assuming that striatum modifies connections from other brain regions to the dorsal prefrontal cortex (dashed arrows in Fig. 2a) rather than directly effecting the dorsal prefrontal cortex, this role of procedural knowledge may be described in terms of effective connectivity (solid arrows in Fig. 2a) [10]. Similar techniques have been suggested for Dynamic Causal Modelling models of the basal ganglia, and they are compatible with the most recent neural network representations of the circuit. Figure 2a shows how our strategy has been applied in its entirety. The positions of the SMM components in the illustration's boxes correspond to the (rough) locations of the corresponding brain circuits as per a conventional mind layout. Figure 1's directional input channels are depicted as solid black arrows, while the procedural module's modulatory effects are shown as dashed lines that end in circles.

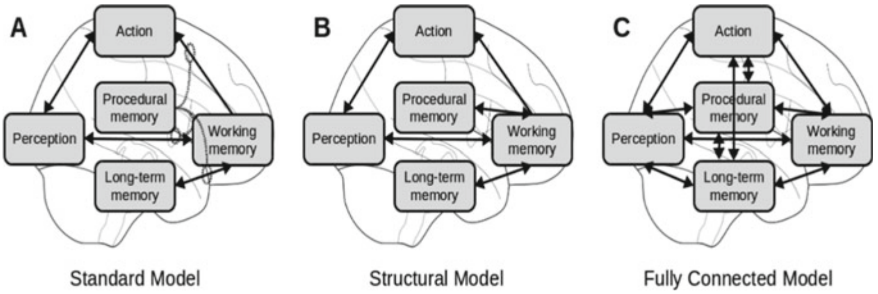


Fig. 2 The three model architecture used in this analysis [3]

5 Analysis of DCM Implementation

The model contains the learnings from creating the general-purpose, intelligent cognitive systems. It is therefore not the best architectural system for a particular purpose. Instead, it stands for the best single functional design that is capable of carrying out a variety of tasks of different complexity. As a result, rather than providing a statistical fit to a specific dataset, the optimum way to evaluate the SMM is to compare it to other designs of same complexity. Additionally, DCM methodology does converge towards accurate network parameters and reliably identifies underlying brain activity generators provided the underlying model is well-specified [11]. The related DCM implementation should result in a similar better fit to the brain data than other, alternative models if the SMM is accurate. In this study, the SMM network will be compared against two other models of effective connectivity between identical places. As an initial assessment, these two models don't provide a complete search of the accessible cognitive modules. They do, however, provide for relevant comparisons because they are at the opposite ends of the spectrum of potential structures.

5.1 The Structural Model

In the given paradigm (Fig. 2b), the direct association between both the procedural module and the working memory module take the role of the modulatory connections of the basal ganglia. As a result, this fulfills the aim of the research. A version of the Standard Model of Mind that corresponds to the general channels of communication in Fig. 1, but it does not correspond to the specific functional job of the procedural module, as suggested by assumptions. It shows whether the practical supposition in regard to the meaning of procedural information is important to portray the progression of assignment in the human mind. Or, to put it another way, this model offers a fascinating contrast for the SMM because to its similarities and enhanced simplicity, providing it a benefit in terms of Bayesian model comparison.

5.2 The Fully Connected Model

In the given model, all the brain regions are associated in the two directions (Fig. 2c). This model carries out an alternate point of view on the idea of brain capability and fills in as a hypothetical direct opposite to the underlying model. While the underlying model recommends that limitations on the directional signals of mind regions are adequate to make sense of patterns of organization action, this model portrays the restricting perspective that there is no genuine "engineering" that is unchanging across exercises and that brain's flexibility comes from the way that all regions are hypothetically associated, as opposed to through a useful association. All in all, there is simply task-driven movement; there is no invariant engineering. The three-model design utilized in this examination is portrayed in Fig. 2.

5.3 Testing Across Multiple Tasks

The three models discussed above were assessed over a range of cognitive abilities, from unstructured, fluid problem-solving to very simple stimulus-response mappings, in order to emphasise the SMM's universality.

6 Materials and Methods

For this study, four functional MRI datasets were employed, all from published research. The four activities were chosen to demonstrate the range of high-level cognitive talents seen in humans which helps us to get some functional imaging parameters. These functional imaging parameters of the four datasets are summarized in Table 2.

Table 2 Details of the four datasets, N = number of member per dataset; slices = number of angled pivotal cuts; TR = redundancy time; TE = reverberation time [3]

Task	Scanner	Functional imaging parameters	Slices	N
RITL	Philips achieva	TR = 2.0 s, TE = 30 ms	36	25
Stroop	Siemens trio	TR = 1.5 s, TE = 30 ms	29	28
RAPM	Philips achieva	TR = 2.0 s, TE = 20 ms	36	24
Flanker	Siemens trio	TR = 2.0 s, TE = 20 ms	40	26

6.1 The Flanker Task

Participants in the Flanker task have to give response to a center arrow-like sign (e.g., "<"), also known as 'less than' sign, with a hand wave that corresponds to the sign's orientation (e.g., left). The primary sign, on the other hand, is flanked by four "flankers" or distractors, which may point towards the same way (congruent trials, for example, "<<<<<") or could in the other direction (incongruent trials, for example, "<<><<"). To handle the interference created by the flankers, incongruent stimuli require further control. This task is potentially the most essential of the trial ideal models used to explore mental control and chief cycles, including least tangible handling, and depending on normal boost reaction mappings. The data is taken from the OpenfMRI public repository [12].

6.2 The Stroop Task

In the Stroop task, participants identify the colour of a written word with a manual reaction. Interference occurs because the words are colour names in and of themselves, resulting in both congruent trails (e.g., "RED" in red) and incongruent trails (e.g., "RED" in blue). The data is taken from the OpenfMRI public repository [13].

6.3 Rapid Instruction Task Learning (RITL)

Even while completing unfamiliar activities, humans are extraordinarily efficient. This aptitude for Rapid Instructed Task Learning (RITL) has lately received a lot of attention. We anticipated that RITL necessitates proactive control, which is based on the advance loading of task-related information into working memory, according to Braver's Dual Mechanisms of Control theory. In this paper, we provide a unified experimental framework for systematic testing of predictions derived from this hypothesis. Participants will learn many new task rules, each in its own experimental miniblock, which will consist of (a) instructions, (b) cuing which subset of the newly instructed rules remains relevant, (c) a NEXT phase in which people who participated press a fixed key to advance the screen and where automatic rule activation is evaluated, and (d) GO—the rule implementation phase. The data is taken from the OpenfMRI public repository [14].

6.4 Raven's Advanced Progressive Matrix (RAPM)

RAPM is a nonverbal test of fluid intelligence and reasoning ability that is frequently utilized. Each problem is represented by a 3-by-3 matrix. The matrix's eight cells each feature a shape made up of distinct elements, while the last cell is vacant. Each figure's visual qualities (like shade or direction) changes among indexes as per certain yet secret rules. Members should conclude the standards and accurately select the shape that finishes the matrix from a set of given four possibilities. This dataset originates from the sole fMRI research that, to the best of our knowledge, employs standard RAPM tasks rather than simplified variants. The data is taken from the OpenfMRI public repository [15].

7 Task Implementation in DCM

Although the DCM models' fundamental design may be established irrespective of the task, the model estimate necessitates of how different stimuli can influence the regions in an explicit explanation. We utilized a similar framework to describe the experimental circumstances throughout the four separate tasks to make the estimating and comparison techniques as universal as feasible. Every stimulus presented in this scheme is encoded to the visual cortex as an input. Visual stimuli might vary greatly in nature in particular activities. Stimuli in RITL can be seven one-digit numbers or five-word instructions, but stimuli in RAPM can be a four-cell or even nine-cell difficult visual issue or a set of four alternative one-cell answers. Distinct categories of stimuli are represented by different inputs to allow for this diversity.

All the four tasks also share a fundamental distinction among "easy" and "difficult" conditions. Difficult tasks require a little more computation or processing as compared to easy tasks. DCM, as a method, is unconcerned about the reason of the stimuli. As a result, the information that particular trials, or situations need more processing must be explicitly expressed. We chose to provide extra input to the working memory area in this study to reflect the demands for processing stimuli in the "difficult" circumstances [16].

This conclusion is based on the pretty uncontroversial notion that increased processing of particular stimuli would result in an increased working memory burden. It should be noted that this assumption remains quite broad, as it does not consider the thought of precise ways in which 'tough' trials may change among tasks.

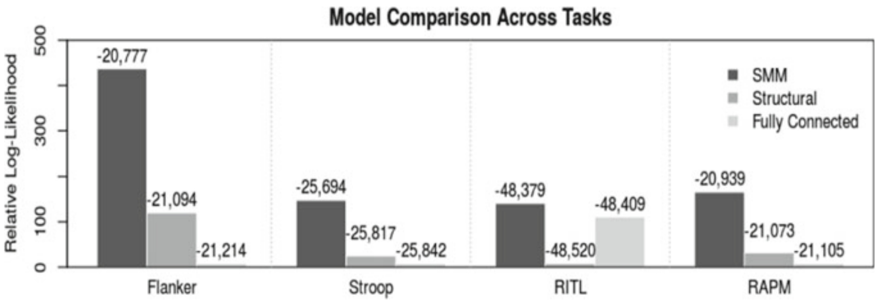


Fig. 3 Bayesian model comparison across tasks [3]

8 Task Results

To look at the three different variety of models by using the four different variety of datasets, we utilized a Bayesian model choice methodology [17]. The consequences of the model correlations are portrayed in Fig. 3. The plots utilize relative log-probability to give an examination across a typical scale, while the names over the bars address genuine log-probability values from the investigation.

Across each of the four different datasets, the Standard Model of Mind gives the best clarification to the information. To place this finding into setting, think about the quantitative contrasts in log-probability among the models. Since DCM figures priors (while additionally representing model intricacy), contrasts in log-probability might be utilized to compute the back likelihood of the one model could be better than all other models. Across correlations, back probability (p) of the Standard Model of Mind remains the best clarification for the information was more than 0.99. Expecting uninformative priors (no starting inclination across models), log-probability contrasts can be deciphered as Bayes factors; subsequently, a distinction of ‘d’ for M1 over M2 shows that M1 is more probable than M2. It was found that d > 30 specifies "solid proof" for one model over another. The Standard Model of Mind beats the underlying model by essentially d > 124 and the completely associated model by basically d > 30 across all undertakings, as displayed in Fig. 3 [18].

9 Limitations

These findings, however, should be seen in light of several possible limitations. The first observation is that, even though we eventually integrated the data into one task-independent system, the topology of the network predicted by the various tasks varies significantly. DCM, on the other hand, is a top-down approach that is confined to evaluating the fit of some of these network architectures. In a conclusion, the extent to which a task-independent design can be determined from varied activities is controversial, reflecting the fundamental premise that brain activity discloses a shared invariant architectural design at a very high level. Of course, this concept is not widely accepted and should be studied individually in future research [19].

10 Conclusion and Future Scope

The speculation of the "Standard Model of the Mind" depends on a correlation of mental designs. Our fundamental examination gives the principal direct observational proof for this hypothesis by showing that the Standard Model of Mind is a conceivable clarification for information designs can be really differentiated to different models. All the more critically, and maybe out of the blue, our discoveries uncover that the SMM's fundamentals stay valid over an enormous number of members (N = 103) and four unique undertakings of different intricacy. While the Standard Model of Mind was constantly picked as the best cognitive model of all, the general significance of the other two cognitive models changed relying upon the task. The Fully Connected model beat the Structural model in the Rapid Instruction Task Learning worldview, which requires a more noteworthy assortment of cognitive capabilities and their transaction. This implies that our discoveries don't have anything to do with any of the models that have a lower deduced probability of fitting the information. Albeit empowering, these discoveries are as yet primer, and more exploration is required.

The SMM's legitimacy, specifically, ought to be surveyed against a bigger arrangement of elective models that cover a more prominent assortment of likely geographies. These discoveries need be imitated across greater datasets enveloping a more extensive scope of spaces, for example, long haul memory, independent direction, and language capacities. At last, association examination ought to be joined with different kinds of investigation that evaluate other SMM suppositions, like utilizing Representational Similarity Analysis to explore information portrayal suspicions. Existing huge scope neuroimaging datasets, like the Human Connectome Study, may give the most interesting field to the advancement of this undertaking [20].

References

1. Laird JE, Lebiere C, Rosenbloom PS (2017) A standard model of the mind: toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. AI Mag 38(4):13–26
2. Hake HS, Sibert C, Stocco A. Inferring a cognitive architecture from multi-task neuroimaging data: a data-driven test of the common model of cognition using granger causality
3. Stocco A, Laird JE, Lebiere C, Rosenbloom PS (2018) Empirical evidence from neuroimaging data for a standard model of the mind. CogSci
4. Ikram S, Dhanda N (2021) American sign language recognition using convolutional neural network. In: 2021 IEEE 4th International conference on computing, power and communication technologies (GUCON). IEEE, pp 1–12
5. Friston K, Moran R, Seth AK (2013) Analysing connectivity with Granger causality and dynamic causal modelling. Curr Opin Neurobiol 23(2):172–178

6. Friston KJ, Harrison L, Penny W (2003) Dynamic causal modelling. Neuroimage 19(4):1273–1302
7. Kane MJ, Engle RW (2002) The role of prefrontal cortex in working-memory capacity, executive attention, and general fluid intelligence: an individual-differences perspective. Psychon Bull Rev 9(4):637–671
8. Squire LR (1992) Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans. Psychol Rev 99(2):195
9. Stocco A, Lebiere C, Anderson JR (2010) Conditional routing of information to the cortex: a model of the basal Ganglias role in cognitive coordination. Psychol Rev 117(2):541
10. Prat CS, Stocco A, Neuhaus E, Kleinhans NM (2016) Basal ganglia impairments in autism spectrum disorder are related to abnormal signal gating to pre-frontal cortex. Neuropsychologia 91:268–281
11. David O, Guillemain I, Saillet S, Reyt S, Deransart C, Segebarth C, De-paulis A (2008) Identifying neural drivers with functional MRI: an electrophysio-logical validation. PLoS Biol 6(12):e315
12. Eriksen BA, Eriksen CW (1974) Effects of noise letters upon the identification of a target letter in a non-search task. Atten Percept Psychophys 16(1):143–149
13. Verstynen TD (2014) The organization and dynamics of corticostriatal pathways link the medial orbitofrontal cortex to future behavioral responses. J Neurophysiol 112(10):2457
14. Cole MW, Laurent P, Stocco A (2013) Rapid instructed task learning: a new window into the human brain's unique capacity for flexible cognitive control. Cogn Affect Behav Neurosci 13(1):1–22
15. Stocco A, Prat CS, Graham LK (2019) Individual differences in reward-based learning predict fluid reasoning abilities
16. Kelly AC, Uddin LQ, Biswal BB, Castellanos FX, Milham MP (2008) Competition between functional brain networks mediates behavioral variability. Neuroimage 39(1):527–537
17. Stephan KE, Penny WD, Daunizeau J, Moran RJ, Friston KJ (2009) Bayesian model selection for group studies. Neuroimage 46(4):1004–1017
18. Kass RE, Raftery AE (1995) Bayes factors. J Am Stat Assoc 90(430):773–795
19. Dhanda N, Datta SS, Dhanda M (2022) Machine learning algorithms. In: Research anthology on machine learning techniques, methods, and applications. IGI Global, pp 849–869
20. Kriegeskorte N, Mur M, Bandettini PA (2008) Representational similarity analysis-connecting the branches of systems neuroscience. Front Syst Neurosci 4

An Overview on Security and Privacy Concerns in IoT-Based Smart Environments

Nitin Anand and Khundrakpam Johnson Singh

Abstract Urban surroundings and human quality of life have significantly improved because of smart environments which include transportation, healthcare, smart buildings, public safety, smart parking, traffic systems, smart agriculture and other areas. They are completely capable of controlling the physical objects in real-time and delivering intelligent information to citizens. Technologies for smart cities can compile personal information. However, security and privacy issues may arise at several architectural levels. Therefore, it is crucial to take these security and privacy issues into account while creating and implementing the applications. In addition to discussing the significant issues of privacy and security throughout the development of the applications for smart cities, the article highlights the main applications of smart cities. We have analysed various possible attacks on IoT networks and moved towards Intrusion detection and highlighted the impact of communication technologies like 4G, 5G and 6G on the security and privacy on the IoT environment.

Keywords Blockchain · DDoS attack · Internet of things · Intrusion detection · Privacy · Security

1 Introduction

Data and services can be transmitted over global networks thanks to the Internet of Things (IoT) which is based on the architecture for Internet. It has an impact on security and privacy, which is a problem in many contexts. To guarantee the resistance of architecture to attacks, data authentication, access control and client privacy, procedures must be put in place. The ideal way to develop an appropriate legal framework that takes into consideration the underlying technology is through an

N. Anand (✉) · K. J. Singh
National Institute of Technology, Manipur, Imphal 795004, Manipur, India
e-mail: nitin1036@gmail.com

K. J. Singh
e-mail: johnkh34@gmail.com

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_21

international legislator, supported by the private sector in accordance with particular needs, and therefore easily adaptable. The right to information, regulations on IT security legislation, measures encouraging the use of IoT technologies, provisions limiting or regulating their use and the creation of a task force. According to Weber [1], all relevant legislation must take into account the legal issues raised by the IoT.

Smart cities as well as smart homes are only two examples out of many applications and services that the IoT offers. IoT smart things collaborate with other elements, like proxies and mobile phones, to deliver new and improved services. User privacy is becoming increasingly important in the IoT context, and as more devices are connecting to the Internet each day, there is a requirement for security solutions. Any security design must take into account the three main security needs, also known as CIA: **Confidentiality, Integrity** and **Availability**. Only the designated user is allowed to read the message due to confidentiality. Integrity ensures that the sent information gets its target without modification, and availability ensures that each service or piece of data is accessible to the user at any time. The most significant danger to the availability of business networks, products and services is DoS/DDoS. The majority of DoS/DDoS attacks can now be defended against by modern security systems, as described by Malliga et al. [2].

2 Research Methodology

This work has been carried out using the following methodology. We have looked through IoT security literature to assess the threats taxonomy and issues associated with IoT security. In order to find pertinent survey articles, the terms IoT and security were searched for in multiple published databases, including ACM, IEEE, Elsevier, Springer and MDPI. Following the completion of these taxonomies, the authors evaluated the various techniques that were presented in those surveys and chose a set of relevant topics that they genuinely think are crucial for network security. These topics were chosen based on our own expertise in the security domain. We have also looked for papers that have presented different solutions that were well known in this field. Finally, we have searched the Internet for intriguing security-related items using search engines.

2.1 Assessment of Research Content.

Finding original reviews of intrusion detection systems as well as gathering all relevant publications required an objective search strategy. The search process should be as complete as well as logical as it is possible to be and search terms need to be provided. We have found that the title of a few works on anomaly-based intrusion detection includes the word ‘intrusion’. Therefore, in order to include publications on anomaly-based intrusion detection, we have defined the search termed 'network intrusion detection'.

2.2 Research Questions

In order to direct our research, we came up with particular research questions (RQs) and include in-depth sub-questions.

The gist of the entire scenario can be seen according to these 8 questions that were raised in the entire course of study:

(a) What issues are addressed by intrusion detection methods?
(b) How do the literatures split up between the various fields?
(c) Why is this distribution the way it is?
(d) How does research in these areas vary from nation to nation?
(e) What are the typical data preparation procedures employed in network intrusion detection?
(f) What technical properties do the pre-processing technologies have and how are they implemented?
(g) How are their applications for intrusion detection distributed?
(h) Which models are used in intrusion detection techniques?

3 Literature Review

Maselli et al. [3] present HyBloSE, a solution that can secure smart buildings even if the cloud management system is attacked. Their strategy is based on a Moving Window Blockchain and Delegated Proof of Authority consensus. It decouples security from IoT application logic, allowing centralized logic to be saved while security management is distributed and it solves these restrictions, particularly in terms of practicality. It can run on low-power devices and does not require any additional hardware. The results show that it has a small overhead (below 0.4 ms), making it fully compatible with existing hardware.

Data security and privacy issues with regard to cloud-based smart city technology have been examined by Khan et al. [4] in their work. Their approach examines security from many stakeholders' points of view and recommends end-to-end security for applications building smart cities that make use of open data and encourage citizen participation. They have offered a framework to solve challenges with data security, privacy as well as trust. In the smart city scenario, such a framework would be useful for providing inhabitants with safe, context-sensitive information services. With encouraging results, the security verification software is used to assess the authentication element of the proposed architecture against potential threats. Other parts of the suggested architecture are now being developed and tested as part of future work, including the secure communication protocol.

IoT security is a topic that both academia and business are currently paying significant attention to. Existing security methods are not always suited for IoT due to high energy consumption and computational complexity, Dorri et al. [5] have proposed the mechanism which uses the Bitcoin BC, which is an immutable record of blocks, to overcome these issues. A smart home was used in the case study to illustrate the concept. The various components of the smart home layer were detailed in this study, as well as the various transactions and procedures involved with it. They have also provided a comprehensive examination of its security and privacy. The overheads caused by their strategy are moderate and reasonable for low-resistance IoT devices, according to the simulation results.

In this study, Waheed et al. [6] analysed the most recent IoT worries and divided them into two categories: security and privacy. In their work, the results, attack types, impact layer and possible solutions were briefly discussed. In order to fill in any possible gaps, a complete overview of the most recent methodology of the research on IoT security and privacy by the means of ML algorithms as well as BC technologies was provided. A modern IoT security and privacy techniques that integrate ML algorithms with BC methodologies have been evaluated in this paper. In an effort to comprehend the security as well as privacy concerns in an ML, they have proposed an idea to exhibit an ML security plan for IoT based on prior studies.

A complete picture of an IoT system that needs an object, cloud, as well as controller is provided by Ling et al. [7] in their publication. For such a system, ten core functions have been identified. According to the risk analysis of various IoT system components, these capabilities need to be properly secured. They have then shown how they had exploited an IP camera system to find three attacks: device scanning, brute force and device spoofing. All of these could completely take control of all IP cameras made by the manufacturer. They have tested the attacks in the real world and found that, regardless of the password, a device spoofing attack has a 98% chance of recovering a user's password.

Shakarami et al. [8] proposed a method for a smart home-based IoT ecosystem, the study provides an architecture-oriented enforcement of access the administration. The system is decentralized, auditable and dependable since it is built on the Ethereum blockchain. Although their implementation outcomes are encouraging and the use of blockchain for operational access control is unpromising, an administrative model can be benefitted from its advantages. For blockchain technology, one of the properties and limitations has also been discussed.

The phrases 'data owners' as well as 'data consumers' were used by Barhamgi et al. [9] to describe users of cyber-physical processes who produce data through communicating with them (for example, residents of smart homes, patients being monitored, etc.) and stakeholders who are seeking to collect and utilizing the data produced, including electricity companies in smart grids, healthcare providers in smart health care networks, government agencies and so forth. In this work, they outline an ongoing endeavour to enable smart cyber-physical system users to independently protect their privacy.

Although security procedures and regulations for IoT-based smart environments differ greatly from those utilized in non-IoT domains, Karie et al. [10] concur that it is now necessary to follow suitable security standards and look into the possibilities for these environments. This is supported by the idea that the secured design for IoT-based smart environments depends on a wide variety of security checks, many of which may not be directly addressed by the present security standards, such as the effective identification approaches described in this study. Due to these issues, the security of IoT-based smart environments is also most likely to be difficult to create, implement, enforce as well as administer, and is largely determined by installations and settings made by usually inexperienced staff members.

Due to the numerous documented issues with IoT-based smart environments, different parties have made their efforts to address specific issues. On the other hand, the contribution in this work is a remarkable effort based on an architecture of challenges for IoT-based smart settings and a review of the literature. The taxonomy can only be applied to the data that the authors of this study have examined. However, as can be seen in Fig. 1, the taxonomy was developed with the primary challenges related to IoT-based smart settings in mind. It is also important to note that the concerns discovered and detailed in this study are by no means exhaustive.

According to Kulyk et al. [11], smart environments have become more prevalent despite a number of possible security and privacy hazards. But are people aware of the implications of embracing smart environments? They performed a survey with 575 people from three countries to answer the research topic about smart homes

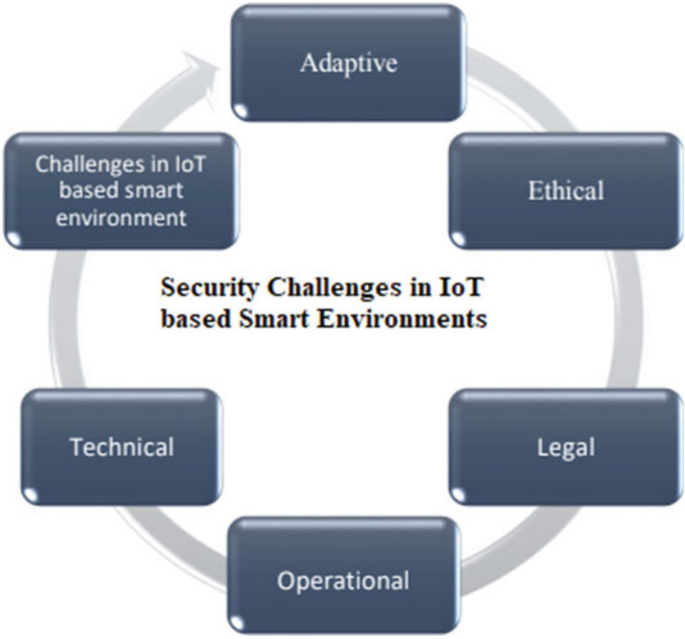

Adaptive
Ethical
Legal
Operational
Technical
Challenges in IoT based smart environment
Security Challenges in IoT based Smart Environments

Fig. 1 Various aspects of IoT-based smart environment

and healthy environments (Germany, Spain and Romania). Less than half of respondents voiced at least one privacy or security worry, with German participants doing so significantly more often than Spanish participants, while Spanish participants noticeably more frequently as Romanian participants. The majority of abstract concern responses mentioned the danger of a cyber-attack, while German participants' key concerns were privacy and data security. The majority of abstract concern responses mentioned the danger of a cyber-attack, while German participants' key concerns were privacy and data security. The majority of participants who reported receiving a threat also voiced privacy worries, such as being observed or spied on in their own home, having information on them collected, or having such information revealed to or provided to another person. The majority of participants who reported receiving a threat also indicated privacy concerns, such as being observed or overheard in their own homes, having information about them collected or having such information made public or transferred to another person.

Big data as well as IoT-based applications involved in making the smart environments are discussed in Hijaji et al. [12]. In these sectors, the goal is to detect the significant application domains, current trends, data architectures and ongoing issues. As per their understanding, this is the first literature review of its own kind, examining the literature published in peer-reviewed journals between 2014 and 2022, using a feature-step selection procedure of identification, screening, eligibility and inclusion. A systematic review was done for investigating these records in which six primary research questions were answered. The results provide new opportunities for real-world smart input signals that monitor, safeguard and enhance natural resources when big data with IoT technologies are combined. Among the topics covered in this survey are smart metering, smart disaster alerts, smart farming and agriculture and smart environment monitoring. We conclude by reviewing the most popular big data and IoT approaches, which will hopefully serve as the basis for future research on smart cities and their environments.

Salim et al. [13]'s explanation of the reasons and factors that lead attackers to select new IoT targets for DDoS attacks. There are many tools available for creating botnets out of IoT devices, and additional tools are being looked into for using IoT bots to conduct DDoS attacks. They investigated the hacking and usage of IoT devices as bots, the classification of DDoS assaults on the IoT environment and they created a clear and systematic categorization of various cloud-based DDoS attack types. They also investigated and gave a complete study of twenty-one state-of-the-art defence techniques for preventing, detecting, as well as reducing DDoS attacks in the academic literature from the past and the present. This survey contributes to a full understanding of the many attack types that can be conducted using specific technologies and explains how to defend, for example, in order to recognize, halt or mitigate attacks.

DDoS (Distributed Denial of Service) attacks, that primarily target a web server, pose serious risks to data centre applications, according to Singh et al. [34]. The requirements of application layer attacks and Flash Events (FE) have not yet been met by the various methods for detecting and mitigating such attacks. The goal of this study is to recognize application layer DDoS attacks as well as distinguish them apart from FE. They have investigated a DDoS attack model and identified the incoming packet characteristics which signify the beginning of the attack. Based on the type of the attack, they have examined the statistical aspects of arriving packets like inter-arrival time, the probability that an IP address will remain unique over time and the lack of HTTP GET requests. The fuzzy classification model receives these variables as input. To offer an optimal variation for the input parameters, we applied the Genetic Algorithm (GA).To identify whether web-accessing clients are behaving in an attack, normal or FE fashion, fuzzy logic has been used to the optimized values. According to their findings, the Fuzzy-GA model accurately detects DDoS attacks with a 98.4% accuracy rate and FE with a 97.3% accuracy rate.

The effect of DDoS attacks was examined in Thongam et al. [35], as well as key aspects that influence the attack. Based on the amount of HTTP GET requests, the entropy of the requests and the variation of the entropy, they have employed a trained MLP using GA learning method to detect the DDoS attack. They have established that entropy has a higher value in the case of typical customers and a lower value in the case of an attack. They've also determined that the variance for an attacking client is nearly zero. It demonstrates that the amount of HTTP GET requests generated for the attacking period does not vary much.

4 Security Challenges in IoT-Based Smart Environments

DDoS attacks have gained popularity recently due to the open accessibility and lack of security of IoT devices. DDoS attacks can be started against other targets using the intrusion, which is incredibly vulnerable to them. Attackers keep picking off different targets until they create a botnet. According to Elrawy et al. [14], the three main security issues in an IoT-based network are **Confidentiality**, **Integrity** and **Availability**. Both the data security and the routing peers participating in data transmission must be authenticated as the part of the authentication procedure. A significant challenge in IoT device authentication has been identified as the effective deployment and maintenance of keys. The complexities in compliance of IoT environments, together with the security of IoT systems, severely impede the existence of smart environments in the real world. DoS and DDoS attacks targeting IoT networks have an impact on IoT resulting in an impact on the services offered by smart environments as well. Researchers look at IoT security challenges from a variety of perspectives, including how susceptible IoT communication protocols are to security risks. Because this paper focuses on IDSs for the IoT architecture regardless of protocol, it highlights the security challenges that IoT systems face based on the IEEE classification and the generic IoT architecture. The security measures used to protect communications utilizing the aforementioned protocols must provide respectable levels. The appropriate information confidentiality, integrity, authentication and non-repudiation requirements must be met by the security techniques used to protect communications using the aforementioned protocols. The security of IoT communications can indeed be examined within the context of the protocol stack itself, as we will see throughout the article, or on the other end using external mechanisms. Other security factors for the IoT must be taken into account, especially for connections with sensing devices. For instance, WSN environments could be exposed to Internet-based assaults like Denial of Service (DoS) attacks, as mentioned in Granjal et al. [15].

4.1 How Communication Technologies like 4G, 5G and 6G Impact the Security and Privacy Concerns in the IoT Environment?

The physical environment can now be closely associated with actuators, sensors, as well as other cognitive factors while maintaining stable network connectivity, thanks to considerable advances in communication technology and the Internet of Things' rapid expansion. A smart environment is made up of computational elements and a physical environment that is constantly connected. A smart environment strives to support and enhance its residents' capacities to carry out tasks, such as guiding elderly people through unfamiliar areas and lifting heavy objects. Ahmed et al. [16]'s efforts to employ IoT for life simplification and research the effects of IoT-based smart environments on human existence. They have also examined ongoing studies with the goal of creating IoT-based smart environments. Wireless technologies used to establish connections over the Internet are known as communication enablers. Wi-Fi, third generation (3G), fourth generation (4G) and satellite are the most widely used wireless Internet Technologies. While Wi-Fi is primarily employed in smart homes, smart cities, smart transportation, smart industries and smart buildings, 3G, 4G, 5G and 6G are typically used in smart city and smart grid contexts. These ecosystems all use satellites.

It is expected that the fifth- as well as sixth-generation (5G and 6G, respectively) communication systems will significantly outperform the current fourth-generation system. Some of the major and frequent concerns relating to the quality of service for 5G and 6G communication systems include high capacity, massive connection, low latency, high security, low energy consumption, outstanding level of performance and reliable connectivity. 6G communication will likely outperform 5G communication in these areas by a wide margin. The IoT, which is built on the multimodal internet, will also be necessary for the 5G-and-beyond (5 GB) communication systems (such as 5G as well as 6G). As a result, 5 GB wireless networks will have a number of challenges in managing the diverse range of heterogeneous traffic.5G communications are accessible starting in 2020. The launch of 6G connection is then scheduled to take place between 2027 and 2030. On the basis of touch internet, it is challenging to realize the goals of 5G/6G and IoT. Only RF-based technologies are unable to meet the strict standards of the forthcoming 5G/6G and IoT networks. OWC technology offers a complementary RF network that is the best. The coexistence of RF and optical wireless devices will help these networks achieve their goals.

Table 1 Comparison of communication technologies used in smart environment

Technology	Frequency	Data rate	Range	Power consumption
Bluetooth	2.4 GHz	25 Mbps	10 m	Low
DASH 7	433 MHz	Up to 200 Kbps	100 m	Low
Zigbee	2.4 GHz	250Kbps	Up to 100 m	Low
Wi-Fi	Up to 5 GHz	Up to 8.75 Gbps	Up to 150 m	Medium
3G	850 MHz	24.8Mbps	5 m	High
4G	Up to 2500 MHz	800 Mbps	6 m	High (twice of 4G)
5G	Up to 40 GHz	Up to 20 Gbps	500 m	High (twice of 4G)
6G	Up to 3THz	Up to 100 Gbps	1 km	Low

Table 1 compares and summarizes the communication techniques utilized in IoT-based smart environments.

4.2 Taxonomy of DDoS Attack on IoT

A cyber-attack, known as a Denial of Service (DoS) attempt, uses the resources or bandwidth of a legitimate user to try to block access to a site. In contrast to Distributed Denial of Service (DDoS), which is a cyber-attack in which incoming traffic emanates from numerous sources, in this instance, an attack is carried out using a single machine without the use of malware. In comparison to a DoS attack, it is more sophisticated and difficult to prevent. It attacks computers with malware. In recent years, DDoS attacks have shown a diversity of attacking strategies, and a lot of alternatives are continuously being investigated. Traditional DDoS attacks and IoT-specific DDoS attacks are very identical to one another. They employ similar techniques to take advantage of flaws in both IoT devices and conventional systems. IoT-specific DDoS attacks, on the other hand, are more varied and sophisticated because of the variety of IoT devices. On the basis of the attacking methods, all of these attacks can be divided into two categories as shown in Fig. 2. One of the most important factors to remember regarding DDoS attack classification is that it is entirely dependent on the attack's impact.

Bandwidth Depletion Attacks and Resource Depletion Attacks are the two primary forms of DDoS attacks.

1. Bandwidth Depletion Attack

This form of attack uses the target's bandwidth that leads to flooding of the network with unsolicited traffic, preventing the legitimate traffic from arriving at the victim. Tools like Trinoo are commonly used to carry out these kinds of attacks. Bandwidth Depletion Attacks are of two types.

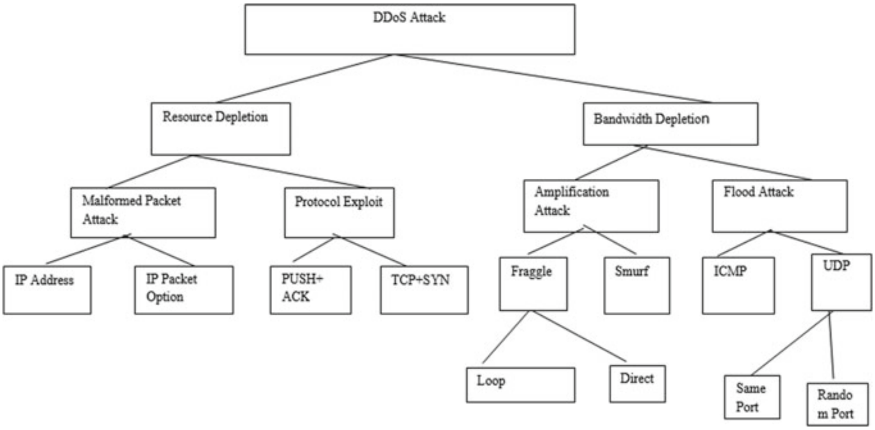


Fig. 2 Taxonomy of DDoS attack

I. Flood Attack

Flood attack is based on the fact that the zombies flood the victim system by IP traffic. The victim system is bombarded by a huge number of packets from the zombies, which causes the victim system to lag, crash or saturate the network capacity. This prevents authorized users from using the victim's resources. UDP packets are transmitted to the victim computer's random or predefined ports during a DDoS UDP Flood attack as mentioned in Specht and Lee [37]. Attacks using UDP flood commonly target arbitrary victim ports. In order to identify the applications that have requested data, the victim system examines the incoming data. If no applications are correctly running on the victim system's machine's targeted port, the victim system will send an ICMP packet to the sender system.

II. Amplification Attack

An example of a DDoS attack is DNS amplification, in which the attacker impersonates the victim's source address. A DNS amplification attack's architecture is shown in Fig. 3. The hacker instructed the bot to send DNS requests and queries to a DNS server using a fictitious IP address and large-text resource records. Recursively, all of these queries are forwarded to the DNS name server. There is no need for handshaking in this process so the DNS query request or response depends on UDP. The reply from the DNS server is then sent to the IP address of the victim. The broadcast IP address feature is used in a DDoS amplification attack to magnify and reflect the attack [36]. It permits a sender system to use a broadcast IP address instead of a specified address as the destination address. This causes packets to be duplicated within the network and sent to all IP addresses. In this sort of DDoS attack, the attacker transmits the broadcast message directly that increases the volume of the attacking traffic. This attack will succeed if the attacker decides to send the broadcast message directly.

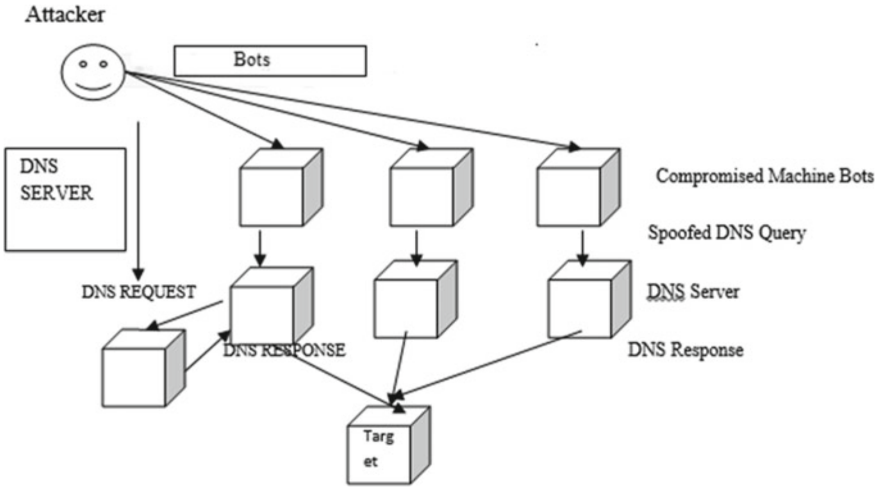


Fig. 3 DNS amplification attack

DDoS attacks have been the subject of numerous surveys in the past. The majority of them, on the other hand, have covered a typical DDoS attack, its variations and established network defence methods. In the literature, there is a wealth of knowledge on defending against DDoS attacks that can be used to lay the foundation for a protection against DDoS attack. In an IoT network, works of Zargar et al. [39] have covered a variety of DDoS flooding attacks, covering various types of botnet-based DDoS attack types which did not cover any fresh virus strains. Our poll looks at present patterns, potential outcomes and tried-and-true methods for stopping DDoS attacks.

New virus attack strategies have been presented by modern DDoS attacks. In order to create a more efficient defence system against these attacks, it is now essential to understand several recently identified malware types and their traits. The poll's results are contrasted with those of recent polls that were also performed while being susceptible to DDoS attacks in Table 2. In order to properly comprehend the required facts, we have collected surveys on DDoS attacks in IoT. But the vast majority of them draw inspiration from conventional DDoS attacks along with related defences.

There are many surveys on DDoS attack defence in conventional networks, but very few on DDoS attack defence in Internet of Things networks. Our evaluation examines all of the key protective factors that have utilized different strategies to thwart DDoS attacks for both conventional and Internet of Things (IoT) networks. Based on the techniques as well as strategies they have employed to combat DDoS, each of these defence strategies have been divided into three categories: mitigation, detection and prevention. Our study is centred on Botnets and Malware, which are the primary contributors to contemporary DDoS attacks in the Internet of Things, as well as defence tactics.

Table 2 Comparison of DDoS attack on IoT networks

Authors	Security Issues in IoT	Nomenclature of DDoS attack	IoT botnets	DDoS defence	Defence mechanism comparison	Challenges
Sonar and Upadhyay [38]	Yes	No	No	No	No	No
Zargar et al. [39]	No	Yes	Yes	Yes	Yes	Yes
Zhang and Green [40]	Yes	No	No	Yes	No	No
Ghani et al. [41]	Yes	Yes	Yes	No	No	Yes
Yang et al. [42]	Yes	Yes	Yes	No	No	No
Vishwakarma and Jain [43]	Yes	Yes	Yes	Yes	Yes	Yes

We've briefly discussed the well-known IoT malware that hit certain well-known servers. There are, undoubtedly, numerous surveys on defending DDoS attacks on traditional networks, but only a handful on defending DDoS in cloud networks. We have identified outstanding obstacles and issues that need to be solved in order to produce an ideal solution against DDoS attacks in an IoT context as a result of our extensive comparison investigation of numerous defence systems.

5 Intrusion Detection and Prevention for IoT-Based DDoS Attacks

It has widely been observed that the services become inaccessible, network defences become impenetrable and the availability factor is put at risk when a DoS attack is launched against an IoT network system and loads the network with a lot of traffic. Regardless of the fact that most Intrusion Detection and Prevention (IDPS) use one or two detection approaches, which may be grouped into two subgroups: anomaly-based and signature-based, they appear to have little chance of surviving a DoS attack. By taking advantage of the weaknesses in the employed techniques, a DoS arrives and forces its attack. The signature-based detection method, also known as rule-based or misuse-based IDS, can spot an attack by examining well-known attack signatures and infection behaviours.

A software package called the Intrusion Detection System (IDS) keeps an eye on and protects against the dangerous behaviour on a network. It can also be viewed as the process for detecting unauthorized access to and breaches into computer networks and information systems. External intruders seek to enter the network and/or information systems from outside the network, whereas internal intruders are legitimate users who try to increase their credentials in order to unlock restricted data or services.

IDS typically consists of a sensor and a reporting system. Information gathering is the main objective of the sensors.

When developing any form of IDS, whether an NIDS or a HIDS, IDS placement is also an important factor to be considered. The IDS's overall efficiency is affected by where it is placed in the IoT network. The two most common IDS deployment options are centralized as well as distributed.

The centralization approach includes the benefit of centralized power, but it also carries the risk of overwhelming the system, that would be detrimental to the Quality of Service (QoS) for IoT networks. The advantage of the dispersed strategy is that it increases processing power while reducing the volume of monitored traffic. However, due to the inherent administrative challenges, establishing an IDS across multiple areas of an IoT network is tough. Not least of all, it requires the updated normal as well as anomaly databases connected with IoT services along with their applications. These databases will be very beneficial in analysing the different types of IDS and tactics in the IoT framework. Whether or not IDS comparison is successful and relevant will depend on these databases.

Figure 4 shows the variety of types of detection techniques which can be incorporated in an IoT environment. The detection strategy is the one that attracts the most emphasis among these three categories, and Wang et al. [17]'s detection strategy descriptions are the sole ones used to build most systems. Signature-based IDS, Anomaly-based IDS, specification-based IDS and hybrid IDS are the subcategories of detection approaches.

