The Journal of Supercomputing (2023) 79:2626–2653
https://doi.org/10.1007/s11227-022-04753-4

Deep learning-based early stage detection (DL-ESD) for routing attacks in Internet of Things networks

Mohammed Albishari[1] · Mingchu Li[1] · Runfa Zhang[1] · Esmail Almosharea[1]

Accepted: 7 August 2022 / Published online: 22 August 2022

Abstract
Security represents one of the main critical issues in the Internet of Things (IoT), especially the routing attacks in the core network where the loss of information becomes very harmful. This paper proposes a novel scheme called deep learning-based early stage detection (DL-ESD) using IoT routing attack dataset (IRAD), including hello flood (HF), decreased rank (DR), and version number (VN) to enhance the detection capability of routing attacks. The experiments have been performed in three phases: (i) features extraction using linear discriminant analysis (LDA), which aims to generate features more distinguishable from each other, (ii) the features normalization using min–max scaling to eliminate the worst overfittings to the existence of fewer data points in training samples, and (iii) selection the substantial features. The binary classification methods have been employed to measure the proposed model's training efficiency. We have performed the training stage on deep learning techniques such as logistic regression (LR), *K*-nearest neighbors (KNN), support vector machine (SVM), naïve Bayes (NB), and multilayer perceptron (MLP). The comparison results illustrate that the proposed MLP classifier has a high training accuracy and the best runtime rate. Consequently, the proposed scheme achieved prediction accuracy reaching 98.85%, precision of 97.50%, recall rate 98.33%, and 97.01% F1 score rate with better performance than state-of-the-art studies.

Keywords IoT · Deep learning · IRAD · Neural network · Routing attacks · RPL protocols

✉ Mingchu Li
mingchul@dlut.edu.cn

Extended author information available on the last page of the article

1 Introduction

The Internet of Things (IoT) is currently leading the charge in the digital landscape. It offers driving forces such as cost reduction, business revenue growth, new business prospects, security, improved decision-making, improved infrastructure, and improved citizen experience [1]. It is a global revolution in the information industry and mighty changes in people's lives by integrating the globally digital and physical into a single ecosystem due to this massive digital development, IoT networks are becoming more vulnerable to cyber-attacks [2, 3], the attackers make hard efforts to cause damage to the infrastructure of networks to carry out hostile acts such as stealing intellectual property and destroying crucial data using developed techniques [4, 5]. Therefore, more than 70 percent of IoT devices are vulnerable to security attacks, and this is now regarded the most open issues given the lack of protection systems that allow attackers to launch serious attacks such as denial of service (DoS) and routing attacks [6, 7]. Many IoT devices, such as sensors and actuators, consume low amounts of power to operate for longer. [8]. Routing layers are a passage port into the targeted devices through IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), an open IoT networking protocol designed by resource-constrained devices [9].

However, routing protocol (RPL) is specified by IETF to handle the specific properties and constraints of networks, several routing attacks occur through malicious node activities over routing among data packets [10], and the rank value increases from the root node to the child node [11]. The attacker can manipulate the Destination Oriented Directed Acyclic Graph (DODAG) issuance system by raising their rank in the hierarchical tree and acquiring multiple children who route the packets through the attacker's parent. Consequently, the attacker can lure multiple child nodes to choose them as a parent by intentionally changing the rank values and thus attracting significant traffic heading to the root node (the parent branch) to flow through itself [12].

Deep learning techniques have made a significant contribution to tracking the behavior of malicious nodes in the routing protocol; the detection and mitigation mechanisms to deal with routing attacks usually are classified either based on modifications to the existing RPL procedures or added procedures to RPL standards [13]. These methods can be classified into mitigation and intrusion detection systems IDSs such as relating the nodes or packets to their locations within the network using GPS, acknowledgment-based methods by sending a message and receiving an acknowledgment, and trust-based methods against the IoT networks [14]. DL-based IDS in IoT environment: IDS also uses deep learning in heterogeneous IoT networks. For instance, Kim et al. [15] trained the IDS model based on the long short-term memory (LSTM) architecture using a recurrent neural network (RNN). The authors ran tests to determine the best hyper-parameter for the best false alarm and detection rates. Similarly, the authors [16] implemented effective and quick anomaly-based IDS in low-power IoT networks using random neural networks (RaNN). The authors proposed a two-layer approach in which the system learns typical behavior at the first layer and detects various illegal memory access (IMA) issues and data integrity attacks on the network at the second layer. The suggested approach is centralized, delivering the results to a single server.

