Ad Hoc Networks 152 (2024) 103331

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

# An ensemble learning framework for the detection of RPL attacks in IoT networks based on the genetic feature selection approach

Musa Osman [a], Jingsha He [a], Nafei Zhu [a,*], Fawaz Mahiuob Mohammed Mokbal [a,b]

[a] Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China
[b] Faculty of Computer Science and Information Technology, Sana'a University, Sana'a, Yemen

ARTICLE INFO

*Keywords:*
Ensemble learning
Internet of things
LLN
RPL protocol
RPL attacks

ABSTRACT

The rapid proliferation of Internet of Things (IoT) devices has raised critical concerns regarding the security of corresponding IoT networks. The Routing Protocol for Low-Power and Lossy Networks (RPL), a foundational element in IoT communication, is susceptible to diverse routing attacks due to IoT nodes' constrained resources and open nature. This underscores the necessity for an Intrusion Detection System (IDS) to safeguard RPL-based IoT networks. Existing anomaly-based IDS suffer from high false alarm rates (FAR). In response to these challenges, this paper presents the Ensemble Learning-based Intrusion Detection System (ELG-IDS), which employs stacking and extreme parameter optimization to detect three RPL internal attacks: version number, decreased rank, and DIS flooding attacks. ELG-IDS employs enhanced feature extraction and genetic algorithm (GA)-based feature selection. Experimental results on a dedicated dataset demonstrate ELG-IDS's remarkable accuracy: 99.18%, 99.38%, 99.66%, and 97.90% for version number, rank attack, DIS flooding, and an average accuracy of 97.90% in multi-classification mode, respectively. This study advances IoT network security through ELG-IDS, enhancing protection against evolving security challenges.

## 1. Introduction

With the tremendous development of our digital world, the IoT has become a significant digital disruption for the future, especially with the emergent 5 G revolution. This technology is rapidly progressing from aspirational intentions to real-world applications. The IoT enables the digital world to be more efficient, saving time and money by allowing businesses, governments, and public authorities to re-think how they can deliver services and manufactured products [1].

IoT is a cutting-edge technology characterized as the network of everything in which objects can connect and communicate over the Internet without human intervention. Moreover, the aim of IoT is to enable people and intelligent devices to communicate with each other without any restrictions in time and place. Therefore, IoT implies that billions of physical devices worldwide are connected to the Internet, controlled with actuators and collecting and sharing data through sensors [2,3,4].

Due to the massive and significant amount of data collected by IoT devices, security has become a major concern. For example, sensors collect crucial data from smart homes (smart home devices), industrial units, smart cities, and other locations. Up to this date, IoT's security record has been abysmal. Due to the limited capabilities of IoT devices, stability and security features, such as data encryption in transit and at rest, have been neglected. Therefore, Vulnerabilities in IoT software are discovered regularly, and many IoT devices can hardly be patched, leaving them indefinitely vulnerable. As a result of their inherent insecurity, hackers aggressively target IoT devices by exploiting vulnerabilities in routing protocols. One of the core IoT protocols is RPL, i.e., IPv6 Routing Protocol for Low Power and Lossy Networks, which is used to connect IoT devices. However, the RPL protocol is not immune to malicious attacks. As a result, an increasing number of attacks have been generated from routers and webcams that are easy to exploit and roll up into massive botnets to hack smart home devices, smartwatches, and so on. Such attacks have made IoT security a paramount concern of the information security research community for gaining the trust of consumers, industrial units, and governments to this emerging technology [5,6,7].

The growing use of IoT devices has increased interest in IoT security research. Researchers focus on attacks against IoT networks, particularly those that utilize the RPL protocol. Despite its popularity in IoT applications due to its suitability for resource-limited devices, the RPL protocol remains vulnerable to various threats. As a result, researchers are

[*] Corresponding author.
*E-mail address:* znf@bjut.edu.cn (N. Zhu).

https://doi.org/10.1016/j.adhoc.2023.103331
Received 15 March 2023; Received in revised form 27 August 2023; Accepted 14 October 2023
Available online 20 October 2023

actively investigating potential risks within Low-Power and Lossy Networks (LLNs) that employ the RPL protocol [8,9]. Moreover, traditional machine learning algorithms have a low detection rate and are incapable of identifying small mutations of current malicious attacks, such as zero-day attacks. Although most of such attacks are minor variations of well-known harmful code (nearly 99% mutations), even the so-called new attacks (1%) rely on past notions and logic. In other words, malignant actions that diverge sufficiently from those seen before will fail to be classified and hence will overpower the undetected system. Ensemble learning techniques have had significant success in the large data arena, and they can also be used to resist cyber threats since attack mutations are similar to minor changes for two reasons. The first is that an ensemble can generate better forecasts and perform better than any single contributing model. The second reduces the spread or dispersion of predictions as well as model performance [10].

This study aims to secure the RPL protocol by equipping it with the capability of countering three types of malicious attacks, i.e., the version number attack (VN), the decreased rank attack (DR) and the DIS flooding attack (DIS). As a result, an intrusion detection system (IDS) named ELG-IDS is proposed based on an ensemble learning approach (stacking) and genetic feature selection method. The proposed ELG-IDS underwent extensive training and testing experiments using a large dataset. Experiment results show that ELG-IDS outperforms state-of-the-art techniques to achieve remarkable accuracy, precision, recall, F-score and true-(positive and negative) rate. Following are the significant contributions:

- A comprehensive and novel dataset named RPL-ELIDS that includes 783,176 unique records has been constructed to detect the three RPL attacks, i.e., VN, DR, DIS flooding attack, developed using the Cooja simulator under the Contiki operating system.
- A genetic algorithm for feature selection is proposed to extract the most optimal set of features from the dataset without compromising computational requirements.
- The proposed ELG-IDS machine learning framework coupled with extreme parameter optimization can detect RPL attacks with high accuracy and detection rate using the ensemble learning technique (stacking).
- The proposed ELG-IDS framework has been evaluated using a dataset to verify its efficiency compared to previous work.

The rest of the paper is organized as follows. Section 2 contains an overview of the RPL routing protocol. Section 3 reviews some related literature. Section 4 describes the proposed methodology. Section 5 summarizes the experimental results of the proposed work using different performance measures. Finally, Section 6 concludes the paper and suggests some future work.

Table 1 lists the main acronyms with explanations used throughout the paper.

## 2. Overview of the RPL routing protocol

The IoT devices are connected using the IPv6 Routing Protocol for Low Power and Lossy Network (RPL) protocol. RPL protocol is a Distance Vector protocol (DV) based on Directed Acyclic Graphs' topological concept (DAGs) [11,12]. Additionally, it structures the nodes using Destination orientated Directed Acyclic Graphs (DODAGs). An RPL network may consist of several DODAG(s), each with a distinct DODAG ID, forming one instance with a unique instance ID. The RPL protocol's main features include auto-configuration, self-healing, loop-avoidance and detection, independence and transparency, and multiple edge routers. Moreover, RPL supports Point to Point (P2P), Point to Multipoint (P2MP) and Multipoint to Multipoint point (MP2P) communication techniques [11,13,14].

RPL creates and constructs the DODAG by using four messages, including DODAG Information Object (DIO), Destination Advertisement Object (DAO), DODAG Information Solicitation (DIS), and Destination Advertisement Object acknowledgement (DAO-ACK). These messages are used to build and maintain a DODAG. The first message contains a DIO (DODAG Information Object), a DODAG information notification sent downstream through the DODAG route, containing related information of the DODAG where the node is located, such as RPL Instance ID, DODAGID, and Rank Value [15,14,16]. DIO message is multicast by the root to create and maintain a new DODAG and create an upward route. The second message is a DAO (DODAG Advertisement Object), a destination information notification that follows the DODAG path. The node sends two DAO messages to two distinct destinations: one in storing mode to the terminal node's parent node and another in non-storing mode to the root node of the DODAG in which the terminal node is situated [17]. The parent or aggregation node establishes a downward route to the node based on the receiving routing declaration. DAO message is used to build the downward route, the third message is a DAO-ACK unicasted by a DAO recipient, and finally, DIS is used to encourage other nodes to send a DIO message [7,18]. The flow and direction of DIO and DAO messages in the RPL network are depicted in Fig. 1.

**Table 1**
Acronyms and explanations.

| Acronym | Explanation |
|---|---|
| ELG-IDS | Ensemble Learning Framework for the Detection of RPL Attacks in IoT Networks Based on the Genetic Feature Selection |
| RPL | IPv6 Routing Protocol for Low Power and Lossy Network |
| GA | Genetic algorithm |
| VN | Version number attack |
| DR | Decreased rank attack |
| DV | Distance Vector protocol |
| DAGs | Directed Acyclic Graphs' |
| DODAG | Destination orientated Directed Acyclic Graphs |
| DIO | DODAG Information Object |
| DAO | Destination Advertisement Object |
| DIS | DODAG Information Solicitation |
| DAO-ACK | Destination Advertisement Object acknowledgement |
| ANN | Artificial neural network |
| MLP | Multi-layer perceptron neural network |
| SVM | Support vector machine |
| PCA | Principal component analysis |
| PDR | Packet delivery ratio |
| DNN | Deep Neural Network |

However, the RPL protocol is not immune to different attacks. RPL attacks can be classified into two groups, i.e., insider attacks and outsider attacks. Between the two, insider attacks are the most severe and the consequence on RPL networks is devastating. The most critical insider attacks are VN, DR and DIS flooding attacks [19].

RPL offers global repair and local repair techniques that work in tandem. When a node detects a network inconsistency (e.g., a failed link between two nodes or detecting a local loop), it initiates a local repair operation. It entails quickly identifying a backup route without trying to fix the DODAG entirely. This alternative healing approach may not be the best option if local repairs are inadequate at rehabilitating the network due to various irregularities. In that case, the DODAG root may conduct a global repair operation, after which it increments the DODAG Version Number and launches a new DODAG Version [16,20]. The global repair operation reorganizes the network topology significantly. Nodes in the new DODAG Version can select a new rank that is not tied or reliant on their previous rank in the old DODAG Version [21,22].