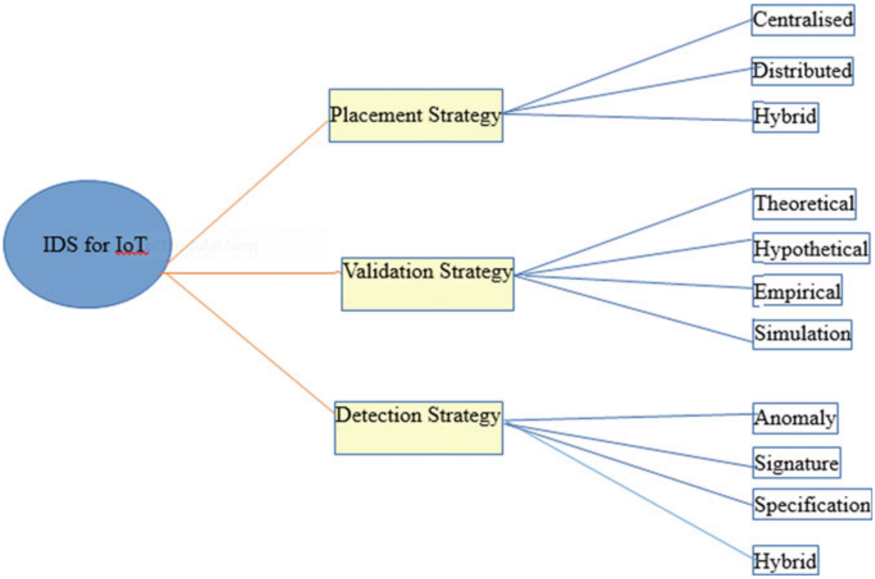


Fig. 4 Intrusion detection system in IoT environment

The Internet of Things (IoT), which is still extremely new, has already caught the attention of a variety of industries, including healthcare, logistics monitoring, smart cities, even autos. As a framework, it is vulnerable to a variety of serious infiltration problems. This study analyses IoT security problems and various methods assessed for its potential to defeat Distributed Denial of Service (DDoS) attacks using internet packet traces. The taxonomy of risky and safe behaviours in the IoT network is the main topic of this study.

5.1 Categories of IoT Threats

There are different categories of IoT threats:

1. Denial of Service (DoS)—By injecting unwanted or worthless traffic, this threat prevents or limits users' network access.
2. Malware—To meddle with IoT network devices, attackers employ software files. They might get illegal access to equipment or gather sensitive data. The attacker can wreak havoc on the IoT architecture by using the devices' software and exposing vulnerabilities in their firmware.
3. Data breaches—a data breach happens whenever private, protected, or secret data is removed from a network. By faking ARP packets, attackers can eavesdrop on peer-to-peer network conversations.
4. Weakening Parameters—IoT network devices are yet to be designed with pervasiveness in mind.

An IDS is a network security device that detects threats and monitors packets. The IDS's functions include providing threat information, taking corrective action when threats are detected and keeping track of all network occurrences.

There are two kinds of intrusion detection systems available as mentioned in Hodo et al. [19] (Table 3):

Table 3 Comparison of HIDS and NIDS performance

Factor	NIDS	HIDS
Intruder redressal	Strong Redressal for outside intruders	Strong Redressal for inside intruders
Evaluation of damage	Very weak	Excellent
Intruder prevention	Good for preventing **outside** intruders	Good for preventing **inside** intruders
Threat response time	Strong response time against outside intruders	Weak response time against outside intruders
Threat anticipation	Good at trending as well as detecting doubtful behaviour pattern	Good at trending as well as detecting doubtful behaviour pattern

- Host-Based Intrusion Detection Systems (HIDS)—These are software-based computer devices that are deployed on a host computer and supervise every operating system, system application and traffic behaviour.
- Network-Based IDS (NIDS)—these are used to capture and analyse data across a whole network.

IoTs are internet-connected gadgets that connect real-world objects with the internet in a number of contexts, such as business, home security and transportation. Users may not like IoT device vulnerabilities or unpredictable faults, which can result in a number of unanticipated outcomes. In IoT devices for vehicles or implanted medical equipment, invasions of privacy like Eavesdropping, Denial of Service attacks or ransomware may result in financial loss and, in severe cases, even death. There are numerous NIDS methods for IoTs that employ attack signatures, but according to Tabassum et al. [20], unknown attacks were not discovered, and False Positive rates are excessive (false alarms). Any component, such as a computer, node or router, can be connected to a host-based IDS to analyse network traffic. HIDS monitors modifications to the file system, system calls and operating system operations, in contrast to NIDS.

5.2 Survey of Intrusion Detection Methods and Datasets

These studies grouped intrusion detection tactics according to technological principles, listed their advantages and disadvantages, however, they did not provide any replicable study ideas or methodology. Further research is made challenging by this. Furthermore, these studies don't go into enough detail to explain intrusion detection techniques. In-depth research has been done in a number of intrusion detection-related fields, incorporating the pre-processing and analytical models along with assessment methodologies. These studies are current in terms of intrusion detection research for a specific target network. Their research is more comprehensive than these because it covers the Internet of Things (IoT), Software Defined Networks (SDN), including Industrial Control Networks (ICN). For the benefit of academics, we also investigate datasets from other fields. The studies described above have also been separated into several categories. The classification of these studies using the following criteria yields the findings shown in Table 4. Many different intrusion detection methods have been suggested. According to the Systematic Literature Review (SLR) technique, we have created a study procedure. This comprises selecting studies, identifying research questions, selecting studies, extracting data and synthesizing data. To better visually express the aforementioned needs, the method is comprised of mixed approaches (qualitative as well as quantitative research techniques).

- Technique: It is stated whether or not the studies are in accordance with SLR methodology.

Table 4 An organized literature review on Intrusion Detection based for IoT-based smart cities

Authors	SLR methodology	Pre-processing	Model	Evaluation	Multifield	Datasets
Ahmed et al. [21]	No	No	Yes	No	No	Yes
Antonia et al. [22]	No	No	Yes	Yes	No	Yes
Bhuyan et al. [23]	No	Yes	Yes	Yes	No	Yes
Buczek and Guven [24]	No	No	Yes	Yes	Yes	Yes
Hande et al. [25]	No	No	Yes	no	No	No
Haq et al. [26]	No	Yes	Yes	Yes	No	Yes
Hodo et al. [27]	No	No	Yes	Yes	No	No
Milenkoski et al. [28]	No	No	No	Yes	No	No
Mishra et al. [29]	No	Yes	Yes	No	No	No
Ring et al. [30]	No	No	No	No	No	Yes
Thakkar and Lohiya [31]	No	No	No	No	No	Yes
Wang and Jones [32]	No	No	Yes	Yes	No	No
Yang et al. [33]	Yes	Yes	Yes	Yes	Yes	Yes
Zarpelao et al. [18]	No	No	Yes	No	Yes	No

- Intrusion detection method: This field describes whether the work mentions intrusion detection methods in addition to pre-processing techniques, analytical models as well as evaluation procedures.
- Multi-field: If the research assessed how intrusion detection systems are currently performing in various network settings.
- Dataset: Whether the study refers to the relevant studies in the dataset is indicated.

6 Conclusion and Future Work

The Internet of Things (IoTs) can be viewed as a group of Internet-connected devices that link real-world objects to the internet in a number of contexts, such as business, home automation, healthcare and environmental monitoring. IoTs improve human operations by rendering routine activities easier, but they also present significant security dangers. Due to the potential vulnerability of IoTs as targets for hackers, businesses are spending billions of dollars to develop a viable system to detect this kind of fraudulent activity in IoT networks. With the use of clever mechanisms including Machine Learning (ML) and Artificial Intelligence (AI), innovative attacks can now be prevented and recognized with the highest degree of precision. The existing Intrusion Detection methods for IoT networks are categorized and described in this paper, with a focus on hybrid and intelligent methods. The advantages and disadvantages of the revealed attack types, placement strategies and current IDS techniques are listed in this study. In particular, for IoTs, real-time validation of the machine learning-based Detection approaches published in the literature to date has not been done. These sophisticated IDS are primarily designed for wireless sensor networks. To create an IDS mechanism that meets the objectives of IPv6-connected IoTs while remaining effective in a limited context, artificial intelligence and deep learning techniques are needed. It has been planned to develop the intrusion detection framework for different IoT networks in the future that is independent of network topology, protocols and existing or undiscovered threats. An intelligent detection mechanism that detects malicious activities without increasing node overhead by secretly analysing network traffic. It also offers mitigating techniques when an intrusion is found. The suggested mechanism must show signs of adaptability to shifting network topologies, threat environments and other networks.

References

1. Weber RH (2010) Internet of things-new security and privacy challenges. Comput Law Secur Rev 26(1):23–30
2. Malliga S, Nandhini PS, Kogilavani SV (2022) A comprehensive review of deep learning techniques for the detection of (distributed) denial of service attacks. Inf Technol Control 51(1):180–215
3. Maselli G, Piva M, Restuccia F (2020) HyBloSE: hybrid block chain for secure-by-design smart environments. In: Proceedings of the 3rd workshop on cryptocurrencies and block chains for distributed systems (pp 23–28)
4. Khan Z, Pervez Z, Ghafoor A (2014) Towards cloud based smart cities data security and privacy management. In: 2014 IEEE/ACM 7th International conference on utility and cloud computing. IEEE, pp 806–811
5. Dorri A, Kanhere SS, Jurdak R, Gauravaram P (2017) Block chain for IoT security and privacy: the case study of a smart home. In: 2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops). IEEE, pp 618–623

6. Waheed N, He X, Ikram M, Usman M, Hashmi SS, Usman M (2020) Security and privacy in IoT using machine learning and blockchain: threats and countermeasures. ACM Comput Surv 53(6), Article 122, 37 p. https://doi.org/10.1145/3417987
7. Ling Z, Liu K, Xu Y, Jin Y, Fu X (2017) An end-to-end view of IoT security and privacy. In: GLOBECOM 2017—2017 IEEE global communications conference, pp 1–7. https://doi.org/10.1109/GLOCOM.2017.8254011
8. Shakarami M, Benson J, Sandhu R (2022) Blockchain-based administration of access in smart home IoT. In: Proceedings of the 2022 ACM workshop on secure and trustworthy cyber-physical systems, pp 57–66
9. Barhamgi M, Yang M, Yu CM, Yu Y, Bandara AK, Benslimane D, Nuseibeh B (2017) Enabling end-users to protect their privacy. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security, pp 905–907
10. Karie NM, Sahri NM, Yang W, Valli C, Kebande VR (2021) A review of security standards and frameworks for IoT-based smart environments. IEEE Access
11. Kulyk O, Reinheimer B, Aldag L, Mayer P, Gerber N, Volkamer M (2020) Security and privacy awareness in smart environments–a cross-country investigation. In: International conference on financial cryptography and data security. Springer, Cham, pp 84–101
12. Hajjaji Y, Boulila W, Farah IR, Romdhani I, Hussain A (2021) Big data and IoT-based applications in smart environments: a systematic review. Comput Sci Rev 39:100318
13. Salim MM, Rathore S, Park JH (2020) Distributed denial of service attacks and its defenses in IoT: a survey. J Supercomput 76(7):5320–5363
14. Elrawy MF, Awad AI, Hamed HF (2018) Intrusion detection systems for IoT-based smart environments: a survey. J Cloud Comput 7(1):1–20
15. Granjal J, Monteiro E, Silva JS (2015) Security for the internet of things: a survey of existing protocols and open research issues. IEEE Commun Surv Tutor 17(3):1294–1312
16. Ahmed E, Yaqoob I, Gani A, Imran M, Guizani M (2016) Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges. IEEE Wirel Commun 23(5):10–16. https://doi.org/10.1109/MWC.2016.772
17. Smys S, Basar A, Wang H (2020) Hybrid intrusion detection system for internet of things (IoT). J ISMAC 2(04):190–199
18. Zarpelão BB, Miani RS, Kawakani CT, de Alvarenga SC (2017) A survey of intrusion detection in Internet of Things. J Netw Comput Appl 84:25–37
19. Hodo E, Bellekens X, Hamilton A, Dubouilh PL, Iorkyase E, Tachtatzis C, Atkinson R (2016) Threat analysis of IoT networks using artificial neural network intrusion detection system. In: 2016 International symposium on networks, computers and communications (ISNCC). IEEE, pp 1–6
20. Tabassum A, Erbad A, Guizani M (2019) A survey on recent approaches in intrusion detection system in IoTs. In: 2019 15th international wireless communications & mobile computing conference (IWCMC). IEEE, pp 1190–1197
21. Ahmed M, Mahmood AN, Hu J (2016) A survey of network anomaly detection techniques. J Netw Comput Appl 60:19–31
22. Nisioti A, Mylonas A, Yoo PD, Katos V (2018) From intrusion detection to attacker attribution: a comprehensive survey of unsupervised methods. IEEE Commun Surv Tutor 20(4):3369–3388
23. Bhuyan MH, Bhattacharyya DK, Kalita JK (2013) Network anomaly detection: methods, systems and tools. IEEE Commun Surv Tutor 16(1):303–336
24. Buczak AL, Guven E (2015) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutor 18(2):1153–1176
25. Hande Y, Muddana A (2021) A survey on intrusion detection system for software defined networks (SDN). In: Research anthology on artificial intelligence applications in security. IGI Global, pp 467–489
26. Haq NF, Onik AR, Hridoy MAK, Rafni M, Shah FM, Farid DM (2015) Application of machine learning approaches in intrusion detection system: a survey. IJARAI-Int J Adv Res Artif Intell 4(3):9–18

27. Hodo E, Bellekens X, Hamilton A, Tachtatzis C, Atkinson R (2017) Shallow and deep networks intrusion detection system: a taxonomy and survey. ArXiv preprint arXiv 1701:02145
28. Milenkoski A, Vieira M, Kounev S, Avritzer A, Payne BD (2015) Evaluating computer intrusion detection systems: a survey of common practices. ACM Comput Surv (CSUR) 48(1):1–41
29. Mishra P, Varadharajan V, Tupakula U, Pilli ES (2018) A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Commun Surv Tutor 21(1):686–728
30. Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A (2019) A survey of network-based intrusion detection data sets. Comput Secur 86:147–167
31. Thakkar A, Lohiya R (2020) A review of the advancement in intrusion detection datasets. Proc Comput Sci 167:636–645
32. Wang L, Jones R (2017) Big data analytics for network intrusion detection: a survey. Int J Netw Commun 7(1):24–31
33. Yang Z, Liu X, Li T, Wu D, Wang J, Zhao Y, Han H (2022) A systematic literature review of methods and datasets for anomaly-based network intrusion detection. Comput Secur 102675
34. Singh KJ, Thongam K, De T (2018) Detection and differentiation of application layer DDoS attack from flash events using fuzzy-GA computation. IET Inf Secur 12(6):502–512
35. Johnson Singh K, Thongam K, De T (2016) Entropy-based application layer DDoS attack detection using artificial neural networks. Entropy 18(10):350
36. Meitei IL, Singh KJ, De T (2016) Detection of DDoS DNS amplification attack using classification algorithm. In: Proceedings of the international conference on informatics and analytics, pp 1–6
37. Specht S, Lee R (2003) Taxonomies of distributed denial of service networks, attacks, tools and countermeasures. In: CEL2003–03. Princeton University, Princeton, NJ, USA
38. Sonar K, Upadhyay H (2014) A survey: DDOS attack on Internet of Things. Int J Eng Res Dev 10(11):58–63
39. Zargar ST, Joshi J, Tipper D (2013) A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Commun Surv Tutor 15(4):2046–2069
40. Zhang C, Green R (2015) Communication security in internet of thing: preventive measure and avoid DDoS attack over IoT network. In: Proceedings of the 18th symposium on communications and networking, pp 8–15
41. Abdul-Ghani HA, Konstantas D, Mahyoub M (2018) A comprehensive IoT attacks survey based on a building-blocked reference model. Int J Adv Comput Sci Appl 9(3)
42. Yang Y, Wu L, Yin G, Li L, Zhao H (2017) A survey on security and privacy issues in Internet-of-Things. IEEE Internet Things J 4(5):1250–1258
43. Vishwakarma R, Jain AK (2020) A survey of DDoS attacking techniques and defence mechanisms in the IoT network. Telecommun Syst 73(1):3–25

Mitigation of Distributed Denial of Service (DDoS) Attack Using Network Function Virtualization (NFV)—A Survey

Gajanan N. Tikhe and Pushpinder Singh Patheja

Abstract One of the most damaging and widely used cyber-attacks is the Distributed Denial of Service (DDoS) attack. A large amount of attack traffic generates traffic congestion and disables online services by disrupting ISP services. DDoS attacks are extremely difficult to detect and mitigate. DDoS attacks are launched using a variety of strategies and approaches, including IP spoofing, botnets and tools. Traditionally hardware, i.e. middle boxes such as Routers, firewall, load balancer and IDS are used to reduce the traffic volume of the DDoS attack. But with the origination of the Network Function Virtualization researchers started to propose various techniques for the mitigation of the volume of the DDoS attack. In this paper, we have reviewed various proposed techniques to defeat DDoS attacks and their efficiency in reducing the attack.

Keywords Distributed denial of service (DDoS) attack · Network function virtualization (NFV) · Virtualized network function (VNF)

1 Introduction

DDoS (Distributed Denial of Service) is one of the most damaging and widely employed cyber-attacks [3]. A large volume of attack traffic generates traffic congestion and disrupts online services by causing ISP services to go down. Any organization is at risk from these types of attacks. Small and medium-sized businesses lack the resources to handle significant traffic volumes. DDoS attacks are getting more powerful and frequent, according to several recent studies [5].

To deal with DDoS attacks, enough resources are essential to prevent excessive traffic caused by DDoS attacks. SYN Flood assaults, for example, can be handled

G. N. Tikhe (✉) · P. S. Patheja
SCSE, VIT Bhopal University, Bhopal, Madhyapradesh, India
e-mail: gajanan.tikhe@gmail.com

P. S. Patheja
e-mail: pspatheja@gmail.com

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_22

by middleboxes such as dedicated high-configuration firewalls, load balancers and so on. However, the middlebox solution is not cost-effective because it requires expensive devices that are not affordable to any small, medium sized organization. If organization redirects its traffic to third-party service solution then privacy is concern. The origination of Network Function Virtualization (NFV) technology [6] can give any organization secure and cost-effective solutions to mitigate the DDoS attack traffic.

IP spoofing and true source IP attack are two approaches to carry out a DDoS attack. The attack is carried out using a false or duplicate IP address in IP spoofing. To carry out a real IP attack, a compromised node known as a botnet is used.

2 A Brief Overview of DDoS Attack and NFV

A. DDoS attack is the most dangerous attack causing the service unavailable to the legitimate users by overloading the servers with huge volume of request. To send the huge volume of request attacker makes use of the IP spoofing, botnets, etc. Distributed Denial of Service (DDoS) attacks can hamper the working of ISPs and online services. It is very dangerous to the organization especially for small and medium-sized organizations with very limited resources. It can consume all the resources and bandwidth of the victim within the second.

Traditionally, DDoS attack is handled by using various middle boxes such as Firewalls, Routers, Switch Intrusion Detection System and Load Balancer but all these devices have limited capacity which is not scalable. With the invent of the Network Function Virtualization we can leverage the various features of NFV such as scalability, flexibility, automation, on the fly deployment and off the shelf server.

B. Network Function Virtualization uses the generalized off the shelf server to implement the Virtualized network functions without using the specialized hardware to handle DDoS attack [20]. Traditionally specialized hardware such as Load Balancer, Firewall, DNS server and Routers are used to handle the DDoS attack but Network Function Virtualization technique uses the off the shelf server to implement the functionality of the specialized hardware that means, function performed by specialized hardware will be implemented by Virtualised Network Functions (VNF).

Using Network Function Virtualization has number of benefits as follows:

1. It uses general purpose server instead of specialized hardware [12].
2. It reduces the expenditure towards hardware components, i.e. CapEx [16].
3. It also reduces the installation and maintenance cost, i.e. OpEX [13].
4. Deployment of flexible network functions depending on the need [14, 15].
5. It reduces the middle boxes [27].
6. Reduces the power consumption [10, 26].
7. Dynamic scaling of the network performance is possible [24, 25].
8. Agility to adapt to market [22, 23].

3 DDoS Attack Challenges

A. Number of strategies for DDoS mitigation have been proposed but very few are applied to the real network environment [2]. Designing and implementing the ideal solution for the mitigation of the DDoS attack is really difficult [7]. The main challenges that must be addressed by any DDoS defence scheme are

(i) Real time mitigation with minimum computation requirement [8].
(ii) Mitigation scheme should be scalable to work with high volume of traffic [11].
(iii) Scheme must handle the unknown DDoS attack [17].
(iv) It must not slow down the processing of the packets coming from legitimate users [18].
(v) Detection method should depend on the minimum number of traffic parameter [19].

B. DDoS attack mitigation approaches categorization

DDoS is very commonly occurring attacks. Numbers of solution are available to mitigate this attack. Solutions can be categorized into three categories as follows [3].

(A) *Traditional DDoS Mitigation*

In these techniques various specialized hardware such as Load Balancer, Intrusion Detection System (IDS) and Firewall are used to mitigate the attack. Organization uses these hardwares to prevent attacks and threats at their premises and datacenters. This will work well if the traffic is not the beyond the capacity of devices used for prevention of the attacks.

(B) *Cloud-Based DDoS Protection Service*

any organization offers DDoS protections through clouds and datacentres due to limitation in capacity of the hardware devices. Customers need to divert their incoming traffic to datacentres available at remote location to detect and mitigate the DDoS attacks. This increases the latency in response of the traffic and also violates the privacy of the customer, because these datacentres or security service companies may observe the behaviours of the traffic.

(C) *NFV-Based Mitigation*

In this approach Network functions and Network Security functions are deployed as software instead of hardware. This will provide all the benefits of the Networks Function Virtualization. Also these functions will be deployed at the customer premises in order to enhance privacy and reduce the latency.

4 DDoS Solutions Using NFV

In *VGuard* [1], VNF is designed, in which external flow is diverted to tunnel based on the priority levels. Legitimate flow is served with all the guaranteed quality of services. Attack/suspicious flow need to compete for the resources. Simulation shows that Satisfying services will be provided to the legitimate flow. It gives static method and Dynamic method for the flow direction. VGuard cannot provide the protection against all types of the DDoS attacks.

In *VFenceiz* [27], for the SYN flood, the authors proposed a DDoS mitigation strategy. Because attack strength varies, the NFV-based defence framework constantly deploys agents to balance the assault load. Numbers of the agents can be increased and decreased depending on the flow of the incoming traffic. This defence solution leverages the NFV features for the implementation of network functions on the fly as per the need. This defence solution creates filtering agents which filters the external DDoS traffic. Load balancer diverts the external traffic to the one of the agents. DDoS filtering Agents works on the traffic and forwards the legitimate traffic to the destination. Evaluation results shows that this method reduces the traffic occurred due to the SYN flood attack. The drawback of this defence solution is that it cannot handle the application layer attack as well as it introduces some extra delay because of the handshaking.

Cofence [9] is another defence solution for the SYN flood attacks. It mitigates the DDoS attack by redirecting the excessive traffic to other domains. Other domain will filter out the traffic and report to the requesting collaborative domain. Cofence also proposes the resource allocation mechanism which limits the allocation of the resources by the domain to the each requestor. Simulation shows that the proposed solution can mitigate impact of the DDoS attack. Drawback of this domain solution is that it can only address SYN flood attack and there is privacy concern as the traffic is diverted to other domains.

Xfirewall [4] is temporary firewall, it will be created only when attack occurs with dynamic configuration rules based on real time traffic analysis. XFirewall protects networks or server against DDoS attack. XFirewall leverages the automation and scalability features of the NFV.

In XFirewall, policy rules change dynamically depending on the traffic flow. Xfirewall will be created by NFV orchestrator only when there is flood of incoming traffic noticed by screener and will be placed at the required locations. This paper does not discuss the algorithm to differentiate the malign traffic and genuine traffic.

Applying NFV/SDN in mitigating DDoS attacks [6]—This study provides a DDoS mitigation architecture that monitors and analyses network traffic using Network Function Virtualization (NFV) and Software Defined Networking (SDN). Framework makes use of the data plane, control plane and application plane of the SDN. The Anamoly Detection module in the Control plane collects data and analyses it before invoking the appropriate DDoS attack mitigation technique in the Application Plane.

The NFV modules manage the virtual machines for each DDoS attack mitigation requirement. It initiates, allocates and instantiates the virtual machine as per the requirement of DDoS attack. Allocated virtual and physical resources are implemented and enabled by SDN module and SDN controller.

Authors applied their framework for the mitigation of DDoS attack in the industrial control system.

Holistic DDoS mitigation using NFV [3]—Paper proposes the two stage DDoS mitigation framework, First stage is screening mechanism and second stage is resource allocation.

It first scans the traffic flows and determines next stage process such as application layer security, network layer security or certain network process needed for traffic flow. Traffic screener screens the incoming traffic using the algorithm, policies and packet features and identifies whether the traffic is legitimate or malicious.

If there is flood of legitimate traffic, the screener makes request to resource allocation module to scale up the resources. The scheme is proposed to deploy at organization's premises to reduce latency and improve privacy.

SDN/NFV-based DDoS Mitigation via *Pushback* [28]—This paper proposes SDN/NFV-based collaborative DDoS attack mitigation scheme. An attribute-Oriented Induction (AOI) algorithm of Machine learning is employed to derive DDoS attack pattern. This mitigation approach is two-phased, first it tries to handle the attack locally. Otherwise pushback is initiated if there is huge flow of DDoS attack traffic. NFV performs pushback of the attack in non-SDN domains. This framework filters the attack close to the source of attack and tries to clearly differentiate between attack traffic and legitimate traffic.

Smart and Lightweight DDoS Detection Using NFV [2]—This paper proposes a mitigation technique which detects all DDoS attack and performs deep inspection to differentiate legitimate traffic from the attack traffic. The proposed mechanism leverages the scalability and automation features of the NFV paradigm. Proposed detection performs fast screening and deep screening. Fast screening detects the abnormal traffic and alerts the deep screening module. Fast screening module cooperates with Resource Allocation Protocol (RAP) to provide the resources to each service. Deep screening module will be invoked to perform deep investigation to detect the attack's parameter.

ARDefense [20]—This paper proposes the solution for individual users especially for the gamers and streamers. It can be also used for online services like website, etc. This scheme performs the server migration and IP spoofing whenever DDoS attacks occur at the application layer.

DeMONS [11] To counteract DDoS attacks, this study presents a hybrid method. A priority classifier, a firewall, an allocation module, a traffic policing module and a manager are the five modules that make up the solution. A priority classifier module examines network traffic and assigns a priority value between 0 and 1. The firewall module blocks zero-priority traffic, while the allocation module assigns the remaining traffic to a separate tunnel. If the low priority flow tunnel is overburdened, the traffic policing module uses an algorithm to limit each flow according to its priority.

5 Conclusions

In this paper, we have discussed the severity of the DDoS attack. Then we discussed the advantages of using the Network Function Virtualization technology for implementing the Network Functions. We also discussed DDoS attack mitigation approaches categorization. NFV is better than all the existing traditional approaches.

Then we discussed the various approaches to handle the traffic volume of DDoS attack. Some of the strategies are handling only SYN attack and not providing the solutions to all types of attacks occurred at the different layers of the networks. Some have the privacy concern as they need to share the traffic to the cloud. Some of the techniques are using NFV.

References

1. Fung CJ, McCormick B (2015) VGuard: a distributed denial of service attack mitigation method using network function virtualization. In: 2015 11th International conference on network and service management (CNSM), Barcelona, pp 64–70. https://doi.org/10.1109/CNSM.2015.7367340
2. Alharbi T, Aljuhani A, Liu H, Hu C (2017) Smart and lightweight DDoS detection using NFV. In: Proceedings of the international conference on compute and data analysis (ICCDA '17). Association for Computing Machinery, New York, NY, USA, pp 220–227. https://doi.org/10.1145/3093241.3093253
3. Alharbi A, Aljuhani A, Liu H (2017) Holistic DDoS mitigation using NFV. In: 2017 IEEE 7th annual computing and communication workshop and conference (CCWC), Las Vegas, NV, pp 1–4. https://doi.org/10.1109/CCWC.2017.7868480
4. Aljuhani A, Alharbi T, Liu H (2017) XFirewall: a dynamic and additional mitigation against DDoS storm. In: Proceedings of the international conference on compute and data analysis (ICCDA '17). ACM, New York, NY, USA, pp 1–5. https://doi.org/10.1145/3093241.3093252
5. Guizani N, Ghafoor A (2020) A network function virtualization system for detecting malware in large IoT based networks. IEEE J Sel Areas Commun 38(6):1218–1228. Network Functions Virtualisation—Introductory White Paper. http://portal.etsi.org/NFV/NFV_White_Paper.pdf
6. Zhou L, Guo H (2017) Applying NFV/SDN in mitigating DDoS attacks. In: TENCON 2017—2017 IEEE region 10 conference, Penang, pp 2061–2066. https://doi.org/10.1109/TENCON.2017.8228200
7. Mijumbi R, Serrat J, Gorricho J-L, Bouten N, De Turck F, Boutaba R (2015) Network function virtualization: state-of-the-art and research challenges. IEEE Commun Surv Tutor
8. Rashidi B, Fung C, Bertino E (2017) A collaborative DDoS defence framework using network function virtualization. IEEE Trans Inf Forensics Secur 12(10):2483–2497. https://doi.org/10.1109/TIFS.2017.2708693
9. Rashidi B, Fung C (2016) CoFence: a collaborative DDoS defence using network function virtualization. In: 2016 12th International conference on network and service management (CNSM), Montreal, QC, pp 160–166. https://doi.org/10.1109/CNSM.2016.7818412
10. Alwakeel AM, Alnaim AK, Fernandez EB (2018) A survey of network function virtualization security. SoutheastCon 2018, St. Petersburg, FL, pp 1–8. https://doi.org/10.1109/SECON.2018.8479121
11. Fülber Garcia V, de Freitas Gaiardo G, da Cruz Marcuzzo L, Ceretta Nunes R, Paula dos Santos CR (2018) DeMONS: a DDoS mitigation NFV solution. In: 2018 IEEE 32nd International conference on advanced information networking and applications (AINA), Krakow, pp 769–776. https://doi.org/10.1109/AINA.2018.00115

12. Li W, Meng W, Kwok LF (2021) Surveying trust-based collaborative intrusion detection: state-of-the-art, challenges and future directions. IEEE Commun Surv Tutor 24(1):280–305
13. Hawilo H, Jammal M, Shami A (2017) Orchestrating network function virtualization platform: migration or re-instantiation? In: 2017 IEEE 6th International conference on cloud networking (CloudNet), Prague, pp 1–6. https://doi.org/10.1109/CloudNet.2017.8071528
14. Chatras (2018) Applying a service-based architecture design style to network functions virtualization. In: 2018 IEEE Conference on standards for communications and networking (CSCN), Paris, pp 1–4. https://doi.org/10.1109/CSCN.2018.8581751
15. Riggio R, Bradai A, Harutyunyan D, Rasheed T, Ahmed T (2016) Scheduling wireless virtual networks functions. IEEE Trans Netw Serv Manage 13(2):240–252. https://doi.org/10.1109/TNSM.2016.2549563
16. Bhosale KS, Nenova M, Iliev G (2017) The distributed denial of service attacks (DDoS) prevention mechanisms on application layer. In: 2017 13th International conference on advanced technologies, systems and services in telecommunications (TELSIKS), Nis, pp 136–139. https://doi.org/10.1109/TELSKS.2017.8246247
17. Nagesh HR, Sekaran KC (2006) Design and development of proactive solutions for mitigating denial-of-service attacks. In 2006 International conference on advanced computing and communications, Surathkal, pp 157–162. https://doi.org/10.1109/ADCOM.2006.4289874
18. Yogesh Patil R, Ragha L (2011) A rate limiting mechanism for defending against flooding based distributed denial of service attack. In: 2011 World congress on information and communication technologies, Mumbai, pp 182–186. https://doi.org/10.1109/WICT.2011.6141240
19. Grant C (2018) Distributed detection and response for the mitigation of distributed denial of service attacks. In: 2018 International conference on information networking (ICOIN), Chiang Mai, pp 495–497. https://doi.org/10.1109/ICOIN.2018.8343168
20. Singh AK, Jaiswal RK, Abdukodir K, Muthanna A (2020) ARDefense: DDoS detection and prevention using NFV and SDN. In: 2020 12th International congress on ultra modern telecommunications and control systems and workshops (ICUMT), Brno, Czech Republic, pp 236–241. https://doi.org/10.1109/ICUMT51630.2020.9222443
21. Bhuyan MH, Kashyap HJ, Bhattacharyya DK, Kalita JK (2014) Detecting distributed denial of service attacks: methods, tools and future directions. Comput J 57(4):537–556. https://doi.org/10.1093/comjnl/bxt031
22. Jin Y, Wen Y (2017) When cloud media meet network function virtualization: challenges and applications. IEEE Multimedia 24(3):72–82. https://doi.org/10.1109/MMUL.2017.3051519
23. Li Y, Chen M (2015) Software-defined network function virtualization: a survey. IEEE Access 3:2542–2553. https://doi.org/10.1109/ACCESS.2015.2499271
24. Chatras B, Ozog FF (2016) Network functions virtualization: the portability challenge. IEEE Netw 30(4):4–8. https://doi.org/10.1109/MNET.2016.7513857
25. Kim S, Kim HS (2017) A high available service based on virtualization technology in NFV. In: 2017 International conference on information networking (ICOIN), Da Nang, pp 649–652. https://doi.org/10.1109/ICOIN.2017.7899578
26. Vilalta R et al (2015) Transport network function virtualization. J Lightwave Technol 33(8):1557–1564. https://doi.org/10.1109/JLT.2015.2390655
27. Jakaria AHM, Yang W, Rashidi B, Fung C, Rahman MA (2016) VFence: a defense against distributed denial of service attacks using network function virtualization. In: 2016 IEEE 40th annual computer software and applications conference (COMPSAC), Atlanta, GA, pp 431–436. https://doi.org/10.1109/COMPSAC.2016.219
28. Bülbül NS, Fischer M (2020) SDN/NFV-based DDoS mitigation via pushback. In: ICC 2020—2020 IEEE International conference on communications (ICC), Dublin, Ireland, pp 1–6. https://doi.org/10.1109/ICC40277.2020.9148717

A Secured Framework for Emergency Care in the E-Healthcare System

Aman Ahmad Ansari, Bharavi Mishra, and Poonam Gera

Abstract The e-healthcare systems provide many services to the patients and the users, including patient health record management, remote patient monitoring, and emergency response system. Nowadays, many e-healthcare systems include emergency care, but the services can only be available to registered patients. Ansari et al. recently proposed a secure and privacy-preserving framework for e-healthcare systems. The emergency phase of the framework provides emergency care only to patients with Wireless Body Area Network (WBAN). This paper extends the above framework to include unregistered patients for emergency care and registered patients without WBAN. The proposed scheme uses hash functions and elliptic curve cryptography (ECC) to provide mutual authentication and security, and it is evaluated using the AVISPA tool and traditional security analysis. The result of security analysis and simulation results demonstrate that the suggested scheme is protected from all known attacks.

Keywords e-healthcare system · Emergency care · Authentication protocol · Security · Privacy

1 Introduction

The e-healthcare system offers many services to patients and the users, like remote patient monitoring, elderly support, patient health record management, and an emergency response system. Many e-healthcare systems adopted cloud architecture to

A. A. Ansari (✉) · B. Mishra · P. Gera
Department of Computer Science and Engineering, LNM Institute of Information Technology, Jaipur, Rajasthan, India
e-mail: 16pcs001@lnmiit.ac.in

B. Mishra
e-mail: bharavi@lnmiit.ac.in

P. Gera
e-mail: poonamgera@lnmiit.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_23

provide storage for electronic health records (EHR) and services to patients and users. Adopting the cloud for e-healthcare also helps medical professionals (MPs). EHR stored in the cloud provides patients' health history to the MPs. It also reduces duplicate records and offers remote medical care, reducing expenses and treatment time [1]. However, all the services the e-healthcare systems provide are only for registered patients, even the emergency response system. In case of a medical emergency to an unregistered person (e.g., a friend or family member of a registered patient or an unknown person), the registered patient may not be able to report to the e-healthcare system. Therefore, such functionality can be added to the emergency response system to include the unregistered patient. Providing emergency care to an unregistered patient can raise security and privacy risks to the e-healthcare system and the registered patient who reported the emergency.

In the literature, a variety of e-healthcare solutions are suggested to offer protected and private health support [2–6]. Jiang et al. [4] developed an authentication and key agreement (AKA) scheme for Telecare Medical Information System (TIMS) in 2013, though Kumari et al. [3] have demonstrated that the protocol is insecure to user impersonation attacks and stolen verifier attacks. Chen et al. [7] offered a different framework for TMIS. They claimed that their system was impervious to all known attacks. However, the architecture developed by Chen et al. [7] lacks adequate message authentication and patient confidentiality. An enhanced technique was also proposed by the authors in [8] to address the problems mentioned in [7]. Later, Chiou et al.'s scheme [8] proved unsafe against stolen verifier attacks in [9]. Additionally, Kumari et al. [9] suggested a new scheme for TMIS.

In 2020, an AKA scheme for a cyber-physical system using the cloud was proposed by Challa et al. [9]. Chaudhry et al. [10] have shown that the protocol given by Challa et al. [9] is insecure to replay attacks. Kumari et al. [11] suggested a cloud-assisted secured and privacy-enabled framework using Elliptic Curve Cryptography (ECC). However, it has been proven unsafe by Khan et al. [12]. They also suggested a biometric-based authentication protocol. A secured and privacy-enabled smart medical system is presented in [13] that is proven unsafe against impersonation attacks and known-session-specific temporary information attacks [14]. Recently, Ansari et al. [15] proposed a privacy-preserving secured framework for e-healthcare systems with emergency care. However, the proposed framework in [15] does not consider patients without WBAN and unregistered patients for emergency care. In this paper, we extended the framework and proposed a secure authentication scheme to include patients without WBAN and unregistered patients for emergency care.

The order of the remaining paper is as follows. In Sect. 2, we covered the proposed scheme. Section 3 presents the security analysis of the suggested framework. Performance evaluation of the suggested framework is covered in Sect. 4. Following that, we gave our conclusion. Table 1 includes a list of all the symbols and notations used in this paper.

Table 1 Notation used in this paper

Notation	Description	Notation	Description
TA	Trusted authority	PW	Password
C	Cloud server	BIO	Biometric imprint
PD	Patient with device	PID	Patient's identity
MP	Medical professional with device	APID	Patient's anonymous identity
HC	Healthcare Center	ERID	Patient's identity for emergency response
WBAN	Wireless body area network	R_e	Patient's public parameter for emergency response
P_{pub}/x	TA's public/private key	P_e/d_e	Patient's public/private key for emergency response
CID	C's identity	$TRID$	Temporary random identity for unregistered patient
R_c	C's public parameter	$\|$	Concatenation
P_c/d_c	C's public/private key	$\mathbf{A}$	Attacker
HID	HC's identity	$h(.)$	Secure one-way hash function
R_h	HC's public parameter	$E_k(m)/D_k(m)$	Encryption/decryption of m using key k
R_h/P_h	HC's public/private key	SK	Session key
MID	MP's identity	$\oplus$	Bitwise XOR
AMID	MP's anonymous identity	σ	Auxiliary bit string generated from the biometric imprint
MPID	associated patient's identity with MP	δ	Secret bit string generated from the biometric imprint

2 Proposed Scheme

This section discusses the proposed scheme for emergency care. This work extends the framework presented in Ansari et al. [15] to extend the emergency care facility to unregistered patients. The proposed framework in [15] is composed of five phases, including the emergency phase (EP), check-up phase (CP), treatment phase (TP), healthcare center upload phase (HUP), and patient data upload phase (PUP). In the EP, the authors of [15] only included patients with Wireless Body Area Network (WBAN). The emergency phase is triggered if the gateway at the WBAN detects that health parameters are over the threshold Fig. 1. The proposed scheme covers medical emergency scenarios for registered patients without WBAN and unregistered patients Fig. 2.

In the proposed scheme, the patient registered with the e-healthcare system initiates the scheme by reporting a medical emergency to the nearest healthcare center (HC). The patient device chooses the healthcare center in real time based on the

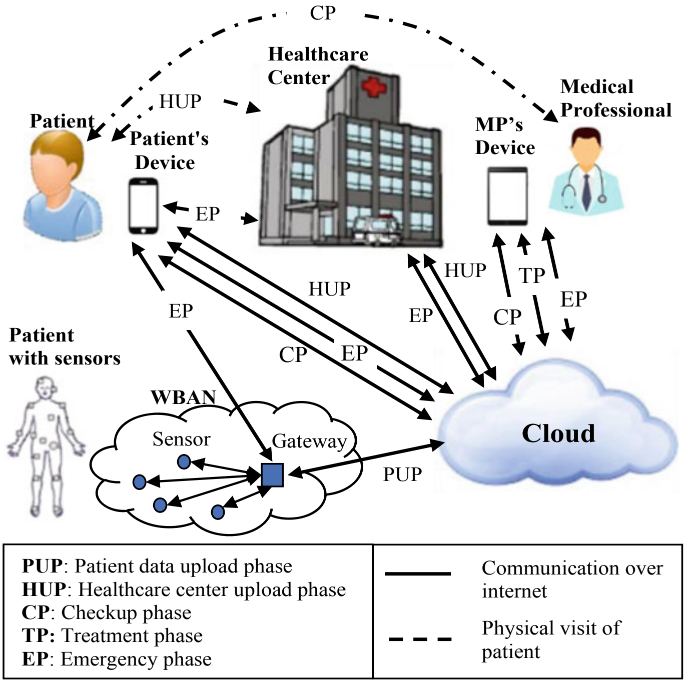


Fig. 1 Architecture of framework proposed by Ansari et al. [15]

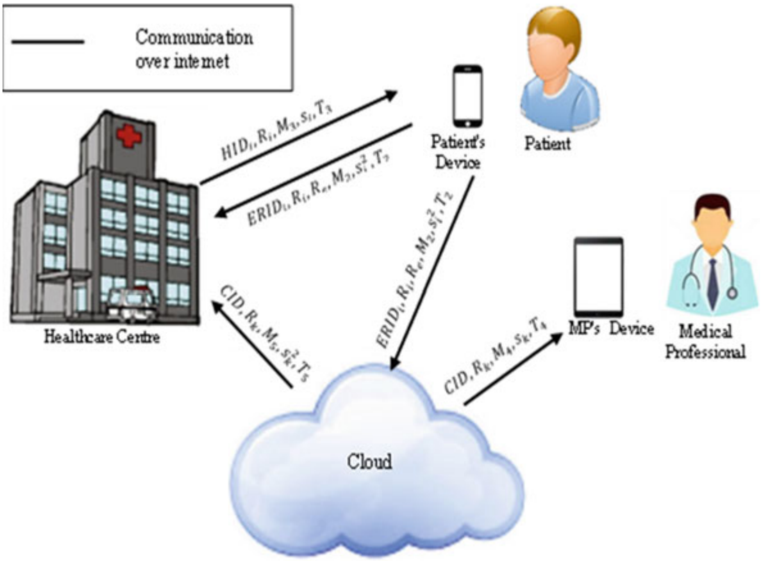

Communication over internet
$HID_i, R_i, M_3, s_i, T_3$
$ERID_i, R_i, R_e, M_2, s_i^2, T_2$
Patient's Device
Patient
$ERID_i, R_i, R_e, M_2, s_i^2, T_2$
Healthcare Centre
$CID, R_k, M_5, s_k^2, T_5$
CID, R_k, M_4, s_k, T_4
MP's Device
Medical Professional
Cloud

Fig. 2 Architecture of proposed scheme

location and stored information about healthcare centers. The patient also informs cloud (C) about the emergency and points out that emergency care is required by him or someone else. The message to the cloud server also includes the chosen healthcare center's identity. After getting the information from the patient, the healthcare center prepares for emergency care and dispatches the ambulance if required by the patient. At the same time, the cloud server prepares the anonymous data of the patient if the registered patient himself requires emergency care, sends the anonymous health data to the HC, and informs the patient's doctor about the emergency.

