Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7386049, 16 pages
https://doi.org/10.1155/2022/7386049

Research Article

Exploiting Machine Learning to Detect Malicious Nodes in Intelligent Sensor-Based Systems Using Blockchain

Maimoona Bint E. Sajid,[1] Sameeh Ullah,[2] Nadeem Javaid,[1,3] Ibrar Ullah,[4] Ali Mustafa Qamar,[5] and Fawad Zaman[6]

[1]*Department of Computer Science, COMSATS University Islamabad, Islamabad 44000, Pakistan*
[2]*School of Information Technology, Illinois State University USA, Normal, IL 61761, USA*
[3]*School of Computer Science, University of Technology Sydney, Ultimo, NSW 2007, Australia*
[4]*Department of EE, University of Engineering and Technology Peshawar, Bannu 28100, Pakistan*
[5]*Department of Computer Science, College of Computer, Qassim University, Buraydah 52571, Saudi Arabia*
[6]*Department of ECE, COMSATS University Islamabad, Islamabad 44000, Pakistan*

Correspondence should be addressed to Nadeem Javaid; nadeemjavaidqau@gmail.com

Received 29 July 2021; Revised 30 November 2021; Accepted 13 December 2021; Published 18 January 2022

Academic Editor: Abdul Basit

In this paper, a blockchain-based secure routing model is proposed for the Internet of Sensor Things (IoST). The blockchain is used to register the nodes and store the data packets' transactions. Moreover, the Proof of Authority (PoA) consensus mechanism is used in the model to avoid the extra overhead incurred due to the use of Proof of Work (PoW) consensus mechanism. Furthermore, during routing of data packets, malicious nodes can exist in the IoST network, which eavesdrop the communication. Therefore, the Genetic Algorithm-based Support Vector Machine (GA-SVM) and Genetic Algorithm-based Decision Tree (GA-DT) models are proposed for malicious node detection. After the malicious node detection, the Dijkstra algorithm is used to find the optimal routing path in the network. The simulation results show the effectiveness of the proposed model. PoA is compared with PoW in terms of the transaction cost in which PoA has consumed 30% less cost than PoW. Furthermore, without Man In The Middle (MITM) attack, GA-SVM consumes 10% less energy than with MITM attack. Moreover, without any attack, GA-SVM consumes 30% less than grayhole attack and 60% less energy than mistreatment. The results of Decision Tree (DT), Support Vector Machine (SVM), GA-DT, and GA-SVM are compared in terms of accuracy and precision. The accuracy of DT, SVM, GA-DT, and GA-SVM is 88%, 93%, 96%, and 98%, respectively. The precision of DT, SVM, GA-DT, and GA-SVM is 100%, 92%, 94%, and 96%, respectively. In addition, the Dijkstra algorithm is compared with Bellman Ford algorithm. The shortest distances calculated by Dijkstra and Bellman are 8 and 11 hops long, respectively. Also, security analysis is performed to check the smart contract's effectiveness against attacks. Moreover, we induced three attacks: grayhole attack, mistreatment attack, and MITM attack to check the resilience of our proposed system model.

1. Introduction

The wireless sensor networks (WSNs) play a vital role in the Internet of Sensor Things (IoST). The IoST networks consist of sensor nodes that are deployed for the environmental monitoring [1]. These sensor nodes are used to sense ambient information like temperature, humidity, and pressure [2]. IoST is an emerging domain, which supports different applications like industrial Internet of Things (IoT), smart cities, air pollution detection, and underwater monitoring.

These networks face well-known issues due to deployment in harsh and unattended environments like attackers can attack the network by compromising the sensor nodes, which have very sensitive information like identification (ID) and location [3, 4]. The credentials are used in different cryptographic functions like encryption and decryption for

generating cipher text. However, these credentials are misused by physically accessing the nodes. As a result, the authors propose different mechanisms to address this issue. The authors store their data at a centralized server, which can securely and privately keep the records using different cryptographic techniques [5, 6]. Moreover, as the systems are controlled by a centralized authority, therefore, it is easy to be manipulated, which can lead to trust issues.

1.1. Problem Statement. The IoST nodes are threatened by the external malicious nodes (MNs), which eavesdrop the communication channels and extract personal information for their interest. Also, the nodes perform malicious activities during routing operations in the network and cause the grayhole attack [7]. To address the issue, the authors in [8] propose a model that consists of two blockchains, encryption schemes, and digital signatures to achieve authentication in the IoST. However, the usage of two blockchains causes extra communication and computational overheads. Moreover, the authors in [9, 10] use blockchain to resolve a single point of failure issue and trace the malicious activities using Merkle tree. The author in [11] uses blockchain-based secure hashing algorithm for authentication of nodes. The algorithm helps to detect MNs in the network. In [12, 13], the authors propose a blockchain-based secure data storage. However, data storage over the blockchain is costly as compared to other centralized storage platforms. In [14], trust aware localized routing is performed to find the secure route. Also, a blockchain-based authentication mechanism is proposed. However, high energy is consumed by the forwarding nodes when delivering the data packets without considering the optimal route. In [15], a blockchain-based nonrepudiation service provisioning scheme is proposed where denial of services by both the provider and client is not possible. The reason is that the proposed scheme uses homomorphic hashing for service verification, which consumes high computational cost.

1.2. Research Objectives. The research objectives (ROs) of this research work are as follows.

RO 1. The CHs are used to verify and aggregate the nodes' data and perform extensive routing. Therefore, CHs die early because of large computational overhead. One of the objectives of this research is to reduce the computational overhead (related research questions (RQs) are 1, 2, and 3).

RO 2. In multihop routing, a source node has multiple paths to forward the data packets towards the destination. However, selecting the longest path increases the delay. Therefore, it is one of our objectives to find the shortest path from source to destination node (related RQ is 4).

RO 3. In multihop routing, the malicious nodes drop the data packets. Therefore, it is our objective to detect the malicious nodes in order to increase the network throughput and Packet Delivery Ratio (PDR) (related RQ is 5).

1.3. Research Questions. RQs are given on the basis of ROs.

RQ 1. How to register the nodes that participate in routing?

RQ 2. Where the routing data will be stored?

RQ 3. How routing data is verified?

RQ 4. How to find the shortest path for forwarding the data?

RQ 5. How to detect the malicious nodes during routing?

1.4. Research Answers. Research answers (RAs) to the RQs are given as follows.

RA 1. The blockchain is used to register nodes in the network (for more details, see Section 3).

RA 2. Routing data, which includes the number of nodes, sent from source to destination and stored in the blockchain (for more details, see Section 3).

RA 3. The PoA consensus mechanism is used to verify the routing data (for more details, see Section 3).

RA 4. The Dijkstra algorithm is used to find the shortest path from source to destination (for more details, see Section 3).

RA 5. The GA-SVM and GA-DT are used to classify the nodes as malicious and legitimate (for more details, see Section 3).

1.5. Contributions. The abovementioned problems are resolved in this paper. The research contributions of the work are given as follows.

(i) A lightweight blockchain-based registration and authentication mechanism is proposed to make the network secure

(ii) The blockchain is used to store routing information in a decentralized fashion so that nobody can alter it

(iii) PoA consensus mechanism is used to reduce the computational cost consumed by PoW because in our scenario, nodes are resource constrained and are not capable of solving the puzzle

(iv) MNs are detected using the Support Vector Machine (SVM) and Decision Tree (DT) algorithms to make the system secure from internal malicious nodes

(v) We have performed security analysis for evaluating the smart contract using Oyente tool. Moreover, the proposed model's robustness against attacks is evaluated by inducing three attacks in the attacker model: grayhole attack, mistreatment attack, and MITM attack

1.6. Blockchain. The blockchain is explored by different authors to make the network free from concept of centralization because blockchain consists of a decentralized and distributed ledger. For decision making, a third party is replaced with a smart contract, which is used to write all the agreements. The data saved in blockchain is immutable and does not face a single point of failure issue [16]. The blockchain has different applications like healthcare [17], banking, energy trading [18], and smart cities [19]. The blockchain has three types: public, private, and consortium. In the public blockchain, any node can join the network and each node has the authority to add and access the data within the blockchain. In private blockchain, the data is not publicly available and only the authorized nodes have the authority to add and access the data, whereas the

consortium blockchain is a semiprivate blockchain, where multiple organizations become part of the network. In this type of blockchain, different nodes have different rights in the network. For example, some nodes have the authority to perform and validate transactions while others are authorized to access the stored data only. Furthermore, different consensus mechanisms like Proof of Work (PoW), Proof of Authority (PoA), and proof of stake are used to make consensus between nodes to add new data blocks in the blockchain. Moreover, in PoW, the miners solve the mathematical puzzle. A miner who solves the puzzle first has the authority to add a new block in the blockchain. In the PoA, preselected nodes perform mining for adding new data blocks in the blockchain in which miners are selected based on the reputation values [7].

The rest of the paper is organized as follows. Section 2 discusses the related work. The proposed model is explained in Section 3. In Section 4, simulation results are presented. Section 5 provides the conclusion.

2. Related Work

In this section, categorization of related work is done on the bases of addressed limitations.