VN attacks exploit the global repair mechanism by manipulating the version number feature in the DIO message, which creates loops and inconsistency in the DODAG [20,21]. DR attacks are used by a malicious node(s) to advertise a fake rank to convince the benign node(s) to select it as the preferred parent [22,17]. The malicious node(s) send many DIS messages in the DIS attacks, consuming the network resources (power, memory, and processing capabilities) [23]. Fig. 2 shows RPL attacks,

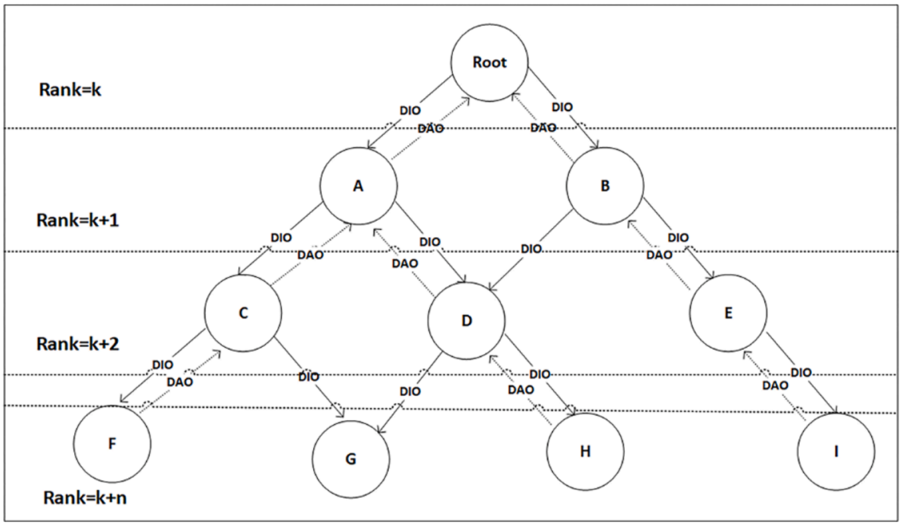
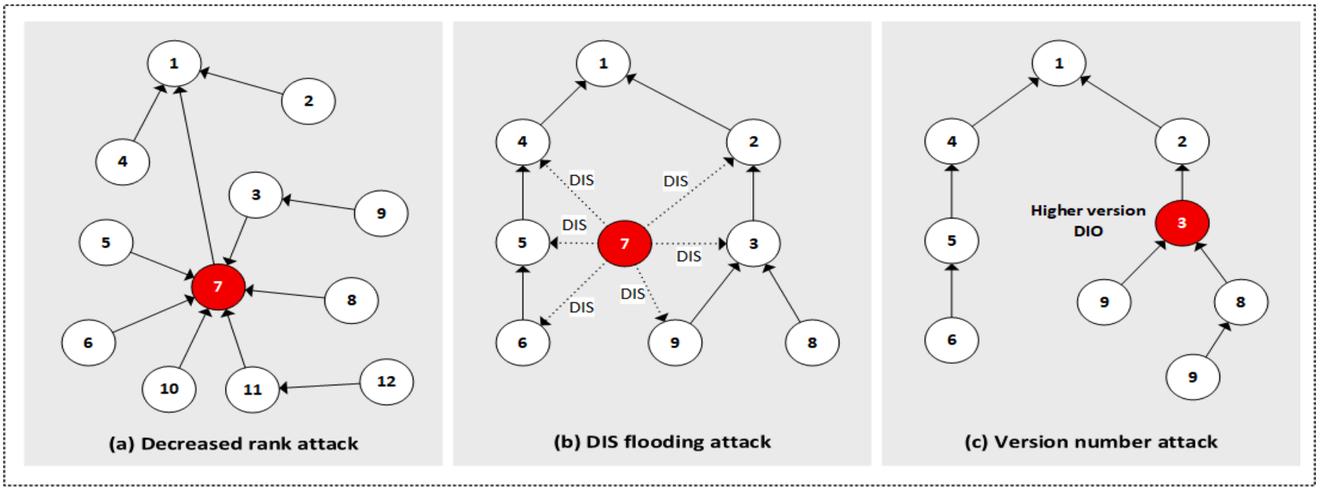


**Fig. 1.** The flow and direction of DIO and DAO messages in the RPL network.



(a) Decreased rank attack

(b) DIS flooding attack

(c) Version number attack

**Fig. 2.** RPL routing attacks.

where is (a) shows Decreased rank attacks, (b) shows DIS flooding attacks and (c) shows Version number attacks.

## 3. Related work

Research in the security and reliability of the RPL protocol is still in its infancy and trim work has been done in this area. Malicious nodes in an IoT network can introduce many vulnerabilities, including packet tampering, malicious corruption, denial of services (DoS), IP address spoofing and eavesdropping. In the following, we briefly review some recent work on the security and reliability of the RPL protocol.

Osman et al. [11] proposed an artificial neural network (ANN) model to detect decreased rank attacks using the IRAD dataset for training and testing. The method can achieve accuracy, precision, false-positive rate and AUC-ROC scores of 97.14%, 97.03%, 0.36% and 98%, respectively. The main limitation of this study is that the ANN is not suitable for IoT devices. Choukri et al. [12] proposed an intrusion detection system based on a multi-layer perceptron (MLP) neural network for detecting RPL rank attacks using a dataset generated by the simulation running in the Cooja simulator for training and testing. The IDS generates accuracy, F1-score and recall values up to 94.57%, 98% and 100%, respectively. Momand et al. [13] presented a machine learning-based approach for detecting three types of RPL attacks, namely, version number, rank and DoS attacks. The author used a support vector machine (SVM) classifier to detect the attack and principal component analysis (PCA) for feature selection, which achieved 76.8% on the packet delivery ratio (PDR).

Diro et al. [14] proposed a distributed attack detection architecture based on a deep learning technique to detect IoT attacks. The authors used the KDDCUP99 dataset to train and validate the model, which achieved accuracy, precision, recall and F1-score value of 99.20%, 99.02%, 99.27% and 99.14%, respectively. The authors claimed that distributed detection was better than centralized detection in dealing with cyber-attacks. Mayzaud et al. [16] proposed a distributed monitoring architecture to detect version number attacks, consisting of three algorithms: monitoring architecture, detection algorithm and monitoring node placement. Monitoring architecture is used to monitor the nodes and send the incremented VN node ID to the root, and this algorithm is applied to all nodes except the root. The detection algorithm and monitoring node are installed in the root node, detecting the attack and collecting all the information in a table. In contrast, the second algorithm is used to identify attackers. The proposed method has a high risk of false positives. Raza et al. [24] presented a specification-based hybrid detection and placement IDS. The IDS consist of three modules. The first module is 6LoWPAN Mapper (6Mapper), responsible for constructing a full DODAG. The second module is the intrusion detection module accountable for detecting the attacks based on signature-based and anomaly-based methods. The last module is a distributed firewall and response module; this module is deployed and installed on every node to prevent the network from external attacks. Another mechanism for detecting version number attacks is proposed by Sahay et al. [25]. The proposed mechanism is a centralized machine learning technique that uses a decision tree, support vector machine, Bernoulli RBM, and LR to evaluate the network's data. The results obtained from the model are 98%, 100%, and 95% for accuracy, precision, and recall, respectively. Additionally, the authors investigated the version number attacks and their impact on the RPL network. The main limitations of the proposed method are unsuitable for a bigdata and the mobility was not addressed. Shafique et al. [18] proposed a method for detecting rank attacks using a root-based statistical intrusion detection system for detecting rank attacks by comparing the ranks of the nodes in the usual conditions. The proposed IDS achieved a high accuracy score for small nodes without mobility. The model's performance deteriorated as the number of nodes increased. The main limitations of the proposed method are the lack of lightweight solutions and do not provide confidentiality and network performance evaluation. Yavuz et al. [23] proposed a deep learning model for detecting routing attacks in IoT using an artificial neural network consisting of seven layers. Moreover, the authors generated a dataset using the Cooja simulator for model training and testing, including DIS flooding attacks, decreased rank attacks, and version number attacks, obtaining the accuracies of 95%, 98%, 94% for decreased rank, DIS flooding, and VN attacks, respectively. Bokka et al. [26] proposed a deep neural network (DNN) based on deep learning to detect anomalies in IoT-based smart home environments. They used a dataset from Kaggle named (DS2OS) for training and testing the model. The model achieves 99.42%, 99%, 99% and 99%, accuracy, precision, recall and F-Score, respectively.

A voting ensemble classifier is a machine learning algorithm that combines the predictions of multiple models to generate a final prediction. In [27], the authors proposed an intrusion detection approach based on an ensemble-based voting classifier for detecting IoT attacks. The IDS combined decision tree, naive bayes, random forest, and K-Nearest neighbors using a voting-based technique. The proposed IDS used available datasets in literature for training and validation. The result outperforms the traditional machine learning methods. In [28] the authors presented an SVM-based lightweight anomaly detection model. For feature selection method the proposed model used a wrapper FS technique based on the combination of GA and grey wolf optimizer (GWO). The proposed features selection method is compared with the pure GA and GWO and the result obtained proved the hybrid algorithm outperform the pure methods. Table 2 provides a summary of the related work.