DL-based attack detection and mitigation: By utilizing the fog ecosystem, Dior et al. [17] developed a DL-based attack detection technique in IoT. In essence, the edge node closest to smart objects is where the attack detection methods are executed. The distributed attack detection mechanisms decide on the learning architecture's output based on the available data, considering various learning mechanism parameters. Abeshu et al. [18] presented a distributed DL-based attack detection technique for the Internet of Things. They implemented DL approaches for threat detection using the fog computing architecture, one of the preferred architectures for implementing IoT applications.

Recent studies on routing attack detection of constrained resource devices have disregarded task distributions and parallel processing of detections scenarios during learning steps, where all the computations of deep learning networks to be addressed in constrained resource devices, any malicious attack of routing attacks on core IoT network can cause enormous loss in network resource consumption. However, it only required tracing malicious nodes in the network, power drain of constrained IoT devices [19]. However, parallelism training in the edge nodes reduces the training stage. Thus, intrusion detection should be in real time. Due to the nature of constrained resource objects in IoT environment, it puts as much computational process and continuous workload on the peripherals as possible.

In order to mitigate the exposure of restricted resources to potential attacks and real-time intrusion detection, the proposed DL-ESD model is designed in a high-level and lightweight method through many stages, starting with data processing which we eliminate the irrelevant features, features extraction using linear discriminant analysis (LDA), aims to generate features more distinguishable from each other, the features normalization using min–max scaling to eliminate the worst overfittings to presence of fewer data points in training samples, then selection the important features. The training accuracy has been achieved the best runtime using binary classification and adopted to keep the service's survival.

The MLP classifier performance is presented in two different phases. (i) ConFigby specifies the optimization algorithm and tracks the loss and other metrics we apply. Initializing the form with these settings requires calling the model. The compile function is as follows: The word "sgd" denotes a random regression ratio. Also, "binary_crossentropy" is defined as a loss function for the outputs with the values 1 or 0—finally, a precision tracking process. (ii) Comprises the network's training process by calling the model and specifying the data for the network training.

1.1 The contributions

In summary, the key contributions of the paper are provided as follows:

(a) Proposing a DL-ESD model-based deep learning for routing attacks detection in early stage before the harmful node can declare on a new version number and create a new DODAG network.

(b) Integration of LDA technique and min–max scaling contribute more distinct features, enhancing DL-ESD model performance in the training and testing stages.
(c) Enhancing the detection accuracy of malicious nodes by the linearity of deep learning reduces the training time.
(d) The binary classification proved that detection efficiency using MLP is higher than other shallow ML algorithms. That offers better prediction, high classification accuracy, and low error rates compared to recent studies.

The remainder of this paper is structured as follows: Section 2 discusses related work of deep learning and IDSs solutions for routing attacks in RPL protocol and attack scenarios in the DODAG network. Section 3 describes the framework architecture sequence, preprocessing data, and the implementation stages of the DL-ESD model. Section 4 shows the analysis and evaluation results, then classification by comparing DL-ESD model with state-of-the-art studies. Finally, Sect. 5 concludes and opens new perspectives for future research.

2 Related work

This section discusses the recent studies on routing attacks and the detection methods. These studies can be classified into two categories: DL-based routing attack detection methods and routing attack scenarios in the DODAG network are widely used in constrained resource devices.

2.1 DL-based RPL protocol in IoT

Recent studies have addressed direct and indirect routing attacks against node resources and countermeasure classifications using emerging mitigation and detection technologies and IDSs in RPL networks. These techniques are categorized as per the following schemes: (i) relating the nodes or routing the packets to their locations within the network using GPS, (ii) acknowledgment-based methods by sending a message and receiving an acknowledgment, and (iii) trust-based methods against the IoT networks.