2.1. Authentication and Privacy Leakage. In [1], the authors motivate the users to use mobile devices by providing incentives. However, the authors ignore the privacy leakage issue. The customers' trust is not developed because of privacy leakage. The privacy of data is the main issue in crowdsensing. In [8], traditional authentication of IoST depends on a third party, which may act maliciously. In [20], IoST is useful for sending data from source to destination. However, data is not secure because of malicious attacks. Sensor nodes are not authentic. Therefore, they act maliciously and drop the data packets. In [15, 21], records are not secured and data privacy is compromised, which leads to trust issues in data trading. Furthermore, IoST generates excessive data, which causes privacy issue. In [22], the authors propose an authentication-based signature mechanism for securing the communication. Moreover, in [23], data trading is performed and high computational power is required for storing and trading data. However, the IoST has limited storage capacity and battery. The authors in [24] mention that single side authentication is not efficient and can be attacked by the outsider nodes. Also, a single centralized site could break down due to any external or internal attack. In contrast to the mentioned problems, the authors propose a blockchain-based decentralized authentication model, which provides the best fault tolerance to the network. The authors in [25] state that in smart cities, network can extract data from outdated devices, which may be operated in unattended environments.

2.2. Malicious Nodes' Detection and Secure Routing. Uddin et al. propose the hierarchical transmission of data in underwater sensor network, where sensors are mobile and data transmission is not secure because of unattended environment [26]. The authors in [27] propose an intrusion detection system on the basis of data generated by the intruders. The authors use the deep learning techniques to train the model for intrusion detection. For hierarchy distribution, fog and cloud are used with secure data transmission. This model ensures the legitimacy of data. Feng et al. state in [28] that traditional models have a centralized database in which temperature, humidity, and gas information of the seafood like shellfish is stored. However, due to the usage of a centralized database, there are high chances of malicious attacks that tamper the data to affect the quality of shellfish. In [10], the WSN faces two types of attacks: internal and external. In the internal attack, internal and legitimate nodes (LNs) become malicious, which send and access the data as legitimate nodes, which affect the overall performance of the network, while in the external attack, the attacker can attack the entire network from outside. In [12], the WSN nodes work as mobile nodes. When any node sends the data from the source to the destination, intermediate mobile routing nodes act as MNs, which tamper and drop the data packets. The authors state in [16, 29] that security and privacy risks increase through modification of the data packets while sending them from the source to destination. In traditional architectures [30], WSNs are centralized and are controlled by the third party, which stores the data. However, a single point of failure issue occurs, and MNs tamper the data. In [31–33], IoST faces the issue of internal attacks that affect the network's lifetime, scalability, and throughput. In [7], MNs broadcast false destination address, which affects the network's performance because data packets are continuously moved in the loop. As a result, the data packets are timed out and are dropped. The authors in [34] assume that BSs could also behave maliciously in the network. Therefore, a blockchain-based secure verification mechanism is proposed to authenticate the BSs. In [11], the author proposes a lightweight authentication and security protocol because sensor and IoT nodes are small in physical size. Also, the author proposes a model for interoperability between different vendors' devices. In [35], MNs are deployed in the network and where they act as beacon nodes. MNs send tampered data to the base station and broadcast the wrong location of the beacon nodes. Therefore, localization errors occur during the localization process. Moreover, wrong localization affects the lifetime of WSN and consumes high energy. In [36], the authors state that the localization of unknown nodes is the main issue in the range free WSNs. However, some beacon nodes act maliciously and broadcast their wrong location. Resultantly, unknown nodes' positions are not accurately calculated. In [37], the authors propose a lightweight routing mechanism based on swarm intelligence for fast and efficient communication in the network.

2.3. Single Point of Failure. The traditional routing protocols are centralized and use a central authority for authentication of IoST [9]. In [29], IoST is widely used in smart city and the number of IoST is increasing on daily basis. However, the model is centralized in which only cloud servers are used for storing and accessing the data. In [21], sensors are used for sensing the data and saving the records on a centralized database. There are high chances of single point of failure. In

[38, 39], the software defined network (SDN) is commonly used for the communication of nodes. The SDN separates the data plane from the control plane. However, the SDN is centralized and faces the issue of single point of failure. Security issue also arises in the centralized system. In [40], IoST also faces the main challenge of security. Moreover, devices are centralized, and issue of single point of failure occurs.

2.4. Resource Constrained Nodes. In traditional models [41–43], blockchain is only used for maintaining the data in WSNs. However, the PoW consensus mechanism consumes high computational power. In [44, 45], mobile sensors cover large areas of smart cities. However, sensor nodes are resource constrained, and their deployment and maintenance cost is very high. In [46], the sensor nodes have to send data to the base station. The sensor nodes are resource constrained and have no computational power to send data over a large area. In [47, 48], IoST uses WSNs for storing the data. However, WSNs are resource constrained.

3. Proposed System Model

In this section, the details of the proposed system model are presented. Firstly, the blockchain is deployed. Secondly, registrations and authentication mechanism is proposed. Thirdly, MNs are detected using two machine learning algorithms: SVM and DT. Fourthly, the details of the shortest path for routing are given, and secure routing is performed, as shown in Figure 1.

3.1. Network Deployment. In the proposed system model, 200 nodes are deployed in the network in which one node is appointed as a base station, some nodes are selected as cluster heads (CHs) based on their resources while remaining nodes are regarded as ordinary nodes. Moreover, blockchain is deployed on the CHs and base station because they have sufficient storage and computational resources. The private blockchain is integrated with IoST to ensure the security of the network and data by keeping record of every transaction in its ledger. Also, the nodes are registered in blockchain where a unique number is assigned to each node. Moreover, the PoA consensus mechanism is suitable for the private blockchain; therefore, it is implemented for verifying the transactions.

3.2. Registration and Authentication. Authentication of nodes is necessary because the nodes are deployed in an unattended and harsh environment. In this environment, nodes are more likely to be attacked by the attackers. These attackers physically access and misuse the connection of particular nodes with the network. Also, attackers can reuse the IDs of the nodes to reenter the network and for doing other malicious activities. Therefore, many authors propose different registration and authentication mechanisms using blockchain where the chances of nodes acting maliciously are minimum because of blockchain's unique characteristics, discussed in Section 1. However, there are still chances of different types of attacks that can be done by manipulating the credentials of the network nodes, although, in most of the papers, hashes are stored on the blockchain, which can be guessed by different methods such as brute force attack and are involved in the malicious activities. Therefore, in this section, a secure and lightweight registration and authentication mechanism is proposed. However, the storage over blockchain is very costly as compared to traditional storage mechanisms. Therefore, the process of registration and authentication is made lightweight to minimize the burden of storage from blockchain. According to our network's requirement, the system will be less benefited because nodes have limited time period for communication. Therefore, nodes' lifetime will finish before accessing and guessing the hashed value of the credentials by the attackers. The detailed and step-wise workflow of the registration and authentication mechanism is as follows. Moreover, Algorithm 1 also depicts the pseudocode of the proposed registration and authentication mechanism.

Step 1. The first step of the registration is to send the credentials to the blockchain B. The package of registration request req $\longrightarrow B$ contains identity of the sensor node ID_S and location of the node L_S.

$$\mathrm{Reg}_{\mathrm{req}\longrightarrow B} = (\mathrm{ID}_S, L_S). \tag{1}$$

Step 2. Blockchain will generate a unique three digit number $\mathrm{No.}_{\mathrm{unique}}$. These digits will be added to the above package to get rid from the collision of hashes. The unique number will be assigned to the node as its password for future correspondence.

$$B \longrightarrow \mathrm{hashing} = \left(\mathrm{ID}_S, L_S, \mathrm{No.}_{\mathrm{unique}}\right)_{\mathrm{hash}}. \tag{2}$$

Step 3. The hash and the unique number are stored in the blockchain for future use like authentication. The hashes will be converted to the cipher text, which would not be guessed by the attacker even if ID and location of the node are obtained.

Step 4. After successful registration of the nodes, if a node wants to access the network, it will send authentication request to blockchain. The authentication request contains the following data in its package.

$$\mathrm{Auth}_{\mathrm{req}\longrightarrow B} = \left(\mathrm{ID}_S, L_S, \mathrm{No.}_{\mathrm{unique}}\right). \tag{3}$$

Step 5. The blockchain will generate the hash of the above given credentials and will compare it with the already stored hash. If the match is successful, blockchain will allow the sensor nodes to communicate successfully. Otherwise, the authentication error message will be popped up. All the steps involved in authentication and registration process are given in Algorithm 1.

Note. It is our assumption that the unique number is securely shared with the node. However, we did not consider the method of sharing the number. Sharing the unique number is necessary because if some other entity knows about this number, we will lose our objective to use this number.