**Table 2**
The detail summary of related work.

| Ref. | Dataset | Methodology | Results | Limitations |
|---|---|---|---|---|
| [11] | IRAD dataset | ANN method for detecting decreased rank attacks | Precision is 97.03% Accuracy 97.14 AUC-ROC is 98% | ANN is not suitable for IoT |
| [12] | The authors' dataset | MLP neural network for detecting RPL rank attacks | Accuracy is 94.57% F1-score is 98% Recall is 100% | MLP is not adequate to IoT devices |
| [14] | KDDCUP99 dataset | Deep learning technique for detecting IoT attacks | Accuracy is 99.20% Precision is 99.02% Recall is 99.27% F1-score is 99.14% | Deep learning techniques are resource-intensive |
| [25] | The authors' dataset | A machine learning technique for IoT attacks detection | Accuracy is 98% precision is 100% Recall is 95% | Dataset is not an IoT dataset |
| [18] | – | A root-based statistical intrusion detection system for detecting rank attacks | The IDS achieved a high accuracy score for small nodes | Performance of the model deteriorates as the number of nodes increases |
| [23] | The authors' dataset | A deep learning model for detecting routing attacks | decreased rank is 95% DIS flooding is 98% VN attacks is 94% | Deep learning models is resource-intensive |
| [26] | DS2OS Kaggle dataset | A deep neural network for detecting IoT-based smart home environments attacks | Accuracy is 99.42% Precision is 99% Recall is 99% F-Score is 99% | Deep neural network is not suitable for tiny devices |

## 4. Detection methodology

In this section, the entire framework is described. The IoT data for different scenarios were collected by transmitted radio data as a PCAP data file using an integrated radio message of Cooja simulator. feature extraction including, cleaning and converted data to a JSON format using Wireshark tool, and parser algorithm we developed to extract the features from the JSON files. In additional to feature selection approach using Genetic algorithm. Finally, stacking-based meta-learning method using stacking technique for built ML model including various algorithms is presented.

### 4.1. The framework

Since the IoT technology has an unbeatable impact on our personal life, environment, and businesses. Therefore, IoT attacks and security threats are increasing alarmingly. RPL attacks are one of these attacks, often leading to severe results. To mitigate the impact of these attacks, we proposed a detection architecture that integrates three modules: data collection, feature extraction, and feature selection, as well as an ensemble learning model (Stacking) for data analysing [29]. Fig. 3 depicts the ELG-IDS framework.

### 4.2. Data collection

To avoid the risk of a lack of a qualitative and adequate training and testing dataset used in developing ELG-IDS detector, we built RPL attacks dataset corresponding to various IoT scenarios . We used the Contiki operating system to emulate the IoT network as a base for the Cooja simulator to collect the raw dataset. Cooja enables the execution of an actual RPL code on the simulated nodes. Moreover, we used the Contiki instance 3.0 with a virtual machine platform with 16 GB of RAM. Each node in our simulation is created as Zolertia mote for benign and malicious purposes, with 8 KB RAM and a 92 KB Flash memory. Cooja enables the collection of transmitted radio data as a PCAP data file using an integrated radio message tool. Moreover, it available some pre-configured nodes in the Cooja simulator. We used the rpl-udp as the base for all the scenarios from these pre-configured nodes. For benign traffic flow, a normal 6LoWPAN network is configured without any attacker node and the, instances output were created in the dataset to represent the benign traffic flow. For malicious instance, the similar setups are used for the three separate attacks described below. In each of these cases, an attacker node is added. Each scenario is run for 10 s.

The variety of network topologies with different scenarios were built. In all the attack scenarios, we employed one sink node in the network with (10, 20) client nodes and (1, 2, 3) malicious nodes. Firstly, for the DIS attack, we decreased the initial interval between successive packet

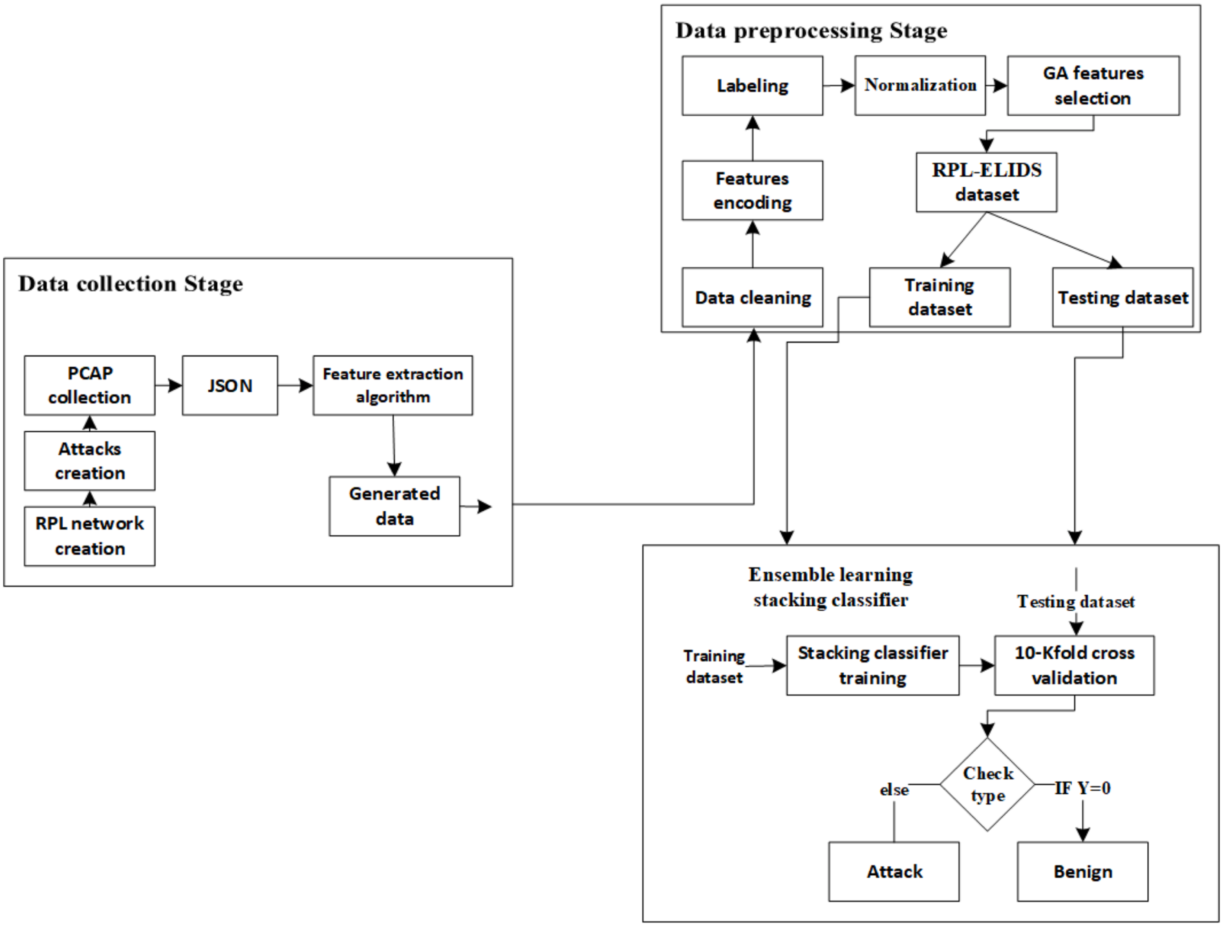


**Fig. 3.** The schematic framework of the proposed technique.

generations in order to increase the number of messages transmitted by the attacking node. Secondly in the decreased rank attacks the malicious node is modified to advertising a low rank to disrupt routing paths. Lastly in the VNA the victim node. When it receives a DIO message, it increases the DAG version and then sends a new (poisoned) DIO message to its neighbors.

At each design, we used the UDP-server and UDP-client node in our network from the rpl-udp example mentioned earlier, and then we modified the Cooja core code to lunch the attacks (for the malicious node). Subsequently, we collected all the PCAP captured data and, using Wireshark, cleaned it to extract solely RPL messages (the captured data contains some data from other protocols) and converted it to a JSON (JavaScript Object Notation) dictionary format.

We developed a Python parser to extract features from the JSON files as shown in Algorithm 1 with the extracted features being saved in CSV files. A total of 113 features are extracted from different layers including the frame layer and the IPv6 layer.

Thus, data preprocessing for the collected data for each attack scenario is implemented. To reduce the feature's dimensionality, two phases were implemented. The first phase removes redundant features such as source IP address and destination IP address and fixed features having constant values, such as WPAN security. As a result, a total of 59 features are remained. In the second stage of the process, the hexadecimal characteristics are eliminated. The source and destination addresses are then transformed from IPv6 configuration to node ID. For instance, the address "f e80: c30c: 0: 0: 03″ is transformed into the simplified representation "3," denoting the third node. To prevent potential overlap with other nodes, the broadcast packets with the address "ff02::1a" are changed to "999." This alteration signifies that the source node is transmitting broadcast packets. As a result of second phase, a total of 17 features are remained. Subsequently, one more column is added as a data label containing two values (0 for benign and 1 for malicious). The dataset's description is summarized in Table 3 and Fig. 4.

**Algorithm 1**
Parsing JSON code.

```
Input: file contains JSON codes
  Output: CSV file contains the features vector
Require: Initialize
  processed_data ← []
  header ← []
  reduced_item ← {}
  data_to_be_processed ← raw_data
  //Reading arguments
  raw_data ← json.loads(json_value)
  for item in data_to_be_ processed
    reduced_item ← {}
    header += reduced_item.keys()
    reduce_item(node, item)
    processed_data.append(reduced_item)
  End for
  //Writing arguments
  open csv file
  header ← list(set(header))
  writer ← csv.DictWriter (header)
  for row in processed_data:
    writer.writerow(row)
  End for
  Return (CSV file contains a total of 113 features extracted from different layers)
  End
```

**Table 3**
The dataset sub-division detail for RPL different attacks.

| Type of attacks | Benign | Malicious | Total number |
|---|---|---|---|
| VN attack | 223,243 | 89,296 | 312,539 |
| DR attack | 208,261 | 22,408 | 230,669 |
| DIS attack | 178,138 | 61,830 | 239,968 |
| Multiclass | 609,642 | 173,534 | 783,176 |

The training and validation extracted features comprise eleven features. The source and destination IP addresses for the 6Lowpan protocol are represented by the 6lowpan.src and 6lowpan.dst features, respectively. The length of the frame sent by the 6Lowpan protocol is indicated by the frame.len feature. The icmpv6.code feature identifies the type of RPL control message. The dio.dtsn feature indicates the flag used to maintain downward routes, while the DODAG Rank of the node sending the DIO message is represented by a dio.rank feature. The DODAG root sets the dio.version unsigned integer to the DODAG version number. The icmpv6.type represents the type of RPL control message. The ipv6.hlim feature indicates the maximum number of links over which the IPv6 packet can travel before being discarded. The wpan.ack_request feature indicates the successful reception of the corresponding data frame. Finally, the path sequence number that may be incremented occasionally to cause a refresh to the Downward routes is represented by a wpan.seq_no feature [30].