The authors [19] researched the effects of the constrained resources consumption and the issue of effecting the routing attacks on energy consumption since the fake control messages and building of loops in the DODAGs reduce the lifetime RPL network. Another related study using the IRAD dataset proposed a reliable DL-based routing attack detection approach; the model considers adversarial training and develops a generative adversarial network classifier (GAN-C) with support vector machine (SVM). This study adopts DL parallel learning [20]. Also, the author [21] suggested a novel secure framework for detection routing attacks networks in IoT networks based on industrial IoT networks. The approach can detect hello flood, version number, black hole, and sinkhole attack. The framework performance is evaluated on performance parameters such as attack detection accuracy, true positive, false-positive rate, and end-to-end delay.

Meanwhile, hello flood causes saturation of routing nodes and traffic congestion in DODAG networks. However, the version number attack increases the control packet overhead, energy usage, and end-to-end delay. It also introduces rank inconsistencies and routing loops. It is worth mentioning that energy usage is critical in IoT networks as most nodes are battery-based, and it sometimes becomes a challenging task to recharge them [22]. Thus, it is highly desirable to conduct such a study to detect the malicious nodes early with less power consumption and network continuity of service. The processed data reduces the training duration time and increasing of training accuracy.

Authors [23] have developed an intelligent intrusion detection system (IDS) by combining deep learning algorithms with network virtualization to detect suspicious behavior on IoT networks. When the DNN detects an unknown intrusion, it saves the corresponding tuple of the only filtered features in the “cache” as feedback. This mechanism is utilized for re-training the DNN model, which contributes to the detection system labeling functionality and feature extraction. This study did not address the significant range of device identifiers. The main study [24] proposed a DNN model can detect attacks based on big data; the study created own real dataset called IoT routing attack dataset (IRAD) includes three types of attacks: hello flood (HF), decreased rank (DR), and version number (VN). The proposed model has been trained based on this IRAD dataset, and the performance results show high accuracy and F1 score up 98%. Another IoT dataset consists of five groups of attacks generated by Kamel SOM et al. [19] and proposed a new model based on convolution neural network (CNN). It predicts the suspicious traffic in IoT networks and detects routing attacks. Three methods have been used to preprocess the generated datasets of features selection, Chi-squared, and weight by tree importance to reduce the overfitting and noise to be a fitting input during training the proposed CNN model.

Authors [25] have also designed a novel scheme for detecting the decreased rank attack and verifying the harmful nodes from the DODAG network using round-trip time. In [10], authors have proposed a security routing been found that the critical point at $N=40$ for many classes appeared in different attacks. A related study by the same authors in [26] enhanced a DNN approach based on supervised machine learning. Several scenarios have been implemented and simulated for the three attacks: hello flood, decreased rank, and version number. The results demonstrate that the malicious node of the hello flood generates the maximum number of packets among neighbor nodes in the DODAG network. Consequently, it raises the power expenditure of neighbors and does not impact the DODAG construction. Another model based on machine learning is presented in [29], consisting of data collection, feature extraction, and two classification methods. The IRAD dataset has been used to train ML-RPL model for new features that have been added manually; ML-RPL indicates an accuracy rate up to 97%. However, all the above approaches are considered models-based on RPL using various classification methods and the same dataset and still suffer from the DODAG Information Object (DIO) control message overhead and the uneven accuracy data of packet delivery ratio. Table 1 depicts the recent works related to IDS system and detection of routing attacks in RPL protocol. The