Step 6. After the authentication process, the source node sends the data packets to the destination node through

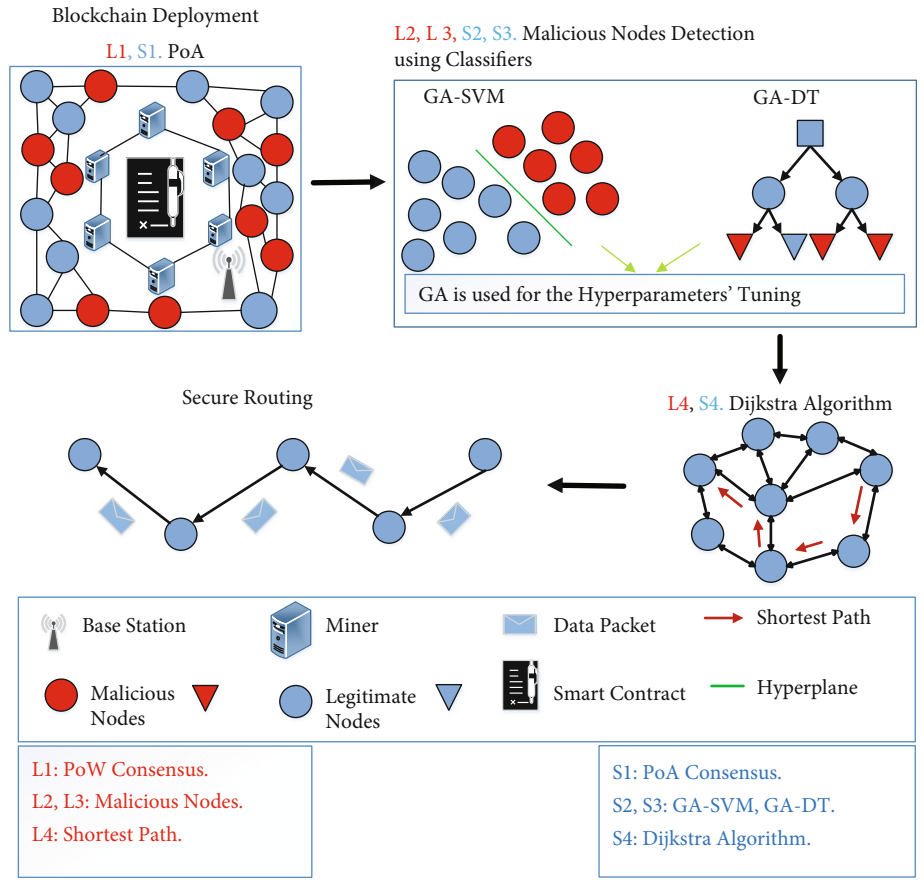

Figure 1: GA-SVM and GA-DT-based MN detection during routing using blockchain.

1 Initialization
▷ Registration
2 Send to Blockchain (ID_S, L_S);
3 Blockchain Generate $No._{unique}$;
4 Blockchain Calculates and Stores the Hash $(ID_S, L_S, No._{unique})$;
▷ Authentication
5 Send to Blockchain $(ID_S, L_S, No._{unique})$;
6 Blockchain Calculates the Hash $(ID_S, L_S, No._{unique})$;
7 if Hash == Already Stored Hash then
8 Node is Authenticated and Allowed to Communicate;
9 else
10 Node is Deauthenticated and Revoked by the Blockchain;
11 end

Algorithm 1: Registration and authentication process.

intermediate nodes like CHs, which perform secure routing by storing the routing data on the blockchain.

3.3. Malicious Node Detection. After the deployment of the network, the presence of MNs in the network affects the network performance. In our proposed model, these nodes are detected by GA-DT and GA-SVM. Before detecting MNs, we first synthesize the data for the algorithms.

3.3.1. Synthetic Data Generation. In the IoST, each node sends the data packets to the destination. After forwarding the data packets to the destination, we calculate how many

data packets are dropped, misrouted, modified by the intermediary MNs, and successfully received by the destination. The following features are used to generate the synthetic data [49].

(a) *Packet drop Ratio (PdR)*. PdR is the ratio of total number of data dropped *Pd* to the total number of data packets coming from the node *Pi*, which is calculated using Equation (4) [49].

$$\text{PdR} = . \tag{4}$$

(b) *Packet Modification Ratio (PMR)*. The PMR is the ratio of the total number of data packets modified Pm to the total number of data packets coming from the source node *Pi*, which is calculated using Equation (5) [49].

$$\text{PMR} = . \tag{5}$$

(c) *Packet Misroute Rate (PMiR)*. The PMiR is the ratio of the total number of data packets misrouted Pmi to the total number of data packets transmitted by the source node *Pf*, which is calculated using Equation (6) [49].

$$\text{PMiR} = \frac{\text{Pmi}}{Pf}. \tag{6}$$

(d) *Packet Delivery Ratio*. The PDR is the ratio of the total number of data packets successfully received at the destination *pt* to the total number of data packets coming from the source node *Pt*, which is calculated using Equation (7) [49].

$$\text{PDR} = . \tag{7}$$

The thresholds are defined for PdR, PMR, PMiR, and PDR in Algorithm 2, and nodes are labeled as malicious and legitimate. The synthetic dataset is used for GA-SVM and GA-DT. Moreover, threshold used in Algorithm 2 is taken from [49].

3.3.2. Classification Using Genetic Algorithm-Decision Tree. DT splits the data into the root, internal, and leaf nodes that form the tree and classifies a particular node as malicious or legitimate. In DT, the dataset is given to root node, which performs testing and splits the dataset into subclasses. The internal nodes represent the outcome of the root node. The leaf nodes that do not split further are the results of classification and represent the class labels. The hyperparameters used in GA-DT are as follows.

```
1 Initialization (nodes =200, data packets =200)
2 for i =1 : nodes do
3 for j =1 : data packets do
4 if receivepackets(i) ≠ inf then
▷S(i) is a structure
5 PDR(i) = (S(i).transmitted data)/j;
6 PdR(i) = S(i).drop packets/j;
7 PMiR(i) = (S(i).misroute packets)/forwarded packets;
8 PMR(i) = (S(i).modified packets)/(S(i).transmitted data);
9 end
10 if PdR(i) ≥2 && PMiR(i) ≥3.9800 &&
PMR(i) ≥0.9975 && PDR(i) ≥98.2500 then
11 MN =0;
12 else
13 LN =1;
14 end
15 end
16 end
```

Algorithm 2: Synthetic data generation.

(a) *Maximum Features*. It shows that to what extent a DT should be splitted. When a tree is splitted deeply, better classification is performed. The maximum depth values are auto, square root (sqrt), log, and none.

(b) *Minimum Sample Leaf*. It is the minimum number of samples present at the leaf node. The parameters are int, float, and none optional.

(c) *Minimum Sample Split*. It is the splitting of the internal node in the minimum range. The parameters are int, float, and optional.

(d) *Criterion*. It is used to measure that how perfectly a DT is splitted. Gini index and entropy are features.

Gini index and entropy criteria are used to measure that either DT is splitted perfectly or not. The data is labeled randomly, and Gini is used to measure the amount of data falsely labeled. If the Gini's value is low, it means classes are perfectly labeled. Moreover, entropy is used for the best splitting of DT. The best classification is performed by tuning the hyperparameters.

The aforementioned parameters are tuned using GA, which evaluates the input values in such a manner that it generates the best output. In GA [50, 51], population is generated randomly, as shown in Algorithm 3. The fittest individual is selected as a parent on the basis of the objective function. After the parent's selection, crossover is performed and offsprings are generated. In the mutation, offspring genes are altered and the optimal solution is generated. As a result, the best hyperparameters are selected from the given features. After tuning the parameters, training and testing are performed. Then, DT classifies LNs and MNs. The detected MNs do not participate in the routing.

```
1 load dataset;
2 split the data 70% = testing and 30% = training;
3 initialize hyperparameters = (Maximum Depth, Minimum Sample Leaf, Minimum Sample Split, Criterion);
4 GA:Maximum Depth, Minimum Sample Leaf, Minimum Sample Split, Criterion;
5 initialize population
6 (generation,populationsize,offspringsize);
7 perform tuning of hyperparameters;
8 perform selection;
9 perform crossover;
10 perform mutation;
▷Decision Tree is DT
11 train DT on selected parameters;
12 DT saves the model;
13 DT tests the model;
14 DT loads the model;
```

Algorithm 3: Detection of malicious nodes with classification by GA-DT.

3.3.3. Classification Using Genetic Algorithm and Support Vector Machine. SVM is a supervised learning technique, and it is the foremost classification model. It performs linear and nonlinear classification. In the proposed model, a linear classification is performed and the hyperplane is created by the line formula [52], as given in Equation (8).

$$y = mx + b, \tag{8}$$

where m is the gradient that shows the height of the y-axis and divided by the distance of the x-axis. The variable b shows the interception of x and y. A hyperplane is created through line equation by defining $x = (x1, x2)$ and $w = (m, -1)$. By putting the values of x and w in Equation (8), Equation (9) is derived [52].

$$w.x + b = 0. \tag{9}$$

The hyperplane is a decision boundary that is used to separate two different classes. Each class margin shows the distance of support vector to the hyperplane. When margin is maximum, the error rate is minimum. In the proposed model, the synthetic dataset is loaded and classification is performed using SVM. The hyperplane classifies both MNs and LNs. SVM hyperparameters are tuned for the best classification.

SVM hyperparameters are as follows.

(a) *C Classification*. Minimum accuracy of classification is high, and perfect decision boundary is created. Its values are 0.1, 1, 2, 5, 7, and 10.

(b) *Kernel*. It tells that how hyperplane is created for one dimension or multiple dimensions. Radial basis function, linear, and polynomial are features of kernel.