On the other hand, if emergency care is required for an unregistered patient, the cloud server will generate a temporary random identity (TRID) for the unregistered patient, associate a storage space with the ID, and send TRID to the healthcare center for further communication about the unregistered patient. The proposed scheme is discussed below in detail. The graphical view of the proposed scheme is given in Fig. 3. The registration phase is taken from [15].

2.1 Registration Phase

2.1.1 Cloud Server Registration

Step 1. Cloud chooses CID and $r_c^{'}$ then compute $R_c^{'} = r_c^{'}P$ and send $< CID, R_c^{'} >$ to Trusted Authority (TA) using a secure channel.

Step 2. On getting the cloud's message $< CID, R_c^{'} >$, TA chooses r_c^{ta} and computes $R_c^{\text{ta}} = r_c^{\text{ta}}P$, $R_c = R_c^{'} + R_c^{\text{ta}}$, and $d_c^{\text{ta}} = r_c^{\text{ta}} + h(R_c \| CID)x$, then sends $< d_c^{\text{ta}}, R_c >$ to C through a secure channel.

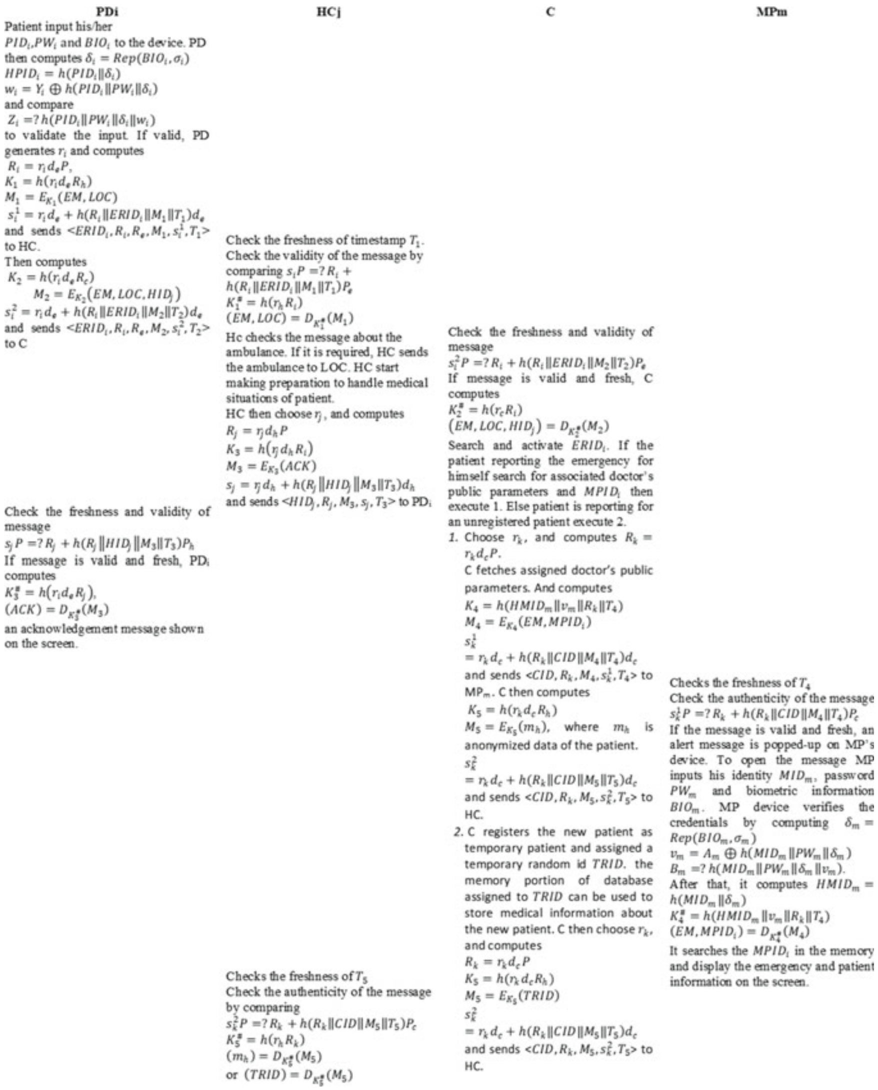
Step 3. On receiving $< d_c^{\text{ta}}, R_c >$ from TA, cloud computes $d_c = d_c^{\text{ta}} + r_c^{'}$ and verifies the private key by comparing $d_cP \overset{?}{=} R_c + h(R_c \| CID)P_{\text{pub}}$. If true, the cloud saves the private key d_c and public parameter R_c for future use.

2.1.2 Medical Professional Registration

Step 1. Initially, MP inputs his chosen identity MID_j and password PW_j with the biometric information BIO_j to the MP device. The device then computes $(\delta_j, \sigma_j) = \text{gen}(\text{BIO}_j)$, $\text{HMID}_j = h(\text{MID}_j \| \delta_j)$ and then sends $< \text{HMID}_j >$ to TA.

Step 2. After receiving $< \text{HMID}_j >$ from MPj, TA generates v_j and computes $\text{AMID}_j = h(\text{HMID}_j \| x \| v_j)$ and then sends $< \text{AMID}_j, v_j, R_c, P_c, \text{CID} >$ to MPj via a secure channel and sends $< \text{HMID}_j, v_j, \text{AMID}_j >$ to C. C will check for collision, encrypt it with C's public key, and store it in the MP_ verifier table.

Step 3. On getting $< AMID_j, v_j, R_c, P_c, CID >$, MPj computes $A_j = h(\text{MID}_j \| \text{PW}_j \| \delta_j) \oplus v_j$, $B_j = h(\text{MID}_j \| \text{PW}_j \| \delta_j \| v_j)$, and stores $\{\text{AMID}_j, A_j, B_j, \sigma_j, \text{CID}, R_c, P_c\}$.



PDi

Patient input his/her PID_i,PW_i and BIO_i to the device. PD then computes $\delta_i = Rep(BIO_i, \sigma_i)$
$HPID_i = h(PID_i \| \delta_i)$
$w_i = Y_i \oplus h(PID_i \| PW_i \| \delta_i)$
and compare
$Z_i =? h(PID_i \| PW_i \| \delta_i \| w_i)$
to validate the input. If valid, PD generates r_i and computes
$R_i = r_i d_e P$,
$K_1 = h(r_i d_e R_h)$
$M_1 = E_{K_1}(EM, LOC)$
$s_i^1 = r_i d_e + h(R_i \| ERID_i \| M_1 \| T_1) d_e$
and sends $<ERID_i, R_i, R_e, M_1, s_i^1, T_1>$ to HC.
Then computes
$K_2 = h(r_i d_e R_c)$
$M_2 = E_{K_2}(EM, LOC, HID_j)$
$s_i^2 = r_i d_e + h(R_i \| ERID_i \| M_2 \| T_2) d_e$
and sends $<ERID_i, R_i, R_e, M_2, s_i^2, T_2>$ to C

Check the freshness and validity of message
$s_j P =? R_j + h(R_j \| HID_j \| M_3 \| T_3) P_h$
If message is valid and fresh, PDi computes
$K_3^{\#} = h(r_i d_e R_j)$,
$(ACK) = D_{K_3^{\#}}(M_3)$
an acknowledgement message shown on the screen.

HCj

Check the freshness of timestamp T_1.
Check the validity of the message by comparing $s_i P =? R_i + h(R_i \| ERID_i \| M_1 \| T_1) P_e$
$K_1^{\#} = h(r_h R_i)$
$(EM, LOC) = D_{K_1^{\#}}(M_1)$
Hc checks the message about the ambulance. If it is required, HC sends the ambulance to LOC. HC start making preparation to handle medical situations of patient.
HC then choose r_j, and computes
$R_j = r_j d_h P$
$K_3 = h(r_j d_h R_i)$
$M_3 = E_{K_3}(ACK)$
$s_j = r_j d_h + h(R_j \| HID_j \| M_3 \| T_3) d_h$
and sends $<HID_j, R_j, M_3, s_j, T_3>$ to PDi

Checks the freshness of T_5
Check the authenticity of the message by comparing
$s_k^2 P =? R_k + h(R_k \| CID \| M_5 \| T_5) P_c$
$K_5^{\#} = h(r_h R_k)$
$(m_h) = D_{K_5^{\#}}(M_5)$
or $(TRID) = D_{K_5^{\#}}(M_5)$

C

Check the freshness and validity of message
$s_i^2 P =? R_i + h(R_i \| ERID_i \| M_2 \| T_2) P_e$
If message is valid and fresh, C computes
$K_2^{\#} = h(r_c R_i)$
$(EM, LOC, HID_j) = D_{K_2^{\#}}(M_2)$
Search and activate $ERID_i$. If the patient reporting the emergency for himself search for associated doctor's public parameters and $MPID_i$ then execute 1. Else patient is reporting for an unregistered patient execute 2.

1. Choose r_k, and computes $R_k = r_k d_c P$.
   C fetches assigned doctor's public parameters. And computes
   $K_4 = h(HMID_m \| v_m \| R_k \| T_4)$
   $M_4 = E_{K_4}(EM, MPID_i)$
   $s_k^1 = r_k d_c + h(R_k \| CID \| M_4 \| T_4) d_c$
   and sends $<CID, R_k, M_4, s_k^1, T_4>$ to MPm. C then computes
   $K_5 = h(r_k d_c R_h)$
   $M_5 = E_{K_5}(m_h)$, where m_h is anonymized data of the patient.
   $s_k^2 = r_k d_c + h(R_k \| CID \| M_5 \| T_5) d_c$
   and sends $<CID, R_k, M_5, s_k^2, T_5>$ to HC.
2. C registers the new patient as temporary patient and assigned a temporary random id $TRID$. the memory portion of database assigned to $TRID$ can be used to store medical information about the new patient. C then choose r_k, and computes
   $R_k = r_k d_c P$
   $K_5 = h(r_k d_c R_h)$
   $M_5 = E_{K_5}(TRID)$
   $s_k^2 = r_k d_c + h(R_k \| CID \| M_5 \| T_5) d_c$
   and sends $<CID, R_k, M_5, s_k^2, T_5>$ to HC.

MPm

Checks the freshness of T_4
Check the authenticity of the message
$s_k^1 P =? R_k + h(R_k \| CID \| M_4 \| T_4) P_c$
If the message is valid and fresh, an alert message is popped-up on MP's device. To open the message MP inputs his identity MID_m, password PW_m and biometric information BIO_m. MP device verifies the credentials by computing $\delta_m = Rep(BIO_m, \sigma_m)$
$v_m = A_m \oplus h(MID_m \| PW_m \| \delta_m)$
$B_m =? h(MID_m \| PW_m \| \delta_m \| v_m)$.
After that, it computes $HMID_m = h(MID_m \| \delta_m)$
$K_4^{\#} = h(HMID_m \| v_m \| R_k \| T_4)$
$(EM, MPID_i) = D_{K_4^{\#}}(M_4)$
It searches the $MPID_i$ in the memory and display the emergency and patient information on the screen.

Fig. 3 Graphical view of login and authentication phase

2.1.3 Healthcare Center Registration

Step 1. HCk chooses HID_k and r_k^h, then computes $R_k^h = r_k^h.P$, and sends $< \text{HID}_k, R_k^h >$ to TA through a secure channel.

Step 2. On getting HCk's message $< \text{HID}_k, R_k^h >$, TA chooses r_k^{ta} and computes $R_k^{\text{ta}} = r_k^{\text{ta}} P$, $R_h = R_k^h + R_k^{\text{ta}}$, and $d_k^{\text{ta}} = r_k^{\text{ta}} + h(R_h \| \text{HID}_k) x$. Then it sends $< d_k^{\text{ta}}, R_h >$ to HCk and sends $< \text{HID}_k, R_h >$ to the cloud via a secure channel. Cloud will check HID_k for collision; if there is no collision, then it stores HID_k in the HC_verifier table.

Step 3. After receiving TA's message $< d_k^{\mathrm{ta}}, R_h >$, HCk computes $d_h = d_k^{\mathrm{ta}} + r_k^h$ then checks if $d_h P =?R_h + h(R_h \| \mathrm{HID}_k) P_{\mathrm{pub}}$ to verify the private key and public parameters. HCk then stores the private key d_h and public parameter R_h.

2.1.4 Patient Registration

Step 1. Patient inputs his chosen identity PID_i, password PW_i with biometric imprint BIO_i to the patient's device. PD_i then generates $(\delta_i, \sigma_i) = \mathrm{gen}(\mathrm{BIO}_i)$ and computes $\mathrm{HPID}_i = h(\mathrm{PID}_i \| \delta_i)$, $\mathrm{HPW}_i = h(\mathrm{PW}_i \| \delta_i)$, and sends $< \mathrm{HPID}_i, \mathrm{HPW}_i >$ to TA through a secure channel.

Step 2. On getting PD_i's message $< \mathrm{HPID}_i, \mathrm{HPW}_i >$, TA generates w_i and computes $APID_i = h(\mathrm{HPID}_i \| x \| w_i)$, $Y_i = h(\mathrm{HPID}_i \| \mathrm{HPW}_i) \oplus w_i$ and then sends $< \mathrm{APID}_i, Y_i, R_c, P_c, \mathrm{CID} >$ to PD_i via a secure channel and $< \mathrm{HPID}_i, w_i, \mathrm{APID}_i >$ to C through a secure channel.

Step 3. The cloud will check for collision and creates a patient record using $\{\mathrm{HPID}_i, w_i, \mathrm{APID}_i\}$, encrypts HPID_i and w_i with C's public key, and stores encrypted HPID_i, w_i with APID_i in the patient verifier table.

Step 4. After receiving $< \mathrm{APID}_i, Y_i, R_c, P_c, \mathrm{CID} >$, PD_i computes $w_i = Y_i \oplus h(\mathrm{HPID}_i \| \mathrm{HPW}_i)$, $Y_i^* = h(\mathrm{PID}_i \| \mathrm{PW}_i \| \delta_i) \oplus w_i$, $Z_i = h(\mathrm{PID}_i \| \mathrm{PW}_i \| \delta_i \| w_i)$, and stores $\{\mathrm{APID}_i, Y_i^*, Z_i, \sigma_i, \mathrm{CID}, R_c, P_c\}$. Then it generates r_i, computes $R_i = r_i P$, $ver_i = h(\mathrm{HPID}_i \| w_i \| R_i \| T_1)$, and then sends $< \mathrm{APID}_i, R_i, ver_i, T_1 >$ to the C.

Step 5. On getting $< \mathrm{APID}_i, R_i, ver_i, T_1 >$ from PDi, C searches for APID_i in the verifier table and verifies the message's authenticity by comparing $ver_i =?h(\mathrm{HPID}_i \| w_i \| R_i \| T_1)$ if T_1 is fresh. After it passes the verification and freshness test, C generates r_j and computes $R_j = r_j P$, $K_1 = h(r_j R_i)$, $\mathrm{ERID}_i = h(\mathrm{HPID}_i \| w_i \| d_c)$, $R_e = R_i + R_j$, $d_j = r_j + h(R_e \| \mathrm{ERID}_i) d_c$, $M_1 = E_{K_1}(\mathrm{ERID}_i, R_e, P_e, d_j)$, and $s_j = r_j + h\big(R_j \| \mathrm{CID} \| M_1 \| T_2\big) d_c$, then sends $< \mathrm{CID}, R_j, s_j, M_1, T_2 >$ to the PD_i, and attaches $\{\mathrm{ERID}_i, R_e\}$ with the patient's data.

Step 6. After getting $< \mathrm{CID}, R_j, s_j, M_1, T_2 >$ from C, PD_i tests freshness. If the timestamp is fresh, PD_i validates the message by comparing $s_j P =?R_j + h(R_j \| \mathrm{CID} \| M_1 \| T_2) P_c$. If successful, PD_i computes $K_1 = h\big(r_i R_j\big)$,$(\mathrm{ERID}_i, R_e, P_e, d_j) = D_{K_1}(M_1)$, $d_e = d_j + r_i$, $d_e P \overset{?}{=} R_e + h(R_e \| \mathrm{ERID}_i) P_c$ and saves $\{\mathrm{ERID}_i, R_e, P_e\}$ where d_e is the private key of the patient. Store d_e at a secure location.

2.2 Login and Authentication Phase

Step 1. To report the emergency, the registered patient first inputs his PID_i, PW_i, and BIO_i to the device. PD_i then computes $\delta_i = \mathrm{Rep}(\mathrm{BIO}_i, \sigma_i)$, $HPID_i = h(\mathrm{PID}_i\|\delta_i)$, and $w_i = Y_i \oplus h(\mathrm{PID}_i\|\mathrm{PW}_i\|\delta_i)$, and compares $Z_i =?h(\mathrm{PID}_i\|\mathrm{PW}_i\|\delta_i\|w_i)$ to verify the patient's credentials. If valid, PD_i chooses r_i and computes $R_i = r_i d_e P$, $K_1 = h(r_i d_e R_h)$, and encrypts the message using the computed key $M_1 = E_{K_1}(\mathrm{EM}, \mathrm{LOC})$. PDi then computes the verifier $s_i^1 = r_i d_e + h(R_i\|ERID_i\|M_1\|T_1)d_e$ and sends <$\mathrm{ERID}_i, R_i, R_e, M_1, s_i^1, T_1$> to HC_j.

Step 2. PD_i computes the key $K_2 = h(r_i d_e R_c)$, and encrypts the message with HCj identity $M_2 = E_{K_2}\big(\mathrm{EM}, \mathrm{LOC}, \mathrm{HID}_j\big)$. PDi then computes the verifier $s_i^2 = r_i d_e + h(R_i\|ERID_i\|M_2\|T_2)d_e$ and sends <$\mathrm{ERID}_i, R_i, R_e, M_2, s_i^2, T_2$> to C.

Step 3. On getting <$\mathrm{ERID}_i, R_i, R_e, M_1, s_i^1, T_1$> from PD_i, HC_j tests freshness, then verifies the message by $s_i P =?R_i + h(R_i\|\mathrm{ERID}_i\|M_1\|T_1)P_e$. If it is fresh and valid, HC_j computes the key $K_1^{\#} = h(r_h R_i)$, and decrypts $(\mathrm{EM}, \mathrm{LOC}) = D_{K_1^{\#}}(M_1)$. HC_j then checks the message about the ambulance. If it is required, HCj sends the ambulance to LOC, and starts preparing to handle the medical situation of the patient. HC_j then chooses r_j, and computes $R_j = r_j d_h P$, $K_3 = h(r_j d_h R_i)$, then encrypts the acknowledgment $M_3 = E_{K_3}(\mathrm{ACK})$. HC_j computes the verifier for the message $s_j = r_j d_h + h(R_j\|\mathrm{HID}_j\|M_3\|T_3)d_h$ and sends <$\mathrm{HID}_j, R_j, M_3, s_j, T_3$> to PD_i.

Step 4. After getting <$\mathrm{ERID}_i, R_i, R_e, M_2, s_i^2, T_2$> from PDi, the cloud first checks T_2's freshness. If it is fresh, C authenticates the message by checking if $s_i^2 P =?R_i + h(R_i\|\mathrm{ERID}_i\|M_2\|T_2)P_e$. If the verifier passes the validity test, C computes the key $K_2^{\#} = h(r_c R_i)$ and decrypts $\big(\mathrm{EM}, \mathrm{LOC}, \mathrm{HID_j}\big) = D_{K_2^{\#}}(M_2)$. C then searches for and activates ERID_i. If the patient is reporting the emergency for himself, C searches for the associated doctor's public parameters and MPID_i then executes *Step 4.1*, or if the patient is reporting for an unregistered patient, it executes *Step 4.2*.

Step 4.1 C chooses r_k, and computes $R_k = r_k d_c P$. C then fetches the assigned doctor's public parameters and computes $K_4 = h(\mathrm{HMID}_m\|v_m\|R_k\|T_4)$. C then encrypts $M_4 = E_{K_4}(\mathrm{EM}, \mathrm{MPID}_i)$ and computes the verifier $s_k^1 = r_k d_c + h(R_k\|\mathrm{CID}\|M_4\|T_4)d_c$. C then sends <$\mathrm{CID}, R_k, M_4, s_k^1, T_4$> to MPm. After that, C computes $K_5 = h(r_k d_c R_h)$, and encrypts $M_5 = E_{K_5}(m_h)$, where m_h is anonymized data of the patient. C then computes the verifier for the message $s_k^2 = r_k d_c + h(R_k\|\mathrm{CID}\|M_5\|T_5)d_c$ and sends <$\mathrm{CID}, R_k, M_5, s_k^2, T_5$> to HC_j.

Step 4.2 C registers the new patient as a temporary patient and assigns a temporary random id TRID. The memory portion of the database assigned to TRID can be used to store medical information about the new patient. C then chooses r_k, and computes

$R_k = r_k d_c P$, $K_5 = h(r_k d_c R_h)$, and encrypts $M_5 = E_{K_5}$(TRID). C then computes the verifier for the message $s_k^2 = r_k d_c + h(R_k \| CID \| M_5 \| T_5) d_c$ and sends <CID, R_k, M_5, s_k^2, T_5> to HC$_j$.

Step 5. After getting <CID, R_k, M_5, s_k^2, T_5> from C, HC$_j$ checks the freshness. If the timestamp passes the freshness test, HC$_j$ performs $s_k^2 P \ {=}?R_k + h(R_k \| CID \| M_5 \| T_5) P_c$ to verify the message. If the message passes the verification, HCj computes $K_5^\# = h(r_h R_k)$ and decrypts $(m_h) = D_{K_5^\#}(M_5)$ or $(TRID) = D_{K_5^\#}(M_5)$.

Step 6. This step is only required when the patient reports the emergency for himself. MPm checks if T_4 is fresh and compares $s_k P \ {=}?R_k + h(R_k \| \text{CID} \| M_4 \| T_4) P_c$ to verify the message. If the message is valid and fresh, MP's device shows an alert message on the screen. To open the message, MP inputs his identity MID$_m$, password PW$_m$, and biometric information BIO$_m$. MP device verifies the credentials by computing $\delta_m = \text{Rep}(\text{BIO}_m, \sigma_m)$, $v_m = A_m \oplus h(\text{MID}_m \| \text{PW}_m \| \delta_m)$, and $B_m \ {=}?h(\text{MID}_m \| \text{PW}_m \| \delta_m \| v_m)$. After that, it computes $\text{HMID}_m = h(\text{MID}_m \| \delta_m)$, $K_4^\# = h(\text{HMID}_m \| v_m \| R_k \| T_4)$, and decrypts (EM, MPID$_i$) $= D_{K_4^\#}(M_4)$. It then searches for the MPID$_i$ in the memory and displays the emergency and patient information on the screen.

3 Security Analysis

3.1 *Informal Security Analysis*

3.1.1 Mutual Authentication

In the proposed scheme, receivers authenticate the senders using the message verifiers with the senders' public key, for example $s_j P \ {=}?R_j + h(R_j \| \text{HID}_j \| M_3 \| T_3) P_h$ where $R_j = r_j d_h P$, $s_j = r_j d_h + h(R_j \| \text{HID}_j \| M_3 \| T_3) d_h$. Computing a verifier s without knowing the private key is impossible. Therefore, if the verifier is valid and the timestamp is fresh, the message is from the claimed sender, and the sender is authenticated using his public key.

3.1.2 Replay Attack

The proposed scheme uses timestamps, and all the messages are validated using a verifier that includes the timestamp to avoid a replay attack, for example, $s_j = r_j d_h + h(R_j \| \text{HID}_j \| M_3 \| T_3) d_h$. Computing a valid verifier requires the sender's private key, so the attacker cannot generate a valid verifier. On the other hand, using a new timestamp with an old verifier will lead to failure in the verification step. Therefore, the proposed scheme is safe against replay attacks.

3.1.3 Man-in-the-Middle Attack

It is a general term for an attacker positioning himself between the parties participating in communication to eavesdrop on or pretend to be one of the participants, seeming as though regular communication is taking place. In the proposed scheme, the message is encrypted using the key generated using the secret parameter of one party with the public parameter of the other party. Computing the key without knowing one of the secrets is impossible for the attacker. On the other hand, to impersonate one of the parties, an attacker needs to compute a valid verifier s for the message. Verifiers are calculated using the sender's session-specific temporary secret and private key and validated using the sender's public key. Calculating a valid verifier without knowing the private key is not possible. Therefore, the proposed scheme is secure against man-in-the-middle attacks.

3.1.4 Stolen Patient Device Attack

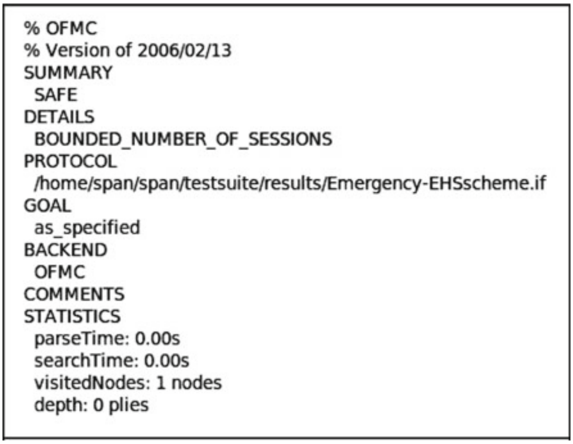
If the patient's device gets stolen from a registered patient, an attacker can extract $\{\mathrm{APID}_i, Z_i, \sigma_i, \mathrm{CID}, R_c\}$ security parameters from the device where APID_i is an anonymous patient identity that gets updated after a single use, $Y_i = h(\mathrm{PID}_i \| \mathrm{PW}_i \| \delta_i) \oplus w_i$, $Z_i = h(\mathrm{PID}_i \| \mathrm{PW}_i \| \delta_i \| w_i)$, σ_i is an auxiliary bit string used to reproduce a secret bit string δ_i with the patient's biometric parameter BIO_i, CID is the cloud server identity, and R_c is the public parameter of the cloud server. PID_i, PW_i, and δ_i are secured using hash functions, and w_i can be only computed using correct PID_i, PW_i, and δ_i. Guessing all three unknown parameters is computationally hard. Therefore, a stolen device does not reveal any long-term secrets to the attacker. So, the suggested scheme is secured against stolen patient device attacks. The same can be applied to stolen MP device attacks.

3.1.5 Impersonation Attack

The receiver in the proposed scheme validates the message using a message verifier with the sender's public key. To impersonate a valid party in the communication, the attacker requires a valid verifier for the message that is computed using the message, session-specific public parameter, and private key of the sender. So, the attacker cannot impersonate the sender without knowing the private key of the sender. Therefore, the proposed scheme is resilient against impersonation attacks.

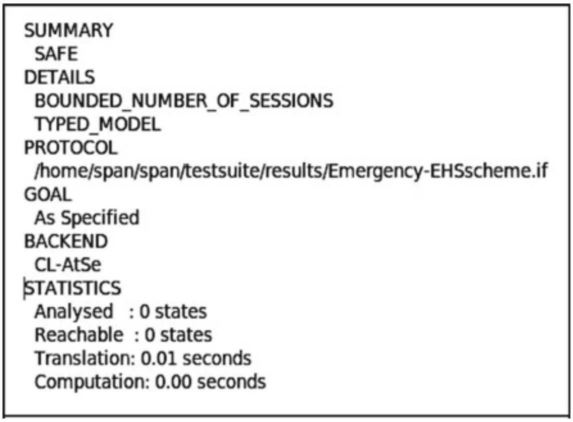
3.2 Formal Security Analysis

This section uses the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool to verify the suggested scheme formally. The Dolev and Yao model [16] is used to simulate and verify the authentication scheme. We simulated the proposed scheme for four secrecy goals and authentication goals. The results of the simulation are given in Figs. 4 and 5.



```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/Emergency-EHSscheme.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.00s
 visitedNodes: 1 nodes
 depth: 0 plies
```

Fig. 4 Simulation result for OFMC back-end



```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL
 /home/span/span/testsuite/results/Emergency-EHSscheme.if
GOAL
 As Specified
BACKEND
 CL-AtSe
STATISTICS
 Analysed   : 0 states
 Reachable  : 0 states
 Translation: 0.01 seconds
 Computation: 0.00 seconds
```

Fig. 5 Simulation result for CL-AtSe back-end

4 Performance Evaluation

4.1 *Security and Functionality Features*

This section compares the proposed scheme's security and functionality attributes to those of other relevant schemes: Chiou et al. [8], Chen et al. [17], Li et al. [18], Chandrakar et al. [19], and Ansari et al. [15]. Table 2 provides the security and functionality comparison of these schemes.

4.2 *Computational and Communication Cost*

This section discusses the proposed scheme's communication and computational costs. For calculating communication and computation costs, we only consider the scheme's login and authentication phase because it is used more frequently than the registration phase. In order to determine the communication cost, we assumed that the length of timestamps, IDs, and random numbers are 48 bits. The size of the cryptographic hash is 160 bits, while the asymmetric/symmetric cryptosystem requires 128 bits. The total communication cost of the proposed scheme is 2816 bits. A detailed description of the proposed scheme's computational and communication cost is given in Table 3.

Table 2 Comparison of security and functionality features of relevant schemes

	[8]	[17]	[18]	[19]	[15]	Proposed
Provides patient unlinkability	N	Y	N	N	Y	Y
Provides doctor unlinkability	Y	Y	Y	Y	Y	Y
Provides session key security	Y	N	N	Y	Y	Y
Provides patient anonymity	N	N	N	Y	Y	Y
Provides data confidentiality	N	Y	N	Y	Y	Y
Resilient against replay attack	Y	Y	Y	Y	Y	Y
Resilient against man-in-the-middle attack	Y	Y	Y	Y	Y	Y
Resilient against impersonation attack	N	Y	N	N	Y	Y
Provides message authentication	Y	Y	N	Y	Y	Y
Resilient against off-line guessing attack	Y	N	Y	Y	Y	Y
Provides emergency care for registered patients with WBAN	N	Y	N	N	Y	Y
Provides emergency care for registered patients without WBAN	N	N	N	N	N	Y
Provides emergency care for unregistered patients	N	N	N	N	N	Y

Table 3 Computational and communication cost

Computational cost				Communication cost
PD	HC	C	MP	
$6T_M + 9T_H + 3T_S$	$8T_M + 6T_H + 3T_S$	$5T_M + 6T_H + 3T_S$	$2T_M + 5T_H + 1T_S$	2816 bits
Total	$21T_M + 26T_H + 10T_S$			

T_M: Time complexity of ECC point multiplication operation; T_H: Time complexity of one-way hash function; T_S: Time complexity of symmetric encryption/decryption operation

5 Conclusion and Future Work

This paper presents an extension of the framework proposed in [15] for emergency care of patients without WBAN implemented on their bodies and for an unregistered person. We performed formal and informal security analyses to demonstrate that the proposed scheme is secured. The AVISPA tool is used to perform formal security analysis. The proposed scheme is computationally efficient and can only work along with the framework presented by Ansari et al. [15] to provide emergency support. We intend to expand our effort in the future to include heterogeneous e-healthcare systems.

References

1. Althebyan Q, Yaseen Q, Jararweh Y, Al-Ayyoub M (2016) Cloud support for large scale e-healthcare systems 71(9–10):503–515. https://doi.org/10.1007/s12243-016-0496-9
2. Nkenyereye L, Islam SMR, Hossain M, Abdullah-Al-Wadud M, Alamri A (2020) Blockchain-enabled EHR framework for internet of medical things. https://doi.org/10.32604/cmc.2021.013796
3. Kumari S, Khan MK, Kumar R (2013) Cryptanalysis and improvement of 'A privacy enhanced scheme for telecare medical information systems.' J Med Syst 37(4):1–11. https://doi.org/10.1007/s10916-013-9952-5
4. Jiang Q, Ma J, Ma Z, Li G (2013) A privacy enhanced authentication scheme for telecare medical information systems. J Med Syst 37(1):1–8. https://doi.org/10.1007/s10916-012-9897-0
5. Abd-El-Atty B, Iliyasu AM, Alaskar H, El-Latif AAA (2020) A robust quasi-quantum walks-based steganography protocol for secure transmission of images on cloud-based e-healthcare platforms. Sensors 20(11):3108. https://doi.org/10.3390/S20113108
6. Abd EL-Latif AA, Abd-El-Atty B, Abou-Nassar EM, Venegas-Andraca SE (2020) Controlled alternate quantum walks based privacy preserving healthcare images in internet of things. Opt Laser Technol 124:105942. https://doi.org/10.1016/J.OPTLASTEC.2019.105942
7. Chen C-LL, Yang T-TT, Chiang M-LL, Shih T-FF (2014) A privacy authentication scheme based on cloud for medical environment. J Med Syst 38(11):143. https://doi.org/10.1007/s10916-014-0143-9
8. Chiou SY, Ying Z, Liu J (2016) Improvement of a privacy authentication scheme based on cloud for medical environment. J Med Syst 40(4):1–15. https://doi.org/10.1007/s10916-016-0453-1
9. Challa S et al (2020) Design and analysis of authenticated key agreement scheme in cloud-assisted cyber–physical systems. Futur Gener Comput Syst 108:1267–1286. https://doi.org/10.1016/j.future.2018.04.019

10. Chaudhry SA, Shon T, Al-Turjman F, Alsharif MH (2020) Correcting design flaws: an improved and cloud assisted key agreement scheme in cyber physical systems. Comput Commun 153:527–537. https://doi.org/10.1016/j.comcom.2020.02.025
11. Kumari A, Kumar V, Abbasi MY (2020) EAAF: ECC-based anonymous authentication framework for cloud-medical system. Int J Comput Appl 0(0):1–10. https://doi.org/10.1080/1206212X.2020.1815334
12. Khan AA, Kumar V, Ahmad M, Rana S (2021) LAKAF: lightweight authentication and key agreement framework for smart grid network. J Syst Archit 116:102053. https://doi.org/10.1016/J.SYSARC.2021.102053
13. Kumari A et al (2020) CSEF: cloud-based secure and efficient framework for smart medical system using ECC. IEEE Access 8:107838–107852. https://doi.org/10.1109/ACCESS.2020.3001152
14. Wu TY, Yang L, Luo JN, Ming-Tai Wu J (2021) A provably secure authentication and key agreement protocol in cloud-based smart healthcare environments. Secur Commun Netw 2021. https://doi.org/10.1155/2021/2299632
15. Ansari AA et al (2022) Privacy-enabling framework for cloud-assisted digital healthcare industry. IEEE Trans Ind Inform. https://doi.org/10.1109/TII.2022.3170148
16. Dolev D, Yao AC (1983) On the security of public key protocols. IEEE Trans Inf Theory 29(2):198–208. https://doi.org/10.1109/TIT.1983.1056650
17. Chen CL, Yang TT, Shih TF (2014) A secure medical data exchange protocol based on cloud environment. J Med Syst 38(9):1–12. https://doi.org/10.1007/s10916-014-0112-3
18. Li CT, Shih DH, Wang CC (2018) Cloud-assisted mutual authentication and privacy preservation protocol for telecare medical information systems. Comput Methods Programs Biomed 157:191–203. https://doi.org/10.1016/j.cmpb.2018.02.002
19. Chandrakar P, Sinha S, Ali R (2020) Cloud-based authenticated protocol for healthcare monitoring system. J Ambient Intell Humaniz Comput 11(8):3431–3447. https://doi.org/10.1007/s12652-019-01537-2

Lithography Hotspot Detection

Lydia R. Darla, Sushma Garawad, and Suneeta V. Budihal

Abstract The paper proposes, creating a learning-based hotspot identification framework. The process of lithography hotspot identification is essential for developing semiconductor designs, but employing an optical simulation method to do so is time-consuming and slows down the process of developing layout designs. Despite the introduction of the geometry-based technique, it still showed poor detection performance and a complex framework. In order to address this problem, we provide a deep convolutional-based neural network hotspot identification technique. The outcomes show that using a learning-based strategy and DCNN, results in performance with a 98.6% accuracy.

Keywords Hotspot detection · PCB defect detection · Machine learning · DC GAN · Convolutional neural network

1 Introduction

Lithography is typically thought to detect open-circuit or short-circuit errors more often. Poor printability across a number of the patterns in a typical layout design can create this problem. Early design GSaw detection is essential since it can help to save a lot of time and money. As the semiconductor design shrinks, the physical design is more confined by the lithography constraint, since the fundamental patterning process still predominantly relies on 193 nm lithography technology. Therefore, it is important to validate semiconductor design in the presence of light and is termed as lithography. The cornerstone of physical design verfication is the optical rule check. Using optical simulation, it generates a projected image on a silicon wafer, from which it extracts troublesome positions. Despite having excellent hotspot detection performance, the optical modelling approach involved in it makes it a costly computing operation resulting in slowed-down process.

L. R. Darla · S. Garawad · S. V. Budihal (✉)
Electronics and Communication, KLE Technological University, Hubli, India
e-mail: suneeta_vb@kletech.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_24

The introduction of geometric verification methods has provided an alternative to optical simulation-based verification, which consumes a lot of time as they directly infer the design arrangement, without the need for optical modelling hence, these approaches offer quick detection speeds. Methods based on machine learning and pattern matching were introduced. The method of pattern matching is effective for locating registered hotspots. By comparing the test layout's pattern geometric similarity to previously registered hotspot patterns, it can identify hotspots. Although it is reliable for known hotspots, it is limited in its ability to locate undiscovered hotspots, which results in poor detection accuracy. For the purpose of locating undiscovered hotspots, machine learning is introduced. Using supervised learning, the system gains the capacity to classify data from known hotspots and non-hotspots. Since it picks up on the properties of the hotspots and the non-hotspots during training, the trained model will be able to spot model flaws.

2 Literature Survey

Harshitha and Dr. Mahesh Rao [1]: The topic of defect identification in many types of printed circuit boards, including single-layer, double-layer and multilayer bare PCBs and assembled PCBs, is covered in this article. The prospects and some of the difficulties that currently exist in this field are discussed. Therefore, the authors believe that a new model has to be developed in order to find a solution to the problems they have discovered.

Ibrahim et al. [2]: This article suggests a PCB fault localization technique for automated visual PCB inspection. Inputs for the classification stage, which comes after fault identification, will be taken from the localized region in the inspected PCB picture. The goal of this research's ongoing work is to put the algorithm on hardware so that the automated visual PCB inspection system can operate with great efficiency in a real-time setting.

Flvio et al. [3]: In an effort to identify PCB failures, this research suggests a novel approach that lessens the computational burden of scanning the whole board. Small photos of the PCB were taken into consideration. After the system located the areas with fatal mistakes, it is feasible. Each little picture is subjected to a link analysis technique. The results show that the approach is workable, but further advancements need to be made before the system can be turned into an industrial real-time system. For instance, we may test each little separated picture using parallel processing. This ought to increase effectiveness and cut down on computation time. Future research will involve updating the techniques for analysing PCBs with components. In this situation, it should be possible to find missing and replaced components, misaligned components, etc.

Liu et al. [4]: SSD, a quick single-shot object detector for several categories, is introduced in this study. Our model's usage of multi-scale convolutional bounding box outputs coupled to several feature mappings at the network's top is a vital component. We can effectively model the space of potential box forms using this approach.

We demonstrate empirically that, given suitable training procedures, a greater number of carefully selected default bounding boxes leads to better performance. In comparison to previous techniques, we construct SSD models with at least an order of magnitude more box predictions sampling position, size and aspect ratio. We show that, given the same VGG-16 basic architecture, SSD performs favourably in terms of accuracy and performance when compared to its cutting-edge object detection competitors. While being three times quicker, our SSD512 model performs noticeably better on PASCAL VOC and COCO in terms of accuracy than the most recent faster R-CNN. Our real-time SSD300 model generates much better detection accuracy while operating at 59 FPS, which is quicker than the existing real-time YOLO alternative.

Ren et al. [5]: The study presents RPNs for quickly and precisely generating region proposals. The downstream detection network and convolutional features are shared, making the region proposal step almost cost-free. With the help of our technique, a deep learning-based object identification system can operate at frame rates that are almost real time. The quality of the region proposals is likewise enhanced by the learnt RPN, increasing the total object detection accuracy.

3 DeepPCB Dataset

We give Deep PCB, which has 1,500 PCB picture pairings spanning six different kinds of PCB defects, to the community. A 640 × 640 free from the defects for the different types of image in the form of templates and a defective testing image forming together to make up each pair. We divide the remaining 500 image pairings into a test set and a training set of 1,000 image pairs.

3.1 Image Collection

This dataset's photos were all taken on a straight scan CCD with just a resolution of around 48 pixels per millimetre., in accordance with standard industrial settings. In the manner described above, the defect-free template picture is a sampling picture that has been carefully cleaned and analysed. For the template and the image that was tested, the original dimensions are roughly 16 k by 16 k pixels. Then, the differences in the images' offsets for translation and rotation are reduced before they are cut into 640 × 640 sub-images and aligned using template matching algorithms. The next step is to carefully choose a threshold to use binarization in order to prevent lighting disturbance. For high-accuracy PCB fault localization and classification, common solutions include image registration and thresholding, even though there are several pre-processing methods depending on the individual PCB defect detecting algorithm [1].

3.2 Image Annotation

For each flaw in the tested photos, we utilize a BB with an axis alignment and a class ID. Figure 1 depicts the six common PCB fault types that we have identified. 1: short-type errors, spur-type errors, open circuit-type errors, pinhole-type errors, spurious copper-type errors and mouse bites. Since the actual tested image only has a few flaws, we manually introduce 3–12 fictitious flaws in each 640 × 640 image by applying artificial flaws in accordance with the PCB defect patterns [2]. In Fig. 2, the total number of PCB flaws is depicted.

3.3 Benchmarks

Average accuracy rate and F-mean are employed for assessment, which is in line with benchmarks on datasets for object or scene text identification [3, 4]. A detection is only accurate if any ground box of the truth with the identified bounding and the same type of class coincide with units (IoU) greater than 0.33.