### 4.3. Feature normalization

Normalization is the process of rescaling the data from its original range to a new range between 0 and 1, which is beneficial when the data has input values with varying scales. Min-Max normalization is one method for transforming the original data range linearly. The primary purpose of normalization is to aid in prediction and forecasting purposes. It depends on the min and max value of the data to create a new range. Eq. (1) illustrates the Min-Max transformation [31].

$$X^{'} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

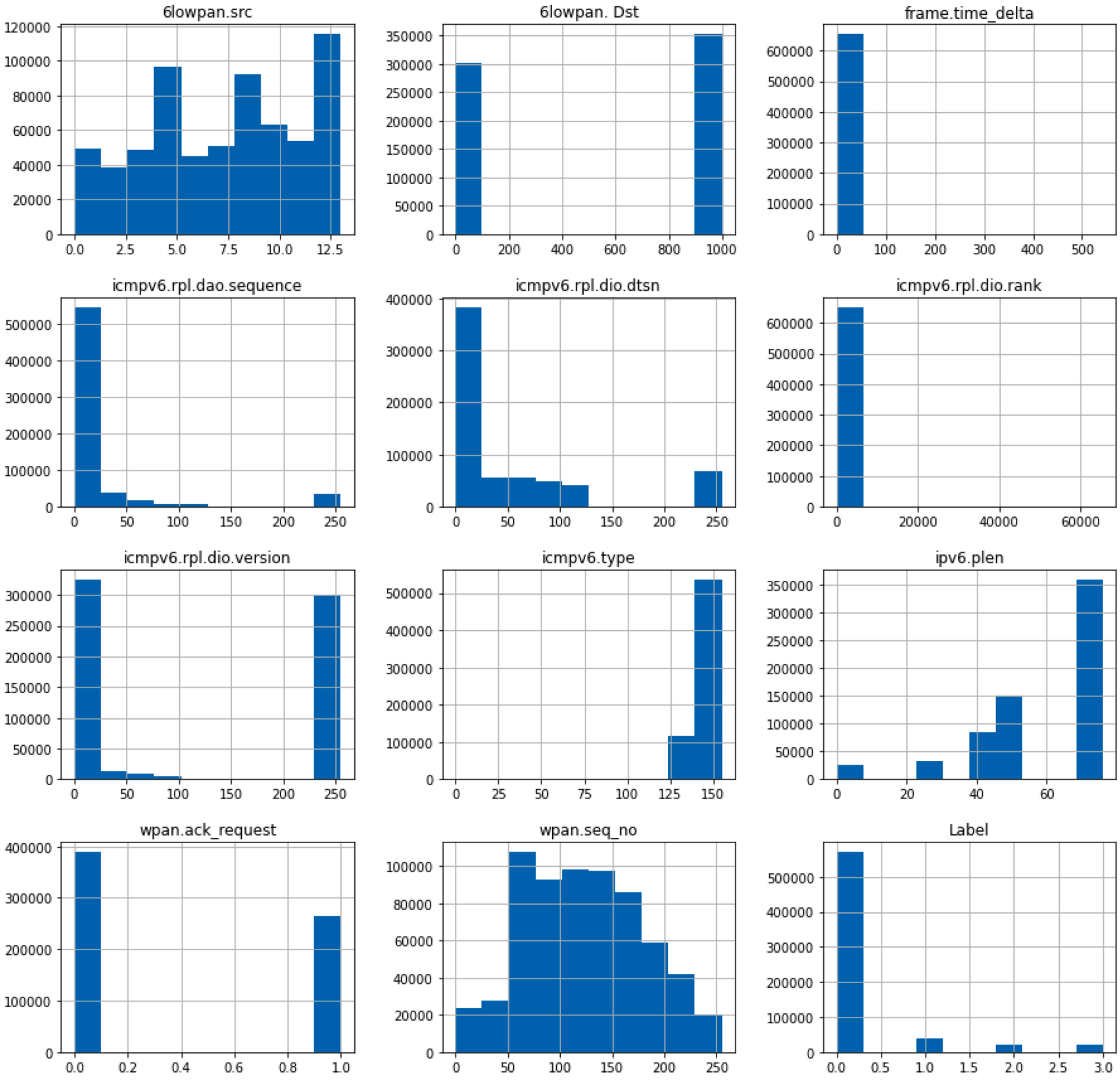


**Fig. 4.** Details of the extracted features for training and validating the proposed model.

### 4.4. Features selection

A dataset with high dimensionality significantly harms the memory storage and the computation costs of data analysis. Features selection or dimensionality reduction is an effective technique for addressing the concern highlighted in the literature. Furthermore, the main objectives of feature selection techniques are to enhance the performance of machine learning models, provide a clean dataset, and construct a more straightforward and comprehensive model [32,33].

Table 4 lists the selected features for the training and validation of the proposed model. A genetic algorithm is a particular case of an evolutionary algorithm that mimics natural evolution based on genetics and natural selection principles. It uses binary string encoding, where each bit indicates the presence or absence of a feature (a “1″ indicates the presence of a feature, while a “0″ signifies its absence). Then, these solutions undergo recombination and mutation. The central idea of Genetic Algorithms (GAs) originates from initiating an initial group of potential solutions termed a “population.” Each candidate solution is assigned a fitness value based on the fitness function, serving as a metric to gauge the effectiveness of solutions. The chosen fitness function in this context is the Mean Squared Error (MSE), formulated as Eq (2). In each iterative step, fitness evaluation for every entity within the population transpires. Consequently, the most proficient individuals are selected probabilistically from the existing population. Notably, the parameters governing crossover and mutation are defined at values of 0.5 and 0.05, respectively. The crossover parameter influences the likelihood of recombination between two chromosomes, creating a new chromosome. Conversely, the mutation parameter dictates the probability of genetic modification within a chromosome [34]. The GA technique is as follows. A population (Y) of n chromosomes is randomly initiated. Each Y chromosome's fitness is calculated. Two chromosomes, C1 and C2, are chosen from the Y population based on their fitness. C1 and C2 are subjected to the single-point crossover operator with crossover probability (Cp) to create the offspring O. The created offspring (O) is then subjected to a uniform mutation operator with a mutation probability (Mp) to generate O′. The new offspring O’ is introduced into a new population. The existing population will undergo selection, crossover, and mutation until the new population is complete [33].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i) \tag{2}$$

Where $n$ represents the total number of data points or samples, $y_i$denotes the actual observed value of data point $i$, and $\widehat{y}_i$ represents the predicted or estimated value of data point $i$ as generated by a model or method.

Genetic Algorithm (GA) can be used for feature selection in both offline and online settings. Each approach has its advantages and disadvantages. In an offline setting, where all of the data is available beforehand (as in our case), GA has the advantage of being able to search the entire solution space and converge on a good solution. This can result in high-quality feature subsets that perform well on the given task. In an online setting, where data is received incrementally, GA has the advantage of being able to adapt to changing data and update its solutions on the fly. This can result in feature subsets that can track changes in the data and remain relevant over time. However, one disadvantage of using GA in an online setting is that it can be more challenging to find good solutions since the algorithm must adapt to changing data and may not have access to all of the information at once, especially with IoT data, which are generated in huge quantities and continuously [35,36].

The main advantages of GA are parallelism, global optimization, a more extensive solution space, less information needed, provides multiple optimal solutions, and probabilistic. The exact number of the selected features to the total number of features is plotted in Fig. 5 based on generation. Fig. 6 shows the fitness scores based on the best score and best average (MSE before the optimization is 0.1814 and MSE after optimization is 0.1151).

**Table 4**
The most suitable subset of features identified by the genetic algorithm (GA) for training and validating the proposed model.

| NO | Selected feature | Feature name |
|---|---|---|
| 1 | 6lowpan.src | Source IP Address |
| 2 | 6lowpan. Dst | Destination IP Adress |
| 3 | frame.len | Frame Length |
| 4 | icmpv6.code | ICMPv6 Code |
| 5 | dio.dtsn | Destination Advertisement Triggered Sequence Number. |
| 6 | dio.rank | DIO Rank |
| 7 | dio.version | DIO Version |
| 8 | icmpv6.type | ICMPv6 Massege Type |
| 9 | ipv6.hlim | IPv6 Hope Limit |
| 10 | wpan. ack_request | Acknowladge Request |
| 11 | wpan.seq_no | Sequence Number |

### 4.5. The ensemble machine learning model

Our proposed model is a stacking-based meta-learning method that aims to integrate the predictions of different algorithms to build a new robust model. Instead of using the original dataset, we create a metadata set that contains the same features provided by base model outputs. In other words, it uses the base model outputs as the input features rather than the original dataset [10,29].

The proposed model contains five learning models as a base model, while the meta-model is Logistic Regression. The base learning model includes KNeighbors, decision trees, random forests, gradient boosting, and linear discriminant classifiers. These five algorithms were selected for the ensemble model because they bring distinct strengths and capabilities. K-Nearest Neighbors (KNN) is a good choice for problems where the data is not normally distributed and is relatively easy to interpret. Decision Trees work well when the data doesn't have a straight-line separation and is also straightforward to interpret. Random forest is often more accurate than individual decision trees and more robust to overfitting. Gradient boosting is very effective at learning complex relationships in the data. LDA is a relatively simple algorithm, but it can sometimes be very effective. Combining these five algorithms in an ensemble model can help improve the model's accuracy and robustness. The different algorithms can learn various aspects of the data, and the ensemble model can then combine these different predictions to make a more accurate final prediction [37]. The outputs of these models are combined by the meta-model (Logistic regression), which is trained on the combined output. As a result, a robust model upon different algorithms was built. Eq (3) shows the combination of the base learners can be expressed as an ensemble probability estimate.