The aforementioned parameters are tuned using GA. As a result, the best values are selected from the given hyperparameters. After tuning the hyperparameters, the model is trained, as given in Algorithm 4. Then, SVM excellently classifies both legitimate and malicious classes. After the classification, GA-SVM and GA-DT results are compared in which GA-DT has 96% accuracy and GA-SVM has 98% accuracy. Therefore, MNs are detected by GA-SVM. On the basis of GA-SVM's accuracy, MNs are revoked and only the LNs are allowed in the network.

3.4. Shortest Path for Routing. After the detection of MNs, in the third step, the source node selects the shortest path to the destination and performs secure routing in the absence of MNs. In the proposed model, the Dijkstra algorithm [53] is used to find the shortest path and it considers all paths from the source to the destination node. The energy consumption of the nodes is reduced after finding the shortest path on the basis of weights assigned to each edge of the node [54, 55]. In Figure 2, the source node is CH1 and the destination node is CH7. The source node has six possible paths going towards the destination. The Dijkstra algorithm finds the shortest path from the given six paths on the basis of the distance. As a result, the shortest path distance is eight hop and CH2, CH3, and CH6 are intermediary nodes used to forward the data packets to the destination, as given in Algorithm 5. After shortest path selection, secure routing is performed.

Table 1 shows the mapping between limitations identified, solutions proposed, and validations done. The first limitation (L1) is that PoW consumes high computational power.

Therefore, in the proposed model, PoA is used for verifying the data because the transaction cost of PoA is less than PoW. The second and third limitations (L2 and L3) are MN detection and grayhole attack, respectively, which are detected using GA-SVM and GA-DT. The features that are used to train the model are PDR, PMiR, PMR, and PdR. GA is used for tuning the hyperparameters of SVM and DT. After tuning the hyperparameters, optimal classification is performed. In the fourth limitation (L4), high energy is consumed using the longest path for routing. The source node sends the data packets to the intermediary nodes who forward the packets to the destination. If the

```
1 load dataset;
2 split the data 70% = testing and 30% = training;
3 initialize hyperparameters = (kernel, c);
4 GA:kernel, c;
5 initialize population
6 (generation,populationsize,offspringsize);
7 performs tuning;
8 Perform selection (c = [0.1, 1, 2, 5, 7, 10]);
9 Perform crossover ([0.1, 1, 2, 5, 7, 10]);
10 Perform mutation ([0.1, 1, 2, 5, 7, 10]);
11 selection (kernel = 'radialbasisfunction','linear','polynomial');
12 crossover ('radialbasisfunction','linear','polynomial');
13 mutation ('radialbasisfunction','linear','polynomial');
14 SVM trains the model on the selected parameters [c=2, kernel=linear, degree=1];
15 SVM saves the model;
16 SVM tests the model;
17 SVM loads the model;
```

Algorithm 4: Detection of malicious nodes with classification by GA-SVM.

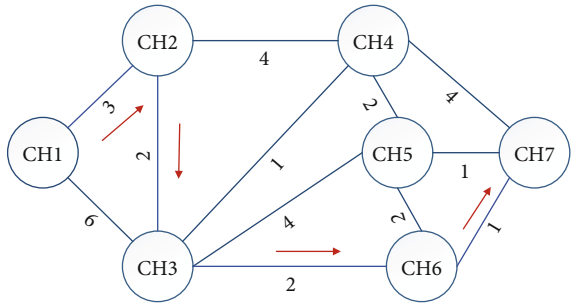

Figure 2: Shortest path selection using the Dijkstra algorithm.

intermediary node is not in the range of destination node, it forwards the data packets to the next intermediary node. In this way, a long routing path is created and energy consumption is increased. In the proposed model, the Dijkstra algorithm is used to find the shortest path. The Dijkstra algorithm computes the distance from all nodes and selects the shortest path. After the selection of the shortest path, secure routing is performed by sending the data packets to the destination.

4. Performance Evaluation

The simulations are performed to validate the performance of the proposed system model. The ML techniques are exploited parallelly working with blockchain. GA-SVM and GA-DT techniques are used to classify LNs and MNs on the basis of their behaviour. The features on which the model is trained are PDR, PdR, PMR, and PMiR. The class labels are malicious and LNs. The dataset is splitted into 70% training and 30% testing sets. Moreover, novelty of system model is depicted through Table 2. The simulation environment used in the proposed work comprises Metamask, Ganache, and Remix-IDE for the implementation of blockchain. Moreover, Python-based Web3.api is used for the interaction of blockchain with networks. ML techniques

```
1 if Source == Destination then
2 Cost = 0;
3 Route=Source;
4 end
5 if Destination==1 then
6 Destination = Source;
7 end
8 Adjacency=exchange the node (Adjacency,1,Source);
9 LengthA=size(Adjacency);
10 */Weight is W*/
11 for i=2 : LengthA do
12 W(1,i)=i;
13 W(2,i)=Adjacency(1,i);
14 end
15 for i=1:Length Adjacency do
16 Destination(i,1)=Graph(1,i);
17 Destination(i,2)=i;
18 end
19 while Cost < = (size(W,1)-1) do
20 Cost=Cost+1;
21 for i=1:size(Distance,1) do
22 Calculate the distance;
23 end
24 Destination=Sort the rows and find Destination;
25 Add the Distance of each Route to the Destination;
26 end
```

Algorithm 5: Dijkstra algorithm.

are implemented in Python to classify LNs and MNs. Moreover, the network parameters are depicted in Table 3.

4.1. Simulation Results and Discussion of Proposed Model. In Figure 3(a), GA-SVM is compared with DT, SVM, and GA-DT to show the effectiveness of the proposed model. DT is used for the classification of LNs and MNs. However, DT consumes large time for training the model because it is complex in nature. Therefore, more computational resources are required to train the model. DT has 88% accuracy, and the hyperparameters used for DT are criterion, minimum

sample leaf, and maximum features. The accuracy is low as compared to other models because DT itself chooses the hyperparameters for tuning. On the other hand, SVM's accuracy is 93%, which is more than DT because SVM performs better linear classification than DT. It also works better with few features and data instances as compared to DT. Furthermore, GA is used to tune the hyperparameters of DT and SVM to improve their accuracy. GA performs tuning and selects the best parameters from the given parameters. As compared to simple DT and SVM, GA does not select the parameters itself while it takes the best hyperparameters using crossover and mutation. In crossover, the best parents are taken from the population and crossover is performed between them. If the resultant offsprings are better than the existing ones, then they are replaced with previous ones. Moreover, mutation is used for making the solutions more diverse. Resultantly, the required results are obtained. The accuracy of GA-DT is 96%, which is higher than SVM and DT. The reason is that SVM has higher accuracy as compared to DT, and when parameters of SVM are tuned using GA, its accuracy further increases. Moreover, the proposed model uses GA-SVM for classification, and it has 98% accuracy, which is higher than DT, SVM, and GA-DT because SVM's accuracy is greater than DT's accuracy. SVM's input parameters are tuned using GA, which helps SVM to achieve better results than simple SVM, DT, and GA-DT. GA-SVM's accuracy is high because GA efficiently tunes C, kernel, and degree hyperparameters. GA evaluates the best value for the given parameters and tunes the parameter according to the selected value. Simple SVM is not good for predicting the range of hyperparameters. It is because SVM selects the hyperparameters on its own, and its accuracy is less than GA-SVM.

The precision is the ratio of true positive to the sum of false positive and true positive. The precision of DT is 100% while SVM has 92% precision. The reason is that SVM has less number of true positives as compared to DT. GA-DT and GA-SVM have 94% and 96% precision, respectively, as shown in Figure 3(b).

Figure 4(a) illustrates that the PdR is decreased when only the LNs are present in the network. In the presence of MNs, the PdR increases to 53% because MNs receive the data packets and do not forward them to the destination. GA-SVM detects 21.75% MNs that are present in the network.

In Figure 4(b), the red bar shows that PMR is 85% in the presence of MNs. MNs tamper the data packets and forward them to the destination. The blue bar shows the number of nodes that do not modify the data packets.

MNs receive the data packets and misroute them to the wrong destination. In the presence of MNs, PMiR increases, as compared to the LNs, as shown in Figure 5(a). When the data packets are misrouted, it causes network congestion.

PDR is higher in the absence of MNs because maximum number of data packets sent from the source are successfully received at the destination, as shown in Figure 5(b). When the PDR is maximum, it increases the performance of the IoST. GA-SVM classifies MNs and LNs. When MNs are revoked from the network, PDR becomes 99.72%.

The source node sends the data packets to the destination node. In Figure 2, seven nodes are selected for the routing. One and seven are source node and destination node, respectively. The Dijkstra algorithm finds the shortest path in the network from the source to the destination. As shown in Table 4, distances are calculated for possible paths from the source to the destination. Distance is calculated on the basis of weights. The selected path's distance is eight, which is smaller as compared to other paths. The source node CH1 sends the data packets using the shortest path, which consumes less energy. The Dijkstra algorithm is compared with the Bellman Ford algorithm. The Bellman Ford's calculated distance is 11 hop, which is greater from Dijkstra for the same source and destination. The Bellman Ford is also more time-consuming than Dijkstra. Bellman Ford follows dynamic programming in which current problem is solved using previous solutions. Moreover, Dijkstra is a greedy algorithm, which solves the problem by finding optimal solution.