3.4 Proposed Methodology

In this segment, we will go through the suggested method for identifying PCB flaws from two input photos in more detail. Instead of only evaluating the difference

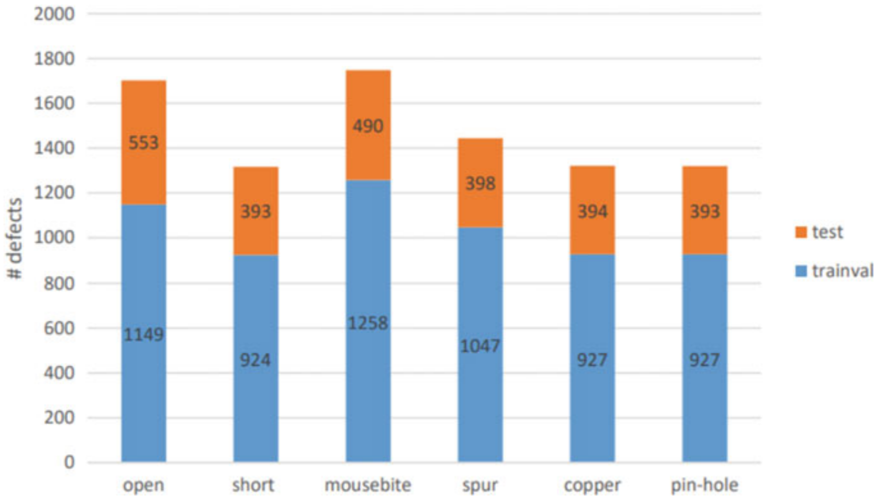


Fig. 1 Defective no. 6 classes of the DeepPCB training or validation including the set of test

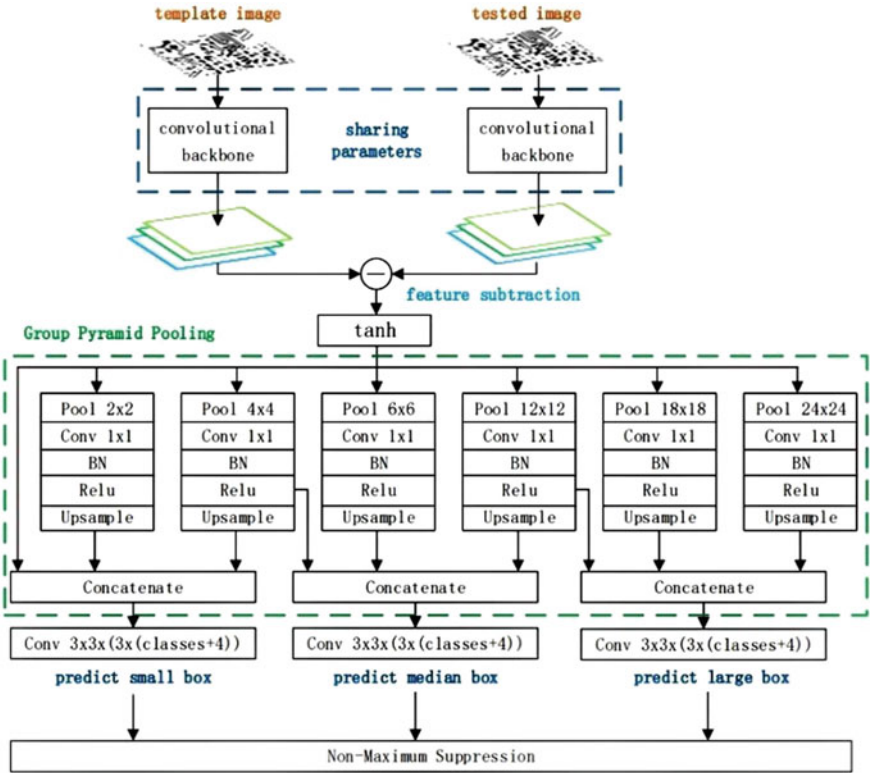


Fig. 2 Structure of group pyramid pooling

between both the input image sets of two, a convolutional core along with maximum operation of pool is first employed to extract features of input image invariance from the input picture pair.

The traits from the template and the evaluated photos are then contrasted. Using a special group pyramid pooling module, features are acquired at various resolutions. We are making projections of various sizes utilizing the backbone's feature maps, much like [5, 6]. In Fig. 2, we show how the recommended PCB defect detection model is generally organized. The cost of storage and processing increases when using the FPN [6], which integrates data at different resolutions ranging from coarse to fine. A pyramid pooling structure is used by the Group Pyramid Pooling (GPP) module for collecting features from different types of resolutions that are available.

Research on a similar pyramid architecture may be found in [7]. Each group is really responsible for forecasting PCB errors from the standard packaging [5] on a certain scale, for example, the first group takes data from pooling sizes of 1 × 1, 4 × 4 and 2 × 2 to anticipate PCB issues with small bounding boxes. These groups also share certain input attributes with the group next to them, which reduces the edge effect.

As seen in Fig. 2, the convolutional layer of the top generated a $m \times n$ [3 [4 + classes]] map for the sake of estimation 3. The m and n positions are the centres of three distinct types of default boxes [5], each of which has an aspect ratio of 0.5, 1.0 or 2.0. In Fig. 1, the small, medium and large boxes in our implementation, the hyper-parameters for the input picture size that are manually chosen for the default box sizes are 0.04, 0.08 and 0.16, respectively.

It generates predictions for (i) localization, it contains I classification at each of the m locations, which includes (ii) six different types of PCB faults and one background class, one translation offset between the centroids and (iii) scaling factor between the default boxes' width and height and the targets' width and height. To obtain the final prediction results, Non-Maximum Suppression (NMS) is used for all the forecasts from various scales.

3.5 Function of Objective

According to the matching process in SSD [5], the default box is compared to every ground truth box first [5] of the biggest Jaccard overlap [8]. The ground truth boxes that have Jaccard overlaps larger than 0.5 are then compared to the default boxes. They fall under the following categories:

$$D = (d, g) | jaccardoverlap(d, g) > 0.5), \tag{1}$$

where the ground truth box is represented by (dcx, dcy, dw, dh) and the centre, height and breadth of the standard box, respectively. As a result, the goal function for box regression is defined as follows:

$$L_{\text{reg}} = \sum_{(d_n, g_n) \in \mathcal{D}} \sum_{i \in cx, cy, w, h} \text{smooth}_{\text{L1}}(l_n^i - t_n^i), \tag{2}$$

SoftMax loss is utilized to determine the PCB fault-type categorization loss, and background default boxes are chosen at random to maintain a roughly 3:1 ratio between the bounding boxes for the background (Bg) and foreground (Fg)

Where the box of the ground truth is represented by (d cx, dcy, dw, dh) and the centre, height and breadth of the standard box, respectively. As a result, the goal function for box regression is defined as follows:

Note that the background box's class index is set to 0.

$$t_n^j = (g_n^j - d_n^j)/d_n^j, \qquad j \in \{cx, cy\},$$

$$t_n^k = \log(\frac{g_n^k}{d_n^k}), \qquad k \in \{w, h\}.$$

$$L_{\text{cls}} = -\sum_{d_n \in \text{Fg}} \log(c_n^p) - \sum_{d_n \in \text{Bg}} \log(c_n^0), \qquad (3)$$

3.6 DC GAN Implementation

After providing it with photos of several PCB fault images, to create new PCB fault pictures, we'll train a GAN. An explicit use of A DCGAN is a simple expansion of the GAN outlined above thanks to the discriminator and generator's convolutional and transpose layers of convolutional, respectively. Un-supervised Presentation Training With DC-GAN, Radford et al. provided the first description of it. LeakyReLU activations, depthwise convolution layers and batching norm layers make up the discriminator. The output is a scalar probability that the input is drawn from the genuine data distribution and the input is a 3 × 64 × 64 input picture. The generator is composed of convolutional transpose stages, a batch of norm sections and a ReLU of activations. A 3 × 64 × 64 RGB picture is produced using a latent vector, which has been taken from a typical normal distribution as the input.

3.7 Weight Initialization

Initializing each model weight at random from a normal distribution with a mean of zero is required, a standard deviation of 0 and a standard deviation of 0.02. When given an initialized model as input, the weights init function re-initializes all conv, conv-transpose and batch normalization layers to satisfy this requirement. Following initialization, the models are immediately subjected to this function.

3.8 Generator

The latent space vector (zz) is intended to be mapped to data space by the generator, GG. As photos make up our data, transforming zz into space of the data entails eventually similar to the practice photos, making an RGB image with those dimensions (i.e. 3 × 64 × 64). There are several layers in this double convolutional transposition, each with a batch norm layer, an activation function of ReLu layer, and are employed in practice to this end. The output of the generator is returned to the [–1,1] [1, 1]

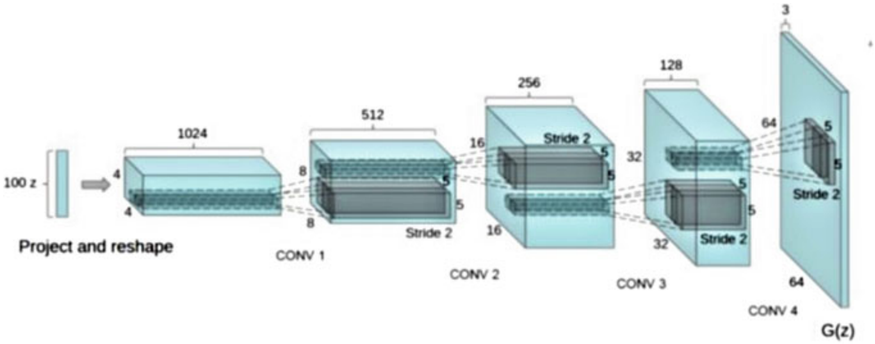


Fig. 3 Example of a figure caption

input data range using the tanh function. It's critical to remember that the DCGAN study made a significant advance by including the convtranspose layers, the batch norm operates. These layers support the gradient flow during training. A producer from the DCGAN study is shown in the image below.

Note how the code generation architecture is affected by the nz, ngf and nc sources that we describe in the input section. The terms nc, nz and ngf refer to the number of streams in the output image, the input vector's length of z and the amount of the extracted features sent by the generator, respectively (set to 3 for RGB images) (Fig. 3).

3.9 Discriminator

Taking in an image, the discriminator is a binary classifier network. as argument and returns a scalar estimate of the reality probability of the input image. This has previously been mentioned. Here, discriminator processes a 3 × 64 × 64 input image to use the Conv2d, BatchNorm2d and LeakyReLU layers before employing a Sigmoid activation function to get the final probability. Despite the fact that this architecture may be modified to include many more layers if the problem calls for it, the addition of strided inversion, BatchNorm and LeakyReLUs is important. The DCGAN study recommends strided convolution for down sampling because it enables the network to create its own pooled function.

3.10 Loss Functions and Optimizers

We can regulate how they learn by using loss functions and optimizers using the DD and GG layout. We'll employ the PyTorch function known as BCELoss which is defined as follows: Next, we define the false label as 0, while the true label is defined as 1. This convention, which was also used in the original GAN study, will be utilized to calculate the losses of DD and GG. We then configured two independent optimizers, one for DD and one for GG. The DCGAN paper describes both Adam optimization techniques with learning rates of 0.0002 and beta1 = 0.5. A fixed batch of latent vectors will be generated by us in order to monitor the generator's learning progress. We will regularly feed this fixed noise into GG during the training loop, and as iterations pass, we will see pictures emerge from the noise.

$$\ell(x, y) = L = \{l_1, \ldots, l_N\}^\top, \quad l_n = -\left[y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n)\right]$$

3.11 Training

For actual and false photos, we will create several mini-batches, and we'll also tweak G's goal function to maximize $\log D(G(z))\log D(G(z))$. There are two fundamental components to training.

Remember that increasing the likelihood that a given input will be correctly classified as authentic or false is the aim of training the discriminator. We want to 'upgrade the discriminator by rising its stochastic gradient', to quote Goodfellow. Maximizing $\log(D(x)) + \log(1D(G(z)))\log(D(x))+\log(1D(G(z)))$ is what we're aiming for practically. We shall compute this in two phases in response to Ganhacks' distinct mini-batch idea. We will first create a collection of real samples taken from the practice set. Next, using the current generator, we'll create a batch of samples, forward pass them through DD, determine their loss$(\log(1 - D(G(z))\log(1D(G(z)))$, and then make a retrograde proceed to gather their gradients. Now, we use the discriminator's optimization to rename a step to a pass rearward. Now, we designate a discriminator's optimizer phase by utilizing the gradients collected from both the batches that were all true.

As mentioned in the original work, our objective is to improve the standard of the fakes produced by the generator by decreasing $\log(1 - D(G(z)))\log(1D(G(z)))$. As previously indicated, as demonstrated that this did not offer enough gradients, particularly in the early stages of learning. Instead, we would like to increase $\log(D(G(z)))$ as a remedy. In the algorithm, We accomplish this by using the discriminator to categorize the generator output from Part 1, using actual labels as GT to compute G's loss.

3.12 Results and Discussion

On the basis of the DeepPCB dataset, this section offers quantitative assessments of several PCB fault detection techniques. Each duo's verified picture and model are generated concurrently using randomization and horizontally/vertically flipped with a chance of 0.5 and then arbitrarily chopped into a dimension of 512 × 512 for the purpose of arguing data for deep learning models. Table 1 shows the evaluation endings based on the DeepPCB dataset. The suggested model increases mean average accuracy from 1.0 to 9.3% for equally sophisticated two-stage models [5, 9] as well as simpler one-stage models [10, 11].

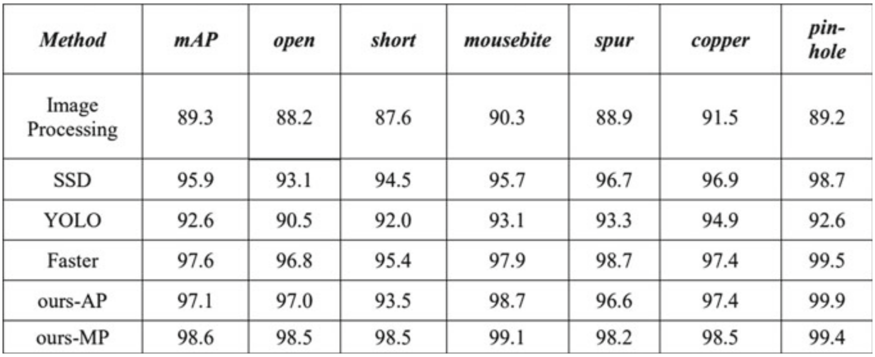
The findings of the experiment also demonstrate that the maximum pooling of GPP enhances mAP by 1.5% more pooling. Even though [10] gets competitive effectiveness, inference requires a lot more time.

Evaluation results for mAP on the DeepPCB dataset are shown in the table above. The contrasting options for maximum pooling versus average pooling operation GPP modules are mentioned by the letters 'AP' or 'MP' (Figs. 4, 5, 6 and 7).

Input image that will be examined is shown in Fig. 8. It has a number of flaws, including pinholes, mouse bites, spurs, copper and open circuits. Figure 9 displays the final picture, which clearly demonstrates all the issues mentioned previously.

The DC-GAN implementation shows us that the system can differentiate the fake image from the real image.

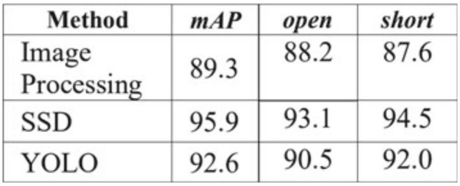
Input image that will be examined is shown in Fig. 8. It has a number of flaws, including pinholes, mouse bites, spurs, copper and open circuits. Figure 9 displays the final picture, which clearly demonstrates all the issues mentioned previously.



Method	*mAP*	*open*	*short*	*mousebite*	*spur*	*copper*	*pin-hole*
Image Processing	89.3	88.2	87.6	90.3	88.9	91.5	89.2
SSD	95.9	93.1	94.5	95.7	96.7	96.9	98.7
YOLO	92.6	90.5	92.0	93.1	93.3	94.9	92.6
Faster	97.6	96.8	95.4	97.9	98.7	97.4	99.5
ours-AP	97.1	97.0	93.5	98.7	96.6	97.4	99.9
ours-MP	98.6	98.5	98.5	99.1	98.2	98.5	99.4

Fig. 4 Results of the mAP evaluation on the DeepPCB dataset.

Fig. 5 Max-pooling procedure evaluation in the GPP subsystem



Method	*mAP*	*open*	*short*
Image Processing	89.3	88.2	87.6
SSD	95.9	93.1	94.5
YOLO	92.6	90.5	92.0

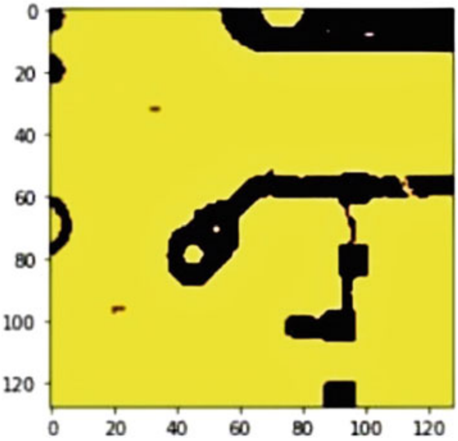


Fig. 6 Image under test

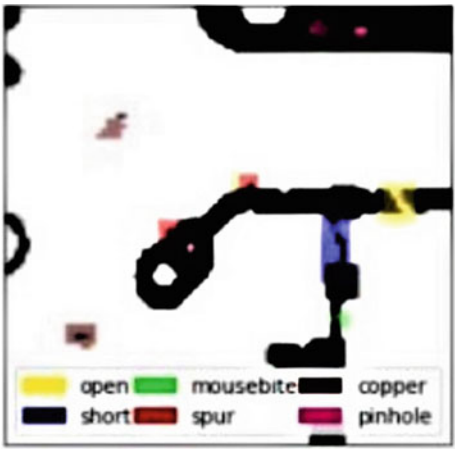


Fig. 7 Test results image

The DC-GAN implementation shows us that the system can differentiate the fake image from the real image.

4 Conclusion

The paper proposes, creating a learning-based hotspot identification framework. Despite the introduction of the geometry-based technique, it still showed poor detection performance and a complex framework. In order to address this problem, we

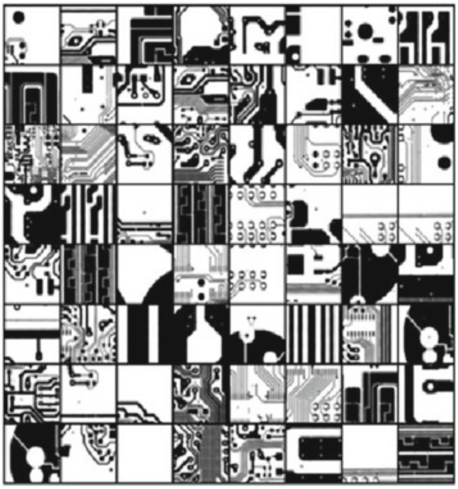


Fig. 8 Real image

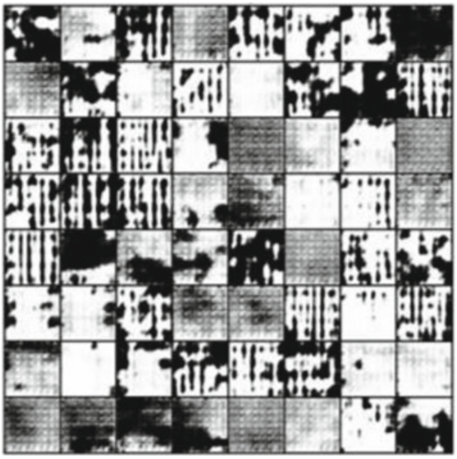


Fig. 9 Fake image

provide a deep convolutional-based neural network hotspot identification technique. With learning-based strategy and DCNN, it results in the performance of 98.6% accuracy.

References

1. Harshitha, Dr. Rao M (2016) Pcb defect detection and sorting using image processing techniques. Int J Eng Res Electron Commun Eng 3(2); Nicole R (in press) Title of paper with only first word capitalized. J Name Stand Abbrev
2. Ibrahim Z, Aspar Z, Al-Attas SAR, Mokji MM (2002) Coarse resolution defect localization algorithm for an auto mated visual printed circuit board inspection. In: IECON, vol 4, pp 2629–2634
3. Leta FR, Feliciano FF, Martins FP (2018) Computer vision system for printed circuit board inspection. In: ABCM symposium series in mechatronics, vol 3, p 623632
4. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision, pp 21–37
5. Ren S, He K, Girshick R, Sun J (2015) Faster rcnn: towards real-time object detection with region proposal networks. In: International conference on neural in formation processing systems, pp 91–99
6. Fu CY, Liu W, Ranga A, Tyagi A, Berg AC (2017) Dssd: deconvolutional single shot detector
7. Lin TY, Goyal P, Girshick R, He K, Dollar P (2017) Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell PP(99):2999–3007
8. Lin TY, Dollar P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid net works for object detection. In: IEEE conference on computer vision and pattern recognition, pp 936–944
9. Moganti M, Ercal F, Dagli CH, Tsunekawa S (1996) Automatic PCB inspection algorithms. Comput Vis Image Understand 63(2):287–313
10. Karatzas D, Gomez-Bigorda L, Nicolaou A, Ghosh S, Bagdanov A, Iwamura M, Matas J, Neumann L, Chandrasekhar VR, Lu S, Shafait F (2015) ICDAR 2015 competition on robust reading. In: International conference on document analysis and recognition, pp 1156–1160
11. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: IEEE conference on computer vision and pattern recognition, pp 6230–6239
12. Tang Z (2012) Hotspot detection by improved adaptive finite element method and its application in high-speed PCB and IC package design. IEEE Trans Compon Packag Manuf Technol 2(10):1659–1665
13. Wei Q et al (2021) The influence and optimization of design parameters on integrated circuits package warpage. In: 2021 22nd international conference on electronic packaging technology (ICEPT), pp 1–5
14. Afripin A, Carpenter B, Hauck T (2021) Finite element analysis of copper pillar interconnect stress of flip-chip chip-scale package. In: 2021 22nd international conference on thermal, mechanical and multi-physics simulation and experiments in microelectronics and microsystems (EuroSimE), pp 1–5
15. Suneeta VB, Purushottam P, Prashantkumar K, Sachin S, Supreet M (2020) Facial expression recognition using supervised learning. In: Smys S, Tavares J, Balas V, Iliyasu A (eds) Computational vision and bio-inspired computing. ICCVBIC 2019. Advances in intelligent systems and computing, vol 1108. Springer, Cham
16. Pavaskar S, Budihal S (2019) Real-time vehicle-type categorization and character extraction from the license plates. In: Mallick P, Balas V, Bhoi A, Zobaa A (eds) Cognitive informatics and soft computing. Advances in intelligent systems and computing, vol 768. Springer, Singapore

Intelligent Hotspot Detection in Layout Patterns

Sushma Garawad and Suneeta V. Budihal

Abstract We propose a framework to identify hotspots in IC design using artificial intelligence. Lithography is the technique of producing the mask pattern on silicon wafer in the context of IC design. One of the most crucial processes is lithography, which enables Moore's law, which states that feature size reduces every two years. Circuits shrink dramatically during this procedure, and the print ability will be on a micro and nano scale, which causes hotspots. Hotspots are nothing more than mistakes or errors on IC mask patterns, which can cause circuits to completely fail. We experimented with other models such as CNN, VGG-11 and achieved 93 percentage in each case using ICCAD 2012 dataset which are of five subsets dataset provided. In the future work, we can increase the sub-datasets' imbalance and model correctness. We could also use various different artificial techniques and architectural designs to implement them in much better results.

Keywords Hotspot detection · Machine learning · Convolutional neural network · VGG-11

1 Introduction

Lithography process is one of the most important stages of IC fabrication. In this process, patterns are generated on the silicon wafers. The mask patterns obtained from an intricate process which means patterns are initially obtained in the form of masks which is nothing but a thin film structured circuit mask; it will be in nanoscale [1], this basically derives from Moore's law. On a wafer to fit more transistors in the same area, size of transistors needs to reduce as time goes on increasing. Hence to follow the law patterning solutions needs to be developed in a cost effective way. The lithography process has been done in many different ways such as Optical

S. Garawad (✉) · S. V. Budihal
Electronics and Communication, KLE Technological University, Hubli, India
e-mail: sushmamgarwad@gmail.com

S. V. Budihal
e-mail: suneeta_vb@kletech.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_25

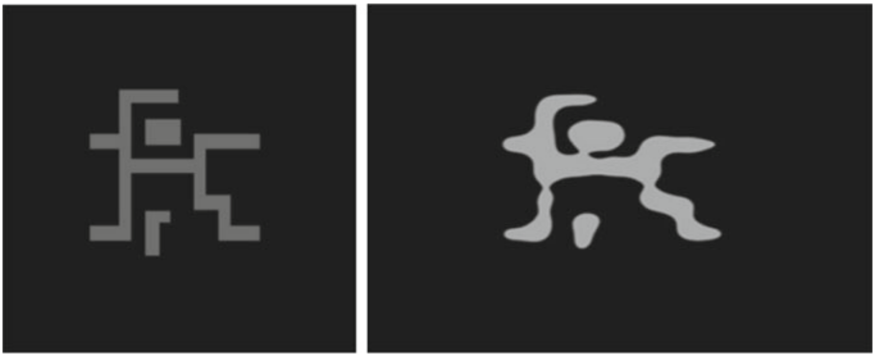


Fig. 1 Wafer images

lithography, electron beam lithography, X-ray lithography, and Ion beam lithography. The most popular method used is optical lithography. In this the feature size is proportional to wavelength.

$$F = \frac{c \cdot \lambda}{n} \tag{1}$$

Here, f is feature size, c is Rayleigh constant, lambda is wavelength, n is numerical aperture. Figure 1 shows the patterns on the mask and how it looks when those masks are printed on wafer die. The most effective way to reduce feature size is to reduce lambda (wavelength of light). In order to reduce feature size, some design rules must be followed by the substrate such as threshold value of edge widths of patterns and minimum spacing between two patterns. The reduction in wavelength may lead to print ability and degradation in resolution as shown in Fig. 1. In the recent studies, machine learning and deep learning-based techniques started to get implemented to identify these hotspots. In this work, we tried to implement different models that can be implemented on the hotspot detection techniques such as CNN and VGG-11 where we got 93 and 91% accuracy results. Lithography process is really a major step in the IC fabrication process; also this process is very expensive. Before the fabrication of the IC chips, if any faulty circuit is obtained with errors then it will be of huge loss to production which also can affect the whole batch of materials. We also studied a vision transformer (ViT) which converts images into patches and then passes them through transformers in order to classify images [2] as future work. The recent deep learning studies have shown prominent results in identifying the hotspot patterns and still improvement in the work is going on in order to get good results in the field of machine learning application in the field of VLSI chip designing.

2 Litrature Survey

Many varieties of Resolution Enhancement Techniques (RETs), including Optical Proximity Correction (OPC) and Sub-Resolution Assist Feature (SRAF), are used to identify lithography hotspots [3]. Different models can then be implemented, including separated into two categories: lithography hotspot detection techniques and Evaluation standards. The SAMPLE lithography simulation method was first introduced in 1979 which gave good results for greater size grids. The PROLITH method was also introduced in 1985; this method was the first method to experiment using personal computers and these positive resist optical lithography methods like improved versions are still used for simulation purposes. Full design although simulation is a highly effective method for identifying hotspots, it is relatively expensive in terms of complexity and computation time [4]. To identify hotspots some design rules must be followed [5, 6]. The likelihood that non-hotspots will be identified as hotspots in this technique is more. The pattern is now identified across the full mask, pixel by pixel. Patterns with some disruptions from the mask are referred to as defective or with hotspots. The string-based techniques that transform two-dimensional layouts into strings are the names given to these single-dimensional structures. Then, to locate hotspot-containing strings, methods utilizing distance arrays are performed. In some deep learning and machine learning models, features must be extracted prior to classification in order to decrease the amount of training data and boost processing speed. There are numerous algorithms, including Topological Classification and Critical feature extraction. If we discuss about deep learning-based techniques then CNNs [7], GANs [8], CNNs with DBSCAN clustering, etc. In this work, we tried to explore the new technique. We can identify the hotspot and non-hotspot using different methods like CNN, VGG-11 and studied the ViT methods, and we find good result in these models. Also there are techniques which are used based on evaluation parameters, and in this we prefer the ROC (receiver operating characteristics) curve that is used for these hotspot evaluation.

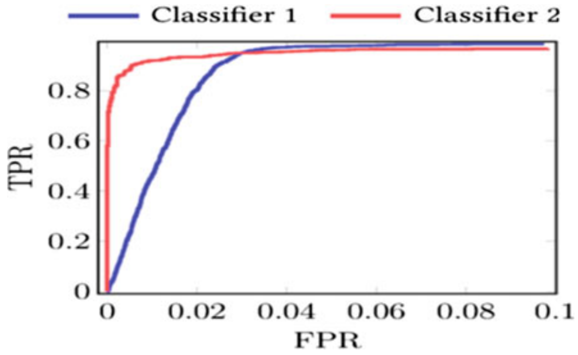
$$\text{Accuracy} = \frac{\text{Total hits}}{\text{number of actual hotspots}} \tag{2}$$

$$\text{False alarm} = \frac{\text{Total extras}}{\text{number of actual hotspots}} \tag{3}$$

Here we can define hit as hotspot that has been correctly identified. Extra is defined as non-hotspot that has been mistakenly classified as hotspot

$$\text{AUC score} = \text{Area under ROC} = \int \text{ROC} \tag{4}$$

Fig. 2 ROC curve



$$\text{Partial AUC score} = \text{Area under ROC (from ab)} = \int_a^b \text{ROC} \tag{5}$$

Receiver operating characteristics (ROC) curve for binary classifier is represented as shown which is applied for imbalance dataset issues. This curve shows true and false positive rates (Fig. 2).

3 Data-Set Used

We have used the ICCAD2012 dataset for this study. There are five sub-datasets with various layouts. The first dataset was produced using a 32 nm technology, whereas the remaining four were produced using a 28 nm technique. Training and test sets are included in each sub-dataset. Here each dataset has been prescribed Training Dataset and a Testing Dataset. These datasets are comprised of patterns from 2 different Product Design Kits (PDKs). The vast majority of these patterns are obtained from a 28 nm PDK, while less than 3 percentage are taken from a 32 nm PDK.

Table 1 shows the details of total number of images present in each sub-datasets including Training and Test datasets; this information is altered as required to our work and data augmentation has been done in this listed so there might be slight changes in the total image numbers and different from given image patterns. To study the ICCAD data set for the reference [9]. The results are those obtained after running our output through a model program iteration check; as a result, there are far fewer hotspots than non-hotspots in the results. A number of strategies, including data augmentation and filtering, have been employed to address the data extreme imbalance. False alarms are one of the issues which this dataset also has. These false alarms have been found to be greatly decreased when synthetic patterns are used to augment the amount of training data. Refer Fig. 3 for sample image dataset from

Table 1 Obtained dataset from ICCAD2012

S.no	Dataset details		
	Data sets	Training set	Test set
1	Sub-dataset-1	1690	8,190
2	Sub-dataset-2	10,958	84,562
3	Sub-dataset-3	11,104	54,582
4	Sub-dataset-4	9094	67,134
5	Sub-dataset-5	4,402	38,736
	Total	37,248	290,934

[a] Above details are obtained from ICCAD2012 conference

hotspot and non-hotspot training and test set of ICCAD1 2012, for example, (a)–(h) images illustrate few example images of hotspot and non-hotspot datasets which are obtained from ICCAD2012 conference. These images have been identified as hotspot and non-hotspot by the relay lines. In case of hotspot images, there are discontinuous relay lines and some overlap images are present, whereas in non-hotspot images have zero defect [10–13] (Fig. 4).

4 Proposed Model

The work was carried out using two models. In the first model we experimented on CNN. Model has multiple convolution stages with fully connected stages and pooling layers are also included. The basic work is feature extraction on input images.Relu that is used to maintain the model is non-linear. Next comes max pooling which reduces the spatial dimension of previous output and control over-fitting, then comes a fully connected layer. Hyper parameters of CNN model we used our work-type of Model is sequential, Batch size of 64, Image shape we used 224, Number of convolution layer are 3, Number of filters used id 12, Kernel size of (3, 3), Activation function is Relu, Number of Max-Pooling layers are 2, Pooling window of (2, 2), Number of dense layers 2, Number of Epochs is ¡= 10, Adam Optimizer, Loss plot calculation sparse categorical cross-entropy, Accuracy plotted in terms of metrics form. The model is binary classification. We have only identified whether the given image has a hotspot or not. Similarly the second model we used is VGG-11 model which is also used to identify the same issue but in this we have used the pre-trained model of VGG-11. The meaning of VGG is Visual Geometry Group. And the basic part of VGG is Alex Net; it is primarily capable of detecting objects. It takes into account over-fitting problems by using drop outs and data augmentation. By integrating ReLU's distinctive properties for over-pooling, it substitutes the tan-h activation function with ReLU. VGG entered the scene as a solution to CNN's lack of depth. It is a pre-trained model that was trained on a specific dataset and has weights that correspond to the features of that dataset. One can save time by employing a

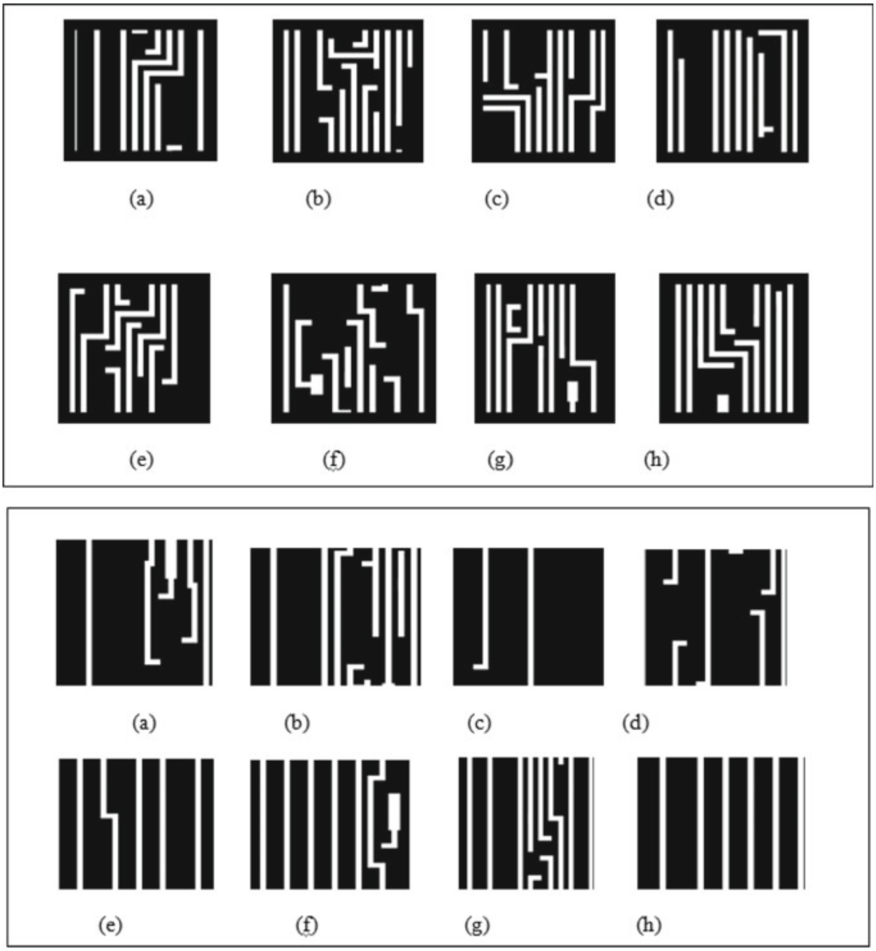


Fig. 3 Images of dataset obtained

pre-trained model. The model will probably profit from the sufficient amount of time and computational resources that have already been used to learn many features. The number following the keyword denotes how many weighted layers there are in the model. VGG models require an RGB image with a resolution of 224 × 224 pixels as input. The reader questions why only 224 of the RGB range's 0–255 pixels were considered in order to maintain a constant image size. The kernel size for the convolutional layer is 3 by 3. This 3 × 3 represents the convolution filter's size. To carry out the appropriate procedure, matrix elements are taken one at a time. Here, the function of ReLU activation is taken into account. From a group of pixels inside a kernel, the max-pooling layer returns the pixel with the highest value. One pixel is the fixed convolution stride. Here, the term "stride" refers to the convolutional

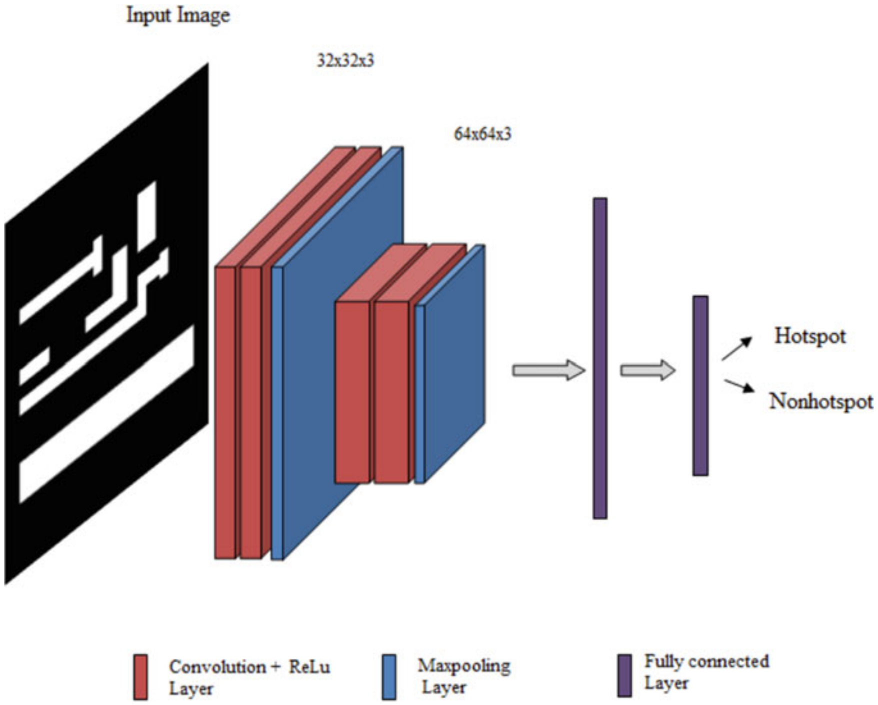


Fig. 4 Architectural representation of model

network step. The selection of the activation function, a non-linear transformation that determines a neural network's output, is a crucial factor in this case. ReLU is chosen because it has Streamlined Calculation, because it shortens the training and evaluation time frames, and huge neural networks are especially well suited to it. Leaky ReLU can prevent ReLU from saturating when the input is less than 0, which happens very seldom. ReLU lowers the computational expense of network training. As a result, larger nets with more parameters can be trained for the same computational cost. Three of the nodes in VGG-11 are completely connected. The diagrammatic representation of the 4096 channels in the upper two fully connected is shown in the image below. The third fully connected layer comprises 1000 channels, each of which belongs to a different class. Regarding VGG-11, it has 11 weighted layers, where weights are used to connect each neuron in one layer to each neuron in the next layer and are close to zero when indicating that changing an input will not change the output. Weights represent the strength of connections between units in adjacent network layers (Fig. 5).

As a feature map with the most noticeable features, the Max-Pooling layer is not counted as a weighted layer. A pooling procedure known as maximum pooling, sometimes known as max pooling, determines the maximum or greatest value in each patch of each feature map. In contrast to average pooling, which highlights

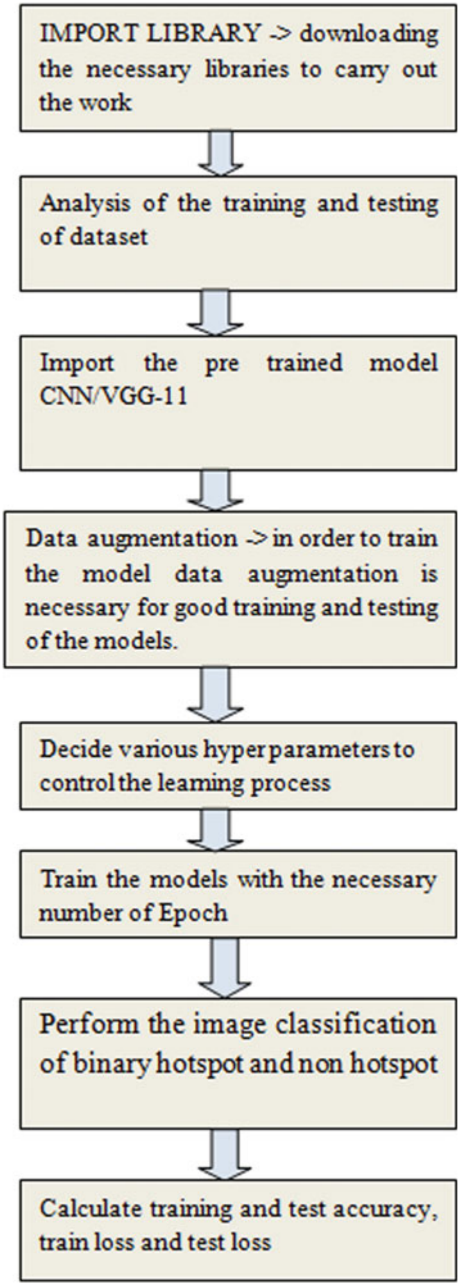

IMPORT LIBRARY -> downloading the necessary libraries to carry out the work
Analysis of the training and testing of dataset
Import the pre trained model CNN/VGG-11
Data augmentation -> in order to train the model data augmentation is necessary for good training and testing of the models.
Decide various hyper parameters to control the learning process
Train the models with the necessary number of Epoch
Perform the image classification of binary hotspot and non hotspot
Calculate training and test accuracy, train loss and test loss

Fig. 5 Steps carried throughout the project

the feature's average presence, the results are down-sampled or pooled feature maps that highlight the feature that is most prevalent in the patch. For computer vision applications like image classification, this has been found to perform better in practice than average pooling. To describe more about layers of VGG-11 it has convolution with 64 filters along with max pooling also 128,256,512 respectively. It has 4096 fully connected. Output layer has soft-max activation of thousand nodes respectively

5 Result and Discussion

CNNs and VGG-11 mode CNNs perform mediocrity at best, and at worst for sub-dataset 3. For sub-datasets 1, 2, 4, 5, and somewhat for sub-dataset, VGG-11 performs the worst 3.

Our proposed model to that of other studies referring to Table 2 the best result is provided. Sub-datasets 3, 4 and 5 provide the highest level of accuracy compared with other models. Although accuracy for sub-dataset 1 and 2 is not the highest of all, it is on par with top-performing models. The proposed model has been compared with old model because it was closer to our work study and it was best suited for our results. The readings are taken in the different iterations of the models and these readings may change for new executions (Table 3).

5.1 Result Graphs

Output graph is obtained after training of the datasets. The accuracy and validation of dataset has been carried out using CNN model and VGG-11 model respectively, whereas VGG-16 model failed to reach expected results hoping for improvement in the future work (Fig. 6).

Table 2 Obtained results from CNN and VGG-11 model

S.no	Accuracy in percentage		
	Data sets	CNN	VGG-11
1	Sub-dataset-1	93.47	87.45
2	Sub-dataset-2	98.81	90.40
3	Sub-dataset-3	90.91	86.90
4	Sub-dataset-4	97.47	89.80
5	Sub-dataset-5	92.89	91.90
6	Overall accuracy in percentage	93.90	88.90

Above readings obtained in different iteration of experiment

Table 3 Comparison with others work

S.no	Average accuracy in percentage		
	Data sets	Proposed model	Yu et al. [7]
1	Sub-dataset-1	93.48	93.81
2	Sub-dataset-2	93.37	98.02
3	Sub-dataset-3	**92.77**	91.95
4	Sub-dataset-4	**92.83**	85.94
5	Sub-dataset-5	**93.80**	92.86
6	Average percentage	**93.80**	92.53

The bold results show better results overall

Graph images in Fig. 5 illustrate the accuracy plot of the sub-dataset 1 and 5. We have put these two plots because of the good accuracy result rate of the model.

6 Conclusion

We used CNNs and VGG-11 and studied ViT to tackle this problem to test if the proposed technique is able to identify the hotspot and non-hotspot image classification. Overall accuracy is given 93% and as listed in the result table. We also compared our study to previously published research, and found that our model performed best overall in terms of accuracy, also performed best or similarly for three out of five sub-data sets, but lagged for two of them. The results show that while the suggested strategy outperforms many state-of-the-art procedures, it falls short of completely replacing all of them for all of the sub-data sets. It is possible to consider our model as a fresh and different approach to locating hotspots in lithography. The technique has a lot of room for growth because it is so innovative. For our issue statement, an increase in accuracy and a reduction in time are also possible using the new techniques. Future study will continue to focus on developing our model, altering the data set to lessen imbalance and increase training data, as well as reducing the number of iterations required for this technique and improving accuracy for sub-data sets 1 and 3. As we tried to apply these data sets, we were not able to get the same results as expected and these data sets do not fit for the VGG-16 model as for now but there might be room for improvement in the future work.