Table 1 Recent works related to RPL protocol attacks in IoT networks

Ref. no.	DL methods	Dataset	Routing attack	Targeted against
[19]	CNN-based deep learning	IoT routing dataset	Selective forward, sinkhole, version, and wormhole attack	Resources and topology
[21]	RPL-based IIoT environments	iIoT dataset	Hello flood, version number, sinkhole, and blackhole attack	Resources and topology
[23]	Based IDS framework	IoT simulation dataset	Blackhole, DDoS attack, sinkhole attack, and wormhole attack, opportunistic attack	Topology
[24]	DL-based detection of routing attacks	IRAD dataset	Hello flood, version number, and decreased rank attacks	Resources and traffic
[25]	Based on round-trip time	Simulated data	Decreased rank attack	Traffic pattern
[10]	SRPL-RP-based rank strategy	Simulated dataset	RPL rank and version number attacks	Resources and traffic
[27]	ANIDS for RPL-based IoT networks	Simulated dataset	Hello flood, blackhole, sinkhole, and selective forwarding	Traffic pattern
[28]	RPL-based system and IoT network	Simulated data	Hello flood, increased version, and decreased rank attack	Resources and traffic
[26]	ANN-based IDS in RPL protocol	Simulated data	Hello flood, version number, and decreased rank	Resources and traffic
[29]	ML based on RPL in IoT network	IRAD dataset	Decreased rank attack	Traffic pattern
[30]	Counter-based detection	Simulated data	Flooding attack	Resources
[20]	DL based on GAN-C and SVM	IRAD dataset	Hello flood, version number and decreased rank	Resources and traffic
Our scheme	DL-based RPL protocol in IoT	IRAD dataset	Hello flood, version number and decreased rank	Resources & Traffic

highlight of recent literature studies was made according to the closest studies that used the same dataset or other data for the same attacks with different features.

2.2 RPL attack scenarios

Many attack scenarios have been simulated to choose the preferred neighbors mote and keep the energy resource of constrained devices along (refer to Fig. 1). Almusaylim et al. [10] referred to choosing the best parent when node (N = 12, 6, and 26) sends a DAO control message to the sink node $\boldsymbol{N}$ starts the distribution module after selecting the preferred parent $\boldsymbol{P}$, while the module calculates the MAC value via the specified parameters. The sink maintains the information table to store four groups of information about all nodes of messages received from the DAO. Likewise, the central unit running is extracted in the incoming information pool via the DAO message for $\boldsymbol{N}$ node. Then, the MAC value is calculated if the two MAC values match. It will ensure that the $\boldsymbol{N}$ node sends the message while maintaining the integrity of the received DAO message [31].

Thus, Palattella et al. [32] referred if $\boldsymbol{N}$ is an intermediate node, the sink node checks the $\boldsymbol{N}$ rank received from the node or child nodes it belongs to. Also, if the order does not match the order that the node received from the DAO message, the source declares $\boldsymbol{N}$ node malicious. A number of routing attacks target resource-constrained devices in IoT networks. In this section, we explain the three most types of attacks. In addition, the MAC value is validated, the sink starts to check the rank of the $\boldsymbol{N}$ node only when it is a leaf node, and then the pool checks for the presence of a low-rank and hyper-level attack [21].

2.2.1 Hello flood

This attack occurs in the routing layers. The malicious node sends DODAG Information Solicitation DIS messages successively to multiple nodes on the RPL network.

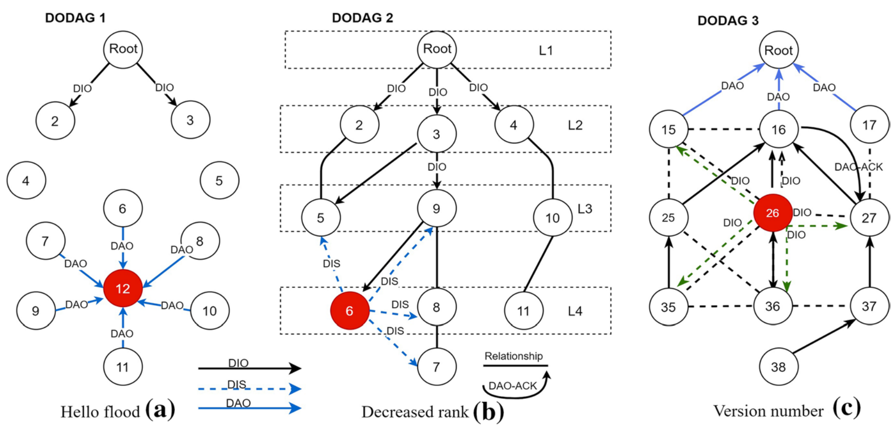

Fig. 1 RPL network constructions: **a** hello flood, **b** decreased rank, and **c** version number