We use PoA consensus mechanism in the blockchain for nodes' registration and data storage. In Figure 6(b), PoA is compared with the PoW. The PoW uses the Ropsten network and PoA uses the Rinkeby network. The PoA consensus is used because private blockchain is deployed in the proposed model. The PoW consensus works efficiently in the public blockchain. As a result, PoW is expensive to deploy the smart contract and consumes higher transaction cost as compared to PoA. Transaction cost occurs when a smart contract is deployed while storing the data of node. When the number of nodes increases, transaction cost also increases because the number of records increases.

4.2. Attacker Model. The security analysis of the proposed model is performed after inducing a grayhole attack [56], a mistreatment attack [57, 58], and MITM attack [59]. Their analysis is shown in Table 5. In the proposed model, we use GA-SVM for detecting malicious nodes and revoked them from the network.

4.2.1. Energy Consumption. Figure 7 shows the effect of grayhole, mistreatment and MITM attacks, and the proposed solution based on energy consumption. During routing, the nodes' energy is consumed when data packets are sent toward the destination. From the figure, it is shown that the grayhole attack has less impact on the energy consumption as compared to mistreatment attack. For the grayhole attack, the malicious nodes do not forward all of the data packets to the destination, and they drop some of the data packets before reaching the destination. Therefore, some of the data packets do not reach CHs, which make it to consume less energy. Furthermore, the mistreatment attack has a high energy consumption as compared to the grayhole and MITM attacks. In this attack, the malicious nodes tampered with the routing table and changed their destination. Therefore, the data packets are forwarded to the wrong destination, and the nodes consume high energy because malicious nodes give long paths and false destinations. Moreover, low energy is consumed in the MITM attack as compared to grayhole attack when sending the tampered

Table 1: Mapping of identified limitations with proposed solutions and validations.

Limitations identified	Solutions proposed	Validations done
L1: PoW utilizes high computational power [10].	S.1: PoA is used that utilizes low computational power.	V.1: transaction cost, as shown in Figure 6
L2: presence of MN in the network [29] L3: grayhole attack is possible on routing nodes [7].	S.2, S.3: GA-SVM and GA-DT are used for the detection of MNs.	V.2, V.3: accuracy, precision, PDR, PMR, PdR, and PMiR, as shown in Figures 3(a) and 3(b), 4(a) and 4(b), and 5(a) and 5(b)
L4: long paths deplete nodes' energy [14].	S.4: the Dijkstra algorithm is used to find the shortest path.	V.4: distance from source to destination is calculated, as shown in Figure 5.
L5: registration consumes more gas due to hybrid blockchain [14].	S.5: lightweight registration and authentication mechanisms	V.5: transaction cost, as shown Figure 6

Table 2: Feature comparison with respect to proposed model.

Contributions	Ref. No.	Similarities	Differences	Limitations
Registration and authentication	[8]	Distributed network authentication	Hybrid blockchain different levels of nodes registered at different platforms	Complex to manage two blockchains
	[30]	Authentication	Use of multiblockchains and cluster manager	Difficult to manage
Routing	[7]	Routing, MN detection	Calculate through Qlearning	Consumes more energy while discovering the route
	[14]	Distributed network, routing	Localization-based routing	Select relatively longer path, which causes rapid energy
MN detection	[10]	Distributed network, registration, MN detection	Detection through network parameters, calculation of node's reputation	Uses computationally complex consensus algorithm
	[29]	Detection through ML, distributed network	SDN-based network, multilevel detection	Considered very few attacks

Table 3: Simulation parameters.

Parameters	Values
Sensing area	$500 \times 500\ m^2$
Deployment	Random
Total nodes	200
Total data packets of each node	200
Consensus	PoA
Network interface	Wireless

data packets toward the destination. The reason is that when the node does not send the correct data packets to the destination, the node again requests for the data from the source nodes. Moreover, less energy is consumed after the detection of malicious nodes using GA-SVM in the proposed model. In GA-SVM, the legitimate nodes perform routing without tampering the data packets and changing routes; therefore, the energy consumed is less as shown in Figure 7.

4.2.2. Dead Nodes. Table 5 shows that the mistreatment attack highly affects the proposed model because it has a low network lifetime. In the mistreatment attack, the nodes consume high energy; therefore, all nodes are dead at 2100 rounds. Furthermore, in the MITM attack, all of the nodes are dead at 2300 rounds because the malicious nodes tampered with the data packets and forwarded the malicious data packets to CHs. Moreover, in the grayhole attack, the nodes do not consume high energy; therefore, their networks' lifetime is higher than both the other attacks. In the proposed model, GA-SVM is used to detect the malicious nodes and revoked them from the network. Using the proposed GA-SVM, the model is free from malicious nodes. The legitimate nodes consume less energy, and as a result, their networks' lifetime became high. Therefore, all of the legitimate nodes are dead at 4000 rounds as shown in Figure 8.

4.2.3. Grayhole Attack. In the grayhole attack, the value of PdR increases by 53% because the malicious nodes received the data packets and do not forward it to the destination. The GA-SVM model detects 21.75% malicious nodes that are present in the network and drops the data packets. Therefore, after the detection of malicious nodes, the value of PdR decreases as shown in Figure 9(a).

The value of PdR for the GA-SVM model is higher than the grayhole attack in the system. In the absence of the malicious nodes, the total number of data packets sent from the source is successfully received at the destination. The value of PdR is minimum in the presence of the malicious nodes because the malicious nodes do not forward the data packets toward the destination as shown in Figure 9(b). However,

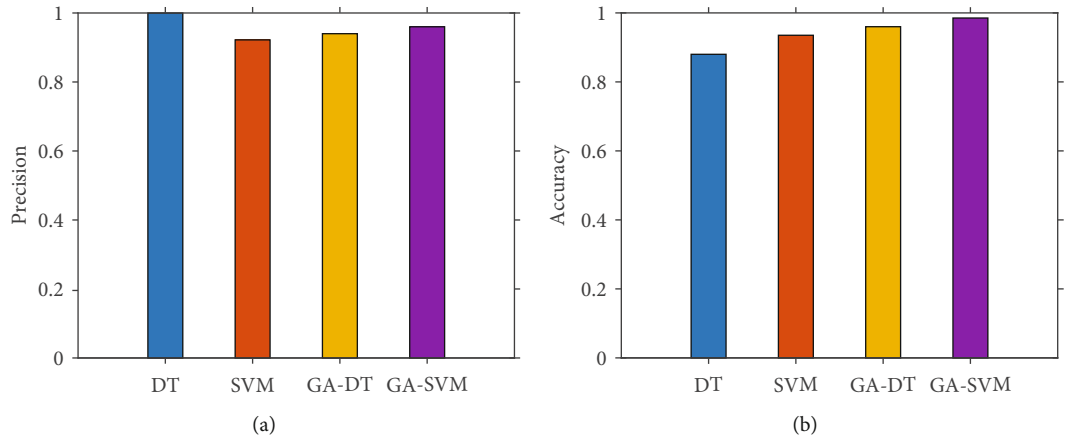

Figure 3: (a) Comparison of accuracy. (b) Comparison of precision.

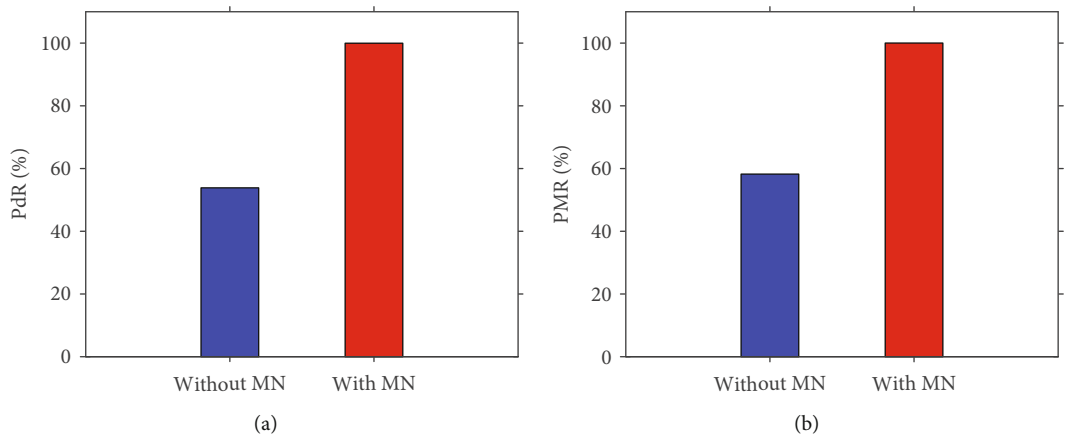

Figure 4: Comparison of (a) PdR between LNs and MNs and (b) PMR between LNs and MNs.