Acknowledgements The work is supported by KLE Technological University Hubballi-580024 under the guidance of Dr. Suneeta V. Budihal.

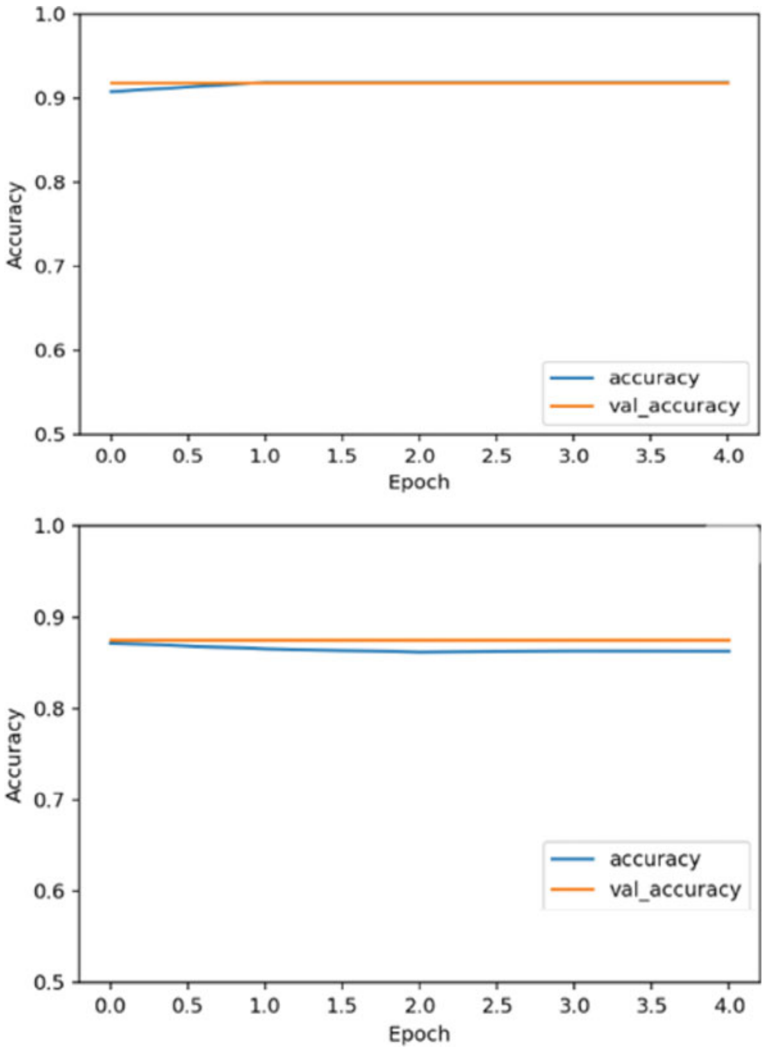


Fig. 6 Output graph of CNN and VGG-11

References

1. Prof. Dasgupta N. Lithography Lecture-1 and Lecture-1. https://nptel.ac.in/courses/117/106/117106093
2. Gkelios S, Boutalis Y, Chatzichristofis SA (2021) Investigating the vision transformer model for image retrieval tasks
3. Tomioka Y, Matsunawa T, Kodama C, Nojima S (2017) Lithography hotspot detection by two-stage cascade classifier employing histogram of directed light propagation

4. Yu Y, Chan Y, Sinha S, Jiang IH, Chiang C (2012) Accurate process-hotspot detection via critical design rule extraction. DAC Des Autom Conf 2012:1163–1168
5. Yang H, Lin Y, Yu B, Young EFY (2017) Lithography hotspot detection: from shallow to deep learning. IEEE international system-on-chip conference (SOCC). Munich, Germany, pp 233–238
6. Ye W, Alawieh MB, Lin Y, Pan DZ (2019) Litho GAN: end-to-end lithography modeling with generative adversarial networks. In: The 56th annual design automation conference 2019. https://doi.org/10.1145/3316781.3317852
7. Ye W, Lin Y, Li M, Liu Q, Pan DZ et al (2019) Litho ROC: lithography hotspot identification with explicit ROC optimization. In: ASPDAC '19: proceedings of the 24th Asia and South Pacific design automation conference. https://doi.org/10.1145/3287624/3288746
8. Torres JA (2012) ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite. IEEE/ACM international conference on computer-aided design (ICCAD) 2012:349–350
9. Suneeta VB, Purushottam P, Prashantkumar K, Sachin S, Supreet M (2020) Facial expression recognization using supervised learning. In: International conference on computational vision and bio inspired computing ICCVBIC 2019: computational vision and bio-inspired computing, pp 275–285
10. Pavaskar S, Budihal S (2018) Real-time vehicle-type categorization and character extraction from the license plates. In: Cognitive informatics and soft computing, pp 557–565
11. Borisov V, Scheible J (2018) Lithography hotspots detection using deep learning. In: 2018 15th international conference on synthesis, modeling, analysis and simulation methods and applications to circuit design (SMACD), pp 145–148. https://doi.org/10.1109/SMACD.2018.8434561
12. Suneeta VB et al (2019) Facial expression recognition using supervised learning. In: International conference on IoT, social, mobile, analytics, and cloud in computational vision and bio engineering (ISMAC-CVB 2019) held on March 13–14, 2019 at Vivekananda College of Technology for Women, Tamil Nadu, India, pp 275–285
13. Choudhari R, Shivakumar M, Nandavar S, Maigur S, Siddamal SV, Budihal SV, Pattanshetti SM (2022) Decentralized and secured voting system with blockchain technology, pp 167–182

Bidirectional Certificateless Searchable Authenticated Encryption for Encrypted Email Application in IoT

Venkata Bhikshapathi Chenam and Syed Taqi Ali

Abstract Today, the Internet of Things (IoT) framework can be incorporated into many fields (i.e., industrial, medical, academic, enterprises, etc.) that require an encrypted email application to exchange information. This application has a huge number of emails generated through the inbox and outboxes of their profiles that can't be maintained by the system users locally, so they use the email gateway server services to store that data. To ensure the security of user data while searching, a Certificateless Searchable Authenticated Encryption (CLSAE) scheme is suitable for the IoT environment. But, the maximum of CLSAE schemes used costly bilinear pairing and was proven secure in the Random Oracle Model (ROM). Later, the researcher introduced another CLSAE system that does not require pairing and ROM, but the searching operation was not flexible enough to reduce the real-time scenario. This paper presents the Bidirectional Certificateless Searchable Authenticated Encryption (BCLSAE) scheme in the Standard Model (SM), which includes individual cipher-keywords algorithms for both sender and receiver, bidirectional search operation, and multi-keyword search. Furthermore, the system provides privacy of keyword information in the trapdoor, which leads to resistance to Keyword Guessing Attacks (KGAs) (launched by both an outsider and an insider). Finally, we compare the BCLSAE with related schemes, which outperform them.

Keywords Certificateless · Multi-keyword search · Bidirectional search · Without pairing · Standard model · IoT

V. Bhikshapathi Chenam (✉) · S. Taqi Ali (✉)
Computer Science and Enginerring Department, Visvesvaraya National Institute of Technology, Nagpur 440010, Maharashtra, India
e-mail: chenamvenkat@gmail.com

S. Taqi Ali
e-mail: sta@cse.vnit.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_26

1 Introduction

Applications for the IoT span many fields and usage scenarios. Nevertheless, there is one element that all integrated IoT systems have in common: they are made up of four distinct parts: devices, connectivity of devices, processing of data, and user interface. 1. *Devices*: Consider different types of smart devices, namely, sensors (i.e., information gathering sensors in all fields), multiple sensors that can be bundled together (i.e., mobiles, tablets, smart devices available in all areas), and wireless networks (i.e., network managed by the connectively by multiple things). The primary purpose of the devices or networks is to collect information from the corresponding environment. 2. *Connectivity of devices*: Here, several techniques are used to link devices to a third party, known as "cloud storage." For instance, connectivity options include: (a) Wireless networks (i.e., cell phones, wireless sensors, satellite communication, microwave, bluetooth, and wireless fidelity). (b) Connecting through a gateway. (c) Connecting directly to the internet through ethernet. 3. *Processing of data*: A certain type of processing is done on the data once it is in the cloud by software. 4. *User interface*: The interface enables the system users to access the system routinely. For example, a user might utilize a mobile app or website to view the feeds of the videos in their home. However, it's not always a one-way track. The user can behave and affect the system depending on the IoT application. For example, A user can remotely modify the temperature in cold storage using an app on their mobile.

Therefore, the IoT framework can be incorporated into different areas that are defined as follows: (1) The Industrial IoT, abbreviated as IIoT, (2) The Medical IoT, abbreviated as MIoT, (3) The Consumer IoT, abbreviated as CIoT, and (4) The Enterprise IoT, abbreviated as EIoT. Any IoT infrastructure described above requires an encrypted email application to exchange emails between IoT users. However, this application provides user profiles that include the inbox and outbox with emails. However, IoT users can send or receive emails to or from other IoT users via outbox or inbox. Unfortunately, users cannot save their entire profile locally; instead, they must outsource it to a third party called an email gateway server. However, users are concerned about sensitive email security because a third party is not trusted. Therefore, the searchable encryption mechanism is appropriate for resolving the issue because it allows for an efficient search on encrypted data without revealing any information about the searched one.

1.1 Literature Survey

The Searchable Encryption (SE) cryptographic approach allows retrieval of data stored on the cloud server without disclosing the information of documents and the searched keywords. Song et al. [1] were created as an SE using symmetric notation, which is referred to as SSE. Unfortunately, a major flaw in SSE systems is the distribution of keys between the system users. According to Boneh et al. [2], who presented an SE scheme using asymmetric cryptographic notation, known as ASE, this solves the key distribution problem. Following that, Baek et al. [3] presented a new ASE system that eliminates the overhead of ASE (i.e., a secure channel is required to transfer the trapdoor to the cloud server). Later, researchers developed many ASEs that are extensions of the [3]th scheme in terms of improving security, withstanding keyword guessing attacks, and searching functionality (i.e., single, conjunctive keyword) in either a ROM [4–6] or SM [7–9]. However, those were based on the Public Key Infrastructure (PKI) mechanism.

To eliminate certificate management issues, Abdalla et al. [10] developed a new SE scheme based on identity-based cryptographic notations, which is referred to as IBSE. Later, researchers developed various IBSE schemes [11–13]. But, the key escrow issue is inherently available in IBSE schemes since a Private Key Generator (PKG) may get all users' private-keys. To avoid the problems of ASE and IBSE, Peng et al. [14] developed the new SE using certificateless cryptographic notations, which is referred to as a CLSE system. In CLSE, the system user's private key comprises a user-generated value (i.e., secret value) and the Key Generation Center (KGC) generated value (i.e., partial private key). In recent years, many CLSE systems [15–25] have been suggested.

Keyword Guessing Attack (KGA): In the KGA attack, the attacker can try to find the keyword information in the trapdoor algorithm, which is easily possible in the standard SE schemes. As a solution to KGA, Rhee et al. [5] developed a designated ASE system that means a testing algorithm executed by a cloud server and provides a sufficient security property to resist the KGA attack, which is called trapdoor indistinguishability. After that, researcher analysis of the dASE scheme resists KGA by an outsider (i.e., except the system users). Unfortunately, the same attack can be performed from inside the system (i.e., a malicious cloud server), which is referred to as an IKGA. As a solution to IKGA, Huang and Li [26] developed a new ASE scheme that includes authentication in the encryption of keywords algorithm, which is referred to as an ASAE and resists KGA attacks. Next, He et al. [15] developed a new CLSE that includes an authenticated property in the cipher-keywords algorithm, abbreviated as CLSAE. Following that, Several CLSAE systems and variations have been suggested [16–24]. According to Pakniat et al. [20], the inadequately specified security models cause [15–17] systems to suffer from major flaws. After that, they created a new CLSAE scheme, defined new security models, and proved them for CLSAE. However, Chennm et al. [27] present the first CLSAE scheme with a conjunctive keyword search for multi-receiver and secure against KGAs.

Motivation and Contribution

IoT systems combine many different gadgets, including sensors, controllers, and smart devices, producing enormous volumes of data. Data security is guaranteed by transmitting and storing this information in encrypted form, but the challenge of ciphertext retrieval hampers flexible data invocation. The SE mechanism is suitable for sending encrypted data and performing a search on the encrypted data. Among the SE schemes, the SAE is the most efficient (i.e., it withstands the KGAs). It also contains a search function for the retrieval of specified ciphertext. The scheme described above is based on the PKI mechanism, which requires extra storage and communication to maintain the certificates of system users. Next, IBC does not require certificate management but has a key escrow issue. Because PKG owns all users' private-keys, its malevolent conduct threatens the information security of all users. To remove key escrow, the CLC scheme has the private key of the system user split into two components: KGC creates one component, and the user chooses another. To summarize, CLSAE is better suited for the IoT environment. CLSAE's benefits have prompted the development of several CLSAE systems [15–24]. Some of these schemes are intended for the IoT [15, 20–22, 24].

The following is a list of the contributions to this article.

- This paper introduces a new BCLSAE scheme for encrypted email systems in the IoT environment that supports bidirectional multi-keyword search and cipher-keywords algorithms.
- Each user in this system has a profile that includes an inbox (i.e., received emails) and an outbox, according to the encrypted email system (i.e., sent emails). As a result, each system user must create a profile in order to utilize the cipher-keywords algorithm, and they must also want to get a certain email from that profile in order to do a search operation. Here, the results of the search operation are more practical since it supports multi-keyword search.
- In the security model, the proposed strategy has been demonstrated to be capable of withstanding KGA against CLC adversaries (i.e., providing the trapdoor indistinguishable security is sufficient). In the security analysis, the adversaries use the real hash algorithms to generate the hash value.
- Finally, the proposed scheme's performance against other related CLSAE methods is compared and avoids performing costly operations, resulting in a computational cost that is often lower than that of previous CLSAE techniques, making it more appropriate for the IoT environment.

2 Definition of the Proposed Scheme

This section first presents some background information that we used in our scheme before presenting the BCLSAE system and security architecture.

Definition 1 Elliptic Curve Group: Let G be an elliptic curve (i.e., $y^2 = x^3 + ax + b$) group defined over the finite field F_q of prime order (i.e., $q > 3$). The number of points in the G is $\{(x, y) \in F_q \times F_q\}$ which includes the point at infinity. Then, a and b are two integers in the curve that belongs to F_q and satisfy the condition always $4a^3 + 27b^2 \neq 0$.

Table 1 Notations are used in the proposed scheme

Notation	Meaning
λ	Security parameter
$param$	Public parameter
msk	System master key
ID_i	Identity of the ith system user ($i \in \{S, R\}$)
S	Sender
R	Receiver
PPK_i	Partial private key of ith system user
PK_i	Public key of ith system user
SK_i	Private key of ith system user
KW	Set of keywords
kw'	Keyword used in the trapdoor
Ge	Generate Email Person (i.e., $Sor R$)
Re	Receive Email Person (i.e., $Sor R$)
Sp	Searching Person (i.e., $Sor R$)
Ep	Searching Related Email Person (i.e., $Sor R$)
Es	Email gateway server

Definition 2 Elliptic Curve Decisional Diffie-Hellman (ECDDH): The following is an explanation of the ECDDH problem. A challenger $\mathcal{C}$ chooses a cyclic elliptic curve prime order (q) group G. Assume P is a generator of G and $\{a, b\} \in Z_q^\star$ at randomly. It is difficult for the adversary $\mathcal{A}$ to differentiate a element $abP \in G$ from a random element $X \in G$ if $\mathcal{C}$ offers it to the adversary as a tuple (P, aP, bP, X). If a polynomial time algorithm (Alg) exists to solve the ECDDH assumption in elliptic curve group G, then the advantage of Alg is negligible.

Figure 1 provides a general architecture of the proposed scheme, and Table 1 describes the notations used in the scheme. The system consists of four distinct users: KGC, Sender, Receiver, and an email gateway server. Below are the responsibilities they played in the system.

1. ***Key Generation Center, KGC***: The KGC is an authorized entity in certificateless cryptography that uses the $Setup$ procedure to generate public system parameters $param$ and a master private key msk. Additionally, it generates a partial private key for each user using msk and shares it securely. The procedure described above is as follows:

   - $Setup(\lambda)$: Input a security parameter λ, and output $param$ and msk.
   - $Extract\text{-}partial\text{-}private\text{-}key(msk, param, ID_i)$: Input a system master key msk, system parameters $param$, and identity of system user $ID_i\big(i \in \{S, R\}\big)$. The output of algorithm is partial private key of system user PPK_i.

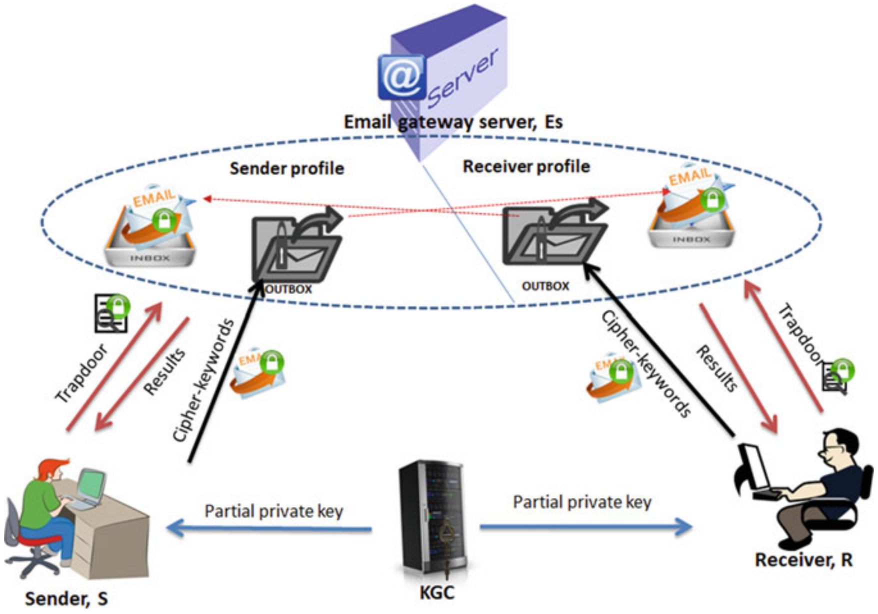


Fig. 1 Architecture of proposed scheme

2. ***Respective system user***, $i \in \{S, R\}$: The relevant system user produces the public key and private key after obtaining a PPK from KGC. The procedure described above is as follows:

   - $Extract\text{-}public\ and\ private\text{-}keys(param, PPK_i, ID_i)$: Input a system parameters $param$, partial private key of system user prk_i', and identity of system user $ID_i\big(i \in \{S, R\}\big)$. Output public key PK_i and Private key SK_i of the respective system user i.

3. ***Sender, S***: The sender has a profile on an email gateway server that contains an inbox and an outbox. The inbox consists of email sent by other users, and the outbox contains emails sent to other users. First, the sender sends the email to the other along with cipher-keywords. Second, S also performs search operations on his profile. The procedure described above is as follows:

   - $Cipher\text{-}keywords(param, SK_S, KW, PK_R) :\longrightarrow C^S_{KW}$
   - $Trapdoor(param, SK_S, kw', PK_R) :\longrightarrow T^S_{kw'}$

4. ***Receiver, R***: Similar to the sender, the receiver can have an inbox and an outbox. The procedure described above is as follows:

   - $Cipher\text{-}keywords(param, SK_R, KW, PK_S) :\longrightarrow C^R_{KW}$
   - $Trapdoor(param, SK_R, kw', PK_S) :\longrightarrow T^R_{kw'}$

5. ***Email gateway server, Es***: The Es is accountable for maintaining system users' $(\{S, R\})$ cipher-keywords and encrypted email. Next, it performs the search operation requested by the system users on their profiles, and a result of 1 or 0 is returned.
    - $Search(param, C_{KW}^{Ge}, T_{kw'}^{Sp})$: Input a system parameters $param$, a cipher-keywords C_{kw}^{Ge} (i.e., $Ge \in \{S, R\}$), and a trapdoor $T_{kw'}^{Sp}$ (i.e., $Sp \in \{S, R\}$). If $kw' \in KW$, output returns 1; otherwise, it returns 0.

Note: In this context, Ge = Generate Email Person, Re = Receive Email Person, Sp = Searching Person, and Ep = Searching Related Email Person. But, never Ge = Re and Sp = EP.

- $Cipher\text{-}keywords\big(param, SK_{Ge}, KW, PK_{Re}\big)$: The cipher-keywords C_{KW}^{Ge} of the keyword set KW is returned when a system parameter $param$, a keyword set KW, a Ge private key SK_{Ge}, and a Re public key PK_{Re} are entered.
- $Trapdoor(param, SK_{Sp}, kw', PK_{Ep})$: Input a system parameters $param$, a keyword kw', a Sp private key SK_R, and a Ep public key PK_S. Output a trapdoor of receiver is $T_{kw'}^{Sp}$ of the keyword kw'.

Note: The attached documents in the email can be encrypted and decrypted using standard asymmetric cryptography.

2.1 Adversaries and Security Model

The system users (S and R) are authenticated users of data to be stored on the email gateway server, so they have no motivation to be attackers on the IoT infrastructure. However, the Es is honest but curious, which implies that the Es will execute search operations and return results honestly but hopes to collaborate with unlawful users to guess the keyword value from cipher-keywords or trapdoor algorithms. As a result, we refer to the Es as an inside attacker and the external users (i.e., except S, R, and Es) as external attackers.

The proposed scheme's security consists of two components: trapdoor indistinguishability against chosen keyword to trapdoor attack ($TI\text{-}CKT$) and cipher-keywords indistinguishability against chosen keyword to cipher-keywords attack ($CI\text{-}CKC$). According to the cryptography of the certificateless [28], the proposed scheme consists of two adversaries, namely Type 1 ($\mathcal{A}_1$) and Type 2 ($\mathcal{A}_2$). $\mathcal{A}_1$ has the ability to replace the system user's public key with a duplicated value but does not have access to the msk. Although $\mathcal{A}_2$ has the ability to get the msk, it cannot replace the value of the system user's public key. A searchable encryption method is deemed more secure if it offers protection from KGA attacks. Furthermore, the authors provide a technique to secure SE schemes: trapdoor security. This is necessary to demonstrate the scheme's security against KGA attacks [5]. Therefore, we are providing only $TI\text{-}CKT$ security for the proposed scheme.

TI-CKT$_{\mathcal{A}}$: The challenger $\mathcal{C}$ and adversary $\mathcal{A} \in \{\mathcal{A}_1, \mathcal{A}_2\}$ are the players in this game.

- `Initialize`: In order to produce the *param* and *msk*, the challenger runs *Setup*. For example, if $\mathcal{A} = \mathcal{A}_1$, the $\mathcal{C}$ sends only *param*; otherwise, the $\mathcal{C}$ sends both to the adversary.
- `Phase1`: $\mathcal{C}$ responds to the following queries and oracles (orcl) made adaptively by $\mathcal{A}$.
  - $ReqPK\text{-}query(ID_i)$: The adversary $\mathcal{A}$ asks public key of an identity ID_i. Then, the challenger $\mathcal{C}$ response PK_{ID_i}.
  - $EPPK\text{-}query(ID_i)$: The adversary $\mathcal{A}_1$ asks partial private key of an identity ID_i. The challenger $\mathcal{C}$ response partial private key of the ID_i .
  - $RepPK\text{-}query(ID_i, \overline{PK_i})$: The adversary $\mathcal{A}_1$ asks to replace the public key of ID_i with a duplicate value. As a result, the public key of ID_i is replaced by $\overline{PK_i}$ and set the corresponding secret value by $\perp$ (i.e., $\perp$ is an empty string).
  - $ExtSV\text{-}query(ID_i)$: The adversary asks the secret value of ID_i. Then challenger responds x_i to the adversary when he does not ask $RepPK$ query on ID_i.
  - $Cipher\text{-}keywords\text{-}orcl(ID_{Ge}, ID_{Re}, KW)$: The adversary asks cipher-keywords oracle on ID_{Ge}, ID_{Re} and KW. Challenger responds as corresponds output value.
  - $Trapdoor\text{-}orcl(ID_{Sp}, ID_{Ep}, kw')$: The adversary asks trapdoor query on ID_{Sp}, ID_{Ep} and kw'. Challenger responds as corresponds output value.
- `Challenge`: $\mathcal{A}$ outputs a Sp identity $ID_{Sp^\star}$, a Ep identity $ID_{Ep^\star}$, and two challenge keywords kw_0 and kw_1 ($\neq kw_0$). Next, the challenger $\mathcal{C}$ randomly selects one keyword from above we call it kw_b and performs $Trapdoor$ algorithm to output the challenge trapdoor $T_{kw_b}^{Sp^\star}$ and returns it to $\mathcal{A}$.
- `Phase2`: The $\mathcal{A}$ can keep asking the same number of queries as it did in `Phase1`.

.

- `Guess`: During this phase, it determines if $b = b'$, then $\mathcal{A}$ returns a guess value b' is correct and wins the game; otherwise, it fails. The following constraints should be considered by the adversary in the preceding game:
  - If $\mathcal{A} = \mathcal{A}_1$ has never been queried $EPPK\text{-}query$ on $ID_{Sp^\star}$ and $ID_{Ep^\star}$ otherwise never been queried $ExtSV\text{-}query$ on $ID_{Sp^\star}$ and $ID_{Ep^\star}$.
  - $\mathcal{A}$ has never been queried $Trapdoor\text{-}orcl(ID_{Sp^\star}, ID_{Ep^\star}, kw_{0\ or\ 1})$.

The advantage of $\mathcal{A}$ is defined as $Adv_{CLBSE,\mathcal{A}_{0\ or\ 1}}^{TI\text{-}CKT} = \left|2Pr[b' = b] - 1\right|$.

3 Proposed Scheme

- *Setup*: KGC takes as input a security parameter λ as input and performs the following:

- It chooses an elliptic curve group G of order q on addition operation over finite filed F_q and a generator $P \in G$. Next, chooses $msk \in Z_q^\star$ at random and computes the master public key $P_{pub} = mskP$.
- It selects the secure hash functions: $h_1 : \{0, 1\}^\star \times G \to Z_q^\star, h_2 : \{0, 1\}^\star \times G^2 \times \{0, 1\}^\star \times G^4 \times \{0, 1\}^\star \to Z_q^\star$, and $h_3 : \{0, 1\}^\star \to Z_q^\star$.
- Finally, KGC publish the public parameters $params = \{\lambda, G, q, P, P_{pub}, h_1 - h_3\}$ and keeps secret msk.

- *Extract-partial-private-key*: KGC takes as input a master secret key msk, public parameters $param$, and the identity of a system user ID_i (i.e., $i \in \{R, S\}$) and performs the following:

  - It selects r_i at random from $Z_q^\star$ and then calculates $pk_i' = r_i P, \alpha_i = h_1(ID_i, pk_i')$, and $sk_i' = r_i + msk\alpha_i$ (mod q).
  - Finally, it initializes $PPK_i = (pk_i', sk_i')$, sends sk_i' through a secure channel to the corresponding system user i ($\in \{R, S\}$), and publishes pk_i'.

- *Extract-public and private-keys*: Respective system user, $i \in \{S, R\}$ takes a public parameter $param$, a partial private key PPK_i, and the system user's identity ID_i as input and then performs the following:

  - It takes a random sk_i'' from the $Z_q^\star$ and computes $pk_i'' = sk_i'' P$.
  - Finally, it initializes $PK_i = (pk_i', pk_i'')$,$SK_i = (sk_i', sk_i'')$ and publishes PK_i.

- *Cipher-keywords*: As inputs, the Ge takes public parameters $param$, a keyword set $KW = \{kw_j\}_{j=1}^m$,the Re identity and public key ID_{Re}, PK_{Re}, and the Ge identity and secret key ID_{Ge}, SK_{Ge}. Then it computes as follows:

  - $\alpha_{Re} = h_1(ID_{Re}, pk_{Re}')$, $K_1 = sk_{Ge}'(pk_{Re}' + \alpha_{Re} P_{pub})$, $K_2 = sk_{Ge}'' pk_{Re}''$.
  - It selects a random c' from the $Z_q^\star$ and then computes $c_{kw_j}' = h_3\big(c', h_2(ID_{Ge}, pk_{Ge}'', pk_{Ge}', ID_{Re}, pk_{Re}'', pk_{Re}', K_1, K_2, kw_j)\big)$ for $j = 1$ *to m*.
  - Finally, it sends $C_{KW}^{Ge} = \{c', c_{kw_1}', c_{kw_2}', \cdots, c_{kw_m}'\}$ to the Es.

- *Trapdoor*: As inputs, the Sp takes public parameters $param$, a keyword kw',the Ep identity and public key ID_{Ep}, PK_{Ep}, and the Sp identity and secret key ID_{Sp}, SK_{Sp}. Then it computes as follows:

  - $\alpha_{Ep} = h_1(ID_{Ep}, pk_{Ep}')$, $K_1' = sk_{Sp}'(pk_{Ep}' + \alpha_{Ep} P_{pub})$, $K_2' = sk_{Sp}'' pk_{Ep}''$, and $T_{kw'}^{Sp} = h_2(ID_{Ep}, pk_{Ep}'', pk_{Ep}', ID_{Sp}, pk_{Sp}'', pk_{Sp}', K_1, K_2, kw')$.
  - Finally, it sends $T_{kw'}^{Sp}$ to the Es.

- *Search*: Es takes $params$, C_{KW}^{Ge}, and $T_{kw'}^{Sp}$ as inputs, then checks whether $c_{kw_j}' = h_3(c', T_{kw'}^{Sp})$ for $j = 1$ *to m* satisfies and outputs 1; otherwise, outputs 0.

Correctness:

If $Ge = Sp$ and $Re = Ep$. Then $K_1 = K'_1$ and $K_2 = K'_2$ are equal values. Otherwise, we are calculating the following way.

1. $K_1 = sk'_{Ge}(pk'_{Re} + \alpha_{Re}P_{pub}) = (r_{Ge} + \alpha_{Ge}msk)(r_{Re} + \alpha_{Re}msk)P = sk'_{Sp}$ $(pk'_{Ep} + \alpha_{Ep}P_{pub}) = K'_1$.
2. $K_2 = sk''_{Ge}pk''_{Re} = sk''_{Ge}sk''_{Re}P{=}sk''_{Sp}pk''_{Ep}P = K'_2$.

After that, we can calculate $h_2(c', T^{Sp}_{kw'})$ and verify it with $\{c'_{kw_j}\}^m_{j=1}$. If a match occurs (i.e., $kw' \in KW$), then the search results are true.

4 Security Proofs

The BCLSAE's TI-CKT security in the standard model is examined in this section.

Theorem 1 *In the standard model, the proposed approach provides TI-CKT security under the ECDDH assumption. However, Lemmas 1 and 2 will be used to demonstrate this theorem.*

Lemma 1 *The challenger $\mathcal{C}$ can solve the ECDDH assumption with an advantage ε' in the standard model if the adversary $\mathcal{A}_1$ can break the proposed scheme with a non-negligible advantage ε.*

Proof. Given a tuple (P, aP, bP, X) as an instance of the ECDDH assumption, the challenger must determine if $X = abP$ with the assistance of the adversary.

- `Initialize:` The challenger $\mathcal{C}$ invokes the $Setup$ algorithm to generate the $params$ and msk, then returns the $params$ to the adversary $\mathcal{A}_1$ while keeping the msk secret.
- `Phase1:` $\mathcal{A}_1$ adversary can asks queries and oracles made by adaptively to $\mathcal{C}$.
  - $EPPK$-$query$: If an adversary $\mathcal{A}_1$ sends an ID_i to the query, then $\mathcal{C}$ keeps a list L_{ppk} of tuples $(ID_i, r_i, pk'_i, sk'_i)$. Later, if $ID_i \in L_{re}$ then $\mathcal{A}_1$ is not permitted to perform this query. Alternatively, $\mathcal{C}$ performs the following:

    If $i \in \{S^\star, R^\star\}$, Then $\mathcal{C}$ adds the two tuples $(ID_{S^\star}, \perp, aP, \perp)$, $(ID_{R^\star}, \perp, bP, \perp)$ to the list L_{ppk} and terminate the process.
    Otherwise, $\mathcal{C}$ finds a tuple in L_{ppk} with entry ID_i and returns sk'_i. If a tuple is not found with entry ID_i, then $\mathcal{C}$ return sk'_i by execute the $Extract$-$partial$-$private$-key algorithm and add the tuple $(ID_i, r_i.pk'_i, sk'_i)$ to the list L_{ppk}.

  - $ExtSV$-$query$: If an adversary $\mathcal{A}_1$ sends an ID_i to the query, then $\mathcal{C}$ keeps a list L_{sv} of tuple (ID_i, sk''_i). Later, if $ID_i \in L_{re}$ then $\mathcal{A}_1$ is not permitted to perform this query. Alternatively,$\mathcal{C}$ performs the following:

If a tuple is finds in the list L_{sv} then return sk_i''.
Otherwise, $\mathcal{C}$ selects sk_i'' at random from the $Z_q^\star$, adds a tuple to the list L_{sv} and retun sk_i''.

- *ReqPK-query*: If an adversary $\mathcal{A}_1$ sends an ID_i to the query, then $\mathcal{C}$ keeps a list L_{pk} of tuple (ID_i, PK_i). Later, if $ID_i \in L_{re}$ then $\mathcal{A}_1$ is not permitted to perform this query. Alternatively,$\mathcal{C}$ performs the following:

  If a tuple is found in the list L_{pk} with entry ID_i then return PK_i.
  Otherwise, refers the lists L_{ppk} and L_{sv}, then computes $pk_i'' = sk_i''P$, set $PK_i = (pk_i', pk_i'')$, and adds the tuple to list L_{pk} and retun PK_i.

- *RepPK-query*: If an adversary $\mathcal{A}_1$ sends an ID_i to the query, then $\mathcal{C}$ keeps a list L_{re} of tuple $(ID_i, \overline{pk_i'}, \overline{pk_i''})$. Next, $\mathcal{C}$ search the entry with ID_i in the list is found to do nothing. Otherwise, $\mathcal{C}$ append a tuple $(ID_i, \overline{pk_i'}, \overline{pk_i''})$ to the list L_{re} and update the lists L_{pk},L_{ppk}, and L_{sv} with values $(ID_i, \overline{PK_i})$, $(ID_i, \perp, \overline{pk_i'}, \perp)$, and $(ID_i, \perp)$, respectively.
- *Cipher-keywords-orcl*: When $\mathcal{A}_1$ submits a tuple (ID_{Ge}, ID_{Re}, KW), then $\mathcal{C}$ does as follows:

  If $\{Ge, Re\} \notin \{S^\star, R^\star\}$ or $\{R^\star, S^\star\}$, then

  · If $\mathcal{A}_1$ not perform the *RepPK-query* on both Ge and Re, then $\mathcal{C}$ generate C_{KW}^{Ge} by calling the *Cipher-keywords* algorithm returns.
  · If $\mathcal{A}_1$ already perform the *RepPK-query* on either Ge or Re, then $\mathcal{C}$ generate C_{KW}^{Ge} by calling the *Cipher-keywords* algorithm except K_1 and K_2 calculated by the private key of either Re or Ge and returns.

  If Ge or Re $\in \{S^\star, R^\star\}$ and $\mathcal{A}_1$ not perform the *RepPK-query* on Re or Ge $\notin \{R^\star, S^\star\}$, then $\mathcal{C}$ generate C_{KW}^{Ge} by calling the *Cipher-keywords* algorithm except K_1 and K_2 calculated by private key of Re or Ge and returns.
  Otherwise, terminates the process.

- *Trapdoor-orcl*: When $\mathcal{A}_1$ submits a tuple (ID_{Sp}, ID_{Ep}, kw'), then $\mathcal{C}$ does as follows:

  If $\{Sp, Ep\} \notin \{S^\star, R^\star\}$ or $\{R^\star, S^\star\}$, then

  · If $\mathcal{A}_1$ not perform the *RepPK-query* on both Sp and Ep, then $\mathcal{C}$ generate $T_{kw'}^{Sp}$ by calling the *Trapdoor* algorithm returns.
  · If $\mathcal{A}_1$ already perform the *RepPK-query* on either Sp or Ep, then $\mathcal{C}$ generate $T_{kw'}^{Sp}$ by calling the *Trapdoor* algorithm except K_1 and K_2 calculated by the private key of either Ep or Sp and returns.

  If Sp or Ep $\in \{S^\star, R^\star\}$ and $\mathcal{A}_1$ not perform the *RepPK-query* on Ep or Sp $\notin \{R^\star, S^\star\}$, then $\mathcal{C}$ generate $T_{kw'}^{Sp}$ by calling the *Trapdoor* algorithm except K_1 and K_2 calculated by private key of Ep or Sp and returns.
  Otherwise, terminates the process.

- `Challenge`: $\mathcal{A}_1$ confirmed the `Phase1` over and outputs a tuple $(ID_{Sp^\star}, ID_{Ep^\star}, kw_0, kw_1)$ to the challenge phase, where $kw_0 \neq kw_1$ and $|kw_0| \neq |kw_1|$. Now, $\mathcal{C}$ choose a bit $b \in \{0, 1\}$ at random, kw_b as input and does as follows:
  - If $\mathcal{A}_1$ not perform the $RepPK\text{-}query$ on both $Sp^\star$ and $Ep^\star$, Then

    Recoveries the lists L_{ppk}, L_{sv}, and L_{pk} with entries $ID_{Sp^\star}$, $ID_{Ep^\star}$ and computes $\alpha_{Ep^\star} = h_1(ID_{Ep^\star}, pk'_{Sp^\star})$, $\alpha_{Sp^\star} = h_1(ID_{Sp^\star}, pk'_{Sp^\star})$. $K'^\star_1 = X + \alpha_{Ep^\star} mskbP + \alpha_{Sp^\star} mskaP + \alpha_{Ep^\star}\alpha_{Sp^\star} mskP_{pub}$, $K'^\star_2 = sk''_{Sp^\star} pk'_{Ep^\star}$, and $T^{Sp^\star}_{kw_b} = h_2(ID_{Sp^\star}, pk''_{Sp^\star}, aP, ID_{Ep^\star}, pk''_{Ep^\star}, bP, K'^\star_1, K'^\star_2, kw_b)$.
  - Otherwise, terminated the process.
    Finally, output the challenge trapdoor $T^{Sp^\star}_{kw_b}$.
- `Phase2`: $\mathcal{A}_1$ cloud performs the various queries and oracles similar to the `Phase1`.
- `Guess`: After $\mathcal{A}_1$ return guessed value b'. The game $TI\text{-}CKT_{\mathcal{A}_1}$ is won if $b = b'$.

Solve the ECDDH problem: Assume that the guess value of $\mathcal{A}_1$ is correct (i.e., $\mathcal{C}$ returns 1 if $b' = b$). As a result, the X value used in the `Challenge` phase is abP, indicating that the challenge trapdoor $T^{Sp^\star}_{kw'_b}$ is valid.

$$\begin{aligned} K'^\star_1 &= X + \alpha_{Ep^\star} mskbP + \alpha_{Sp^\star} mskaP + \alpha_{Ep^\star}\alpha_{Sp^\star} mskP_{pub} \\ &= (a + \alpha_{Sp^\star msk})(b + \alpha_{Ep^\star msk})P \\ &= sk'_{Sp^\star}(pk'_{Ep^\star} + \alpha_{Ep^\star} P_{pub}) \end{aligned}$$

It is similar to K'_1 in the $Trapdoor$ algorithm.
Probability analysis of the Lemma 1: The challenger $\mathcal{C}$ evaluates the probability of all events involved in the above game as follows:

e_1: $\mathcal{A}_1$ did not make the $EPPK\text{-}query$ for $ID_{S^\star}$ and $ID_{R^\star}$.
e_2: $\mathcal{C}$ did not terminate the $Cipher\text{-}keywords\text{-}orcl$.
e_3: $\mathcal{C}$ did not terminate the $Trapdoor\text{-}orcl$.
e_4: $\mathcal{A}_1$ did not perform the $ReqPK\text{-}query$ on $ID_{S^\star}$ and $ID_{R^\star}$.

The total probability of the game $TI\text{-}CKT_{\mathcal{A}_1}$ is as follows:

$$Pr[e_1 \wedge e_2 \wedge e_3 \wedge e_4] = \frac{q_{ppt} - 2}{q_{ppt}} \times \Big(\frac{q_{ppt} - 2}{q_{ppt}} + \frac{q_{re} - 2}{q_{re}}\Big)^{q_E} \times \Big(\frac{q_{ppt} - 2}{q_{ppt}} + \frac{q_{re} - 2}{q_{re}}\Big)^{q_t} \times \frac{q_{re} - 2}{q_{re}}$$

Consider that $\mathcal{A}_1$ has advantage ε to win the game is $Pr[\mathcal{C} \rightarrow 1 | X = abP] = Pr[b' = b | X = abP] = \frac{1}{2} + \varepsilon \geq 2\varepsilon$.

The probability of $\mathcal{C}$ to solve the ECDDH assumption with help of $\mathcal{A}_1$ as subroutine is

$$\varepsilon' \geq 2\varepsilon \times \frac{q_{ppt}-2}{q_{ppt}} \times \left(\frac{q_{ppt}-2}{q_{ppt}} + \frac{q_{re}-2}{q_{re}}\right)^{q_E} \times \left(\frac{q_{ppt}-2}{q_{ppt}} + \frac{q_{re}-2}{q_{re}}\right)^{q_t} \times \frac{q_{re}-2}{q_{re}}$$

where q_{ppt} = number of $EPPK\text{-}query$, q_{re} = number of $RepPK\text{-}query$, q_E = number of $Cipher\text{-}keywords\text{-}orcl$, and q_t = number of $Trapdoor\text{-}orcl$.

Lemma 2 *The challenger $\mathcal{C}$ can solve the ECDDH assumption with an advantage ε' in the standard model if the adversary $\mathcal{A}_2$ can break the proposed scheme with a non-negligible advantage ε.*

Proof. Given a tuple (P, aP, bP, X) as an instance of the ECDDH assumption, the challenger must determine if $X = abP$ with the assistance of the adversary.

- `Initialize:`The challenger $\mathcal{C}$ invokes the $Setup$ algorithm to generate the $params$ and msk, then returns the $params$ and msk to the adversary $\mathcal{A}_2$.
- `Phase1:` $\mathcal{A}_2$ adversary can asks queries and oracles made by adaptively to $\mathcal{C}$.
  - $EPPK\text{-}query$: If an adversary $\mathcal{A}_2$ sends an ID_i to the query, then $\mathcal{C}$ keeps a list L_{ppk} of tuples $(ID_i, r_i, pk_i', sk_i')$. $\mathcal{C}$ performs the following:

    $\mathcal{C}$ finds a tuple in L_{ppk} with entry ID_i and returns sk_i'.
    Otherwise, $\mathcal{C}$ return sk_i' by execute the $Extract\text{-}partial\text{-}private\text{-}key$ algorithm and add the tuple $(ID_i, r_i, pk_i', sk_i')$ to the list L_{ppk}.

  - $ExtSV\text{-}query$: If an adversary $\mathcal{A}_2$ sends an ID_i to the query, then $\mathcal{C}$ keeps a list L_{sv} of tuple (ID_i, sk_i''). $\mathcal{C}$ performs the following:

    If $i \in \{S^\star, R^\star\}$, Then $\mathcal{C}$ adds the two tuples $(ID_{S^\star}, \perp)$, $(ID_{R^\star}, \perp)$ to the list L_{sv} and terminate the process.
    If a tuple is finds in the list L_{sv} then return sk_i''. Otherwise, $\mathcal{C}$ selects sk_i'' at random from the $Z_q^\star$, adds a tuple to the list L_{sv} and retun sk_i''.