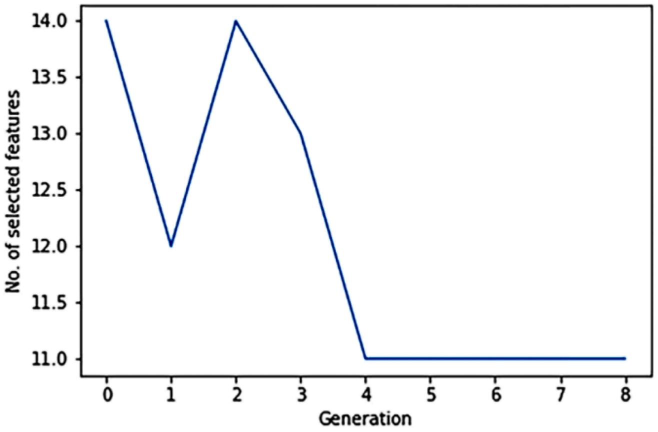


**Fig. 5.** The number of selected features vs. the rounds of generations.

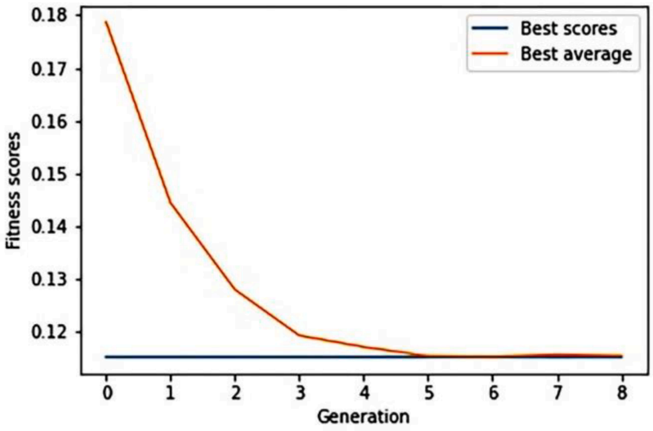


**Fig. 6.** The fitness scores.

$$p(y|x) = \sum_{i=1}^{N} w_i \; p_i(y|x) \tag{3}$$

where $p(y|x)$ is the likelihood for N base learners, a $w_i$ is a weight given to each learner N based on the training data samples, and $y$ is the class label to be estimated.

## 5. Experiment and result analysis

In this section, we present the proposed model's and other models' results on several datasets, including the Decreased Rank (DR) attack dataset, DIS attack dataset and the Version Number (VN) attack dataset, and compare the results in binary classification and multiple classifications. We also perform the statistical test with our proposed model and several other models on the same dataset to verify the results.

### 5.1. Performance evaluation metrics

This study chooses distinct evaluation metrics to evaluate the model performance, such as accuracy, precision, F-score, false-positive rate, and false-negative rate. These metrics are based on the confusion matrix parameters. True negative (TN) refers to the benign cases detected correctly as benign, false-positive (FP) refers to the benign cases mistakenly labelled as malignant, false-negative (FN) refers to the malignant cases wrongly labelled as benign, and true positive (TP) refers to the malignant cases that are detected correctly [38,39]. Eqs. (4)-9 define the metrics.

$$Accuracy = \frac{(TP \; + \; TN)}{(TP \; + \; TN \; + \; FP \; + \; FN)} \tag{4}$$

$$\text{Precision} = \frac{TP}{(TP \; + \; FP)} \tag{5}$$

$$\text{FN rate (FNR)} = \frac{FN}{(FN \; + \; FP)} \tag{6}$$

$$\text{Detection rate (DR) or Recall} = \frac{TP}{(TP \; + \; FN)} \tag{7}$$

$$\text{FP rate (FPR)} = \frac{FP}{(TN \; + \; FP)} \tag{8}$$

$$F1 - score = 2 \times \left( \frac{(Recall \; \times Precision)}{(Recall \; + Precision)} \right) \tag{9}$$

### 5.2. Experiment results

For training and forecasting purposes, the ensemble learning stacking method is used. This is primarily due to stacking's efficiency in distinct dataset types, including balance and imbalance [10]. We conduct numerous experiments in a separate dataset with different parameters in binary classification. Furthermore, multi-classification experiments are conducted where all the datasets are combined in one dataset. All performance evaluation was conducted on a machine running 64-bit Windows 10 and equipped with an Intel(R) Core (TM) i7–8550 U CPU running at 2.00 GHz and 16 GB of main memory. Python 3.7.9 is used in developing and evaluating classifiers.

#### 5.2.1. Results of binary classification

The suggested detection framework is reliable and efficient, as evidenced by the notable perfection and higher performance metrics scores of several tests, including both classes (attack and benign). The F-score demonstrated it achieved a high recall-high precision ratio concurrently. After fine-tuning the model with selected parameters, the final configuration was optimized to yield the best results. We obtained 99%, 100%, and 99% accuracy scores for VNA, DRA, and DIS, respectively. The classification report results with various datasets are summarized in Table 5. Furthermore, Table 6 shows the overall accuracy of other methods compared to the ensemble learning classifier (stacking). These findings reveal that ensemble learning outperforms state-of-the-art methods.

Figs. 7-9 show the results on the VNA dataset. The stacking performance (accuracy) in the training and the testing datasets is plotted in Fig. 7. The receiver operating characteristic (ROC) is shown in Fig. 8 with the area under the curve (AUC) being 98.9%. Fig. 9 depicts the confusion matrix on the VNA dataset. As we can see, the FPR is 0.01456 while the FNR is 0.014264.

Figs. 10-12 illustrate the findings obtained from the DRA dataset. Fig. 10 shows the overall accuracy from the training and testing DRA dataset, Fig. 11 depicts the ROC curve and Fig. 12 depicts the confusion matrix. As can be seen, the area under the curve is 98.3% of the ROC curve. In addition, the FPR is 0.033673 while the FNR is 0.02997.

Figs. 13-15 show the results obtained on the DIS flooding attacks dataset in which the area under the curve is 99.7%, and the FNR value is 0.005513 while the FPR is 0.000038.

#### 5.2.2. Results of multi-classification

Moreover, to evaluate the proposed scheme, all the datasets are combined into one dataset for performing multi-classification. Also, we used the same model with the same parameters. Table 7 contains the classification report obtained from the multi-classification experiments. The experimental results show that the ensemble-based model outperforms the other models in terms of accuracy, false-positive rate, false-negative rate, and area under the curve for detecting three RPL attacks.

### 5.3. Comparison between feature selection and GA feature selection

For further clarification, several feature selection methods are compared with the GA method. The ELG-IDS is used as a classifier. The

**Table 5**
Stacking results obtained from different datasets.

| Classifier | Non-attack | | | Attack | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Stacking (VNA) | 99% | 99% | 99% | 99% | 99% | 99% |
| Stacking (DRA) | 100% | 100% | 100% | 97% | 97% | 97% |
| Stacking (DIS) | 99% | 100% | 100% | 100% | 99% | 100% |

**Table 6**
Accuracy results of different classifiers.

| Classifier | VNA | DRA | DIS | Multi classification |
|---|---|---|---|---|
| K-Nearest Neighbors | 99.13% | 99.26% | 99.58% | 97.72% |
| Naive Bayes | 72.75% | 91.47% | 99.42% | 61.00% |
| Decision Tree | 95.58% | 99.15% | 99.65% | 97.22% |
| Random Forest | 97.18% | 98.68% | 99.50% | 97.24% |
| Adaboost | 80.05% | 98.24% | 99.49% | 47.73% |
| Gradient Boosting | 96.67% | 92.73% | 99.32% | 92.95% |
| LDA | 73.90% | 92.01% | 99.42% | 89.52% |
| Logistic Regression | 74.06% | 95.91% | 99.42% | 92.50% |
| **Stacking** | **99.18%** | **99.38%** | **99.66%** | **97.90%** |

result obtained from GA outperforms other feature selection methods, as shown in Table 8. The GA performs better in terms of feature selection than other methods for several reasons. One of the reasons is that it can search a large and complex solution space, making it able to search the solution space more efficiently than other methods. GA uses a population-based approach, where multiple potential solutions are evaluated simultaneously. This allows GA to explore multiple regions of the solution space at once, increasing the chances of finding a good solution, especially in an offline setting, where all of the data is available beforehand [36,33].

It should be noted that, in forward feature selection, the number of selected features is the same as GA but different in the feature's properties.

### 5.4. Statistical test and comparison

We also statistically tested the proposed model by using the null hypothesis test to determine whether there was a significant difference between the output of our proposed model and that of other machine learning approaches with the same dataset. A hypothesis test is a statistical technique that compares two mutually exclusive statements about a population to determine which statement is statistically most consistent with the sample data [39]. Our experiment's null and alternate hypothesis statements are as follows:

H0: Both models perform identically on the same dataset.
H1: Both models perform differently on the same dataset.