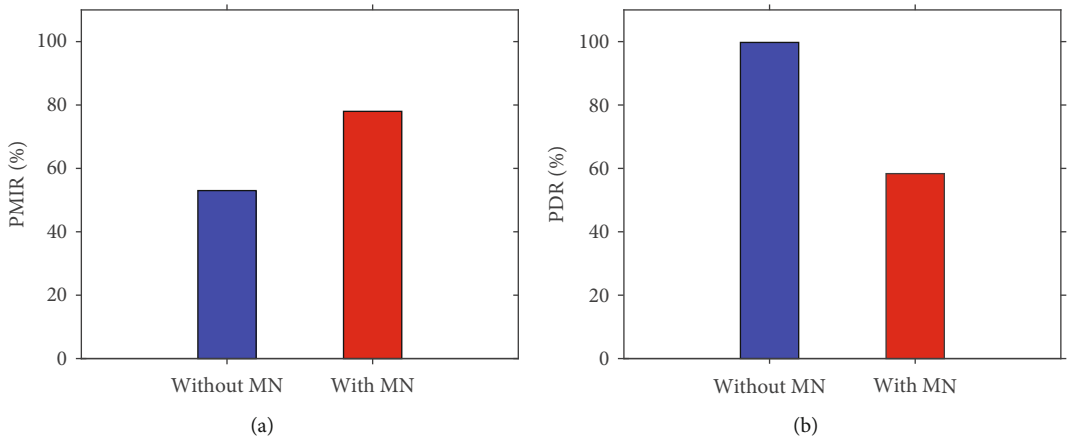

Figure 5: (a) Comparison of PMiR between LNs and MNs. (b) Comparison of PDR between LNs and MNs.

when the value of PdR is maximum, the performance of the IoST network is increased. Moreover, the GA-SVM model classifies the malicious and legitimate nodes in order to remove the malicious node from the network. After the malicious nodes are revoked from the network, the value of PdR increases up to 99.72%.

4.2.4. Mistreatment. In the misrouted attack, the malicious nodes received the data packets from source nodes; however, they are misrouted. In the presence of malicious nodes, the value of mistreatment is high as shown in Figure 10(a). In the network, if the data packets are misrouted, the network becomes congested. Otherwise, it will not be congested.

Table 4: Shortest path using the Dijkstra algorithm.

Paths	Distances (hop)
[1, 3, 6, 7]	9
[1, 3, 4, 7]	11
[1, 2, 4, 5, 7]	10
[1, 3, 5–7]	13
[1–3, 6, 7]	8
[1, 2, 4–7]	12

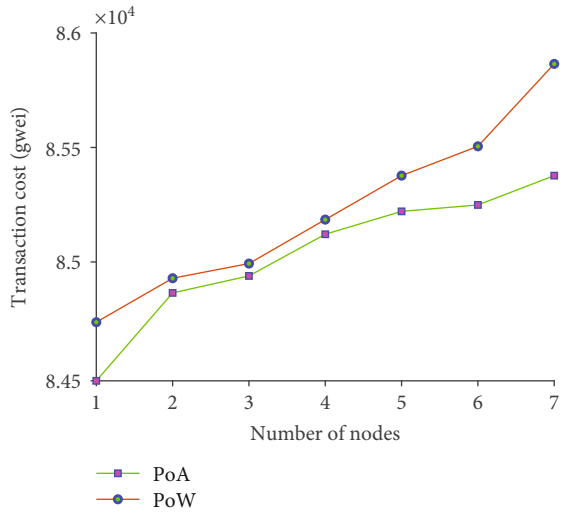

Figure 6: Transaction cost (TC) of PoA and PoW.

Table 5: Performance analysis of dead nodes in grayhole attack, mistreatment attack, MITM attack, and GA-SVM.

		First node dead	Last node dead
Attacks	Grayhole attack	400 rounds	3800 rounds
	Mistreatment attack	300 rounds	2100 rounds
	MITM attack	200 rounds	2300 rounds
Without attack	GA-SVM	700 rounds	4000 rounds

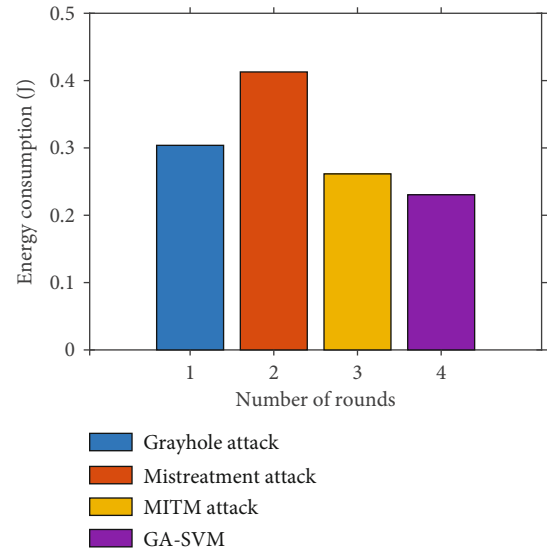

Figure 7: Performance analysis of energy consumption under grayhole, mistreatment, MITM attacks, and proposed GA-SVM solution.

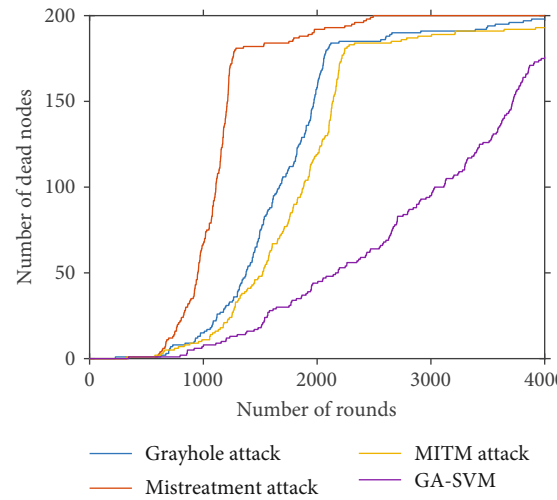

Figure 8: Performance analysis of dead nodes under grayhole, mistreatment, MITM attacks, and proposed solution GA-SVM.

4.2.5. MITM. In Figure 10(b), MITM attack is induced in the proposed model. Therefore, the value of PMR becomes high in the presence of malicious nodes, i.e., 89%. It is high because the malicious nodes tampered with the data packets and forwarded them toward the destination. After the detection of malicious nodes using GA-SVM, the value of PMR becomes low. The purple bar of GA-SVM shows the number of nodes that do not modify the data packets. The proposed system model is robust against the following two attacks.

(1) Spoofing Attack. In the spoofing attack, the malicious node takes the identity (ID) of the legitimate node and acts as a trustworthy node. This attack is not possible because all of the nodes are registered in the blockchain, which makes it difficult to compromise the system.

(2) Bad Mouthing Attack. In this attack, the malicious nodes change the reputation value of the forwarder nodes; as a result, the nodes become untrusted in the system. Therefore, this attack is not possible in the proposed model because the forwarder nodes are not selected based on their reputation values

4.3. Security Analysis of Smart Contract. The proposed smart contract is analyzed using an Oyente tool [4]. The tool is used to analyze the smart contract using symbolic

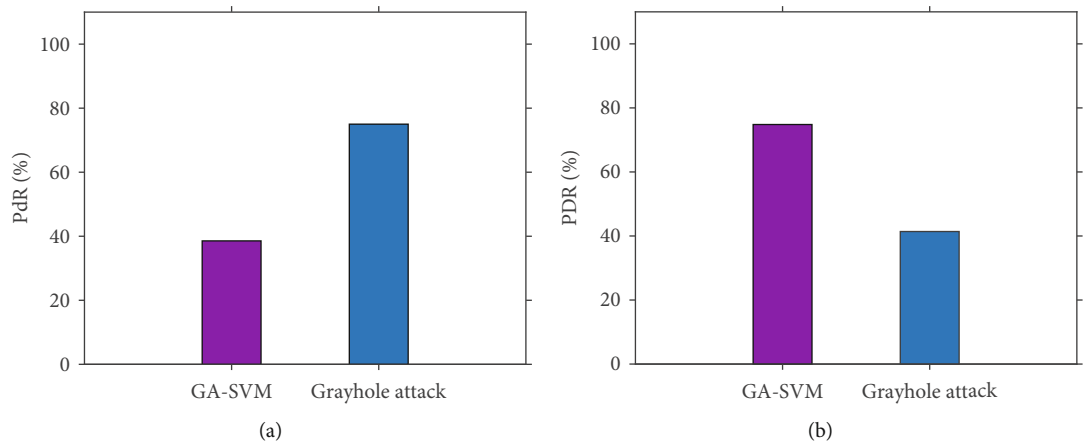

Figure 9: Comparison of (a) PdR between grayhole attack and GA-SVM and (b) PDR between grayhole attack and GA-SVM.