  - $ReqPK\text{-}query$: If an adversary $\mathcal{A}_2$ sends an ID_i to the query, then $\mathcal{C}$ keeps a list L_{pk} of tuple (ID_i, PK_i). $\mathcal{C}$ performs the following:

    If $i \in \{S^\star, R^\star\}$, Then $\mathcal{C}$ refers the list L_{ppk} and computes $PK_{S^\star} = (pk_{S^\star}', aP)$, $PK_{R^\star} = (pk_{R^\star}', bP)$. Next, adds the two tuples $(ID_{S^\star}, PK_{S^\star})$, $(ID_{R^\star}, PK_{R^\star})$ to the list L_{pk} and returns $PK_{S^\star or R^\star}$.
    If a tuple is found in the list L_{pk} with entry ID_i then return PK_i. Otherwise, refers the lists L_{ppk} and L_{sv}, then computes $pk_i'' = sk_i''P$, set $PK_i = (pk_i', pk_i'')$, and adds the tuple to list L_{pk} and retun PK_i.

  - $Cipher\text{-}keywords\text{-}orcl$: When $\mathcal{A}_2$ submits a tuple (ID_{Ge}, ID_{Re}, KW), then $\mathcal{C}$ does as follows:

    If $\{Ge, Re\} \notin \{S^\star, R^\star\}$ or $\{R^\star, S^\star\}$, then $\mathcal{C}$ generate C_{KW}^{Ge} by calling the $Cipher\text{-}keywords$ algorithm returns.
    If Ge or Re $\in \{S^\star, R^\star\}$, then $\mathcal{C}$ generate C_{KW}^{Ge} by calling the $Cipher\text{-}keywords$ algorithm except K_2 calculated by private key of Re or Ge and returns.
    Otherwise, terminates the process.

- *Trapdoor-orcl*: When $\mathcal{A}_2$ submits a tuple (ID_{Sp}, ID_{Ep}, kw'), then $\mathcal{C}$ does as follows:

  If $\{Sp, Ep\} \notin \{S^\star, R^\star\}$ or $\{R^\star, S^\star\}$, then $\mathcal{C}$ generate $T^{Sp}_{kw'}$ by calling the *Trapdoor* algorithm returns.
  If Sp or Ep $\in \{S^\star, R^\star\}$, then $\mathcal{C}$ generate $T^{Sp}_{kw'}$ by calling the *Trapdoor* algorithm except K'_2 calculated by private key of Ep or Sp and returns.
  Otherwise, terminates the process.

- `Challenge`: $\mathcal{A}_2$ confirmed the `Phase1` over and outputs a tuple $(ID_{Sp^\star}, ID_{Ep^\star}, kw_0, kw_1)$ to the challenge phase, where $kw_0 \neq kw_1$ and $|kw_0| \neq |kw_1|$. Now, $\mathcal{C}$ chooses a bit $b \in \{0, 1\}$ at random, kw_b as input and does as follows:
  - If $\{Sp^\star, Ep^\star\} \in \{S^\star, R^\star\}$ or $\{R^\star, S^\star\}$, then $\mathcal{C}$ as follows:

    Recoveries the lists L_{ppk}, L_{sv}, and L_{pk} with entries $ID_{Sp^\star}$, $ID_{Ep^\star}$ and computes $\alpha_{Ep^\star} = h_1(ID_{Ep^\star}, pk'_{Sp^\star})$.
    $K'^\star_1 = sk'_{Sp^\star}(pk'_{Ep^\star} + \alpha_{Ep^\star} P_{pub})$, $K'^\star_2 = X$, and $T^{Sp^\star}_{kw_b} = h_2(ID_{Sp^\star},$
    $pk'_{Sp^\star}, aP, ID_{Ep^\star}, pk'_{Ep^\star}, bP, K'^\star_1,$
    $K'^\star_2, kw_b)$.
  - Otherwise, terminated the process.
    Finally, output the challenge trapdoor $T^{Sp^\star}_{kw_b}$.
- `Phase2`: $\mathcal{A}_2$ cloud performs the various queries and oracles similar to the `Phase1`.
- `Guess`: After $\mathcal{A}_2$ return guessed value b'. The game $TI\text{-}CKT_{\mathcal{A}_2}$ is won if $b = b'$.

Solve the ECDDH problem: Assume that the guess value of $\mathcal{A}_2$ is correct (i.e., $\mathcal{C}$ returns 1 if $b' = b$). As a result, the X value used in the `Challenge` phase is abP, indicating that the challenge trapdoor $T^{Sp^\star}_{kw'_b}$ is valid.

$$K'^\star_2 = X = abP = sk''_{Sp^\star} sk''_{Ep^\star} P.$$

It is similar to K'_2 in the *Trapdoor* algorithm.
Probability analysis of the Lemma 2: The challenger $\mathcal{C}$ evaluates the probability of all events involved in the above game as follows:

e_1: $\mathcal{A}_2$ did not make the *ExtSV-query* for $ID_{S^\star}$ and $ID_{R^\star}$.
e_2: $\mathcal{C}$ did not terminate the *Cipher-keywords-orcl*.
e_3: $\mathcal{C}$ did not terminate the *Trapdoor-orcl*.

The total probability of the game $TI\text{-}CKT_{\mathcal{A}_2}$ is as follows:

$$Pr[e_1 \wedge e_2 \wedge e_3] = \frac{q_{sv} - 2}{q_{sv}} \times \left(\frac{q_{sv} - 2}{q_{sv}}\right)^{q_E} \times \left(\frac{q_{sv} - 2}{q_{sv}}\right)^{q_t}$$

Consider that $\mathcal{A}_2$ has advantage ε to win the game is $Pr[\mathcal{C} \rightarrow 1 | X = abP] = Pr[b' = b | X = abP] = \frac{1}{2} + \varepsilon \geq 2\varepsilon$.

Table 2 Comparisons of security properties

Properties	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[6]	[9]	Ours
Certificateless	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✔
Trapdoor IND	✘	✘	✘	✔	✘	✔	✔	✔	✔	✔	✘	✔	✔
Withstand OKGA	✘	✘	✘	✔	✘	✔	✔	✔	✔	✔	✘	✔	✔
Withstand IKGA	✘	✘	✘	✔	✘	✔	✔	✔	✔	✔	✘	✔	✔
Pairing-free	✘	✘	✘	✘	✘	✘	✘	✔	✔	✔	✘	✔	✔
Standard model	✘	✘	✘	✘	✘	✘	✔	✘	✘	✔	✘	✔	✔
Bidirectional search	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	✔	✔
Multi-keyword search	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔

The probability of $\mathcal{C}$ to solve the ECDDH assumption with help of $\mathcal{A}_2$ as subroutine is

$$\varepsilon' \geq 2\varepsilon \times \frac{q_{sv}-2}{q_{sv}} \times \left(\frac{q_{sv}-2}{q_{sv}}\right)^{q_E} \times \left(\frac{q_{sv}-2}{q_{sv}}\right)^{q_t}$$

where q_{sv} = number of $ExtSV\text{-}query$, q_E = number of $Cipher\text{-}keywords\text{-}orcl$, and q_t = number of $Trapdoor\text{-}orcl$.

5 Performance Evaluation

This section provides the performance evaluation by comparing the proposed scheme to similar related systems on the basis of security properties, computation, and storage. On the basis of security properties, Table 2 compares the BCLSAE with similar relevant systems. As a result, the BCLSAE satisfies all the security properties.

Next, we compare the BCLSAE with similar twelve searchable encryption schemes based on computation and storage costs on the basis of bench mark operations. Among those searchable encryption schemes, some are bilinear pairing based and some are non-pairing based. Table 3 depicts the bench mark operation values for non-pairing-based and pairing-based schemes, using the elliptical curve "secp160r1" and supersingular Type-A curve, respectively. Additionally, the SHA-256 method was used to carry out a general hash algorithm.

Table 3 Bench mark values of basic operations

Operation	Meaning	Time/size
Bp	Bilinear pairing	4.154 ms
Pa_1	Point addition in G_1	0.013 ms
Sm_1	Scalar multiplication in G_1	1.631 ms
H	Hash-to-point	4.362 ms
pa	Point addition in G	0.003 ms
Sm	Scalar multiplication in G	0.509 ms
Mm	Modular multiplication in $Z_q^\star$	0.001 ms
Mi	Modular inverse in $Z_q^\star$	0.013 ms
h	Cryptographic hash	0.004 ms
$\|G_1\|$	Size of a element in G_1	64 bytes
$\|G_T\|$	Size of a element of G_T	128 bytes
$\|G\|$	Size of an element of G	40 bytes
$\|Z_q^\star\|$	Size of an element of $Z_q^\star$	20 bytes
$\|h\|$	Output of a hash algorithm	32 bytes
q	Order of bilinear group	64 bytes

The comparison is performed on searchable encryption schemes, which are implemented for IoT scenarios using different functionalities like keyword search, receivers, and bidirectional searches. In order to estimate the computational cost of an algorithm, we employ an approach that involves adding up the execution times of all the operations that make up the algorithm. For example, in the proposed scheme, the $Cipher\text{-}keywords$ algorithm requires three scalar multiplications and one point addition in elliptical group G, $(1 + mh)$ hash operations; as a result, the computational cost of $Cipher\text{-}keywords$ is $3Sm + Pa + (1 + 2m) = 1.574$ ms, assuming m = number of keywords = 5. However, the storage of searchable encryption is then calculated as the total of the number of hash values and group points. To upload a cipher-keywords to the cloud server, a sender in our scheme must send one number in $Z_q^\star$ and one hash value; as a result, the storage of cipher-keywords is $|Z_q^\star| + m|h| = 52$ bytes. For further information of computation and storage of related searchable encryption schemes, see Table 4 and Figs. 2 and 3.

Finally, some of the related schemes include bilinear pairing in their implementation. As a result, their computation cost is higher than other schemes. Furthermore, the performance of the proposed scheme is similar to the scheme [24] if $m = 1$, otherwise lower; the scheme [22] trapdoor computation cost is lower than all.

Table 4 Comparisons of related schemes with proposed scheme

Scheme	Computation cost			Storage cost	
	Cipher-keywords	*Trapdoor*	*Search*	*Cipher-keywords*	*Trapdoor*
[15]	$Sm_1 + 3Bp + Mm + H + 3h$	$Bp + 3Sm_1 + 2Pa_1 + Mm + H + 3h$	$2Bp + 2Sm_1 + 2Pa_1 + 2h$	$\|G_1\|$	$\|G_T\|$
[16]	$5Sm_1 + 3Pa_1 + Mm + H + 3h$	$Bp + 3Sm_1 + 2Pa_1 + Mm + H + 3h$	$2Bp + 2Sm_1 + 2Pa_1 + 2h$	$2\|G_1\|$	$\|G_T\|$
[17]	$5Sm_1 + 3Pa_1 + Mm + H + 3h$	$Bp + 7Sm_1 + 4Pa_1 + Mm + H + 3h$	$2Bp + 3Sm_1 + 2Pa_1 + Mm + 2h$	$2\|G_1\|$	$2\|G_1\| + \|G_T\|$
[18]	$3Bp + 6Sm_1 + 2H + h$	$Bp + 5Sm_1 + 4Pa_1 + H + h + Mi$	$3Bp + 3sm_1 + H + h + Mi$	$3\|G_1\| + 2\|G_T\|$	$2\|G_1\|$
[19]	$5Sm_1 + 2Pa_1 + Mm$ $2H + h + 2Mi$	$5Sm_1 + 2Pa_1 + 2Mm + h$	$3Bp + Pa_1$	$4\|G_1\|$	$3\|G_1\|$
[20]	$Bp + 3Sm_1 + Mm + H + 2h$	$Bp + Sm_1 + H + h$	$Sm_1 + h$	$2\|G_1\|$	$\|h\|$
[21]	$3Sm_1 + Pa_1 + Mm + 4h$	$Sm_1 + Pa_1 + 2Mm + 4h + Mi$	$2Bp + 2Sm_1 + h$	$2\|G_1\|$	$2\|G_1\|$
[22]	$5Sm + 2Pa + Mm + 4h + Mi$	$2Sm + 2Pa + Mm + 2h$	$2Sm + 2h$	$\|G\| + Z_q^\star + \|h\|$	$\|G\| + Z_q^\star$
[23]	$4Sm + Pa + 3h + Mm$	$3Sm + Pa + 2h$	$Sm + h$	$2\|G\|$	$\|h\|$
[24]	$3Sm + Pa$ $3h$	$3Sm + Pa$ $2h$	h	$\|h\| + Z_q^\star$	$\|h\|$
[6]	$2Bp + H + h$	$Bp + H$	$Bp + h$	$\|G_1\| + logq$	$\|G_1\|$
[9]	$4Sm + 5h$	$2Sm + 5h$	$2Sm + Mi + h$	$\|G\| + \|Z_q^\star\|$	$Z_q^\star$
Ours	$3Sm + Pa$ $(1 + 2m)h$	$3Sm + Pa$ $2h$	h	$m\|h\| + \|Z_q^\star\|$	$\|h\|$

5.1 Experiment Results

In this subsection, the authors must conduct an experiment using an IoT device. Through experimentation, the authors will examine the execution times of each algorithm in our system. In this experiment, we used a Raspberry Pi 4 Model B Rev. 1.5 device installed with the Charm crypto library to implement the BCLSAE using the elliptical curve "secp160r1". Table 5 describes the execution of each algorithm in the BCLSAE. Additionally, the BCLSAE allows IoT applications in the fields of e-health, e-tendering, banking, and insurance, as well as encrypted search engines, etc.

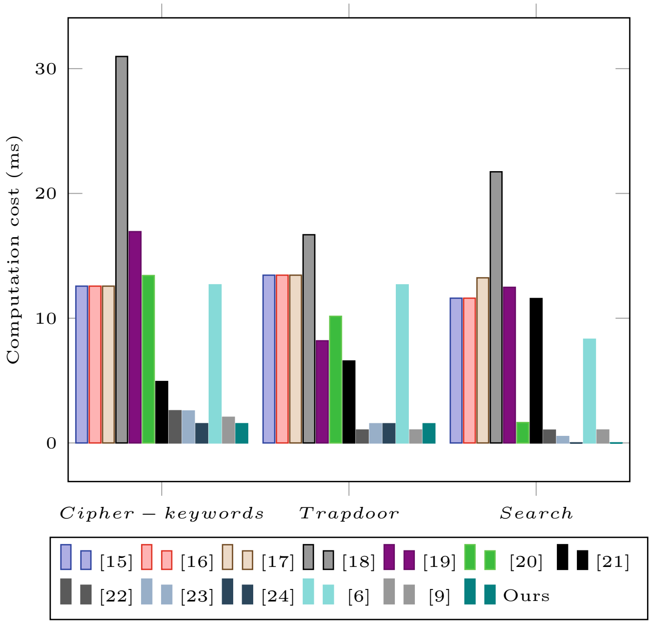


Fig. 2 Comparison results based on computation cost

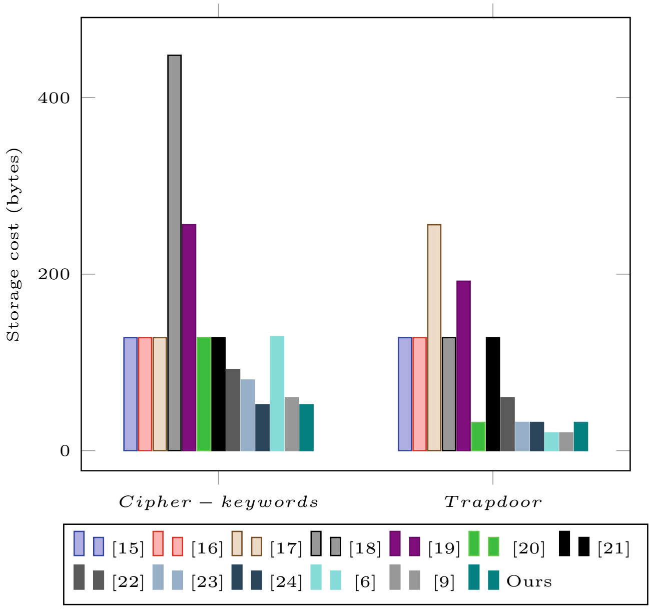


Fig. 3 Comparison results based on storage cost

Table 5 Execution of the proposed scheme

Algorithm	Execution time (ms)
Setup	2.1
Keygeneration	3.49
Cipher-keywords	2.62
Trapdoor	2.709
Search	0.0234

6 Conclusion

Most of the existing searchable cryptographic schemes use a bilinear pairing operation (i.e., high-cost) and implement their security in the ROM. As a result, they are not suitable for encrypted email applications in an IoT environment. This is because IoT devices have limited power, computing, and storage. Therefore, searchable encryption schemes without bilinear pairing (i.e., suitable for IoT) do not provide a practical environment for an encrypted email application. For example, a real-time application provides both the sender and the receiver with the ability to compose an email and search for the email in their profiles (i.e., it includes inbox and outbox). As a result, this paper proposes an efficient bidirectional certificateless searchable encryption system whose security can be demonstrated in the standard model, ensuring a realistic scenario of encrypted email applications in an IoT infrastructure. In addition, it supports multi-keyword searching. Finally, we compare the proposed scheme's performance to that of other current searchable encryption schemes. The results demonstrate that the proposed scheme is effective at searching for and encrypting keywords.

References

1. Song DX, Wagner D, Perrig A (2000) Practical techniques for searches on encrypted data. In: Proceeding 2000 IEEE symposium on security and privacy. S&P 2000. IEEE, pp 44–55
2. Boneh D, Crescenzo GD, Ostrovsky R, Persiano G (2004) Public key encryption with keyword search. In: International conference on the theory and applications of cryptographic techniques. Springer, pp 506–522
3. Baek J, Safavi-Naini R, Susilo W (2008) Public key encryption with keyword search revisited. In: International conference on computational science and its applications. Springer, pp 1249–1259
4. Jiang Z, Zhang K, Wang L, Ning J (2022) Forward secure public-key authenticated encryption with conjunctive keyword search. Comput J

5. Rhee HS, Park JH, Susilo W, Lee DH (2010) Trapdoor security in a searchable public-key encryption scheme with a designated tester. J Syst Softw 83(5):763–771
6. Zhang W, Qin B, Dong X, Tian A (2021) Public-key encryption with bidirectional keyword search and its application to encrypted emails. Comput Stand Interfaces 78:103542
7. Lu Y, Wang G, Li J (2019) Keyword guessing attacks on a public key encryption with keyword search scheme without random oracle and its improvement. Inf Sci 479:270–276
8. Guangbo W, Feng L, Liwen F, Haicheng L (2021) An efficient SCF-PEKS without random oracle under simple assumption. Chin J Electron 30(1):77–84
9. Lee CY, Liu ZY, Tso R, Tseng YF (2022) Privacy-preserving bidirectional keyword search over encrypted data for cloud-assisted IIoT. J Syst Arch 130:102642
10. Abdalla M, Bellare M, Catalano D, Kiltz E, Kohno T, Lange T, Malone-Lee J, Neven G, Paillier P, Shi H (2008) Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions. J Cryptol 21(3):350–391
11. Vaanchig N, Qin Z, Ragchaasuren B (2022) Constructing secure-channel free identity-based encryption with equality test for vehicle-data sharing in cloud computing. Trans Emerg Telecommun Technol 33(5):e3896
12. Zhang X, Huang C, Gu D, Zhang J, Wang H (2021) BIB-MKS: post-quantum secure biometric identity-based multi-keyword search over encrypted data in cloud storage systems. IEEE Trans Serv Comput
13. Liu ZY, Tseng YF, Tso R, Chen YC, Mambo M (2021) Identity-certifying authority-aided identity-based searchable encryption framework in cloud systems. IEEE Syst J
14. Yanguo P, Jiangtao C, Changgen P, Zuobin Y (2014) Certificateless public key encryption with keyword search. China Commun 11(11):100–113
15. He D, Ma M, Zeadally S, Kumar N, Liang K (2017) Certificateless public key authenticated encryption with keyword search for industrial internet of things. IEEE Trans Ind Inform 14(8):3618–3627
16. Liu X, Li H, Yang G, Susilo W, Tonien J, Huang Q (2019) Towards enhanced security for certificateless public-key authenticated encryption with keyword search. In: International conference on provable security. Springer, pp 113–129
17. Wu L, Zhang Y, Ma M, Kumar N, He D (2019) Certificateless searchable public key authenticated encryption with designated tester for cloud-assisted medical internet of things. Ann Telecommun 74(7):423–434
18. Zhang Y, Wen L, Zhang Y, Wang C (2019) Designated server certificateless deniably authenticated encryption with keyword search. IEEE Access 7:146542–146551
19. Yang X, Chen G, Wang M, Li T, Wang C (2020) Multi-keyword certificateless searchable public key authenticated encryption scheme based on blockchain. IEEE Access 8:158765–158777
20. Pakniat N, Shiraly D, Eslami Z (2020) Certificateless authenticated encryption with keyword search: enhanced security model and a concrete construction for industrial IoT. J Inf Secur Appl 53:102525
21. Karati A, Fan CI, Zhuang ES (2021) Reliable data sharing by certificateless encryption supporting keyword search against vulnerable KGC in industrial internet of things. IEEE Trans Ind Inform 18(6):3661–3669
22. Lu Y, Li J, Zhang Y (2019) Privacy-preserving and pairing-free multirecipient certificateless encryption with keyword search for cloud-assisted iiot. IEEE Internet Things J 7(4):2553–2562
23. Shiraly D, Pakniat N, Noroozi M, Eslami Z (2022) Pairing-free certificateless authenticated encryption with keyword search. J Syst Arch 124:102390
24. Hu Z, Deng L, Wu Y, Shi H, Gao Y (2022) Secure and efficient certificateless searchable authenticated encryption scheme without random oracle for industrial internet of things. IEEE Syst J
25. Kamble S, Bhikshapathi CV, Ali ST (2022) A study on fuzzy keywords search techniques and incorporating certificateless cryptography. In: 2022 international conference on computing, communication, security and intelligent systems (IC3SIS). IEEE, pp 1–6
26. Huang Q, Li H (2017) An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. Inf Sci 403:1–14

27. Chenam VB, Ali ST (2022) A designated cloud server-based multi-user certificateless public key authenticated encryption with conjunctive keyword search against IKGA. Comput Stand Interfaces 81:103603
28. Al-Riyami SS, Paterson K.G (2003) Certificateless public key cryptography. In: International conference on the theory and application of cryptology and information security. Springer, pp 452–473

A Motive Towards Enforcement of Attribute-Based Access Control Models in Dynamic Environments

Udai Pratap Rao, Pooja Choksy, and Akhil Chaurasia

Abstract Access control is one of the most basic information security requirements, which prevents unauthorized people from accessing the system or facilities. The access control process relies on specified policies and rules for any access. Access control is a crucial security feature that assures the safety of data and resources. There is a need for a framework to govern the proper use of information related to persons in an age of distributed computing, where enormous volumes of data are being transmitted and shared. This is where access control comes into play, ensuring that only authorized users can access the data. Many such modules have been proposed for a variety of sectors, including the extremely well-known Discretionary Access Control Model (DAC), Role-Based Access Control (RBAC) Model, Mandatory Access Control Model (MAC), and Attribute-Based Access Control (ABAC). We have shown recent trends in access control models in this research work. Furthermore, we present a few case examples that describe ABAC frameworks and how they are used in the university context. Finally, we use a comparison table to demonstrate ABAC's utility in dynamic and open contexts such as the cloud, IoT, etc.

Keywords Access control · Attribute-Based Access Control (ABAC) · Role-Based Access Control (RBAC) · Extensible Access Control Markup Language (XACML) · REST · Policy Machine (PM)

1 Introduction

The CIA Triad is a set of three major information security needs, which are as follows: "confidentiality" refers to the prohibition of unlawful disclosure of data or a resource. Integrity refers to the ban on illegal data or resource modification.

U. P. Rao (✉)
Computer Science and Engineering Department, NIT Patna, Patna, India
e-mail: udai.cs@nitp.ac.in

P. Choksy · A. Chaurasia
Computer Science and Engineering Department, SVNIT Surat, Surat, India

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_27

Availability guarantees that data or resources are always available to authorized users where access control is required. The appropriate use of information is significant when dealing with Personally Identifiable Information, abbreviated as PII, due to the pervasive nature of modern computing infrastructures [1]. Of course, Personal Identifiable Information (PII) reveals information about persons, and this is where privacy and security come into play.

In this context, access management is a crucial component of security that protects data from unwanted access. Controlling user access to a system or its resources is known as access control [2–4]. As shown in Fig. 1, there are multiple phases to the Access Control procedure.

- **Access Control Policies**: This level lays out the high-level regulations that must be followed regarding access control. Students, for example, can read materials. These policies are genuine authorizations and access controls that must be observed [5].
- **Access Control Models**: These formally represent how an access control policy works and is implemented. It presents a conceptual model that the access control system can implement [1]. This model can be displayed in various ways, including DAC, MAC, RBAC, ABAC, etc.
- **Access Control Mechanisms**: It specifies low-level functionality where policies and models are imposed for control access.

The fundamental architecture of access control is represented in Fig. 2, consisting of the three components below.

- **Subjects**: Subjects are any people or programs considered active entities in the access control process requesting permits to access any data or resources. Subjects initiate requests to access specific resources or data (Objects).
- **Objects**: Objects are non-active entities that include files, documents, databases, directories, programs, computer systems, and other resources that must be safeguarded from unwanted access.

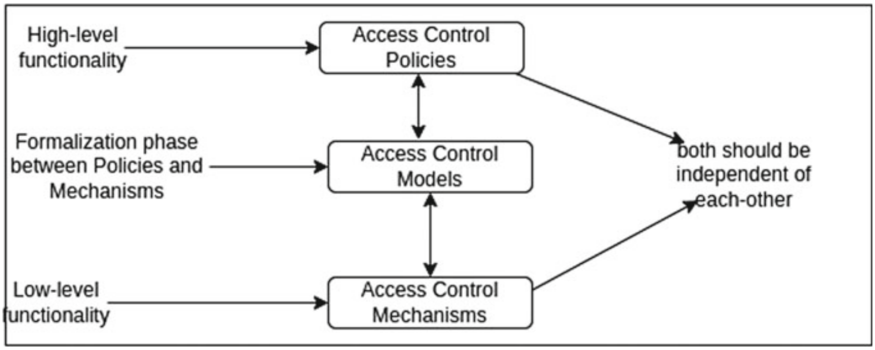

High-level functionality
Access Control Policies
Formalization phase between Policies and Mechanisms
Access Control Models
both should be independent of each-other
Low-level functionality
Access Control Mechanisms

Fig. 1 Multiple phases to the access control procedure

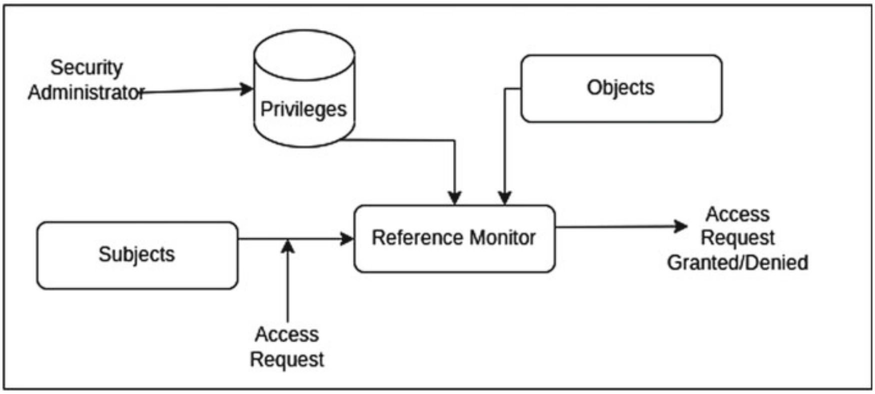


Fig. 2 Fundamental architecture of access control

- **Reference Monitor**: Any access control system must include a reference monitor. It is depicted in Fig. 2 where a dependable component intercepts all system requests [5, 6].

The remainder of the paper is laid out: Classical access control approaches such as DAC, MAC, RBAC, and ABAC are discussed in Sect. 2. In Sect. 3, we've included case studies of the Attribute-Based Access Control (ABAC) concept. Section 4 explores the different traditional access control approaches regarding parameters. Finally, Sect. 5 describes the process of future decision-making and its scope.

2 Analysis of Access Control Models

The classification of four traditional access control models is shown in Fig. 3. In the tree-based method, each user connects to an access structure, frequently referred to as an access tree over data attributes or files.

Additionally, no such tree is kept for data access in a non-tree-based technique [2].

2.1 Discretionary Access Control (DAC) Model

Discretionary access control (DAC) is a security access control that uses an access policy defined by the object's owner group and subjects to grant or restrict object access [7, 8]. DAC mechanism controls are established by identifying a user using credentials supplied at the moment of authentication, such as a username and password. The owner decides the benefits of the object's accessibility. It simplifies making

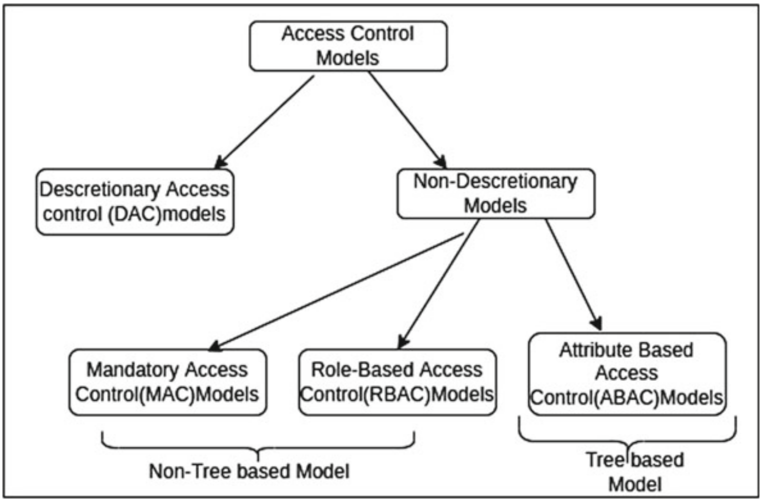

Access Control Models
Descretionary Access control (DAC)models
Non-Descretionary Models
Mandatory Access Control(MAC)Models
Role-Based Access Control(RBAC)Models
Attribute Based Access Control(ABAC)Models
Non-Tree based Model
Tree based Model

Fig. 3 Classification of access control models

policies and grants permissions for each access point. Access control complexity is kept to a minimum to improve the administration of the network's resources. Scalability for a large number of users is also a challenge.

2.2 Mandatory Access Control (MAC) Model

The access control system in which access to resources is restricted based on information sensitivity is called Mandatory access control. The user gets permission to access information according to the security level. MAC is a hierarchical model in which all users are given a security level, and all objects are attached to a security label. The system allows users to access objects with a security level equal to or lower than their security level.

Security levels connected with people and objects are a hierarchically ordered set, as defined in [7]. For example, there are four layers of security to consider: Eq. (1) shows a partially ordered list of High Secret (HST), Secret (ST), Classified (CF), and Unclassified (UF).

$$\text{HST} > \text{ST} > \text{CF} > \text{UF} \tag{1}$$

According to Eq. (1), the match of sensitivity label, Access control allows to share and restrict the object with sensitivity label to the subject of the system. In contrast, a group of categories is an unordered list. Access classes partially ordered levels are known as dominates and are symbolically denoted by "≤". Equation (2) below illustrates how the "dominates" connection in the MAC model is defined. In Eq. (2), SC_i stands for security levels and CL_i for a list of categories with $i = 1$ and 2, respectively.

$$(SC_1,\ CL_1)\ \leq\ (SC_2,\ CL_2) \Leftrightarrow (SC_1 \geq SC_2) \wedge (CL_1 \supseteq CL_2) \tag{2}$$

For example, if the subject of the system has the label "secret," it will not be permitted to access objects having a sensitivity label. This example clarifies the working of the MAC system. Only particular criteria depending on security levels allow a subject to access an object. This, of course, regulates the flow of data. The MAC-based models Bell–LaPadula (BLP) [9] and Biba [10] provide confidentiality and integrity, respectively. However, MAC is difficult to install due to a single point of failure caused by a centralized administrator. It is the case where a centralized admin should handle the entire burden of maintenance and configuration of the system. As systems grow large and become more complex, the admin becomes overwhelmed.

2.3 Role-Based Access Control Model (RBAC) Model

The RBAC model, which is based on the organizational structure of businesses, has been presented in [11–14]. This paradigm assigns diverse roles to job functions. Subjects are given authority to carry out specific access activities following their designated roles. Subjects (or users) are given a certain role to obtain access authority to carry out particular tasks. Users in this access model can get access authority through the roles allocated to them. It is clear from this that managing a user's privileges involves giving them moral responsibilities, making it easy to add new users, or simply altering a user's department [13]. The National Institute of Standards and Technology (NIST) represented the RBAC model in its standard form [14]. According to the RBAC hierarchy paradigm, lower-level roles own permissions that higher-level roles subsume [15, 16]. RBAC is particularly effective in assigning permissions systematically and consistently compared to other access models. The system's requirements for secrecy and privacy can be effectively handled by RBAC [17]. Additionally, RBAC may be implemented using APIs and can add and alter functions. RBAC can be regarded as a non-DAC approach [2, 18]. RBAC can benefit from a security policy for central administration. The system model of RBAC is shown in Fig. 4.

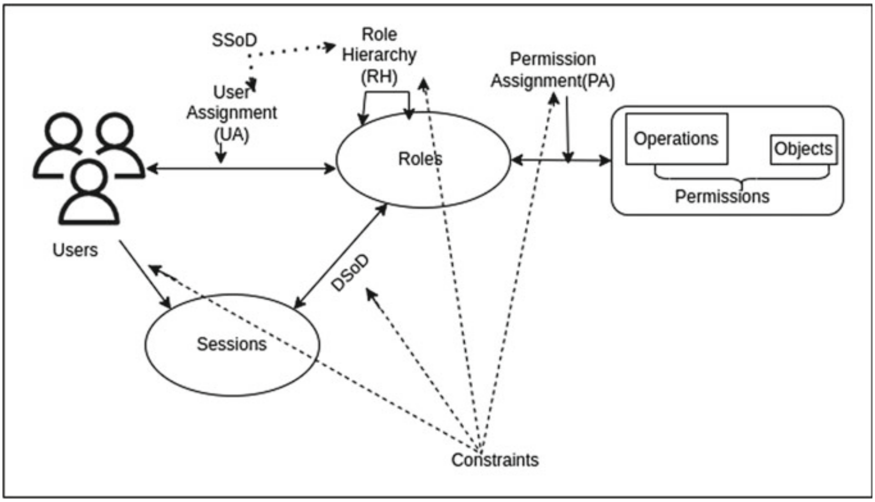


Fig. 4 Role-Based Access Control (RBAC) model [13, 19]

RBAC model is described in the following three forms.

1. Basic RBAC Model –

   {Users + Sessions + Roles + Permissions} ⇒ Basic RBAC

2. Hierarchical RBAC (RLH) Model –

   {Basic RBAC + Role Hierarchy} ⇒ Hierarchical RBAC

3. Constraints RBAC Model –

   {Hierarchical RBAC + Constraints} ⇒ Constraints RBAC
   Components of the RBAC Model depicted in Fig. 4 are detailed below:

- **Role**: A role can be described as a job or task with certain semantics of responsibility and authority inside the organizational system
- **Sessions (S)**: A session or collection of sessions occurs when a user logs into a system for a specific time.
- **Role Hierarchy (RLH)**: Here, any privileges held by younger positions may be inherited by senior roles. Once more, it demonstrates a many-to-many link between classes.
- **Permission Assignment (PRA)**: The many-to-many link between roles and permissions is demonstrated by PRA.
- **Static Separation of Duty (SSoD)**: This constraint on role separation is strong-exclusive [1]. SSoD restricts User-Role Assignments within the parameters of the security policy, which is not time- or space-bound. SSoD is another name for authorization SoD.

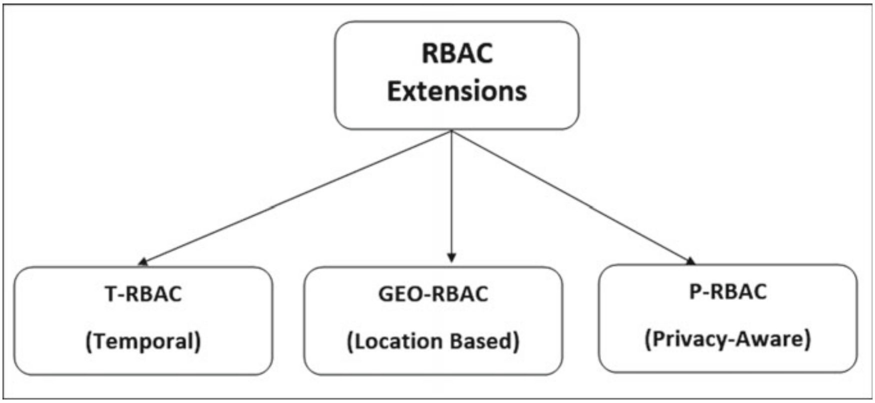


Fig. 5 RBAC extensions

- **Dynamic separation of Duty (DSoD)**: The prerequisite for weak-exclusive Roles separation is DSoD. The goal of DSoD is the same as that of SSoD, but DSoD policy also limits the roles given to a user during a particular session. As a result, we referred to DSoD as a "runtime SSoD."

2.3.1 RBAC Extensions

RBAC is widely acknowledged in both business and academics due to its ease of use, adaptability, and static nature. To offer fine-grained access, additional information, such as temporal or location context, must be considered using permissions. Figure 5 shows models of RBAC extensions based on contextual data.

- **T-RBAC**: RBAC [12] limits the utilization of permissions provided to roles up until specific time frames [1]. Users may only use authorized roles during the timespan(s) stated, such as only on Sundays between 9 AM and 4 PM.
- **GEO-RBAC**: When combined with location-based services, mobile apps, and users, GEO-RBAC [20] satisfies security criteria. Many businesses or institutions want their sensitive data only to be accessible on their property and in safe places.
- **P-RBAC**: By incorporating privacy-conscious procedures with roles, P-RBAC [21] expands RBAC. P-RBAC adds a privacy-related policy to secure sensitive information to the authorization process.

Role explosion is the most difficult task in any RBAC-based system, where there are more roles than users [4]. The central administrator must deal with the resulting issues with role management and interrelationships. RBAC is unsuitable for dynamic situations due to its coarse-grained access control flaws.

2.4 Attribute-Based Access Control (ABAC) Model

Role Engineering enables the use of Attribute-Based Access Control (ABAC). This more recent access control model provides access to objects based on authorization rules and is evaluated against attributes of subjects, objects, and environments. Therefore, it overcomes RBAC limitations like role explosion. An attribute-based access control model (ABAC) that integrates the traditional access control methods previously mentioned (DAC, MAC, and RBAC) [16, 22]. ABAC is scalable because more characteristics and policies can be added or withdrawn, and it is flexible since it can accommodate various attributes. Any procedure used by a system to confirm the identity of a person attempting to access the system is known as NIST. Authentication is necessary for successful security since access control is often based on the identity of the user requesting access to a resource. The ABAC model's primary component is the attributes belonging to the involved entities. An attribute is a set of keys and values. For instance, the user's name is "abc."

The National Institute of Standards and Technology (NIST) released the first ABAC document, which includes detailed guidelines on various business-related aspects [23]. Figure 6 illustrates the ABAC model's operation based on the organization's characteristics [19]. The following elements make up the ABAC model:

- Subjects and Subject attributes, such as Name = "abc."
- Objects and Object attributes, such as Location = "SVNIT" or fileName = "Program1,"
- Environmental Attributes, for example, access Date = "10- July-2022."
- Collectively, subjects descriptor and permissions define access rights.
- The term "object descriptor" refers to a combination of criteria and object properties.
- The operations and object descriptors provide the basis for permissions.
- The demonstration of the ABAC model's definition of rules [24] is described below.

Here, ABS_x, ABO_y, and ABE_z be sets of the subject, object, and environments attributes, respectively, where $1 \leq x \leq X$, $1 \leq y \leq Y$, and $1 \leq z \leq Z$ and ABT (i), ABT (j), and ABT (k) attribute assignment for each entity:

$$ABT(i) \subseteq ABS_1 \times ABS_2 \times .... \times ABS_X$$

$$ABT(j) \subseteq ABO_1 \times ABO_2 \times .... \times ABO_Y$$

$$ABT(k) \subseteq ABE_1 \times ABE_2 \times .... \times ABE_Z$$

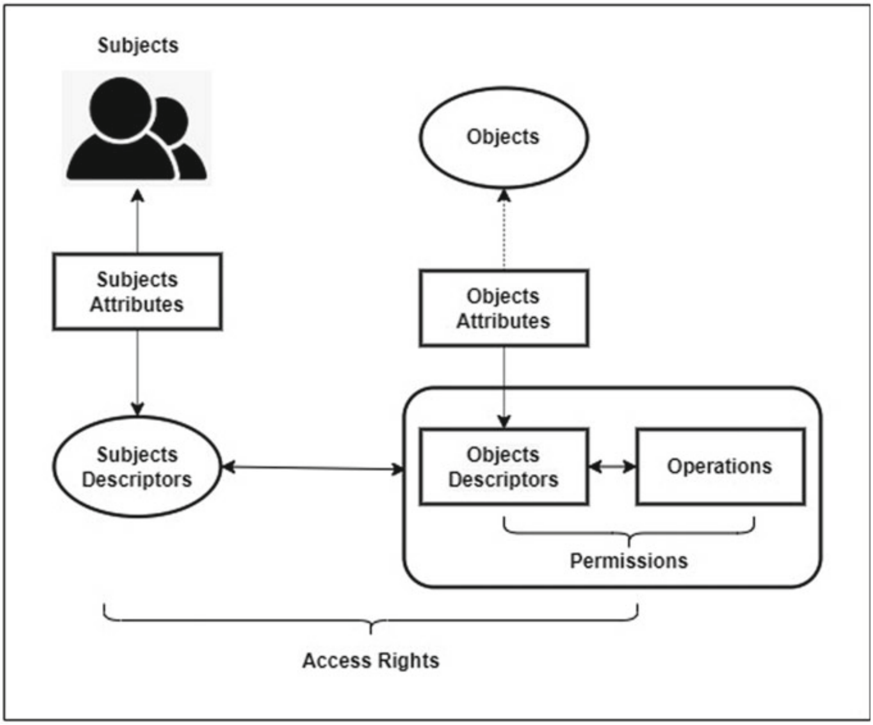


Fig. 6 Attribute-based access control model [19]

A policy rule that determines whether or not a subject can access an object o can be defined as a boolean function F. The given function F is as follows:

$$\text{Rule}_{\text{i}} : \text{ Grant (i, j, k)} \leftarrow \text{F(ABT (i), ABT (j), ABT (k))} \tag{3}$$

If F is true, access to the object is permitted; otherwise, the request is denied. One way to express Rule 1: "Users having the role of Doctor" may access the "Patient-Records" is as follows:

$$\text{Rule}_1 : \text{Grant(i, j, k)} \leftarrow (\text{Role(i)} = \text{Doctor}) \wedge (\text{Name(j)} = \text{Patient} - \text{Records}) \tag{4}$$

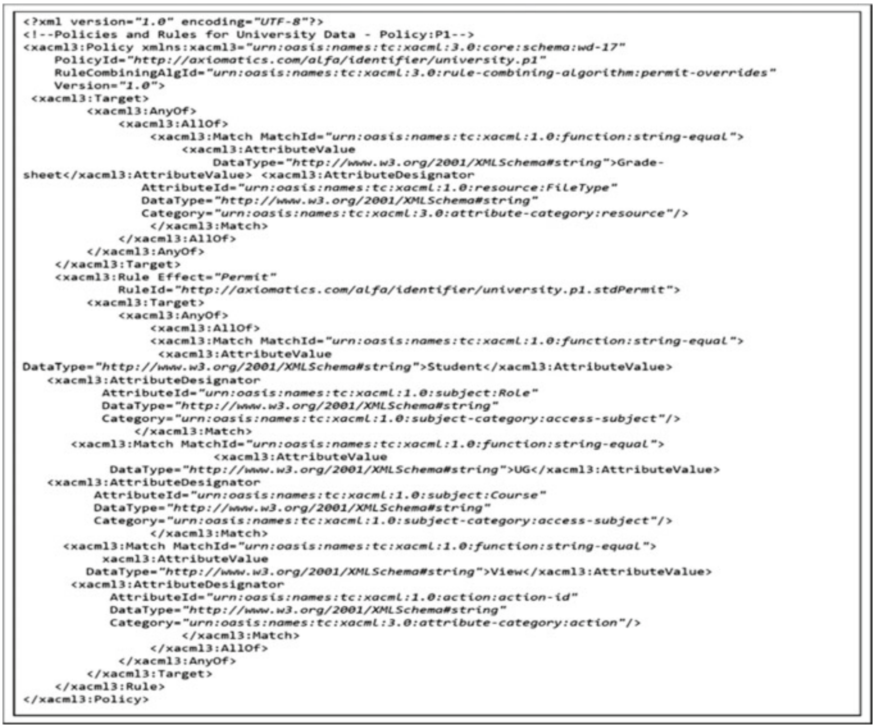
where Role and Name, concerning their respective values, are subject and object attributes.