The hello flood attack shortens the interval between each two successive DIS messages [28]. After infiltrating the RPL network, the malicious node will immediately begin sending out multiple DIS messages to all of their neighboring nodes in the network, as shown in Fig. 1a. Consequently, adjacent nodes receiving DIS messages must respond with DIO messages, resulting in a set of timer and repeated DIO messages that waste a significant amount of power from neighboring devices receiving a request from the malicious node [10]. Agiollo et al. [33] developed an intrusion detection system that can deal with multiple attacks to avoid the overhead of RPL. So, in hello flood and DIS attack scenarios, the malicious node has an abnormal amount of control packets. DETANOR's attack classification mechanism identifies the attackers as those devices transmitting an abnormal amount of control packets.

2.2.2 Decreased rank

The harmful node in a decreased rank attack declares its false low rank through the DIO control message to attract traffic to its neighboring nodes, as shown in Fig. 1b. The (root node) takes the 1st rank in DODAG construction [10]. Node 1 (root) sends multicast DIO messages containing all the information of its neighbor nodes. The neighbor nodes in rank 1 choose the root node as a parent. Therefore, after connecting DODAG, the adjacent nodes of root nodes 2, 3, and 4 multicasts their DIO messages by setting the rank to 2nd. The rank of nodes increases in a descending direction. If nodes discard high-value DIO messages from the rank value, they visualize the DIO message coming from child nodes (down) [25]. Node 3 can add nodes 2 and 4 as a preferred parent as in the node three range. Moreover, all descending nodes receive DIO messages from neighbor nodes but decide on harmful nodes as the preferred parent based on the best rank.

Node 6 is harmful and declares a false rank value (rank = 1st) to enable neighboring nodes 5, 9, 8, and 7 to move toward the harmful node, indicated by dotted arrows [21]. The sixth node means that its rank value is rank 1st, while its actual ranking value is 4. In the current circumstances, nodes 5, 7, 8, and 9 decide the harmful node six as the favorite parent and reroute the traffic through node 6, as shown in Fig. 1b.

2.2.3 Version number

This attack is one of the most efficient attacks in routing layers; particularly in the network layer, the malicious node alters a DIO message [32]. In contrast, the malicious node receives a DIO message in the IoT network. The DODAG version number is incremented in a DIO message, and the malicious node forwards the infected DIO message [25]. These require overhauling the entire DODAG architecture. These frequently forced DODAG re-assessment also wastes the key parameter "power" from all nodes belonging to DODAG construction. Thus, the nodes in the network lose their energy rapidly, as shown in Fig. 1c. As a result, the life of the network is greatly affected. A. Mayzaud et al. [34] proposed a monitoring strategy with dedicated algorithms for detecting version number attacks; the solution's performance has been evaluated through experiments and quantified with the sup. Almusaylim et al. [10] proposed a security routing protocol (SRPL-RP) for RPL rank and version number attacks. The proposed protocol detects and isolates attacks and adds them to the blocklist. The detection is based on a comparison of the ranking mechanism. The analysis results indicate that the PDR packet delivery rate of (98.48%) and SRPL-RP achieved an accuracy rate of (99.92%) under version number attacks. Sahay et al. [35] proposed an inclusive framework for the prediction of version number; the framework includes a feed-forward neural network that uses the traffic as an input for prediction version number attack. Therefore, the framework uses the smart contract-fortified blockchain technique to establish secure channels to access in IoT resources.

3 Proposed DL-ESD model

This section introduces the phases of the proposed model, describes the DL-ESD structure and implementation, also provides a detailed explanation of data processing, and then builds the deep neural network.