We assume that the error rate to be allowed is 5%, i.e., alpha (α) =

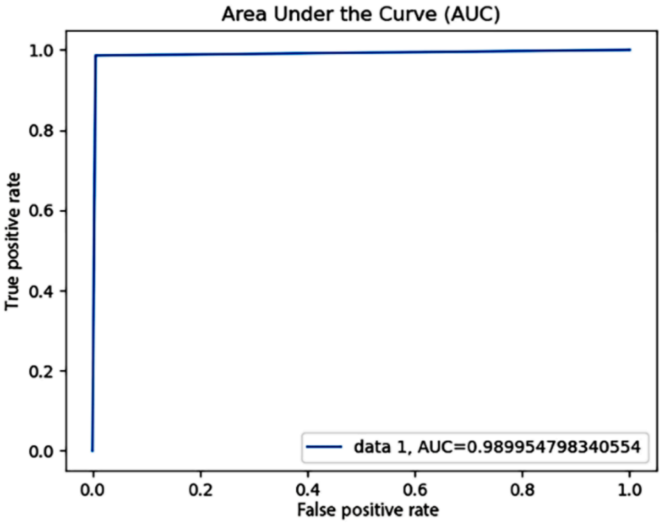


**Fig. 8.** Stacking model ROC curve (VNA).

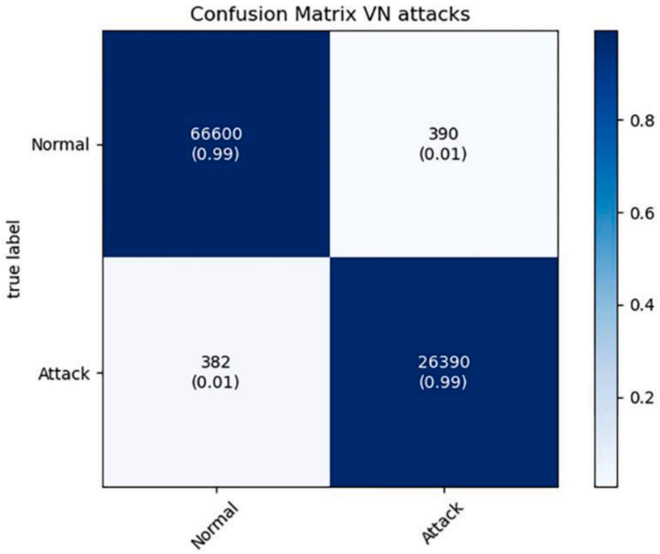


**Fig. 9.** Stacking model confusion matrix in VNA attacks.

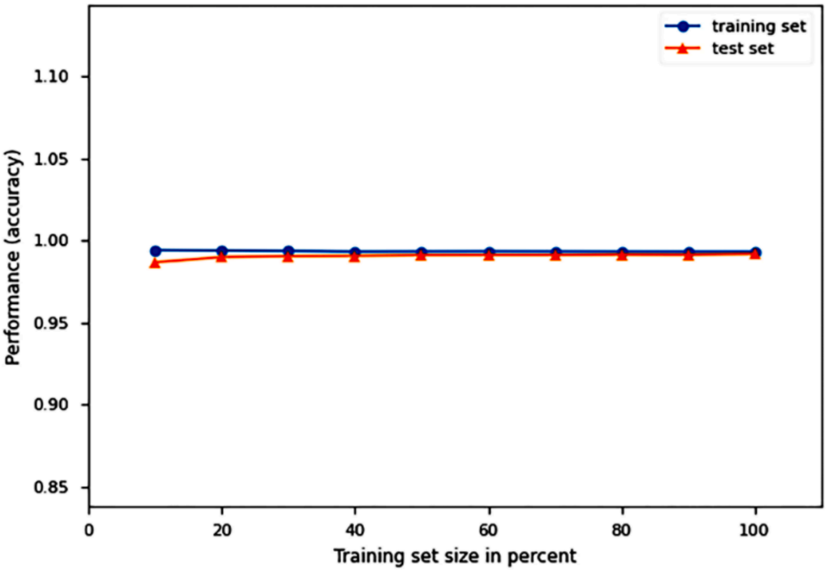


**Fig. 7.** Stacking model accuracy performance based on training set size (VNA).

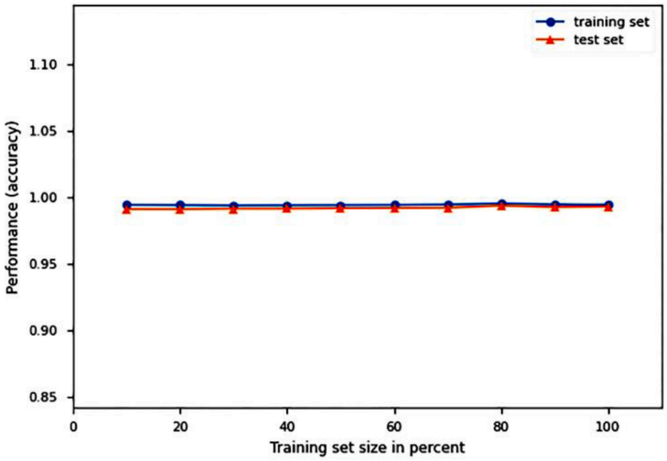


**Fig. 10.** Stacking model accuracy performance based on training set size (DRA).

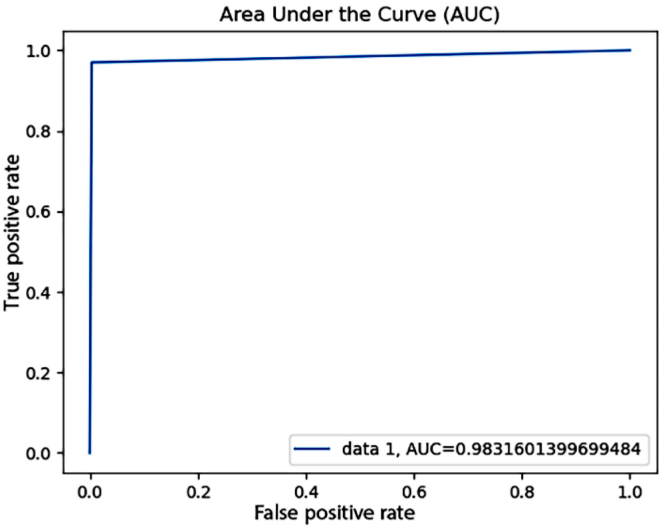


**Fig. 11.** Stacking model ROC curve (DRA).

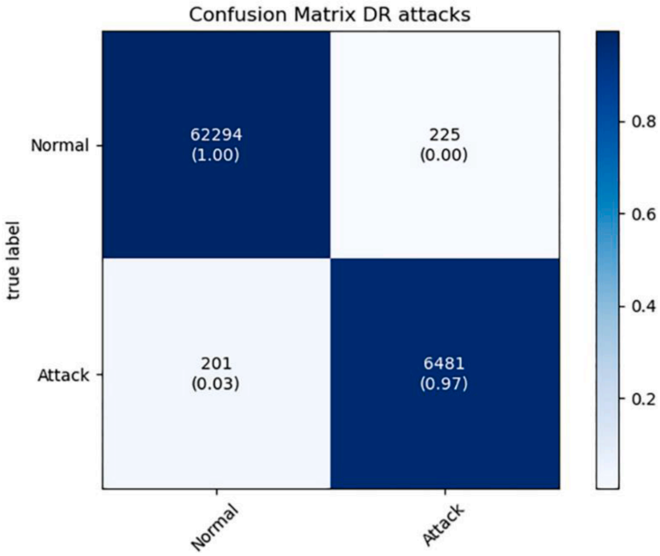


**Fig. 12.** Stacking model confusion matrix in DRA attacks.

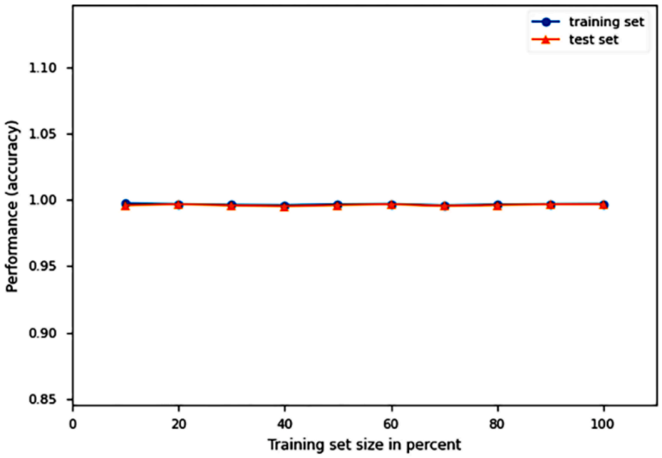


**Fig. 13.** Stacking model performance (DIS Attacks).

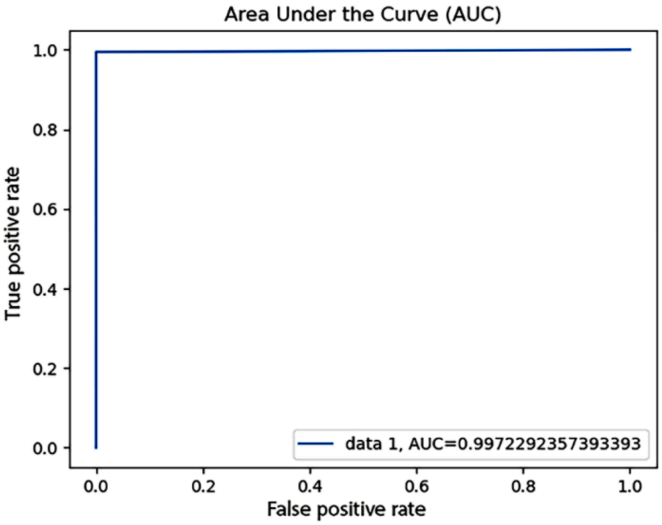


**Fig. 14.** Stacking model ROC curve (DIS Attacks).

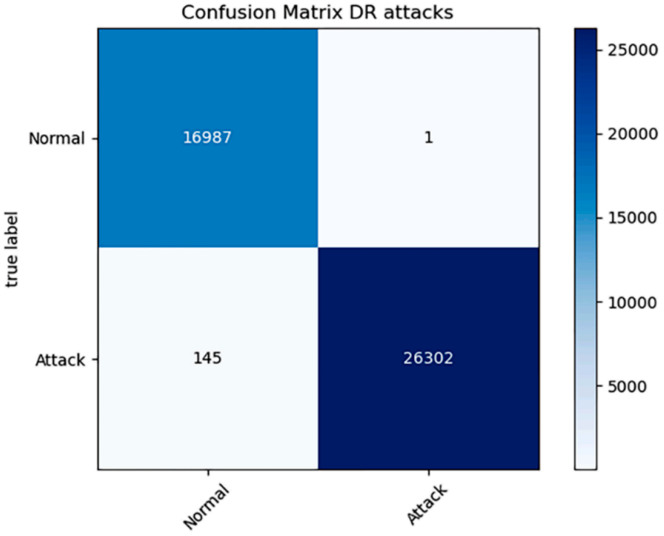


**Fig. 15.** Stacking model confusion matrix in DIS attacks.

**Table 7**
Classification report for the multiclass dataset.

| Classes | Precision | Recall | F-score |
|---|---|---|---|
| Normal | 99% | 98% | 99% |
| VNA | 90% | 94% | 92% |
| DRA | 94% | 99% | 97% |
| DIS | 85% | 91% | 88% |