3 Case Studies Using Various Attribute-Based Frameworks

3.1 University Scenario Using XACML

An access control system called XACML (Extensible Access Control Markup Language) specifies attributes as key-value pairs. Figure 7 shows how to use XACML components to generate XACML policies. The essential elements of XACML are shown in the following figure.

- **Policy-set**: Another policy set, policies, a target, and a policy-combining algorithm make up a policy set (like permit overrides or deny overrides). Only if the target matches will the policy established be in effect.
- **Policy**: Target and rule-combining algorithms make up policy (to resolve the conflict between two or rules).
- **Rule**: Regarding the objective, effect, criteria, etc., rules are formed.
- **Obligation expressions**: These expressions optionally may be included by the author of a rule.



```
<?xml version="1.0" encoding="UTF-8"?>
<!--Policies and Rules for University Data - Policy:P1-->
<xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
    PolicyId="http://axiomatics.com/alfa/identifier/university.p1"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:permit-overrides"
    Version="1.0">
 <xacml3:Target>
        <xacml3:AnyOf>
            <xacml3:AllOf>
                <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <xacml3:AttributeValue
                        DataType="http://www.w3.org/2001/XMLSchema#string">Grade-
sheet</xacml3:AttributeValue> <xacml3:AttributeDesignator
                 AttributeId="urn:oasis:names:tc:xacml:1.0:resource:FileType"
                 DataType="http://www.w3.org/2001/XMLSchema#string"
                 Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"/>
                  </xacml3:Match>
            </xacml3:AllOf>
        </xacml3:AnyOf>
    </xacml3:Target>
    <xacml3:Rule Effect="Permit"
            RuleId="http://axiomatics.com/alfa/identifier/university.p1.stdPermit">
        <xacml3:Target>
            <xacml3:AnyOf>
                <xacml3:AllOf>
                <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                 <xacml3:AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Student</xacml3:AttributeValue>
   <xacml3:AttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Role"
          DataType="http://www.w3.org/2001/XMLSchema#string"
          Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"/>
              </xacml3:Match>
      <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <xacml3:AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">UG</xacml3:AttributeValue>
   <xacml3:AttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Course"
         DataType="http://www.w3.org/2001/XMLSchema#string"
         Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"/>
                </xacml3:Match>
     <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          xacml3:AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">View</xacml3:AttributeValue>
      <xacml3:AttributeDesignator
           AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
           DataType="http://www.w3.org/2001/XMLSchema#string"
           Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"/>
                    </xacml3:Match>
                </xacml3:AllOf>
            </xacml3:AnyOf>
        </xacml3:Target>
    </xacml3:Rule>
</xacml3:Policy>
```

Fig. 7 XACML university policy (university.P1)

- **Advice Expressions**: Upon encountering a rule with obligation expression, the PDP evaluates it into obligations and returns some of these to the PEP in response to the context.
- **Condition**: A Boolean expression known as condition yields true or false.

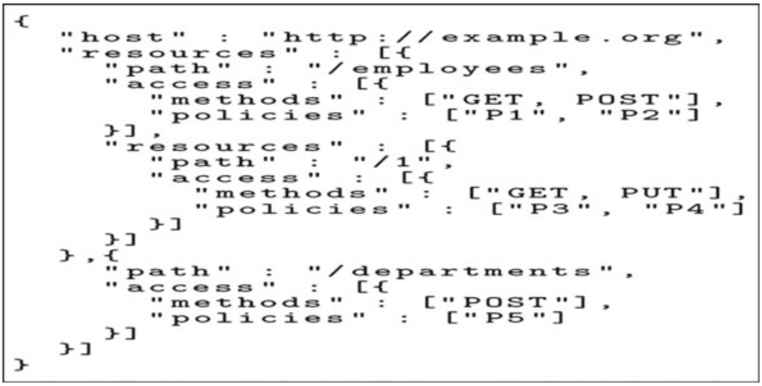
We have taken the university example, and these XACML-based policies are explained below. University (policy-set) has the following policies, as illustrated in Fig. 7: college.P1. ALFA [25] This XACML policy is created with the help of an eclipse plug-in. An XML-based file contains this policy. Policy (university.p1) P1: Role and course attributes are used for user identification in this policy (or subject). The following rule applies to the FileType and AccessLocation properties for objects and environments, respectively: (university.p1.stdPermit).

3.2 RESTful Web Services Scenario Using JSON [26]

Through RESTful Web services, attribute-based access policies can also be implemented. According to [26], each request (either via a browser or a RESTful client like Postman) must pass through an access control mechanism before being delivered to the server.

As seen in Fig. 8, JSON is an alternative method for ABAC enforcement: Every JSON file's foundation is its key.

Figure 8 shows several possible keys like host, resources, path, access, methods, and policies. Each of these keys has a set of values associated with it, such as "GET" and "POST" for the key "methods." As seen in Fig. 9, only users with compatible credentials and policies based on attributes are allowed access to resources (i.e., URIs).



```
{
    "host" : "http://example.org",
    "resources" : [{
        "path" : "/employees",
        "access" : [{
            "methods" : ["GET, POST"],
            "policies" : ["P1", "P2"]
        }],
        "resources" : [{
            "path" : "/1",
            "access" : [{
                "methods" : ["GET, PUT"],
                "policies" : ["P3", "P4"]
            }]
        }]
    },{
        "path" : "/departments",
        "access" : [{
            "methods" : ["POST"],
            "policies" : ["P5"]
        }]
    }]
}
```

Fig. 8 JSON libraries for RESTFul web service [26]

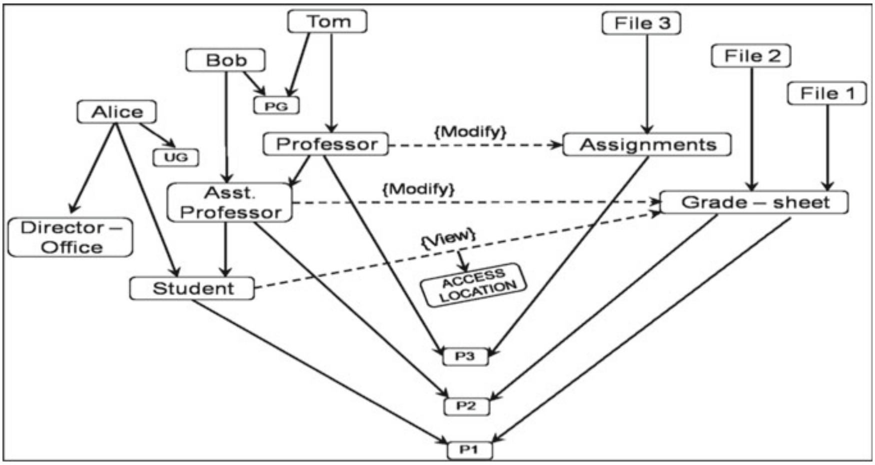


Fig. 9 Policy Machine (PM) assignments example for university

3.3 University Scenario Using Policy Machine (PM)

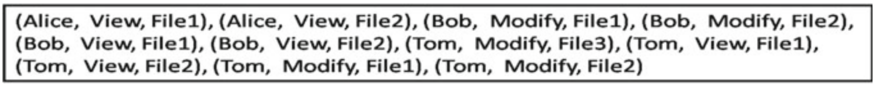
PM [27, 28] is a framework for access control that enables the configuration of access control policies. PM enables the definition of data, a collection of relations, and functions for policy expression and enforcement. The primary goal of PM is to establish a consistent framework that will support a variety of attribute-based policies. Additionally, PM asserts policy combinations by a solitary technique that demands adjustments to its data setup. Three different kinds of PM relations exist:

- **Assignments**: This assists in identifying and securing privileges.
- **Prohibitions**: User and process deny relations are expressed using this.
- **Obligations**: In addition to specifying conditions, this is used to define pattern-response relations.

We have indeed considered the relations between PM assignments for simplicity. The four abstractions that make up PM are listed below.

- **User Attributes (UA)**: A collection of users and a set of capabilities (op, o) are determined by a many-to-many relationship known as UA.
- **Object Attributes (OA)**: A set of objects and a set of access control entries (u, op) are determined by the many-to-many relation known as OA.
- **Operation sets (OPS)**: OPS is a collection of user-performed operations on objects.
- **Policy Classes (PC)**: PC refers to attribute and policy mappings.

We have taken the example of the university for showing PM relations shown in Fig. 9, which explains PM assignments for the university scenario. There are three policy classes (PC), P1, P2, and P3. Users (Tom, Bob, Alice) belong to User Attributes



(Alice, View, File1), (Alice, View, File2), (Bob, Modify, File1), (Bob, Modify, File2), (Bob, View, File1), (Bob, View, File2), (Tom, Modify, File3), (Tom, View, File1), (Tom, View, File2), (Tom, Modify, File1), (Tom, Modify, File2)

Fig. 10 Derived privileges of PM assignment for university

(UA) (Role and Course). Objects (File1, File2, and File3) are Object Attributes (OA) such as Assignments and Grade-sheet.

To illustrate PM relations, we used the university example in Fig. 9. PM assignments for a university context are explained in Fig. 9. Three policy classes (PC), P1, P2, and P3, exist. Tom, Bob, and Alice are members of the User Attributes (UA) group (Role and Course). Files 1, 2, and 3 are objects that relate to object attributes (OA), like assignments and grade sheets. Figure 10 lists the derived privileges of the Policy Machine assignments depicted in Fig. 9. Due to hierarchy, a professor can acquire all assistant professor and student privileges that are met.

Due to the massive amount of data, including photos, videos, and other types of media shared by numerous users on OSNs, the demonstration of how PM enables the specification of administrative policies and access control policies in a uniform manner while also enabling access control configuration changes in dynamic systems like Online Social Networks (OSN) [29, 30].

4 Access Control Models Comparison

Table 1 compares and analyses DAC, MAC, RBAC, and ABAC in dynamic environments like the cloud and IoT.

5 Conclusion and Future Scope

In this study, access control models were thoroughly examined. The fundamental ideas, architectures, and models for implementing present and future technology in the research field are defined to highlight the significance of access control models. ABAC use case, examples and the access control models DAC, MAC, RBAC, and ABAC are examined. Therefore, new research areas in this direction are yet conceivable. Finally, we compared these four models and found that ABAC is more effective for open and dynamic environments like the Internet of Things and cloud computing.

Table 1 Comparative analysis of access control models

Features (based on Cloud and IoT)	DAC	MAC	RBAC	ABAC
Dynamicity	No	No	No	Yes
Flexibility	No	No	No	Yes
Separation of duty	No	No	Yes	Yes
Least privileges	No	No	Yes	Yes
Granularity	Fine	Coarse	Coarse	Fine
Role-explosion problem	NA	NA	Yes	No
Context-awareness	No	No	Yes	Yes
Scalability	No	No	Up to some extent	Yes
Authorization decision	Locally	Locally	Locally	Globally
Changing privileges	Simple	Simple	Complex	Simple
Auditing	Easy	Easy	Easy	Complex
Privacy awareness	No	No	Up to some extent	No
Suite for open environments	No	No	No	Yes

Acknowledgements This work is supported by a research project funded by IHUB NTIHAC Foundation (Sanction number: IHUB-NTIHAC/2021/01/8), IITK under the aegis of the National Mission on Interdisciplinary Cyber-Physical System (NM- ICPS), DST, GoI.

References

1. Michael K (2012) Handbook on securing cyber-physical critical infrastructure. Elsevier Inc., ch. Policies, access control, and formal methods. Lo NW, Yang TC, Guo MH. An attribute-role based access control mechanism for multi-tenancy cloud environment. Wirel Pers Commun 84(3):2119–2134
2. Clerk Maxwell J (1892) A treatise on electricity and magnetism, 3rd edn, vol 2. Clarendon, Oxford, pp 68–73
3. Khan AR (2012) Access control in the cloud computing environment. ARPN J Eng Appl Sci 7(5):613–615. Elissa K. Title of paper if known. unpublished
4. Bang AO, Rao UP, Visconti A, Brighente A, Conti M (2022) An IoT inventory before deployment: a survey on IoT protocols, communication technologies, vulnerabilities, attacks, and future research directions. Comput Secur 10:102914
5. Samarati P, Di Vimercati SDC (2001) Access control: policies, models, and mechanisms. Lecture Notes in Computer Science (LNCS). Springer, pp 137–196
6. Anderson JP (1972) Computer security technology planning study, vol 2. DTIC Document, Technical Report
7. Damiani E, Ardagna CA, El Ioini N (2008) Open source systems security certification. Springer Science & Business Media
8. Latham DC (1986) Department of defense trusted computer system evaluation criteria. Department of Defense
9. Bell DE, LaPadula LJ (1973) Secure computer systems: mathematical foundations. DTIC Document, Technical Report

10. Biba KJ (1977) Integrity considerations for secure computer systems. DTIC Document, Technical Report
11. Ahn G-J, Sandhu R (2000) Role-based authorization constraints specification. ACM Trans Inf Syst Secur (TISSEC) 3(4):207–226
12. Bertino E, Bonatti PA, Ferrari E (2001) Trbac: a temporal role-based access control model. ACM Trans Inf Syst Secur (TISSEC) 4(3):191–233
13. Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. Computer 2:38–47
14. Ferraiolo DF, Sandhu R, Gavrila S, Kuhn DR, Chandramouli R (2001) Proposed NIST standard for role-based access control. ACM Trans Inf Syst Secur (TISSEC) 4(3):224–274
15. Ravidas S, Lekidis A, Paci F, Zannone N (2019) Access control in Internet-of-Things: a survey. J Netw Comput Appl 15(144):79–101
16. Hu CT, Ferraiolo DF, Kuhn DR, Schnitzer A, Sandlin K, Miller R, Scarfone K (2019) Guide to attribute based access control (ABAC) definition and considerations [includes updates as of 02-25-2019]. No. Special Publication (NIST SP)-800-162
17. Asaf Z, Asad M, Ahmed S, Rasheed W, Bashir T (2014) Role-based access control architectural design issues in large organizations. In: Open source systems and technologies (ICOSST), 2014 international conference on. IEEE, pp 197–205
18. Ni Q, Bertino E, Lobo J, Calo SB (2009) Privacy-aware role-based access control. IEEE Secur Priv 4:35–43
19. Sandhu R, Ferraiolo D, Kuhn R (2000) The NIST model for role-based access control: towards a unified standard. In: ACM workshop on role-based access control, vol 2000
20. Jin X, Krishnan R, Sandhu RS (2012) A unified attribute-based access control model covering DAC, MAC and RBAC. DBSec 12:41–55
21. Hu VC, Ferraiolo D, Kuhn R, Friedman AR, Lang AJ, Cogdell MM, Schnitzer A, Sandlin K, Miller R, Scarfone K et al (2013) Guide to attribute-based access control (ABAC) definition and considerations (draft). NIST Special Publication, vol 800, p 162
22. Qiu J, Tian Z, Du C, Zuo Q, Su S, Fang B (2020) A survey on access control in the age of internet of things. IEEE Internet Things J 7(6):4682–4696
23. Ferraiolo D, Chandramouli R, Kuhn R, Hu V (2016) Extensible access control markup language (XACML) and next generation access control (NGAC). In: Proceedings of the 2016 ACM international workshop on attribute based access control. ACM, pp 13–24
24. Alfa eclipse plugin for XACML policies. https://www.axiomatics.com/alfa-plugin-for-eclipse.html
25. Ferraiolo D, Atluri V, Gavrila S (2011) The policy machine: a novel architecture and framework for access control policy specification and enforcement. J Syst Architect 57(4):412–424
26. Patra L, Rao UP (2016) Internet of Things—Architecture, applications, security and other major challenges. In: 2016 3rd international conference on computing for sustainable global development (INDIACom) 2016 Mar 16. IEEE, pp 1201–1206
27. Hsu AC, Ray I (2016) Specification and enforcement of location-aware attribute-based access control for online social networks. In: Proceedings of the 2016 ACM international workshop on attribute based access control. ACM, pp 25–34
28. Bennett P, Ray I, France R (2015) Modeling of online social network policies using an attribute-based access control framework. In: International conference on information systems security. Springer, pp 79–97
29. Servos D, Osborn SL (2017) Current research and open problems in attribute-based access control. ACM Comput Surv (CSUR). 49(4):1–45
30. Ouaddah A, Mousannif H, Abou Elkalam A, Ouahman AA (2017) Access control in the Internet of Things: big challenges and new opportunities. Comput Netw 15(112):237–262

A Critical Analysis of Learning Approaches for Image Annotation Based on Semantic Correlation

Vikas Palekar and L. Sathish Kumar

Abstract Automatic Image Annotation (AIA) is a critical research topic and plays an important role in image retrieval. Recently, AIA approaches are focused to label the relevant data to the images. The annotations give the labels to images. Most of the image annotation techniques initially extract the features from the testing and training images, and then the annotation framework provides the accurate annotation based on the training data. This research provides a comprehensive overview of the most current stage in the development of AIA methodologies by combining 32 literature published over the last three decades. AIA approaches are divided into five groups: (1) Kernel Logistic Regression (KLR), (2) Triple relational Graph (TG), (3) Semantically Regularized CNN-RNN (S-CNN-RNN), (4) Label Correlation guided Deep Multiple view (DMLC), and (5) Multiple Modal Semantic Hash Learning (MMSHL). Acquiring meaningful low-level visual characteristics and generating the task of high-level semantic correlation is difficult for (AIA). The key finding through analysis is that semantic correlation gap of various image representation features for large-scale datasets is still a challenging issue. Exhaustive large-scale image annotation can be done effectively by efficiently indentifying good semantic correlation among image-image, image-label and label-label features. Then the performance evaluation of mentioned learning methods is based on precision and recall for popular datasets like MIR flicker, DeviantArt images, NUS-WIDE, MSCOCO and Coral5k. The performance of the mentioned methodology is compared based on idea, framework, complexity and accuracy in terms of different performance metrics to show differentiation in performance efficiency.

Keywords Deep learning · Hash learning · Image annotation · Logistic regression · Relational graph

V. Palekar (✉) · L. Sathish Kumar
School of Computing Science and Engineering, VIT Bhopal University, Kothri Kalan, Sehore, Madhya Pradesh, India
e-mail: vikaspalekar@gmail.com

L. Sathish Kumar
e-mail: sathish.kumar@vitbhopal.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_28

1 Introduction

Automatic Image Annotation (AIA) aims to predict the annotation of unknown images based on a set of rules. Relationships between the semantic concept space and the visual feature space of images have gained much more attention in the multimedia research community. Annotation and labelling of images are highly in demand owing to growth in Artificial Intelligence (AI) and Machine Learning (ML) developments. Traditional image annotation technique, where model learning is done by manual semantic level labelling, is not applicable on an exhaustive large scale.

Many of the researchers are using supervised or completely automatic ways of annotating images. With due research, automatic image annotation has achieved makeable gain. Support Vector Machine (SVM) supervises the classification algorithm and separates the labels dataset into two classes. SVM classifier minimizes the margin between classes using appropriate separating hyper plane. SVM usually reduces the loss caused due to higher margin separating hyper plane classifier. This loss is known as hinge loss which is a convex function [1]. Hinge loss is only limited to two classes; it cannot be generalized for multi-class classifiers.

On the other hand, the supervised approach which deals with manual labelling of images is not applicable for an extensive large number of images. One of the solutions is semi-automatic image labelling employing semi-supervised learning (SSL). Generalization ability for limited labelled images can be improved with SSL. SSL explores intrinsic structure from labelled and unlabelled images from the training dataset. The SSL approach of manifold regularization investigates the geometry of intrinsic data probability distributions that affect potential and objective function [2–4]. In the last decade, generative methods have been utilized to close the gap between low-level visual features and high-level semantics [5].

Our contribution to this critical analysis is to focus more on label correlation AIA methods by a deep investigation of their mathematical model. These AIA methods are compared on the basis of the main idea, framework of the model, complexity of computation, time complexity and accuracy in annotation; comparative analysis for various AIA methods is done. Section 1 gives an overall introduction to AIA methodologies and the major challenges that need to be addressed in regard to AIA methodologies. Section 2 tells the related work objectives of this analysis as well as the research gap. Section 3 shows the mathematical model used by five AIA methodologies. Section 4 gives the comparative analysis of five methodologies based on precision, recall and F1-score. Section 5 discusses some challenges, open issues and promising future directives for AIA development based on this analysis.

2 Related Work

According to the ability of the framework, exact annotations are generated based on the extracted features. Existing different machine learning techniques are used for image annotation. Although these machine learning and deep learning models attained adequate results, their accuracy is still lower. They require a lot of improvements to get greater interpretability for image annotation.

Research Gap/Objective:

Acquiring meaningful low-level visual characteristics and generating the task of high-level semantic correlation are difficult for (AIA). Exhaustive large-scale image annotation can be done effectively by efficiently indentifying semantic correlating image-image, image-label and label-label features. Web images are equipped with additional textual description; this description can be utilized for effective annotation and efficient retrieval. Usually, description is defective which creates a barrier to applying the correct annotation method. A flexible image annotation model is required for large-scale worldwide images, to achieve expected prediction. Research and investigation on reducing semantic gap is highly recommended which requires guidance on neural network choice to train deep computational models for improved efficiency.

3 Methodology

Researchers had developed many machine learning approaches for image annotation in the last decade. Most of the methods belong to supervised and semi-supervised learning (Fig. 1).

3.1 Manifold Regularized Kernel Logistic Regression (KLR)

Various machine learning algorithms were developed for images. The margin between two classes can be maximized by finding hyper plane employing a support vector machine (SVM). A maximum margined classifier is trained by minimizing a hinge loss using SVM. Semi-automatic image annotation is done by applying semi-supervised learning (SSL) [1, 6–8].

Executing SVM which uses a hyper plane to maximize the margin between two class exercise loss. This loss is called hinge loss. Manifold regularized KLR which is a semi-supervised approach has instant advantages: first, it has a smooth loss function; second, it is in place of a class label and generates a probability guess that is explicit; third, it is generalized for multi-class cases; and the information distribution's inherent structure can be well utilized by Laplacian regularization [2].

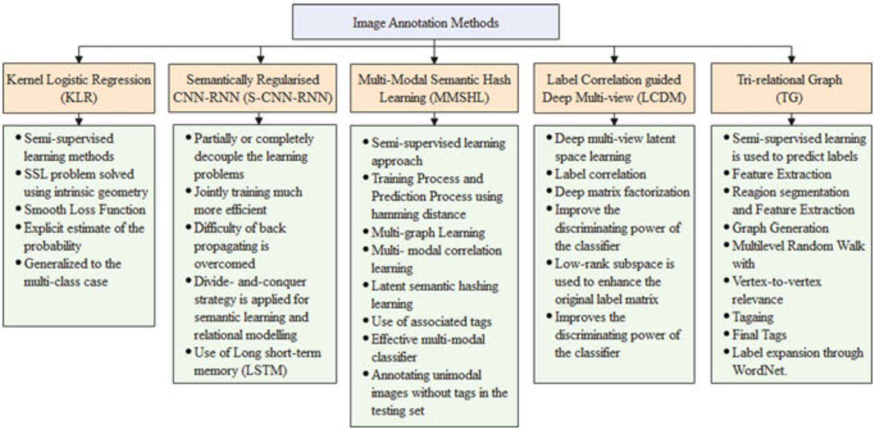


Image Annotation Methods
Kernel Logistic Regression (KLR)
Semantically Regularised CNN-RNN (S-CNN-RNN)
Multi-Modal Semantic Hash Learning (MMSHL)
Label Correlation guided Deep Multi-view (LCDM)
Tri-relational Graph (TG)
• Semi-supervised learning methods
• SSL problem solved using intrinsic geometry
• Smooth Loss Function
• Explicit estimate of the probability
• Generalized to the multi-class case
• Partially or completely decouple the learning problems
• Jointly training much more efficient
• Difficulty of back propagating is overcomed
• Divide- and-conquer strategy is applied for semantic learning and relational modelling
• Use of Long short-term memory (LSTM)
• Semi-supervised learning approach
• Training Process and Prediction Process using hamming distance
• Multi-graph Learning
• Multi- modal correlation learning
• Latent semantic hashing learning
• Use of associated tags
• Effective multi-modal classifier
• Annotating unimodal images without tags in the testing set
• Deep multi-view latent space learning
• Label correlation
• Deep matrix factorization
• Improve the discriminating power of the classifier
• Low-rank subspace is used to enhance the original label matrix
• Improves the discriminating power of the classifier
• Semi-supervised learning is used to predict labels
• Feature Extraction
• Reagion segmentation and Feature Extraction
• Graph Generation
• Multilevel Random Walk with
• Vertex-to-vertex relevance
• Tagaing
• Final Tags
• Label expansion through WordNet

Fig. 1 Image annotation methods

Objective function optimization problems with to take advantage of the intrinsic geometry, an additional regularization term is written as [2–4, 9]:

$$\min_{f\in H_k}\frac{1}{1}\sum_{i=1}^{1}\varphi\left(f, x_i, y_i\right) + \lambda_1||f||_K^2 + \lambda_2||f||_I^2 \tag{1}$$

where

φ—Generalized function of loss, $||f||_K^2$—K castigate the difficulty of the classifier in proper kernel replication Hilbert's domain (KRHD) H_k, $||f||_I^2$—term to castigate *f* along the underlying manifold, use the manifold regularization.
λ2 and λ1 balance the regularization terms and function of loss $||f||_I^2$ and $||f||_K^2$, respectively. The local similarity is ensured by Laplacian regularization, even though the manifold regularization terms $||f||_I^2$ can be chosen in a variety of ways.

Local similarities are preserved by Laplacian regularization. This research is used for annotating web images with Laplacian regularized kernel logistic regression [10]. The logistic loss represented by $\log\left(1+e^{-f}\right)$ is used to build a kernel logistic regression as a loss function (KLR) model [11]. In comparison with supervised SVM, KLR has similar performance.

Therefore, an equivalent optimization problem was obtained by adding a Laplacian regularized term to the objective function with logistic loss:

$$\min_{f\in H_k}\frac{1}{1}\sum_{i=1}^{1}(y_i\log\frac{1}{\left(1+e^{-f(x_i)}\right)} + (1-y_i)\log(1-\frac{1}{\left(1+e^{-f(x_i)}\right)}) + \lambda_1||f||_K^2 + \lambda_{21}f^TLf \tag{2}$$

where $f = [f(x_1), f(x_2), \ldots f(x_{l+u})]^T$, and L—Laplacian graph (given by $L = D-W$).

Here, D is a diagonal matrix (given by $\mathrm{D_{ii}} = \sum_{\mathrm{i}=1}^{\mathrm{l+u}} \mathrm{W_{ij}}$),
where W—the data adjacency graph edge weight matrix [9].

3.2 CNN-RNN Regularized on a Semantic Level (S-CNN-RNN)

CNN-RNN End-to-end Semantic Regularization for Recurrent Image Annotation has been used in a number of earlier studies, because of the embedding of images that provides the interface between the CNN and RNN. Existing models use the weakly semantic CNN hidden layer or its transform. RNN is overburdened with the task of predicting visual concepts and modelling their relationships in order to generate structured annotation output. Because of the challenge of back-propagating gradients through the RNN to coach the CNN, end-to-end training of the CNN and RNN is slow and unsuccessful [12]. Because the interface between the CNN and the RNN is semantically regularized, a semantically regularized embedding layer is required. Regularizing the interface can partially or entirely dissociate the training issues, allowing for more effective training of each and more efficient joint training [9, 13].

3.2.1 CNN-RNN

It is merely important to understand the working of Convolution Neural Network and Recurrent Neural Network (CNN-RNN) before making its user for regularization at the semantic level. A CNN-RNN method is divided into encoding and decoding. Encoder embeds the recognized visual features of an image and based on embedded features as input, the decoder generates the sequences of tags and labels.

Image Embedding is represented by I_e. It is a fixed-length vector $I_e \in \mathbb{R}^{d\times 1}$. Image Encoder is represented by f_{enc}. The embedding function is represented by $I_e = f_{enc}(I)$ as encoding recognizes the visual features and embeds them into an image, and I_e may be treated as feature transformation [14–17]. In this method, semantic representation is enforced to interact with RNN as it is used as a decoder.

Embedded feature from image (I_e) will be passed as context to the decoder RNN and a predictive path will be generated. Multi-label classification may be involved during decoding. Predictive path $\pi = (a_1, a_2, \ldots, a_{n_s})$. While generating a predictive path, importance is given to the sequence of labels in the case of multi-label classification. n_s indicates the estimated number of semantic labels (a_i) guessed for an image. During image captioning, the token word is a_i from a sentence with length a_i. Labels are to be converted in sequence by defining the priority to encounter label imbalance problems. Mostly, the LSTM-RNN decoder is used as a decoder with various CNN as encoders. Training of the previous RNN model was affected by messages supporting ups and downs of the gradient problem. Long Short-Term

Memory (LSTM) is widely used because of such message controlling mechanism. Cell and hidden are the two states represented by *c* and *h,* respectively [18, 19].

Following [18, 20], the forward pass of the LSTM-RNN decoder at time *t* with input x_t is calculated as

Input gate $i_t = \sigma(h_{t-1} \cdot W_{i,h} + c_{t-1} \cdot W_{i,c} + x_t \cdot W_{i,x} + b_i)$
Forget gate $f_t = \sigma(W_{f,h} \cdot h_{t-1} + W_{f,c} \cdot c_{t-1} + W_{f,x} \cdot x_t + b_f)$
Output gate $o_t = \sigma(h_{t-1} \cdot W_{o,h} + c_{t-1} \cdot W_{o,c} + W_{o,x} \cdot x_t + b_o)$
Output activation $g_t = \delta(W_{g,h} \cdot h_{t-1} + W_{g,c} \cdot c_{t-1} + W_{g,x} \cdot x_t + b_g)$
Cell state $c_t = f_t \odot c_{t-1} \odot i_t \odot g_t - \mathrm{e}$
Hidden state $h_t = o_t \odot \delta(c_t)$
W, h, W, c—weights of recurrent, *W, x*—weight of input, *b* are the biases.
σ(·) is the sigmoid function and δ activation function of the output.

At time step t, the decoder takes the last prediction a_{t-1} as input and computes a distribution over possible outputs:

$$x_t = E \cdot a_{t-1}, \quad h_t = LSTM(x_t, h_{t-1}, c_{t-1}),$$
$$y_t = softmax(W \cdot h_t + b),$$

where *E*—matrix of embedded word,
h_{t-1}—the recurrent units' concealed state at $t - 1$,
W, b—output layer weight and bias,
a_{t-1}—one-shot coding of the most recent prediction a_{t-1} and
LSTM(·) is a unit forward step.
The output y_t represents a distribution of possible actions from which the next action a_{t-1} is chosen.

3.2.2 Regularized Semantically CNN-RNN

Likely, CNN-RNN has divided the task into two parts, semantically regularized CNN-RNN has reduced the operational load of RNN by dividing it into semantic concept learning and relational modelling. CNN model takes image and associated information as input and generates an estimated probabilistic semantic concept. RNN generates the relational model which takes in the estimated probability concept and establishes the correlations for generating the sequence of label/words. CNN label concept prediction layer of an Inception net [21] was used instead of the feature embedding layer I_e. This embedding has clear semantic concept as it is being trained with ground-truth labels/visual concepts.

For predicting the semantic concept, CNN has a huge label space. For multi-label classification, approximately 1 k sizes of labels are available. Semantic concepts are predicted these label space $\hat{s} \in \mathcal{R} \wedge (k \times 1)$. *K* is the number of semantic concepts. The number of visual conceptions is *k,* are used which normally smaller than the word size in the vocabulary for captioning the image. RNN generates predictive sequence path π from input $\widehat{S}$. The point to be noted here is that, at both embedding layer $\widehat{s}$ and RNN output layer, supervision can be added which results in concept prediction $\mathcal{L}_u(s|\widehat{s})$ and relational modelling $\mathcal{L}_r(\pi, \pi^*|\widehat{s})$ loss.

Formally, we have $\mathcal{L} = \mathcal{L}_u(s|\widehat{s}) + \mathcal{L}_r(\pi, \pi^*|\widehat{s})$.

3.3 Multi-modal Semantic Hash Learning (MMSHL)

This method uses a semi-supervised machine learning approach for image annotation. The MMSHL model is trained by using labelled and unlabelled image datasets. The researcher has used NUS-WIDE and MRI flicker datasets for experimental results. MRI flicker dataset which has 12,500 training and testing samples, 2500 annotated samples, 12,500 testing samples, 38 semantic concepts, 457 textual features and 500 image features is compared with the NUS-WIDE dataset having 161,789 training and 107,858 testing samples, 32,357 annotated samples, 81 semantic concepts, 1000 textual features and 100 image feature.

MMSHL model is effective for classifying the labelled and unlabelled pair of image-text from the training dataset. The intention of this method is to annotate unmoral images without tags in its testing set. The annotation method is divided into two steps. First, the hash function is learned using MMSHL model on labelled and unlabelled images. The hash function uses three inputs: multi-graph, factorization matrix and multi-modal correlation. Second, the KNN classifier is trained to annotate the image. As this method is using a hash function which has efficient storage and computation capacity, it can be used for larger scale image datasets. Associative use of labels and tags can achieve good results. Modalities of semantic correlations are preserved by this framework [22].

3.3.1 Multi-graph Learning

In multipath learning, waited image graph and test graph are used. Semantic correlation between different modalities is identified and a multi-modal hashing framework is constructed. Graph matrices for various methods are prepared first and meagre based on their modal graph matrices. This method gives better performance as compared to traditional techniques of early and late fusion [23].

The multi-graph learning function is given as [22]

$$MinTr\left(F^T \sum_{m=1}^{2} \propto_m^2 L_m F\right)$$

where F—multi-modal semantic matrix, $\propto_m$—the modality's weight m and $L_m = I - A_m$ is modality m's Laplacian matrix.

A_m is the graph matrix of modality m, which is built as follows using the anchor graph:

$$A_m = Z_m \Lambda_m^{-1} Z_m^T$$

where diagonal matrix $\Lambda_m = diag\left(Z_m^T 1\right)$, and Z_m is the similarity matrix of modality m, which is computed based on anchors as follows:

$$Z_{ij}^m = \exp(-dist_m\left(x_i - x_i^a\right)/\sigma)$$

where $x_l^a|_{j=1}^{N=a}$ is a vector of the anchor.

The modality m's feature distance is determined by $dist_m(\cdot)$.

The Euclidean distance and the histogram distance were used for the image and text modalities, respectively. Multiple graph learning process is sped up by semantic matrix $F = ZU$, where U is the matrix of semantic mapping, $Z = [Z_1, Z_2]$; to make the solution of the modified objective function easier, consider the restriction $U^T U = I$ [22]:

$$MinTr\left(\sum_{m=1}^{2} \propto_m^2 U^T Z^T Z_m \Lambda_m^{-1} Z_m^T Z U\right)$$

s.t. $U^T U = I, \propto_1 + \propto_2 = 1$

3.4 Label Correlation Guided Deep Multi-view (DMLC)

In the existing multi-view annotation method, the labelled correlation and diversified complex multi-view features are ignored which can be found in social platform images. Image annotation can be improved by a comprehensive description of images. Research had exposed to correlation of labels in multi-view images [24]. Various features of images are preserved by capturing additional information in data representation. This method explores the correlation of labels by training low-level features from the label matrix. The originality of the label matrix has improved from low-level label subset. This technique reduces the missing and noisy labels [25]. Explored label correlation used for training the classifier. Two similar classes are identified using label correlations which improve the distinguishing ability of classifier.

3.4.1 Deep Multi-view Latent Space Learning

Image with multiple views is always represented with compressed data. The representation of this complex data is called latent space. The object in such images may have similarities which makes it difficult to annotate the object. This method of deep multiple-view latent space learning has represented unified multiple-view data $\{X_v\}_{v=1}^{V}$ in deep matrix factorization model. Due to this representation, the coefficient and 4 basis matrices are learned layer by layer. Unified multi-view data of all the views is represented by consistent coefficient matrix H [23].

We use a deep matrix factorization model to learn the basic matrices and coefficient matrices layer by layer to generate a unified data representation from multi-view data $\{X_v\}_{v=1}^{V}$, and the unified data representation is obtained by adding a consistent coefficient matrix H across all views. To better encode intra-view correlations, the minimization objective function is employed to reduce reconstruction error [5].

The optimization is presented as

$$\min_{H,\propto^v} \sum_{v=1}^{V} (\propto^v)^r \|X^v - Z_1^v Z_2^v \cdots Z_m^v H\|_F^2 \ s.t. \sum_{v=1}^{V} \propto^v = 1, \propto^v > 0, H \geq 0$$

where Z_i^v—the ith layer's base matrix for view v, m—layer count and $\propto^v$—control the importance of the vth view using the weight parameter; the vth view's relevance is controlled by the weight parameter.

By capturing inter-view relations and solving the above equation, complementary inter-view information can be retained, as each view has a common representation H. Weight parameter $\propto^v$ gives the respective view with accuracy due to less embedding loss.

3.4.2 Deep Multiple View Image Annotation with Label Correlation (DMLC) Method

Label correlation image annotation depends on labelling accuracy. Noisy and missing labels degrade the quality of image annotation. Labelling accuracy can be improved by identifying missing and noisy labels. To improve the performance, this method mainly focused on two tasks. First, Robust label correlation can compete with the missing labels and correct the noisy labels. Second, a feature-based classifier predicts the correct labels by correlation. Comparative results of label correlation, and similarity of class are identified. The class identified for two related labels can provide more concrete features than unrelated labels.

The objective function for image annotation:

$$\min_{S,P} \|Y - YS\|_F^2 + \beta\|S\|_* + \eta\|PH - S^T Y^T\|_F^2 + \lambda Tr\left(P^T LP\right) \ s.t. S \geq 0$$

The first two terms are used to learn a low-rank subspace $S \in \mathbb{R}^{C \times C}$ from the label Y. We use the constraint S $\geq$ 0 to ensure that the solution is meaningful because S captures the correlations of labels. The stronger the correlation between two labels, the higher the value of S_{ij}. The third term is to use a linear classifier to predict picture labels, and the classifier parameters are $P \in \mathbb{R}^{C\times k}$. The classifier for label l_i is represented by P_i, which is the *i*th column of P. Relationship between labels S is utilized to improve the original picture labels, while $S^T Y^T$ is the aim of classifier training. The final term is a graph regularization requirement that the classifiers must follow.

This research introduced the affinity matrix of labels $W = \frac{S+S^T}{2}$, and its Laplacian graph is L = D−W, where D is a matrix of dimension, stated as $D_{ii} = \sum_j W_{ij}$ by using l_i and l_j. When the correlation between the classifier parameters P_i and P_j increases, the classifier parameters P_i and P_j become more similar. Control the importance with β, η and λ [5].

3.5 Tri-relational Graph (TG)

Researchers have observed during this decade that image understanding regarding semantic labelling was explored at its peak and achieved great attention [26]. Implementing atomization in annotation has used image-level semantic concepts rather than region-level. Visual feature at the Image level has limited discrimination power which has less ability to predict small objects [27]. This identified gap has been fulfilled by representing the image semantic concept at the region level. Visual features at the region level are described more correctly by annotating at the region level. But annotating images at the region will lead to a new problem that images may have several labels due to various regions. These multiple labels are semantically correlated. This problem was addressed by label classification and refinement. The Tri-relational Graph (TG) is mainly designed for web images because relatively good textual description is available for images [28, 29].

3.5.1 Traditional Graph Versus Triple Relational Graph Learning

Traditional graph-based learning uses the data acquired from images only, which uses semi-supervised learning [30, 31], but image data is insufficient in the image acquiring process. To overcome this, a bidirectional graph (BG) is introduced where the relationship between multiple labels is explored [32]. Image data are represented with multiple labels which require strong semantic relationships. Semantic ambiguity is the issue when correlated multiple labels are assigned based on image data. Semantic ambiguity can be resolved by identifying the various regions in the image where multiple labels are assigned, which is the motivation to introduce a tri-relational graph.

In the triple relational graph annotation method, the image is divided into different regions and a set of various regions T is prepared [28, 33]. A set of semantic labels C and Image set X is prepared. Sets of images, regions and labels are prepared based on similarity. The model is trained with Image set I, Semantic labels C and Region set T.

Based on data of image, labels and region, respective graphs are prepared. New graphs invade from image segmentation and label allocation by connecting these sub-graphs.

The importance of the TG model is vertex-to-vertex relevance. A random walk restart algorithm is used to find the relation between image, data and label graph, where the visual correlation between images and regions, the semantic relationship between multiple labels including the relevance of image-to-label, label-to-region and region-to-image [28, 34]. Semi-supervised approach is used for prediction. Regions with non-labelled images are inserted in TG to predict regions of unannotated images. Researchers have used WordNet [35] for label expansion with the help of nouns and semantics of additional information with web images.

Tri-regional graph semi-supervised image annotation has three steps: first, generation of tri graph, second, annotating region of the image based on additional context analysis, third, expanding the label using WordNet [28] (Fig. 2).

Mainly, the image is represented in segmented regions to extract visual features at a low level. Extracted visual features are analysed and compared and then the region graph for TG is prepared. To generate an Image graph, the visual similarity of all the regions is calculated and compared.

Looking at the concept, the segmentation of the image is an important task in TG which is achieved through a Texture-enhanced JSEG algorithm which is dependent on

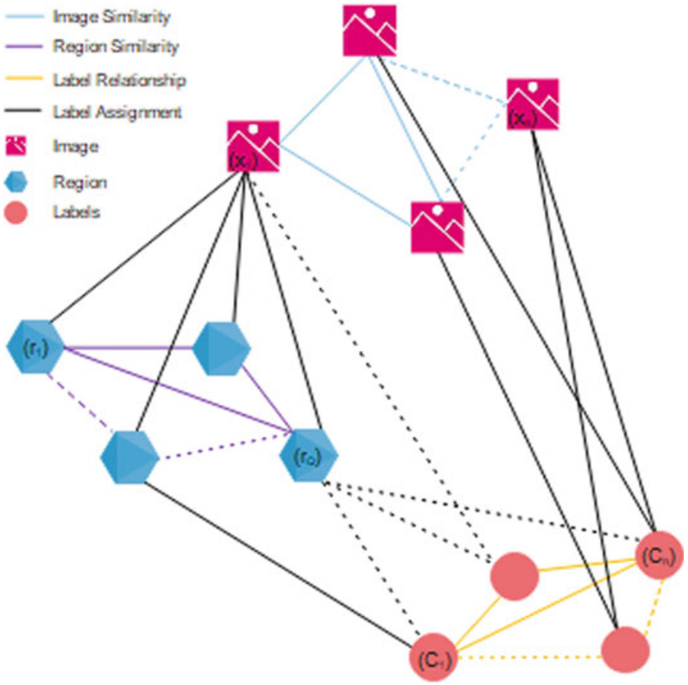


Fig. 2 Three relational graph [28]

regional latent semantic dependency [33]. For correct relatively independent segmentation, texture and colour class map are combined by texture-enhanced segmentation (TJSEG). Unnecessary segmentation is more then also performance may penalize, this will happen due to over-segmentation. This issue was effectively addressed by point line region (PLR).