3.1 Framework overview

The framework structure consists of three levels: data preprocessing, deep learning networks, and classification, as depicted in Fig. 2. It describes the framework structure as follows: Data processing is divided into three phases; feature selection, the linear discriminant analysis (LDA) has been used for feature extraction and a linear projected transformation utilized for feature extractions in different aspects. It means that feature extraction based on machine learning techniques can obtain an optimal contrast level between the extracted features and improve the performance of the training stage. Data normalization and visualization; in this phase, min–max scaling methods normalize the dataset and adopt the standard quintile conversion to disperse marginal values, and then, the correlation coefficient is also measured to select the dependency level for best features. In the third stage, the dataset is split into a 75% training and 25% testing set using scikit-learning and Pandas function (). In this stage, features are scaled to be compared on a common basis, and then, the preprocessed data is fitted into our classifier to extract the most important features. Therefore, the experiments are performed the deep learning techniques. In the last level, the performance of deep learning techniques is measured and compared to the MLP technique. To achieve the research objectives in capability detection for the malicious nodes in the routing layer, we have proposed a novel deep learning-based early stage detection (DL-ESD) using IoT routing attack dataset (IRAD). The deep learning techniques have been compared to make detecting attacks most accessible. However, MLP technique has proved ability highly in training accuracy and duration, which is the most contribution of our model to improving detection accuracy. Binary classification methods also have been employed to improve performance efficiency.

DL-ESD model is presented under two different phases. The first phase: ConFigby specifies the optimization using adam optimizer as faster training in less time and more efficiency and tracks the loss and other metrics we apply.

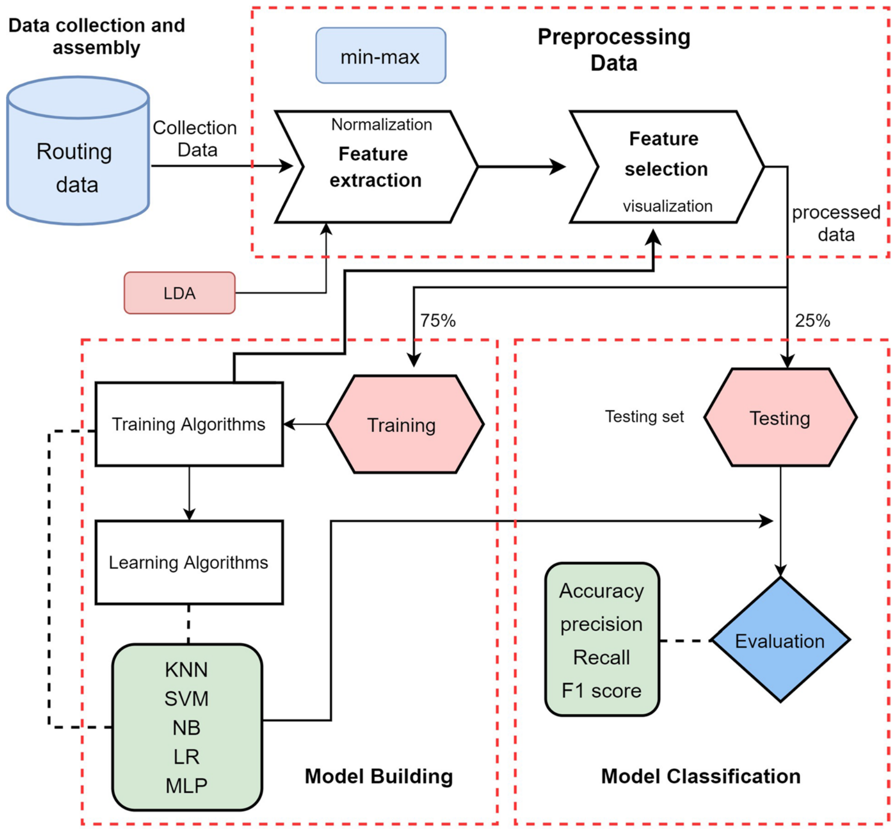

Fig. 2 Proposed framework for detection of routing attack of RPL-based IoT networks

Initializing the form with these settings requires calling the model. The compile function is as follows: The word "sgd" denotes a random regression ratio. In the second line, "binary_crossentropy" is defined as a loss function for the outputs with the values 1 or 0—finally, a precision tracking process. The second phase includes the training process by calling the model to specify the data we want to train the network on, namely X_train and Y_train, and then setting the mini-batch size at 32 and choosing the training time epochs = 100. With ten iterations. Finally, we decide on our verification data which leads measure model performance that can verify at each point of the verification data.