**Table 8**
Performance comparison between the genetic algorithm and some other feature selection methods.

| Algorithm | Number of selected features | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|---|
| **Forward feature selection** | 11 | 95% | 98% | 97% | 98.19% |
| **Chi2** | 13 | 98% | 98% | 98% | 98.06% |
| **Recursive feature elimination** | 10 | 95% | 95% | 95% | 94.70% |
| **Genetic Algorithm** | 11 | 99% | 99% | 99% | 99.18% |

(0.05), meaning that the significance level is 95%. A common paired sample *t*-test method is used to determine the difference's significance between our proposed model and other machine learning algorithms. A *t*-test is a type of inferential statistic used to determine whether there is a significant difference using the means of two groups that share some characteristics [40]. The estimated probability, aka the P-value, is used to determine the observed, or more extreme, findings when our null hypothesis (H0) assertion is true at the significance threshold (alpha=0.05). Thus, if P-value>alpha, the null hypothesis that both models perform equally on the same dataset cannot be rejected. However, if the P-value is less than 0.05 ($p<=0.05$), the null hypothesis is rejected, meaning that the performance of the two models is significantly different. Table 9 compares the *t*-test for the proposal and other models.

### *5.5. Comparison to conventional methods*

To evaluate the framework objectively, ELG-IDS is compared to the methods proposed by Verma et al. [41], Sharma et al. [42], Yavuz et al. [23], Verma et al. [43], and Thamilarasu et al. [44]. The proposed framework (ELG-IDS) outperformed the previously proposed models by a significant margin. Table 10 presents the comparison results of the performance of our model with that of recent studies that have produced their datasets for infiltration detection.

Compared with the results reported in [41], our ELG-IDS model achieved an accuracy rate of 97.9%, higher by 3.4%. Moreover, [41] did not report any results on the rest of the performance measures, especially the detection rate. Comparing the performance of ELG-IDS with the work presented in [42] for accuracy and detection rate, the ELG-IDS model achieved an accuracy and detection rate of 97.9%, 98.0%, higher by 2.6% and 2.6%, respectively.

**Table 10**
ELG-IDS performance comparison with related work.

| Model | Accuracy | Precision | Recall | F-score |
|---|---|---|---|---|
| Verma et al. [41] | 94.5% | – | – | – |
| Sharma et al. [42] | 95.3% | 96.0% | 95.0% | – |
| Yavuz et al. [23] | 94.7% | 94.0% | 94.0% | 95.0% |
| Verma et al. [43] | 94.0% | – | – | – |
| Thamilarasu et al. [44] | 95.0% | 96.8% | 98.2% | 97.4% |
| M.Osman et al. [11] | 97.1% | 97.3% | 97.0% | 97.0% |
| W. Choukri et al. [12] | 94.5% | – | 100% | 98.0% |
| **Proposed ELG-IDS** | **97.9%** | **98.0%** | **98.0%** | **98.0%** |

In addition, ELG-IDS achieved a higher accuracy rate of 3.2% and a higher detection rate of 4% compared to the work presented in [23], not to mention the computational cost of their approach (deep learning). For [43], they achieved an accuracy rate of 94%, which is lower than the performance of ELG-IDS by 3.9%. Moreover, ELG-IDS provided a significant performance compared to feed-forward neural networks in [44], achieving a performance accuracy of 95%, the work presented in [11] achieved an accuracy of 96.59%, and the work in [12] performed an accuracy of 95%. 94.5%. Our ELG-IDS model achieves an accuracy of 97.9%, an accuracy rate of 98%, a recall rate of 98%, and an F1 score rate of 98%, all of which are evidence of the scalability and effectiveness of our model. Furthermore, it is proof of high-quality data as well as the quality of the feature extraction and selection method proposed in this paper.

## 6. Conclusion

This paper proposed the ELG-IDS-based framework to detect RPL attacks that rely on the version number, decreased rank and DIS flooding. The proposed framework has demonstrated its efficiency in attaining high accuracy and detection rate with near-zero false-positive and false-negative rates. A large dataset was used to train and test the detection framework with a proposed feature extraction and selection technique. The proposed model underwent numerous analyses in terms of quality and testing at different stages. The tests were designed to determine the proposed model's efficiency and advantages and focused on high-performance consistency by using several measures from both categories in three RPL attack datasets. Test results indicated that the proposed ELG-IDS model outperformed most of the presented studies in various aspects, including efficiency and accuracy. In Future work, we will investigate the ELG-IDS model's performance under varying network conditions in dynamic IoT scenarios, including expanding the attack dataset in dynamic IoT scenarios to enhance model assessment

**Table 9**
Comparison of T-test for the proposal and other models for alpha=0.05.

| Detectors pairs | Detector | Mean Accuracy | Mean std | P-value | t-statistics | Reject/ Accept |
|---|---|---|---|---|---|---|
| ELG-IDS & KNN | ELG-IDS | 97.85% | +/-(0.0012) | 0.0019 | 5.9743 | Reject |
| | KNN | 96.74% | +/-(0.0019) | | | |
| ELG-IDS & AdaBoost | ELG-IDS | 97.85% | +/-(0.0012) | 0.0022 | 5.7737 | Reject |
| | AdaBoost | 62.74% | +/-(0.0325) | | | |
| ELG-IDS & SVM | ELG-IDS | 97.85% | +/-(0.0012) | 0.0000 | 72.4499 | Reject |
| | SVM | 93.35% | +/-(0.0016 | | | |
| ELG-IDS & DT | ELG-IDS | 97.85% | +/-(0.0012) | 0.1182 | 1.8843 | Reject |
| | DT | 97.60% | +/-(0.0011) | | | |
| ELG-IDS & GaussianNB | ELG-IDS | 97.85% | +/-(0.0012) | 0.0000 | 172.2573 | Reject |
| | GaussianNB | 61.44% | +/-(0.0039) | | | |
| ELG-IDS & LDA | ELG-IDS | 97.85% | +/-(0.0012) | 0.0000 | 29.1324 | Reject |
| | LDA | 89.57% | +/-(0.0016) | | | |

and conduct diverse experiments involving network dynamics and environmental fluctuations such as network topology, node mobility, and environmental variables. Furthermore, the adaptability and effectiveness of the ELG-IDS model in highly dynamic IoT scenarios will also be evaluated. Furthermore, further research may be needed to fully assess the model's robustness against evasion techniques or attempts to bypass detection under various adversarial scenarios.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.adhoc.2023.103331.

## References

[1] B.B. Zarpelão, R.S. Miani, C.T. Kawakani, S.C. de Alvarenga, A survey of intrusion detection in Internet of Things, J. Netw. Comput. Appl. 84 (2017) 25–37, https://doi.org/10.1016/j.jnca.2017.02.009.

[2] M.A. M.Sadeeq, S.R.M. Zeebaree, R. Qashi, S.H. Ahmed, K. Jacksi, Internet of Things Security: a Survey, in: 2018 International Conference on Advanced Science and Engineering (ICOASE), Oct. 2018, pp. 162–166, https://doi.org/10.1109/ICOASE.2018.8548785.

[3] M. binti Mohamad Noor, W.H. Hassan, Current research on Internet of Things (IoT) security: a survey, Comput. Networks 148 (Jan. 2019) 283–294, https://doi.org/10.1016/j.comnet.2018.11.025.

[4] M. El-hajj, M. Chamoun, A. Fadlallah, A. Serhrouchni, Analysis of authentication techniques in Internet of Things (IoT, in: 2017 1st Cyber Security in Networking Conference (CSNet), Oct. 2017, pp. 1–3, https://doi.org/10.1109/CSNET.2017.8242006.

[5] M. El-hajj, A. Fadlallah, M. Chamoun, A. Serhrouchni, A Survey of Internet of Things (IoT) Authentication Schemes, Sensors 19 (5) (Mar. 2019) 1141, https://doi.org/10.3390/s19051141.

[6] M. El-hajj, M. Chamoun, A. Fadlallah, A. Serhrouchni, Taxonomy of authentication techniques in Internet of Things (IoT), in: 2017 IEEE 15th Student Conference on Research and Development (SCOReD), Dec. 2017, pp. 67–71, https://doi.org/10.1109/SCORED.2017.8305419.

[7] L. Atzori, A. Iera, G. Morabito, The Internet of Things: a survey, Comput. Networks 54 (15) (2010) 2787–2805, https://doi.org/10.1016/j.comnet.2010.05.010.

[8] C. Pu, L. Carpenter, Digital Signature Based Countermeasure Against Puppet Attack in the Internet of Things, in: 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA), Sep. 2019, pp. 1–4, https://doi.org/10.1109/NCA.2019.8935010.

[9] T.A. Al-Amiedy, M. Anbar, B. Belaton, A.H.H. Kabla, I.H. Hasbullah, Z.R. Alashhab, A Systematic Literature Review on Machine and Deep Learning Approaches for Detecting Attacks in RPL-Based 6LoWPAN of Internet of Things, Sensors 22 (9) (Apr. 2022) 3400, https://doi.org/10.3390/s22093400.