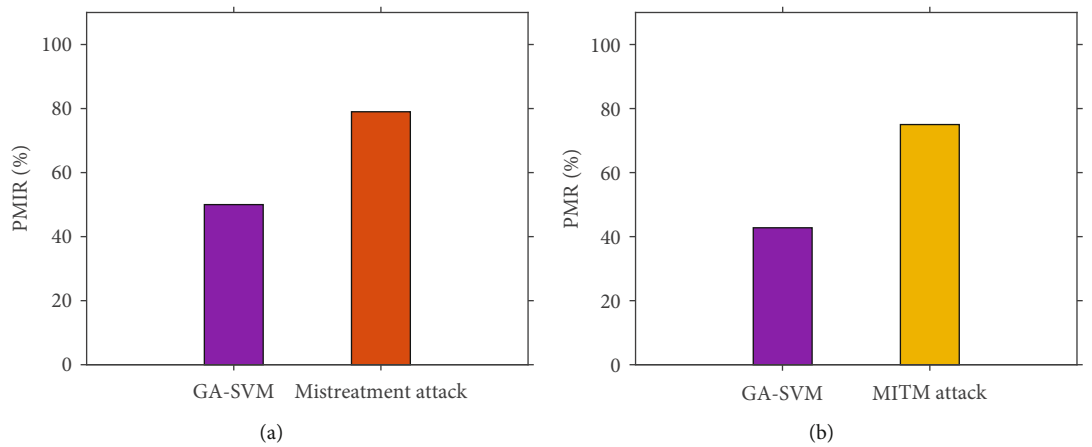

Figure 10: Comparison of (a) PMIR between mistreatment attack and GA-SVM and (b) PMR between MITM attack and GA-SVM.

```
root@68c239632c96: /oyente/oyente#                    Phython oyente.py -s LRAB-SC.sol
WARNING: root: You are using evm version 1.8.2. The supported version is 1.7.3
WARNING: root: You are using solc version 0.4.21, The latest supported version is
0.4.19
INFO:root:contract LRAB-SC.sol:LAB-SC:
INFO:symExec:        ============ Results ===========
 INFO:symExec:          EVM code coverage:                          8.0%
 INFO:symExec:          Integer Underflow:                          False
 INFO:symExec:          Integer Underflow:                          False
INFO:symExec:          Parity multisig Bug 2:                      False
INFO:symExec:          Callstack depth attack vulnerability:       False
INFO:symExec:          Transaction-ordering dependence (TOD):      False
INFO:symExec:          Timestamp Dependency:                       False
INFO:symExec:          Re-Entrancy vulnerability:                  False
INFO:symExec:        ====== Analysis completed ======
root@68c239632c96: /oyente/oyente#
```

Figure 11: Security analysis of smart contract using Oyente tool.

execution techniques. From Figure 11, it is shown that the outputs of all the analysis report are "false," which means that the smart contract is robust against the vulnerabilities. The following vulnerabilities and attacks are analyzed in the proposed model.

4.3.1. Integer Underflow and Overflow. The proposed smart contract is robust against this attack. This attack arises when the integer value is less than 1 bit or more than 256 bits. Therefore, the proposed smart contract gives an error and failed to deploy. Therefore, the proposed smart contract is robust against this attack.

4.3.2. Parity Multisig Bug 2. In this attack, the malicious node creates multiple accounts and generates fake signatures in the smart contract. When a malicious node performs

transactions using the fake accounts, then the smart contract will execute and deploy in the proposed system. Therefore, the proposed smart contract is robust against this attack as shown in Figure 11.

4.3.3. Call Stack Attack. In a smart contract, when a function is called by other functions, its depth is 1023 frames. In this attack, the malicious node exceeds the frame size to 1024 frames. Therefore, the function is failed to call, and the smart contract stopped working. However, the smart contract analysis shows that this attack is not possible in the proposed system model.

4.3.4. Timestamp Dependency. This attack is performed on miners. The miner who mines the block in minimum time is allowed to mine the next block. In the attack, the malicious node changes the mining time from the timestamp table. This attack is not possible on the proposed smart contract because PoA is used, and each miner has a copy of ledger. Therefore, if any malicious node changes the time, it can be traced and detected by other miners.

4.3.5. Reentrancy Vulnerability. In this attack, the malicious node calls the same function repeatedly and does not return any value. When a function is executed, then other functions cannot be executed. However, the proposed smart contract is robust against this attack as shown in Figure 11.

5. Conclusion

In this paper, ML techniques are exploited using blockchain for enhancing the security of the network. Firstly, registration is performed for the unauthenticated nodes, which can prove harmful for the network. Blockchain is used to register the nodes that are taking part in the routing process. Secondly, it stores the routing information that is generated during routing process. Moreover, a PoA consensus mechanism is used for validating the transactions because PoW requires more computational resources. Thirdly, ML techniques, GA-SVM and GA-DT, are exploited for MN detection. GA-SVM is used for the detection of MNs because of its higher accuracy as compared to GA-DT. MNs that are detected using GA-SVM are revoked from the network by deleting their registration from the blockchain, which allows the LNs to take part in the routing process. The model is trained using the following: PDR, PdR, PMR, and PMiR. Moreover, routing is performed by finding the shortest path using the Dijkstra algorithm. Once the route is calculated, source is able to securely and efficiently send the data packets to the destination. The proposed model is validated through extensive simulations. Moreover, security analysis using Oyente tool and three attacker models' induction is performed to evaluate the smart contract and the proposed system model, respectively. We compare the PoA and PoW consensus mechanisms using transaction cost in which PoA performs well and consumes up to 30% less cost. Furthermore, without MITM attack, GA-SVM consumes 10% less energy than with MITM attack. Moreover, without any attack, GA-SVM consumes 30% less energy than with grayhole attack and 60% less energy than mistreatment. The Dijkstra algorithm is compared with Bellman Ford in which the Dijkstra algorithm consumes less time to find the shortest path for routing. Dijkstra's and Bellman Ford's shortest paths are 8 and 11 hops, respectively. DT, SVM, GA-DT, and GA-SVM results are compared on the bases of accuracy and precision. The accuracy of DT, SVM, GA-DT, and GA-SVM is 88%, 93%, 96%, and 98%, respectively, while the precision of DT, SVM, GA-DT, and GA-SVM is 100%, 92%, 94%, and 96%, respectively.

Acronyms

CHs: Cluster heads
DT: Decision Tree
GA: Genetic Algorithm
GA-DT: Genetic Algorithm-Decision Tree
GA-SVM: Genetic Algorithm-Support Vector Machine
MITM: Man In The Middle
IoST: Internet of Sensor Things
PDR: Packet Delivery Ratio
PdR: Packet drop Ratio
PMiR: Packet Misroute Rate
PMR: Packet Modification Ratio
PoA: Proof of Authority
PoW: Proof of Work
RAs: Research answers
ROs: Research objectives
RQs: Research questions
SDN: Software defined network
SVM: Support Vector Machine
WSNs: Wireless sensor networks.

Data Availability

No supporting data exists separately.

Conflicts of Interest

The authors declare that there is no known conflict of interest.

References

[1] B. Jia, T. Zhou, W. Li, Z. Liu, and J. Zhang, "A blockchain-based location privacy protection incentive mechanism in crowd sensing networks," *Sensors*, vol. 18, no. 11, p. 3894, 2018.

[2] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.

[3] A. Shankar, P. Pandiaraja, K. Sumathi, T. Stephan, and P. Sharma, "Privacy preserving E-voting cloud system based on ID based encryption," *Peer-to-Peer Networking and Applications*, vol. 14, no. 4, pp. 2399–2409, 2021.

[4] S. Abbas, N. Javaid, A. Almogren, S. M. Gulfam, A. Ahmed, and A. Radwan, "Securing genetic algorithm enabled SDN routing for blockchain based Internet of Things," *IEEE Access*, vol. 9, pp. 139739–139754, 2021.

[5] A. P. Atmaja, A. El Hakim, A. P. A. Wibowo, and L. A. Pratama, "Communication systems of smart agriculture based on wireless sensor networks in IoT," *Journal of Robotics and Control (JRC)*, vol. 2, no. 4, pp. 297–301, 2021.

[6] A. S. Yahaya, N. Javaid, A. Almogren, A. Ahmed, S. M. Gulfam, and A. Radwan, "A two-stage privacy preservation and secure peer-to-peer energy trading model using blockchain and cloud-based aggregator," *IEEE Access*, vol. 9, pp. 143121–143137, 2021.

[7] J. Yang, S. He, Y. Xu, L. Chen, and J. Ren, "A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks," *Sensors*, vol. 19, no. 4, p. 970, 2019.

[8] Z. Cui, X. U. E. Fei, S. Zhang et al., "A hybrid blockchain-based identity authentication scheme for multi-WSN," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 241–251, 2020.

[9] G. Ramezan and C. Leung, *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 4029591, Wireless Communications and Mobile Computing, 2018.

[10] W. She, Q. Liu, Z. Tian, J. S. Chen, B. Wang, and W. Liu, "Blockchain trust model for malicious node detection in wireless sensor networks," *IEEE Access*, vol. 7, pp. 38947–38956, 2019.

[11] S. Hong, "P2P networking based Internet of Things (IoT) sensor node authentication by blockchain," *Peer-to-Peer Networking and Applications*, vol. 13, no. 2, pp. 579–589, 2020.

[12] K. Haseeb, N. Islam, A. Almogren, and I. U. Din, "Intrusion prevention framework for secure routing in WSN-based mobile Internet of Things," *IEEE Access*, vol. 7, pp. 185496–185505, 2019.

[13] Y. Liu, K. Wang, Y. Lin, and W. Xu, "LightChain: a lightweight blockchain system for industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3571–3581, 2019.

[14] M. H. Kumar, V. Mohanraj, Y. Suresh, J. Senthilkumar, and G. Nagalalli, "Trust aware localized routing and class based dynamic block chain encryption scheme for improved security in WSN," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 5, pp. 5287–5295, 2021.

[15] Y. Xu, J. Ren, G. Wang, C. Zhang, J. Yang, and Y. Zhang, "A blockchain-based nonrepudiation network computing service scheme for industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3632–3641, 2019.