Gabor texture, HSVH, CM and SIFT feature methods are used to represent the region feature and construct the visual word. $M_0 \times M_0$ pixel grid segments are used.

Additional information associated with web images like titles, comments and description context is semantically analysed with WorldNet and image contents are described.

As semantic labels are assigned to various regions of image, the union of three properties, image, region and labels, is represented as

$$G_q = r_q \cup \{X_i | Y_1(i, q) = 1\} \cup \{c_k | Y_3(q, k) = 1\} \tag{I}$$

Here, the Random Walk Restart [34] (RWR) algorithm which is used in birelational graph is modified by using a relationship among the image, region and labels. Semantics of each group (image, region and labels) are defined as

$$h_q = \begin{bmatrix} \gamma h_q^x \\ (1 - \gamma - \lambda) h_q^R \\ \gamma h_q^L \end{bmatrix} \epsilon R_+^{n+K+} \tag{II}$$

where h_q $(1 \leq q \leq Q)$

The triple-level random walk expression is formulated as

$$P_q^{(t+1)}(j) = (1 - \propto) \sum \propto h_q(i) + p_q^{(t)}(i) M(i, j) \tag{III}$$

The P_q^* is the final distribution which is decided by $P_q^{\infty} = (1 - \propto) M^y P_q^{(\infty)} + \alpha\, h_q$ which can be rewritten as

$$P_q^* = \alpha \left[I - (1 - \propto) M^T \right]^{-1} h_q \tag{IV}$$

Tri-level RWR Algorithm for image annotation [28]:

```
Input: Tri-relation Graph: g;
Transition probability matrix: M;
Testing image $X_i$ and its regions $X_i \rightarrow R_i$, $r_q$ ∈\,$R_i$.
Output: The labels $L_i$ for testing image $X_i$, $c_k \rightarrow r_q$.
1: Insert the test image $X_i$ and its segmented regions $R_i$ into the TG.
2: Analyse the semantic context of the test image according to Equation I.
3: Construct semantic group $H_q$ according to Equation II.
4: Set $t = 1$, $P_q^t = h_q$.
5: Repeat.
```

```
6: Calculate $\mathrm{P_q^{t+1}}$ according to Equation III.
7: $\mathrm{t = t + 1}$.
8: Until Equation IV sets up.
9: $\mathrm{c_k = c_{argmaxp_q^*(i)}}$.
```

Contributions of analysis:

1. KLR has solved semi-supervised learning problems using intrinsic geometry. The loss function is improvised. Probability is estimated correctly. Multi-class cases are generalized.
2. CNN-RNN regularized on a semantic level (S-CNN-RNN) has completely decoupled the learning problem. Jointly, the training module is more efficient. The complexity of backpropagation is resolved. Semantic learning and relational modelling are done using the divide and conquer strategy. Long short-term memory (LSTM) was used.
3. Multi-Modal Semantic Hash Learning (MMSHL) has used the semi-supervised machine learning technique. Hamming distance is used for training and prediction. The model is trained using the correlation between multiple graphs and latent semantic hash learning.

   Associative tags are used for effective multi-model classifier design having unimodal annotation capability.
4. Label correlation guided deep multi-view (DMLC) has used latent space learning with label correlation. Deep matrix factorization is used to improve the distinguishing power of the classifier. Original labels are enhanced by low-rank subspace.
5. Tri-relational graph is a semi-supervised learning approach to predict the labels. Features of images and their regions are extracted. Accurate correlation of semantic group and images is done by improving the random walk restart algorithm. Image correlation graph is prepared by semantic correlation of image graph, region graph and label graph. WordNet along with additional web information is used for label expansion.

4 Performance Analysis

Here, we have evaluated the performance of popular semantic-based learning methods on popular datasets like MIR flicker, DeviantArt, Coral5k, NUS-WIDE and Microsoft COCO. Performance analysis is done using precision and recall. Evaluated methods considered: (1) Kernel Logistic Regression (KLR) [9], (2) Semantically Regularized CNN-RNN (S-CNN-RNN) [18], (3) Multiple Modal Semantic Hash Learning (MMSHL) [22], (4) Label Correlation guided Deep Multiple view (DMLC) [24] and (5) Triple relational Graph (TG) [28]. A popular semantic method's performance is shown in Table 1.

Table 1 Popular AIA model's performance including KLR, TG, S-CNN-RNN, DMLC and MMSHL

Methods	Coral 5 k (%)			MIR flicker (%)			DeviantArt (%)			NUS-WIDE (%)			MSCOCO (%)		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
KLR [9]	18	21	19	–	–	–	–	–	–	–	–	–	–	–	–
S-CNN-RNN [18]	25	31	28	–	–	–	–	–	–	34	38	36	55	62	58
MMSHL [22]	18	25	21	–	–	–	–	–	–	82	63	71	77	67	72
DMLC [24]	29	34	32	26	21	23	31	21	25	43	33	37	68	58	63
TG[28]	26	27	26	20	21	20	26	25	25	–	–	–	–	–	–

Precision is the ability to reject unrelated information. Assume m1 as available pre-tagged images in the test dataset. m2 is the total number of images whose labels are tagged correctly. The number of images labelled with the label using the ground drive information is denoted by m3, and then precision is calculated as m2/m1. Recall is the ability to obtain important information, which will be calculated as m3/m2. As precision and recall are incompatible to justify the performance of automatic image annotation as it may produce biased values. Unification of precision and recall can generate the F1 score, which will be used to assess the AIA methodologies' resilience:

$$\mathrm{F1} = (2 * \mathrm{P} * \ \mathrm{R})/(\mathrm{P} * \mathrm{R})$$

A higher F1 score will indicate the more resilient method. Bold values indicate the highest F1 score on the respective dataset among all the methods. Observation indicates the effective performance of the DMLC method on Coral5k, MIR flicker and Deviant art datasets. The MMSHL method outperforms on NUS-WIDE and MSCOCO datasets. Table 2 shows some of the datasets that were used in the evaluation.

Datasets used:

The DeviantArt dataset consists of 3000 images. It is divided into two parts such as 1800 training images and 1200 testing images. Both the training and testing sets

Table 2 The five standard datasets' summary statistics

Used dataset	Trained set	Testing set	Size of set	Word size	Class
DeviantArt	1800	1200	3000	35	3
Corel5K	4500	499	4999	260	50
MIR Flicker	12,500	12,500	25,000	3000	38
MSCOCO	56,414	30,774	87,188	1000	80
NUS-WIDE	161,789	107,859	269,648	5018	81

contain 35 labels. The Corel5K dataset consists of 4999 images collected by the Corel Company. It is divided into two parts such as 4500 training images and 499 testing images. Each image in the dataset consists of 1–5 labels and both training and testing sets contain 260 same labels. The MIR FLICKR dataset consists of 25,000 images from the Flickr website. Tags assigned by Flickr users and EXIF information fields are available for each image in the dataset.

5 Challenges and Future Directions in AIA

The following are the major challenges that need to be addressed in regard to AIA methodologies:

1. Identify the relationship between visual and semantic features
2. Interpretation of semantic features in the same image
3. Identification of obscure semantic labels
4. Resolve the problem of unbalanced class labels
5. Complication in the learning of less frequent labels present in the image.

Although these machine learning and deep learning models attain adequate results, their accuracy is still lower. They require a lot of improvements to get greater interpretability for image annotation. Most of the existing approaches are not effective in finding the exact image labels. The other is low-level extracted features, which are not illustrating the image meanings correctly. Therefore, this research gives directions to use a hybrid deep learning model to improve the accuracy image annotation.

6 Conclusion

In general, two different factors that affect image annotation performance are the extracted features and the annotation framework. Therefore, an effective set of features and a sophisticated annotation framework are important for accurate annotation. If we compare the precision, recall and F1 score, on the MSCOCO dataset, MMSHL is giving the highest F1 score of 72%, DMLC is giving 63% and S-CNN-RNN is giving 58%. Methods are mainly divided into learning-based, training-based and model-based methods. During this decade, research shifted to semi-supervised learning. The semantic correlation gap of various image representation features remains a difficult problem. However, overgrowth of image data, lack of image context and irrelevant descriptions are still unexplored barriers to semi-supervised automatic image annotation. Image recognition from exhaustive large-scale image recognition models faces the difficulty of sufficient training images. Identifying classes that are invisible without training data is a future direction. Almost all generative model-based AIA methods are inferior to AIA methods for other types. This is because generative models cannot capture the complex dependencies between image features and labels. Deep learning-based AIA methods show the best performance compared to other methods due to their ability to obtain robust properties. Deep learning-based AIA methods typically annotate images with an automatically determined number of class labels. You have to deal with the "label imbalance" problem. Therefore, this research gives directions to use a hybrid deep learning model to improve the accuracy of image annotation.

References

1. Tao D, Tang X, Li X et al (2006) Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Trans Pattern Anal Mach Intell 28(7):1088–1099
2. Guan N, Tao D, Luo Z, Yuan B (2011) Non-negative patch alignment framework. IEEE Trans Neural Netw 22/8:1218–1230
3. Luo Y, Tao D, Geng Bo et al (2013) Manifold regularized multitask learning for semi-supervised multilabel image classification. IEEE Trans Image Process 22/2:523–536
4. Belkin M, Niyogi P, Sindhwan VI (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7/11:2399–2434
5. Xue Z, Du J, Zuo M, Li G, Huang Q (2019) Label correlation guided deep multi-view image annotation. IEEE Access 7:134707–134717. https://doi.org/10.1109/ACCESS.2019.2941542
6. Liu W, Tao D (2013) Multiview Hessian regularization for image annotation. IEEE Trans Image Process 22(7):2676–2687. https://doi.org/10.1109/TIP.2013.2255302
7. Liu W, Tao D et al (2014) Multiview Hessian discriminative sparse coding for image annotation. Comput Vision Image Understand 118:50–60. https://doi.org/10.1016/j.cviu.2013.03.007
8. Luo Y, Tao D, Chang X et al (2013) Multi-view vector-valued manifold regularization for multi-label image classification. IEEE Trans Neural Netw Learn Syst 24(5):709–722
9. Liu W, Liu H, Tao D et al (2016) Manifold regularized kernel logistic regression for web image annotation. Neurocomputing 172:3–8. ISSN 0925-2312
10. Schölkopf B, Herbrich R, Smola AJ (2001) A generalized representer theorem. Comput Learn Theory Lect Notes Comput Sci 2111:416–426
11. Zhu J, Hastie T (2005) Kernel logistic regression and the import vector machine. J Comput Graph Stat 14(1):185–205. https://doi.org/10.1198/106186005X25619
12. Zheng S, Jayasumana S, Romera-Paredes B, Vineet V, Su Z, Du D, Huang C, Torr P (2015) Conditional random fields as recurrent neural networks 1529–1537. https://doi.org/10.1109/ICCV.2015.179.
13. Chua T-S, Tang J, Hong R, Li H, Luo Z, Zheng Y (2009) NUS-WIDE: A real-world web image database from National University of Singapore. https://doi.org/10.1145/1646396.1646452
14. Jin J, Nakayama H (2016) Annotation order matters: recurrent image annotator for arbitrary length image tagging. In: 2016 23rd international conference on pattern recognition (ICPR), pp 2452–2457. https://doi.org/10.1109/ICPR.2016.7900004
15. Mao J, Xu W, Yang Y, Wang J, Yuille A (2015) Deep captioning with multimodal recurrent neural networks (m-RNN). arXiv: Computer Vision and Pattern Recognition
16. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556
17. Wang J, Yang Y, Mao J, Huang Z, Huang C, Xu W (2016) CNN-RNN: a unified framework for multi-label image classification 2285–2294. https://doi.org/10.1109/CVPR.2016.251
18. Liu F, Xiang T, Hospedales TM, Yang W, Sun C (2017)Semantic regularisation for recurrent image annotation. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, pp 4160–4168. https://doi.org/10.1109/CVPR.2017.443

19. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780.https://doi.org/10.1162/neco.1997.9.8.1735
20. Graves A, Mohamed A, Hinton G (2013)Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing, pp 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947
21. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, pp 2818–2826. https://doi.org/10.1109/CVPR.2016.308
22. Wang J, Li G (2017)A multi-modal hashing learning framework for automatic image annotation. In: 2017 IEEE second international conference on data science in cyberspace (DSC), Shenzhen, pp 14–21. https://doi.org/10.1109/DSC.2017.48
23. Snoek CGM, Worring M, Smeulders AWM (2005) Early versus late fusion in semantic video analysis. In: ACM international conference on multimedia, Singapore. DBLP, November, pp 399–402
24. Zhu P, Hu Q, Hu Q, Zhang C, Feng Z (2018) Multi-view label embedding. Pattern Recogn 84:126–135. https://doi.org/10.1016/j.patcog.2018.07.009
25. Lin Z, Ding G, Hu M, Wang J, Ye X (2013) Image tag completion via image-specific and tag-specific linear sparse reconstructions. In: Proceedings of IEEE conference computing vision pattern recognition (CVPR), June, pp 1618–1625
26. Palekar V, L SK (2021)Label dependency classifier using multi feature graph convolution networks for automatic image annotation. In: 2021 international conference on computational performance evaluation (ComPE). IEEE, pp 619–624. https://doi.org/10.1109/ComPE53109.2021.9752236
27. Zhang J, Wu Q, Shen C, Zhang J, Lu J (2016) Multi-label image classification with regional latent semantic dependencies. CoRR. abs/1612.01082, arXiv:1612.01082
28. Zhang J, Tao T, Mu Y, Sun H, Li D, Wang Z (2019) Web image annotation based on Tri-relational Graph and semantic context analysis. Eng Appl Artif Intell 81:313–322. ISSN 0952-1976
29. Palekar V, Ali M, Meghe R (2012) Deep web data extraction using web-programming-language-independent approach. J Data Mining Knowl Discov 3/2:69–73. ISSN: 2229-6662, ISSN: 2229-6670
30. Chen G, Song Y, Wang F, Zhang C. Semi-supervised multi-label learning by solving a sylvester equation. In: Proceedings of the 2008 siam international conference on data mining. SIAM, pp 410–419
31. Tong H, Faloutsos C, Pan J-Y (2006) Fast random walk with restart and its applications. https://doi.org/10.1109/ICDM.2006.70
32. Wang H, Huang H, Ding C (2011) Image annotation using bi-relational graph of images and semantic labels. In: Computer vision and pattern recognition (CVPR), 2011 IEEE conference on. IEEE, pp 793–800. https://doi.org/10.1109/CVPR.2011.5995379
33. Zhang J, Mu Y, Feng S, Li K, Yuan Y-B, Lee C-H (2018) Image region annotation based on segmentation and semantic correlation analysis. IET Image Process 12(8):1331–1337
34. Fellbaum C, Miller G (1998)Wordnet: an electronic lexical database. The MIT Press.https://doi.org/10.7551/mitpress/7287.001.0001
35. DeviantArt. http://www.deviantart.com

A Clustering-Based Approach for Effective Prevention of SQL Injections

Smit P. Shah, V. Achyuta Krishna, V. L. Kartheek, and R. Gururaj

Abstract SQL injection attack happens due to the insertion of a malicious SQL query into the application server via the input data from the attacker. A successful SQL injection attack can read and/or modify sensitive data stored in the database, execute administrative commands on the database such as DB shutdown, and in some cases issue harmful commands to the operating system. Owing to this, injection-based security vulnerabilities rank in the top ten security pitfalls identified by Open Web Application Security Project. Due to the significance of the security of web-based database applications, prevention of SQL injection at the right time is of paramount importance. A lot of research has been carried out on SQL injection attack detection and prevention. However, there is not much work done on clustering SQL injections into different types. Since different types of SQL injections can be countered effectively using different techniques, grouping of SQL injections can help organizations to allocate resources judiciously for effective prevention of SQL injection attacks. In this paper, we present an empirical study to identify the best suited unsupervised machine learning algorithm for clustering SQL injections, with the intention to find the frequency of occurrence of different types of injections. This approach would aid organizations to strengthen their security measures against particular types of injections in a more focused way. We also give a detailed analysis of the efficacy of the clustering algorithms used.

Keywords SQL injection · Unsupervised learning · Detection · Prevention · Clustering

S. P. Shah · V. A. Krishna · V. L. Kartheek (✉) · R. Gururaj
BITS Pilani, Hyderabad, India
e-mail: p20210105@hyderabad.bits-pilani.ac.in

S. P. Shah
e-mail: f20170080h@alumni.bits-pilani.ac.in

V. A. Krishna
e-mail: f20180165@hyderabad.bits-pilani.ac.in

R. Gururaj
e-mail: gururaj@hyderabad.bits-pilani.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes
in Electrical Engineering 1049, https://doi.org/10.1007/978-981-99-3569-7_29

1 Introduction

Database security is a vital part of information security. It is estimated that a significant number of web application attacks are caused by SQL injections. Recent SQL injection attacks include the 2020 data breach of millions of usernames and passwords [1], the 2020 attack to steal credit card data [2], and the 2017 attack on more than 60 universities and governments worldwide [2]. Gaining unauthorized access to the enterprise databases allows attackers to have greater control over pivotal data. This allows attackers to do nearly anything with the data, including alteration and deletion of data. The SQL Injection Attack (SQLIA) will occur when the malicious SQL query is received at the server through input forms. As blazoned by the Open Web Application Security Project (OWASP) association, injection attack has been the leading security threat in 2013 and 2017, and SQL injection attack is one of the most prominent types [3]. Hence, mitigating these injections using robust solutions becomes of utmost importance, otherwise there is a serious threat to the entire community. In this work, we propose to use an unsupervised learning approach for clustering SQL injections so that organizations can minimize the impact at the onset of the issue itself.

1.1 SQL Injection Attack

SQL stands for Structured Query Language. It is a language used for querying and modifying data in a relational database. SQL Injection is a malicious query statement formulated by an attacker with the intent to extract or update records in the database. For example, consider the following query:

select studentmarks, studentid, studentname, FROM students where studentname = 'Wiley' and studentId = 12
If an attacker were to input the value of studentName as a string- *'Wiley' OR 1 = 1–*, the resultant query would be
select studentmarks, studentid, studentname, FROM students where studentName = 'Wiley' oR 1 = 1—' and StudentId = 12

This query would return the marks of every student in the database.

Though there exist multiple types of SQL injections, in this work we have considered the following important types:

- Union-based injection: The attacker uses the UNION keyword to concatenate a malicious query to the original query.
- Tautology-based injection: In tautology, authentication is bypassed by injecting malicious input so that one or more conditional statements is intended to be true.
- Logically Incorrect query: In this attack, intentionally the attacker tries to throw an error from the database to understand the database schema.

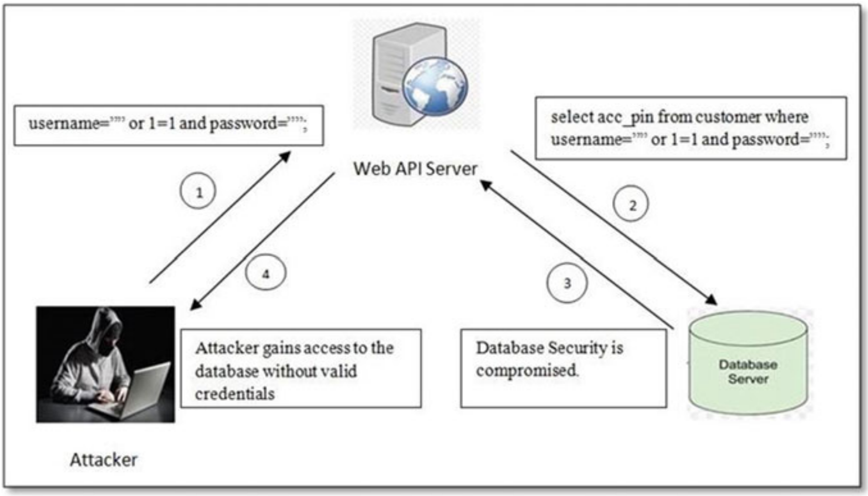

username="'" or 1=1 and password="'";
Web API Server
select acc_pin from customer where
username="'" or 1=1 and password="'";
1
2
3
4
Attacker gains access to the database without valid credentials
Database Security is compromised.
Database Server
Attacker

Fig. 1 Illustration of an SQL injection attack

– Timing-based injection: In this attack, the aim is to delay database responses by using keywords like *sleep, wait,* etc.
– Alternately encoded injection: In this type, the injection query is hidden by masking it with ASCII/hexadecimal characters.

For instance, in Fig. 1, the attacker bypassed authentication on the database server resulting in access to the database because the submitted SQL query was always evaluated to be true by adding one or more conditional expressions such as $1 = 1$. This is a simple example of a tautology-based SQL injection attack.

1.2 How Does SQLIA Affect the Database Security

An SQLIA can compromise database security in multiple ways. The number of SQLIAs have increased drastically in the recent past. As per a report, these attacks accounted for almost half of all web operation attacks in 2017 [4]. SQL injection attacks compromise the security, confidentiality, privacy, and integrity of the database. This might lead to a financial impact on the organizations or a data breach, ultimately leading to a huge loss of trust among the customers.

In the recent past, researchers have attempted to detect SQL injections through machine learning models which enabled organizations to take counter-measures [5–7]. But we observed that there is less research work carried out on clustering the injections. In the work presented in this paper, we conduct an empirical study on three unsupervised learning clustering algorithms to identify the one which is best suited for clustering SQL injection attacks.

The rest of the paper is organized as follows. We discuss related work along with a motivation for our proposal in Sect. 2. In Sect. 3, we give a detailed account of our proposal along with the methodology. Section 4 gives analytical insights into the clustering algorithms used. Finally in Sect. 5, we conclude the paper.

2 Related Work

There has been extensive research on using machine learning and deep learning-based techniques to classify SQL statements into injections and non-injections.

Zhang uses various machine learning algorithms such as random forest, SVM, and deep learning-based algorithms like convolutional neural networks and multilayer perceptron to detect the presence of an SQL injection [5]. Tripathy et al. classify SQL statements into injection and non-injection by the use of deep learning techniques like AdaBoost and Deep ANN [6]. Farooq uses an ensembling of gradient boosting machine (GBM), AdaBoost, Extended gradient boosting machine (XGBM), and Light gradient boosting machine (LGBM) to detect SQL injections [7]. Wang et al. have tried to establish a detection method based on static and dynamic analysis approach [8]. Kindy et al. in their survey paper have tried to categorize SQL injections and present the prevention techniques [9]. They have done a comparative analysis of the prevention techniques and presented details of the different schemes and defense mechanisms against each kind of SQL injection. Similarly, Halfond et al. have tried to compare which techniques are effective against which kind of SQL injections [10]. Kasim et al. in their work have provided an ensemble approach for detecting the severity level of SQL injections into three broad categories: simple, unified, and lateral attack [11]. Courant proposes a developer-friendly approach using abstract syntax trees which helps developers create dynamic queries with ease to prevent developers from unknowingly writing a malicious SQL injection [12]. Gandhi et al. propose a convolutional neural network-based solution based on CNN-BiLSTM for detecting SQL injections accurately. It provides a comparison between traditional CNN, LSTM, and Bi-LSTM and concludes Bi-LSTM as the winner [13]. Hirani et al. propose another deep learning-based solution where they compare multiple algorithms for detection of SQL injection and conclude that CNN is the best among all [14]. The majority of the work is aligned toward the detection of a SQL injection. There is no concrete work in classifying them into different types. From the research work carried out by Jemal et al., it is evident that there exist multiple techniques for the prevention of SQL injections, and it is also clear that a given technique is more suitable for preventing specific types of attacks [15].

We observe that the majority of research work carried out so far in this area focuses more on the detection of the presence of SQL injection attacks rather than finding the type of attack. The main problem with this approach is that some organizations might have spent most of their resources on deploying a particular technique, which might not be an effective solution against handling other injection types. On the other hand, some organizations might have deployed all counter-measures which would result in ineffective utilization of resources because some of the types of attacks they have considered may not be frequent ones. Hence, we propose that if an organization can get information about the frequency of occurrence of different types of attacks, it would help in the optimal and effective allocation of resources to prevent SQL injection attacks. Owing to the above reasons, in this paper we propose a machine learning-based model to cluster SQL injections that have occurred, which would enable the organizations to allocate resources and spend more developer cycles in an effective way to counter SQL injections.

3 Proposed Work

With the motivation explained in the previous section, we propose to conduct a detailed study on three unsupervised machine learning algorithms for clustering SQL injections. Now we present the overall process involved, as depicted in Fig. 2.

3.1 Data Collection

We chose an existing Kaggle dataset of 30,873 points and performed unsupervised learning on the same [16]. The dataset consists of around 38% SQL injections and the rest are not SQL injections. We filtered out the SQL injections from the dataset and then extracted features from it.

3.2 Pre-processing the Dataset

Feature extraction: Based on our domain knowledge about the types of SQL injections, we have considered 14 features as given in Fig. 3.

These are the fourteen features which we have tried to extract for each data point. Except for feature number three and four, the rest all are Boolean values. The keywords *sleep*, *wait,* and *benchmark* are typically used to delay the execution of queries, which is a characteristic of timing-based injection attacks.

The keyword *union select* represents the number of unions followed by a select keyword which characterizes a union-based injection. The keyword *ASCII, char, 0 ×* denotes the presence of conversion of a string to other types, and can be considered an indicator for alternately encoded injection. One way of generating a logically incorrect query is by converting a row object into another primitive data type. This can be achieved using keywords *convert* and *xtype.*

Normalizing the dataset: Once we have the features extracted for each of the data points, we normalize the dataset. Normalization helps in transforming our attribute values to a similar scale. This will help in achieving uniformity and can improve the

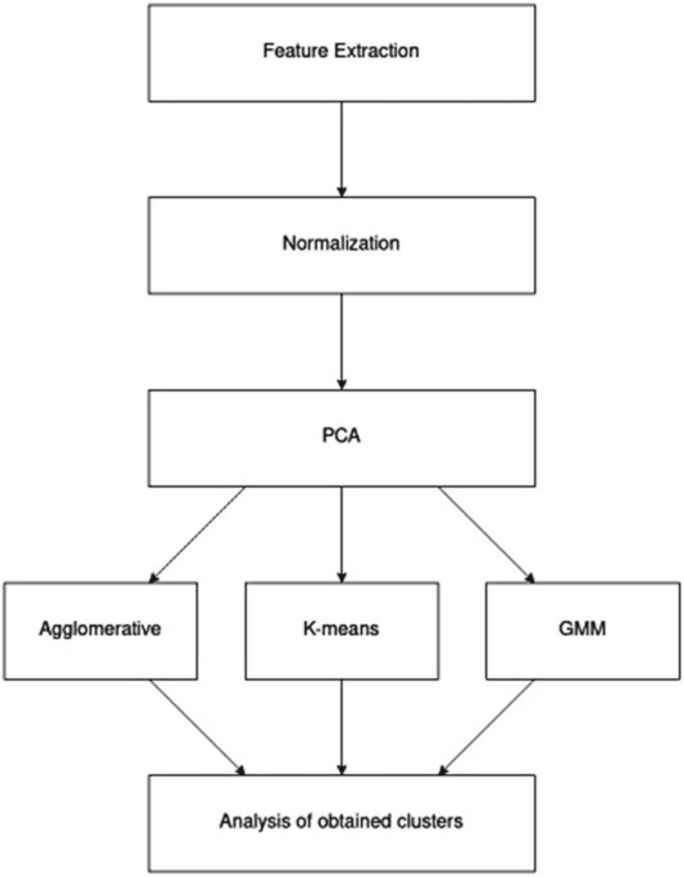
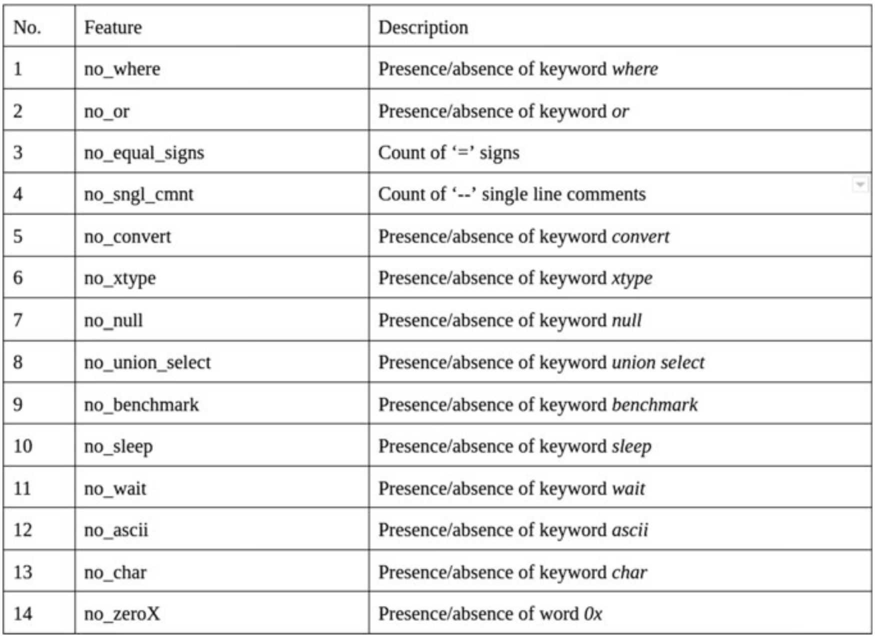


Fig. 2 Flowchart illustrating our proposed methodology

performance of the model. We have considered Z-score normalization which helps to ensure that our mean is 0 and standard deviation is 1. The Z-score is calculated as

$$Z = \frac{(x - mean)}{std.deviation} \tag{1}$$

Principal Component Analysis: Once we have normalized the dataset, we apply Principal Component Analysis (PCA) to the features. The advantage of using PCA is that it reduces the number of features while preserving maximum information about the dataset. From the initial 14 features, we reduce it to 8, preserving about 75% of the information. Smaller datasets are easier to explore and models are less expensive in terms of computation.



No.	Feature	Description
1	no_where	Presence/absence of keyword *where*
2	no_or	Presence/absence of keyword *or*
3	no_equal_signs	Count of '=' signs
4	no_sngl_cmnt	Count of '--' single line comments
5	no_convert	Presence/absence of keyword *convert*
6	no_xtype	Presence/absence of keyword *xtype*
7	no_null	Presence/absence of keyword *null*
8	no_union_select	Presence/absence of keyword *union select*
9	no_benchmark	Presence/absence of keyword *benchmark*
10	no_sleep	Presence/absence of keyword *sleep*
11	no_wait	Presence/absence of keyword *wait*
12	no_ascii	Presence/absence of keyword *ascii*
13	no_char	Presence/absence of keyword *char*
14	no_zeroX	Presence/absence of word *0x*

Fig. 3 Feature set

3.3 Learning the Patterns in the Dataset

After pre-processing the dataset, we apply clustering algorithms to it. We have used three unsupervised machine learning algorithms. Unsupervised learning involves learning from information which is neither classified nor labeled. The model achieves the intended goal of clustering from the available information without any guidance. Clustering is one important aspect of unsupervised machine learning which is meant for grouping similar objects to form clusters. We have used the following algorithms in our study:

- Agglomerative clustering—It is a type of hierarchical clustering algorithm wherein initially each point is an individual cluster and further each cluster is merged until all points fall in one big cluster. It is a bottom-up clustering algorithm.
- K-Means—It is a type of unsupervised clustering algorithm where the k value is predefined to be the number of clusters and each point is assigned to the nearest center. All such points club together to form a cluster.
- Gaussian Mixture Model (GMM)—It involves the mixture of multiple probability distributions, and it can accommodate clusters that have different sizes and shapes.

3.4 Optimal Number of Clusters

The goodness of clustering depends on two main factors: the points within a cluster should be as similar as possible and the points in different clusters should be as distinct as possible, i.e., inter-cluster distance should be high and intra-cluster distance between points should be low. One statistical measure to find the goodness of clustering algorithms is *Silhouette coefficient.*

Silhouette coefficient (S) in simple terms can be explained as

$$\mathrm{S} = \frac{(b - a)}{\max(a, b)} \tag{2}$$

where a is average intra-cluster distance and b is average inter-cluster distance.

The value of the Silhouette coefficient ranges from -1 to $+1$ where $+1$ means that the clusters are separate and distinguishable, 0 means that the clustering is decent but there is scope for improvement, and -1 means that the clustering is incorrect.

For each of the three models, we computed the Silhouette coefficient to determine the optimal number of clusters and obtained the following results. Agglomerative clustering returns the optimal number of clusters to be 5 with a Silhouette coefficient of 0.41 as depicted in Fig. 4. K-Means returns a Silhouette coefficient score of 0.42 for 3 clusters as seen in Fig. 5. GMM returns the optimal number of clusters as 2, with the Silhouette coefficient for the same being 0.42 as shown in Fig. 6.

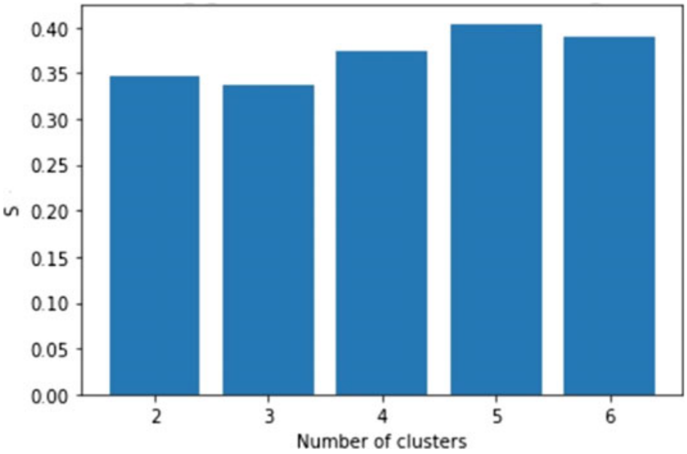


Fig. 4 Silhouette coefficients for agglomerative clustering

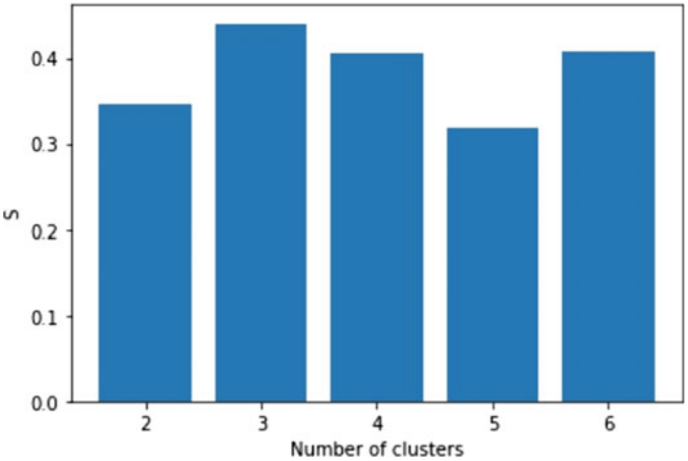


Fig. 5 Silhouette coefficients for K-means

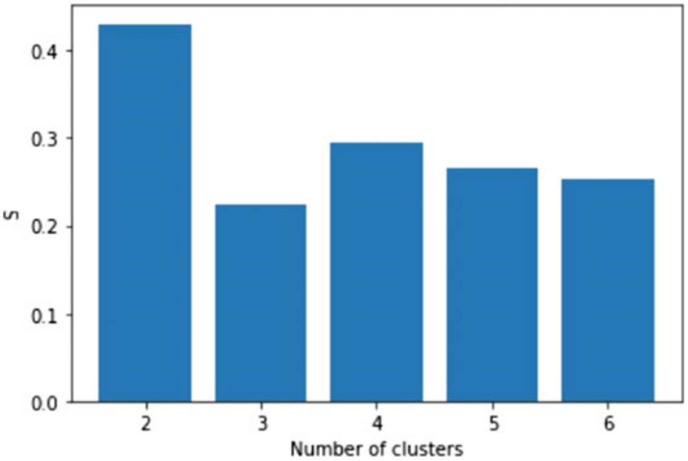


Fig. 6 Silhouette coefficients for Gaussian mixture model

4 Analysis

We analyzed the different clusters formed post which we attempted to map the clusters formed to a few specific type of SQL injections. Based on the values of specific features for each cluster, we determined the presence or absence of a particular injection type in the cluster. The results for GMM were not very promising with 2 being the optimal number of clusters. K-Means algorithm returned 3 as the optimal clusters but since this algorithm could not represent some of the types, we had to drop it. The 5 clusters obtained from agglomerative clustering gave more useful insights than the other two.

On analyzing the results for agglomerative clustering which can be seen in Fig. 7, we found the following. All 318 points in cluster 1 were timing-based injections with some being alternately encoded. All points in cluster 2 were union-based injections out of which 112 points were alternately encoded. Cluster 3 had a majority of alternately encoded injections and 108 of them were logically incorrect. Cluster 4 contained the maximum number of points, and it contained a mix of timing-based injections and alternately encoded injections. Cluster 5 had only 6 points, all of which seem to be outliers in the dataset.

The attributes *no_sleep*, *no_wait,* and *no_benchmark* were used to identify timing-based injection attacks. On analyzing the data points that have a presence of these keywords, it was found that all the points in cluster 1 had the presence of either one of the three keywords corresponding to timing-based attacks. It is possible for an SQL injection to be of multiple types. For example, a SQL query can be both union-based injection and alternately encoded. This can be seen from the results, where cluster 2 consists of SQL statements which represent both union-based and alternately encoded SQL injections. Similarly, the attributes *no_convert* and *no_xtype* were used for the identification of logically incorrect queries. The attributes *no_union_select* and *no_null* were used to group union-based injections. For alternately encoded injections *no_ascii*, *no_char*, and *no_zeroX* were used to detect alternately encoded injections.

Apart from the Silhouette coefficient, we further verified our algorithms with two other metrics, Davies–Bouldin index (DB index) and Calinski-Harabasz index. For agglomerative clustering, both DB index and Calinski-Harabasz gave 5 and 6 as the

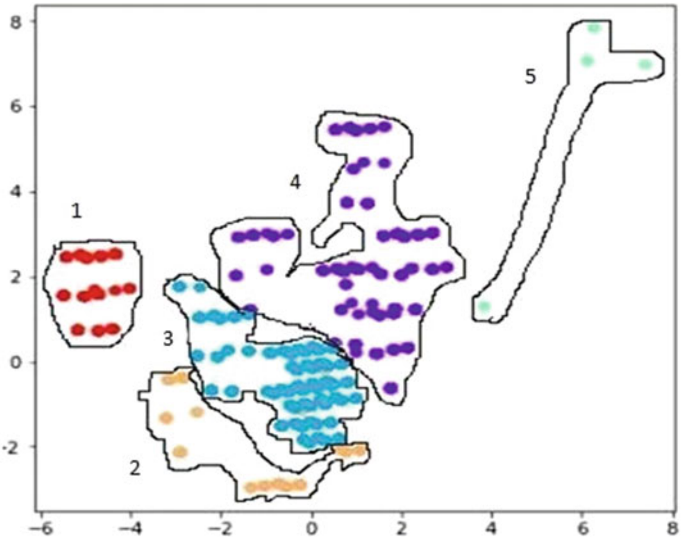


Fig. 7 Scatter plot with clusters

optimal number of clusters with very small difference in the values. For K-Means and GMM clustering, the DB index aligned with the previously obtained Silhouette index result of 3 and 2 clusters. respectively. Calinski score for K-Means and GMM gave a higher number of clusters and so we discarded it.

Thus, as per our empirical analysis, agglomerative clustering works better than the other two algorithms. Our model primarily aims to aid organizations to find out the frequency of types of injections. This model can be run in a batch mode after a substantial amount of injections are encountered (for example, after every 5000 injections) or in a timed mode (for example, after every 6 months) as per organizations' convenience. However, one limitation of this approach is that some manual effort in mapping the clusters is still required. If we have a dataset wherein we know the type of SQL injections, we can build a more accurate model by using supervised machine learning algorithms.

5 Conclusion

In this work, we considered three unsupervised machine learning algorithms for clustering SQL injections. Our empirical study proved that the agglomerative clustering algorithm is better suited for our dataset. Since the data points are unlabeled w.r.t. the type of injections, some manual inspection is still required even after forming clusters to map the clusters to the specific types. This technique of clustering SQL injections can help organizations to figure out the frequency of occurrence of each type of attack. This would result in the judicious deployment of resources to prevent SQL injection attacks. We understand that there is still scope for performing clustering using other advanced machine learning techniques which may give better results. In the coming future, we also plan to work on labeling SQL injections for the benefit of the research community.

References

1. Security statement by Freepik (2020). https://www.freepikcompany.com/newsroom/statement-on-security-incident-atfreepik-company/
2. SQL injection attack: a major application security threat (2020). June. https://www.kratikal.com/blog/sql-injection-attack-a-major-application-securitythreat/
3. OWASP Top 10 2017 (2017). https://owasp.org/www-project-top-ten/2017/
4. SQL injections cyber attacks (2017). September. https://outpost24.com/blog/SQL-injections-cyberattacks
5. Zhang K (2019) A machine learning based approach to identify SQL injection vulnerabilities. In: 2019 34th IEEE/ACM international conference on automated software engineering (ASE), November. IEEE, pp 1286–1288
6. Tripathy D, Gohil R, Halabi T (2020) Detecting SQL injection attacks in cloud SaaS using machine learning. In: 2020 IEEE 6th international conference on big data security on cloud (BigDataSecurity), IEEE international conference on high performance and smart computing,

(HPSC) and IEEE international conference on intelligent data and security (IDS), May. IEEE, pp 145–150

7. Farooq U (2021) Ensemble machine learning approaches for detection of SQL injection attack. Tehnički glasnik 15(1):112–120
8. Wang Y, Wang D, Zhao W, Liu Y (2015, July) Detecting SQL vulnerability attack based on the dynamic and static analysis technology. In: 2015 IEEE 39th annual computer software and applications conference, vol 3. IEEE, pp 604–607
9. Kindy DA, Pathan ASK (2011). A survey on SQL injection: vulnerabilities, attacks, and prevention techniques. In: 2011 IEEE 15th international symposium on consumer electronics (ISCE), June. IEEE, pp 468–471
10. Halfond WG, Viegas J, Orso A (2006, March) A classification of SQL injection attacks and counter measures. In: Proceedings of the IEEE international symposium on secure software engineering, vol 1. IEEE, pp. 13–15
11. Kasim Ö (2021) An ensemble classification-based approach to detect attack level of SQL injections. J Inf Secur Appl 59:102852
12. Courant J (2020) Developer-proof prevention of SQL injections. In: International symposium on foundations and practice of security, December. Springer, Cham, pp 82–99
13. Gandhi N, Patel J, Sisodiya R, Doshi N, Mishra S (2021) ACNN-BiLSTM based approach for detection of SQL injection attacks. In: 2021 international conference on computational intelligence and knowledge economy (ICCIKE), March. IEEE, pp 378–383
14. Falor A, Hirani M, Vedant H, Mehta P, Krishnan D (2022) A deep learning approach for detection of SQL injection attacks using convolutional neural networks. In: Proceedings of data analytics and management. Springer, Singapore, pp 293–304
15. Jemal I, Cheikhrouhou O, Hamam H, Mahfoudhi A (2020) SQL injection attack detection and prevention techniques using machine learning. Int J Appl Eng Res 15(6): 569–580
16. SQL injection dataset (2021). https://www.kaggle.com/datasets/syedsaqlainhussain/sql-injection-dataset