[10] O. Sagi, L. Rokach, Ensemble learning: a survey, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 8 (4) (Jul. 2018), https://doi.org/10.1002/widm.1249.

[11] M. Osman, J. He, F.M.M. Mokbal, and N. Zhu, "Artificial Neural Network Model for Decreased Rank Attack Detection in RPL Based on IoT Networks," doi: 10.6633/IJNS.202105 23(3). 15.

[12] W. Choukri, H. Lamaazi, N. Benamar, RPL rank attack detection using Deep Learning, in: 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), Dec. 2020, pp. 1–6, https://doi.org/10.1109/3ICT51146.2020.9311983.

[13] M.D. Momand, M. Khan Mohsin, Ihsanulhaq, Machine Learning-based Multiple Attack Detection in RPL over IoT, in: 2021 International Conference on Computer Communication and Informatics (ICCCI), Jan. 2021, pp. 1–8, https://doi.org/10.1109/ICCCI50826.2021.9402388.

[14] A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for Internet of Things, Futur. Gener. Comput. Syst. 82 (May 2018) 761–768, https://doi.org/10.1016/j.future.2017.08.043.

[15] A. Vasseur, "RPL : the IP routing protocol designed for low power and lossy networks Internet Protocol for Smart Objects (IPSO)," 2011.

[16] A. Mayzaud, R. Badonnel, I. Chrisment, A Distributed Monitoring Strategy for Detecting Version Number Attacks in RPL-Based Networks, IEEE Trans. Netw. Serv. Manag. 14 (2) (Jun. 2017) 472–486, https://doi.org/10.1109/TNSM.2017.2705290.

[17] M.A. Boudouaia, A. Ali-Pacha, A. Abouaissa, P. Lorenz, Security Against Rank Attack in RPL Protocol, IEEE Netw 34 (4) (Jul. 2020) 133–139, https://doi.org/10.1109/MNET.011.1900651.

[18] U. Shafique, A. Khan, A. Rehman, F. Bashir, M. Alam, Detection of rank attack in routing protocol for Low Power and Lossy Networks, Ann. Telecommun. 73 (7) (Aug. 2018) 429–438, https://doi.org/10.1007/s12243-018-0645-4.

[19] A. Mayzaud, R. Badonnel, and I. Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things," 2016.

[20] A. Aris, S.F. Oktug, S. Berna Ors Yalcin, RPL version number attacks: in-depth study, in: NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, Apr. 2016, pp. 776–779, https://doi.org/10.1109/NOMS.2016.7502897.

[21] A. Aris, S.F. Oktug, Analysis of the RPL Version Number Attack with Multiple Attackers, in: 2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), Jun. 2020, pp. 1–8, https://doi.org/10.1109/CyberSA49311.2020.9139695.

[22] M. Nikravan, A. Movaghar, M. Hosseinzadeh, A Lightweight Defense Approach to Mitigate Version Number and Rank Attacks in Low-Power and Lossy Networks, Wirel. Pers. Commun. 99 (2) (Mar. 2018) 1035–1059, https://doi.org/10.1007/s11277-017-5165-4.

[23] F.Y. Yavuz, D. Ünal, E. Gül, Deep learning for detection of routing attacks in the internet of things, Int. J. Comput. Intell. Syst. 12 (1) (2018) 39–58, https://doi.org/10.2991/ijcis.2018.25905181.

[24] S. Raza, L. Wallgren, T. Voigt, SVELTE: real-time intrusion detection in the Internet of Things, Ad Hoc Netw 11 (8) (Nov. 2013) 2661–2674, https://doi.org/10.1016/j.adhoc.2013.04.014.

[25] R. Sahay, G. Geethakumari, B. Mitra, and I. Sahoo, "Efficient Framework for Detection of Version Number Attack in Internet of Things," 2020, pp. 480–492.

[26] R. Bokka and T. Sadasivam, "Deep Learning Model for Detection of Attacks in the Internet of Things Based Smart Home Environment," 2021, pp. 725–735.

[27] M.A. Khan et al., "Voting Classifier-Based Intrusion Detection for IoT Networks," 2022, pp. 313–328.

[28] A. Davahli, M. Shamsi, G. Abaei, A lightweight Anomaly detection model using SVM for WSNs in IoT through a hybrid feature selection algorithm based on GA and GWO, J. Comput. Secur. 7 (1) (2020) 63–79.

[29] S. Cui, Y. Yin, D. Wang, Z. Li, Y. Wang, A stacking-based ensemble learning method for earthquake casualty prediction, Appl. Soft Comput. 101 (Mar. 2021), 107038, https://doi.org/10.1016/j.asoc.2020.107038.

[30] O. Gaddour, A. Koubâa, RPL in a nutshell: a survey, Comput. Networks 56 (14) (Sep. 2012) 3163–3178, https://doi.org/10.1016/j.comnet.2012.06.016.

[31] S.G.K. Patro and K.K. Sahu, "Normalization: a Preprocessing Stage." 2015.

[32] J. Li, et al., Feature Selection, ACM Comput. Surv. 50 (6) (Jan. 2018) 1–45, https://doi.org/10.1145/3136625.

[33] J. Guo, J. White, G. Wang, J. Li, Y. Wang, A genetic algorithm for optimized feature selection with resource constraints in software product lines, J. Syst. Softw. 84 (12) (Dec. 2011) 2208–2221, https://doi.org/10.1016/j.jss.2011.06.026.

[34] Hanchuan Peng, Fuhui Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (Aug. 2005) 1226–1238, https://doi.org/10.1109/TPAMI.2005.159.

[35] O.H. Babatunde, L. Armstrong, J. Leng, and D. Diepeveen, "A genetic algorithm-based feature selection," 2014.

[36] S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, Multimed. Tools Appl. 80 (5) (Feb. 2021) 8091–8126, https://doi.org/10.1007/s11042-020-10139-6.

[37] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, Springer US, New York, NY, 2021.

[38] F.M.M. Mokbal, D. Wang, X. Wang, L. Fu, Data augmentation-based conditional Wasserstein generative adversarial network-gradient penalty for XSS attack detection system, PeerJ Comput. Sci. 6 (Dec. 2020) e328, https://doi.org/10.7717/peerj-cs.328.

[39] F.M.M. Mokbal, W. Dan, W. Xiaoxi, Z. Wenbin, F. Lihua, XGBXSS: an Extreme Gradient Boosting Detection Framework for Cross-Site Scripting Attacks Based on Hybrid Feature Selection Approach and Parameters Optimization, J. Inf. Secur. Appl. 58 (May 2021), 102813, https://doi.org/10.1016/j.jisa.2021.102813.

[40] G. Liang, W. Fu, K. Wang, Analysis of *t*-test misuses and SPSS operations in medical research papers, Burn. Trauma 7 (2019).

[41] A. Verma, V. Ranga, Evaluation of Network Intrusion Detection Systems for RPL Based 6LoWPAN Networks in IoT, Wirel. Pers. Commun. 108 (3) (Oct. 2019) 1571–1594, https://doi.org/10.1007/s11277-019-06485-w.

[42] M. Sharma, H. Elmiligi, F. Gebali, A. Verma, Simulating Attacks for RPL and Generating Multiclass Dataset for Supervised Machine Learning, in: 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Oct. 2019, pp. 0020–0026, https://doi.org/10.1109/IEMCON.2019.8936142.

[43] A. Verma, V. Ranga, ELNIDS: ensemble Learning based Network Intrusion Detection System for RPL based Internet of Things, in: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Apr. 2019, pp. 1–6, https://doi.org/10.1109/IoT-SIU.2019.8777504.

[44] G. Thamilarasu, S. Chawla, Towards Deep-Learning-Driven Intrusion Detection for the Internet of Things, Sensors 19 (9) (1977), https://doi.org/10.3390/s19091977. Apr. 2019.

**Musa Osman** received the B.Sc. degree in computer science from the University of Gezira, Sudan, and the M.Sc. degree in information system from Osmania University, India. He is currently pursuing the Ph.D. degree with the Beijing University of Technology (BJUT), China. His main research interests include security issues in the Internet of Things, primarily based on RPL protocol, machine learning, and articial neural networks.

**Jingsha He** (Member, IEEE) received the bachelor's degree in computer science from Xi'an Jiaotong University, China, and the master's and Ph.D. degrees in computer engineering from The University of Maryland, College Park, MD, USA. He worked for several multinational companies in USA, including IBM Corporation, MCI Communications Corporation, and Fujitsu Laboratories. He is currently a Professor with the Faculty of Information Technology, Beijing University of Technology (BJUT), Beijing. He has published more than ten articles and authored nine books. Since August 2003, he has been publishing over 300 articles in scholarly journals and international conferences. He holds 12 U.S. patents. He also holds over 84 patents and 57 software copyrights in China. He was a principal investigator of more than 40 research and development projects. His research interests include information security, wireless networks, and digital forensics.

**Nafei Zhu** received the B.S. and M.S. degrees from Central South University, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science and technology from the Beijing University of Technology, Beijing, China, in 2012. She was a Postdoctoral Research Fellow with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, from 2015 to 2017. She is currently an Associate Professor with the Faculty of Information Technology, Beijing University of Technology. She has published over 20 research articles in scholarly journals and international conferences. Her research interests include information security and privacy, wireless communications, and network measurement.

**Fawaz Mahiuob Mohammed Mokbal** received the B.S. degree in computer science from Thamar University, Yemen, and the M.S. degree in information technology from the University of Agriculture, Pakistan. He is currently a Ph.D. Researcher in computer science and technology with the Beijing University of Technology, China. He is also a Research Associate with the Faculty of Computer Science, ILMA University, Pakistan. He served as the Head of the Technical Team of Information center Project for the local authority for two years, and the Manager of information systems with the Ministry of Local Administration for ve years. He is the author and a reviewer of various SCI, EI, and Scopus indexed journals. His research interests include machine and deep learning, medical images, braincomputer interface,Web application security, and the IoT security issues.