[16] P. K. Sharma and J. H. Park, "Blockchain based hybrid network architecture for the smart city," *Future Generation Computer System s*, vol. 86, pp. 650–655, 2018.

[17] A. U. Khan, A. Shahid, F. Tariq et al., "Enhanced decentralized management of patient-driven interoperability based on blockchain," in *Lecture Notes in Networks and Systems*, pp. 815–827, Springer, Cham, 2019.

[18] M. U. Javed, N. Javaid, M. W. Malik et al., "Blockchain based secure, efficient and coordinated energy trading and data sharing between electric vehicles," in *Cluster Computing*, pp. 1–29, Springer, 2021.

[19] R. Khalid, M. W. Malik, T. A. Alghamdi, and N. Javaid, "A consortium blockchain based energy trading scheme for electric vehicles in smart cities," *Journal of Information Security and Applications*, vol. 63, article 102998, 2021.

[20] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing.*, p. 1, 2020.

[21] G. Rathee, M. Balasaraswathi, K. P. Chandran, S. D. Gupta, and C. S. Boopathi, "A secure IoT sensors communication in industry 4.0 using blockchain technology," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1–13, 2020.

[22] R. Fotohi and F. S. Aliee, "Securing communication between things using blockchain technology based on authentication and SHA-256 to improving scalability in large-scale IoT," *Computer Networks*, vol. 197, article 108331, 2021.

[23] R. Khalid, N. Javaid, A. Almogren, M. U. Javed, S. Javaid, and M. Zuair, "A blockchain-based load balancing in decentralized hybrid P2P energy trading market in smart grid," *IEEE Access*, vol. 8, pp. 47047–47062, 2020.

[24] M. Zhaofeng, M. Jialin, W. Jihui, and S. Zhiguang, "Blockchain-based decentralized authentication modeling scheme in edge and IoT environment," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2116–2123, 2021.

[25] C. M. Ferreira, C. T. Garrocho, R. A. Oliveira, J. S. Silva, and C. F. Cavalcanti, "IoT registration and authentication in smart city applications with blockchain," *Sensors*, vol. 21, no. 4, p. 1323, 2021.

[26] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasurbramanian, "A lightweight blockchain based framework for underwater iot," *Electronics*, vol. 8, no. 12, p. 1552, 2019.

[27] M. Mahdavisharif, S. Jamali, and R. Fotohi, "Big data-aware intrusion detection system in communication networks: a deep learning approach," *Journal of Grid Computing*, vol. 19, no. 4, pp. 1–28, 2021.

[28] H. Feng, W. Wang, B. Chen, and X. Zhang, "Evaluation on frozen shellfish quality by blockchain based multi-sensors monitoring and SVM algorithm during cold storage," *IEEE Access*, vol. 8, pp. 54361–54370, 2020.

[29] S. Rathore, B. W. Kwon, and J. H. Park, "BlockSecIoTNet: blockchain-based decentralized security architecture for IoT network," *Journal of Network and Computer Applications*, vol. 143, pp. 167–177, 2019.

[30] A. Mubarakali, "An efficient authentication scheme using blockchain technology for wireless sensor networks," in *Wireless Personal Communications*, pp. 1–15, Springer, 2021.

[31] N. Tariq, M. Asim, F. A. Khan, T. Baker, U. Khalid, and A. Derhab, "A blockchain-based multi-mobile code-driven trust mechanism for detecting internal attacks in Internet of Things," *Sensors*, vol. 21, no. 1, p. 23, 2021.

[32] H. Honar Pajooh, M. Rashid, F. Alam, and S. Demidenko, "Multi-layer blockchain-based security architecture for Internet of Things," *Sensors*, vol. 21, no. 3, p. 772, 2021.

[33] X. Wu and J. Liang, "A blockchain-based trust management method for Internet of Things," *Pervasive and Mobile Computing*, vol. 72, article 101330, 2021.

[34] J. Wang, Y. Liu, S. Niu, H. Song, W. Jing, and J. Yuan, "Blockchain enabled verification for cellular-connected unmanned aircraft system networking," *Future Generation Computer Systems*, vol. 123, pp. 233–244, 2021.

[35] T. H. Kim, R. Goyat, M. K. Rai et al., "A novel trust evaluation process for secure localization using a decentralized blockchain in wireless sensor networks," *IEEE Access*, vol. 7, pp. 184133–184144, 2019.

[36] R. Goyat, G. Kumar, M. K. Rai, R. Saha, R. Thomas, and T. H. Kim, "Blockchain powered secure range-free localization in

wireless sensor networks," *Arabian Journal for Science and Engineering*, vol. 45, no. 8, pp. 6139–6155, 2020.

[37] J. Wang, Y. Liu, S. Niu, and H. Song, "Lightweight blockchain assisted secure routing of swarm UAS networking," *Computer Communications*, vol. 165, pp. 131–140, 2021.

[38] A. Rahman, M. J. Islam, A. Montieri et al., "SmartBlock-SDN: an optimized blockchain-SDN framework for resource management in IoT," *IEEE Access*, vol. 9, pp. 28361–28376, 2021.

[39] W. Meng, W. Li, and J. Zhou, "Enhancing the security of blockchain-based software defined networking through trust-based traffic fusion and filtration," *Information Fusion*, vol. 70, pp. 60–71, 2021.

[40] C. M. Chen, X. Deng, W. Gan, J. Chen, and S. H. Islam, "A secure blockchain-based group key agreement protocol for IoT," *The Journal of Supercomputing*, vol. 77, no. 8, pp. 9046–9068, 2021.

[41] K. Sergii and F. Prieto-Castrillo, "A rolling blockchain for a dynamic WSNs in a smart city," 2018, http://arxiv.org/abs/1806.11399.

[42] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11008–11021, 2018.

[43] P. Danzi, A. E. Kalor, C. Stefanovic, and P. Popovski, "Delay and communication tradeoffs for blockchain systems with lightweight IoT clients," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2354–2365, 2019.

[44] G. Kolumban-Antal, V. Lasak, R. Bogdan, and B. Groza, "A secure and portable multi-sensor module for distributed air pollution monitoring," *Sensors*, vol. 20, no. 2, p. 403, 2020.

[45] M. Naz, F. A. Al-zahrani, R. Khalid et al., "A secure data sharing platform using blockchain and interplanetary file system," *Sustainability*, vol. 11, no. 24, p. 7054, 2019.

[46] M. Radhika and P. Sivakumar, "Energy optimized micro genetic algorithm based LEACH protocol for WSN," *Wireless Networks*, vol. 27, no. 1, pp. 27–40, 2021.

[47] L. Zhang, F. Li, P. Wang, R. Su, and Z. Chi, "A blockchain-assisted massive IoT data collection intelligent framework," *IEEE Internet of Things Journal.*, 2021.

[48] B. M. Yakubu, M. I. Khan, N. Javaid, and A. Khan, "Blockchain-based secure multi-resource trading model for smart marketplace," *Computing*, vol. 103, no. 3, pp. 379–400, 2021.

[49] P. Kautoo, P. K. Shukla, and S. Silakari, "Trust formulization in dynamic source routing protocol using SVM," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 6, no. 8, pp. 43–50, 2014.

[50] R. D. R. Popli, "A worm hole attack detection in mobile ad-hoc network using GA and SVM," *International Journal of Engineering Applied Sciences and Technology*, vol. 5, no. 3, pp. 582–588, 2020.

[51] A. Alsarhan, M. Alauthman, E. A. Alshdaifat, A. R. al-Ghuwairi, and A. al-Dubai, "Machine learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–10, 2021.

[52] S. Suthaharan, "Support Vector Machine," in *Machine Learning Models and Algorithms for Big Data Classification*, pp. 207–235, Springer, Boston, MA, 2016.

[53] D. Rachmawati and L. Gustin, "Analysis of Dijkstra's algorithm and A∗ algorithm in shortest path problem," in *In Journal of Physics: Conference Series*, vol. 1566, no. 1p. 012061, IOP Publishing, 2020.

[54] M. Bakshi and A. Srivastava, "Magnify lifeless nodes in WSN using shortest path ALGO for reducing energy diversion," *International Journal of Modern Communication Technologies and Research*, vol. 6, no. 6, 2018.

[55] F. Dad, N. Amin, S. T. Shah, F. Badshah, Z. U. Rahman, and ur Rahman, I, "Optimal path selection using Dijkstra's algorithm in cluster-based LEACH protocol," *Journal of Applied Environmental and Biological Sciences*, vol. 7, no. 2, pp. 194–198, 2018.

[56] M. Tripathi, M. S. Gaur, and V. Laxmi, "Comparing the impact of black hole and gray hole attack on LEACH in WSN," *Procedia Computer Science*, vol. 19, pp. 1101–1107, 2013.

[57] A. Chakrabarti and G. Manimaran, "Internet infrastructure security: a taxonomy," *IEEE Network*, vol. 16, no. 6, pp. 13–21, 2002.

[58] K. H. Yeung and W. K. Fung, "Attacking routers by packet misrouting," *WSEAS Transactions on Communications*, vol. 3, no. 2, pp. 493–498, 2004.

[59] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-Middle attack to the HTTPS protocol," *IEEE Security & Privacy*, vol. 7, no. 1, pp. 78–81, 2009.