Packet sniffer: Enter the interface's name to sniff node information that can be seen now.

Training and testing our neural network: Enter the name of the CSV dataset file you wish to use. If you want to load a previous model, enter "y" and the model's name. Otherwise, just press Enter. Based on the size of the dataset and model topology, the process may take a bit of time. Once completed, enter “y” to see the weights and intersections of the model after training, input “y” again to be saved model (end as “.sav,” must).

Data viewer: It allows displaying data within a dataset. Enter the name of (.CSV) dataset that you would display, input “a” to see all, input “n” to see numeric data only, and “c” to see categorical data only.

Live deep neural network: DNN uses a trainer to detect routing attacks from the menu. Enter the name of the interface that would detect RPL attacks when the input of the trained model’s file name, as that will run until stopped or an attack is detected.

Visualizing loss and accuracy: Displays a visual representation of how the ANN model sounds. It can change code, currently showing an input layer of $9n$, two hidden layers of $100n$, and “*1n*” as an output layer.

3.2 Data preprocessing

The IRAD dataset has been used in this study for training and testing stages within various scenarios. Three datasets samples are used, and each sample contains two classes: the malicious and benign samples listed in Table 2 [24]. Therefore, when the completion of the simulation stage. The packet capture (PCAP) files have been converted into a comma separated value (CSV) format using a Wireshark analyzer and developed a preprocessing script for Python data that applies a feature extraction process for the converted CSV files [24].

3.2.1 Features extraction

This phase aims to reduce the number of dataset features by discarding the original overfitting features, creating new features from existing ones, and summarizing the most information in raw features. To eliminate the overfitting and get issue-oriented attributes to distinguish between routing attack and normal RPL traffic. Moreover, it reduces the running of training and validation time. The LDA method reduces the dimensions and shows feature samples on a straight line to produce more distinct features. The number of extracted features must equal one since each subset has two classes of attack and benign [36]. The flow identifiers such as Source IP, destination IP, source port, destination port, packet length, time, and protocol type are eliminated to avoid bias toward malicious or legitimate nodes. The IRAD datasets contain qualitative and quantitative features. Our learning algorithm allows for quantitative

Table 2 IRAD datasets values

Dataset	No. of values	(GB)
Hello flood attack	64,178,435	0.75
Version number attack	22,868,210	0.27
Decreased rank attack	49,873,385	0.58

values only. However, we applied feature conversion qualitative features to convert its integrated format. Thus, the selected features such as DAO are used for unicasting destination information according to the parents selected.

In RPL also, DIO is the message type. It holds the current sequence for the node and uses the specified metrics as distance or hop count to decide the optimal route over the base node. DIS is another message form, and nodes use DIS to join WSN. Other types of IRAD datasets are RPL nodes that are simulated data nodes. Firstly, we have calculated several transmitted and received packets for every node in 100 s in the scheduled time and then split these values into 1000 ms to obtain each node's DIO transmission and receiving rates (DTR, DRR), respectively. In all time ranges, the time it takes for each node to be sent and received is calculated. It can also calculate the total transmission time and receiving time by adding up each transmission and receiving time, a 1000 ms data packet and each node's transmission and receiving time. The number of control data packets was calculated in the window size for each node and extracted features as per the steps outlined above. The benign and hostile datasets have the same structure when mixed.

3.2.2 Features normalization

Normalization is one of the most used methods for shifting values between 0 and 1 in a given range. It cleans up the data and lowers bias, resulting in high detection accuracy and improving the performance and training stability of the model [37]. Thus, we have performed feature normalization to drag datasets into the same range. The min–max scaling methods proved the easiest, most intuitive, and more flexible for normalizing the values in the selected features, which **X*** is the new feature from 0 to 1, *RPL.FEATURES* is the original feature value and $RPL.FEATURE_{min}$ and $RPL.FEATURE_{max}$ are the maximum and minimum values of the selected features as shown in Eq. (1), respectively.

$$X_{*} = \frac{\left(RPL.FEATURE - RPL.FEATURE_{min}\right)}{\left(RPL.FEATURE_{max} - RPL.FEATURE_{min}\right)} \tag{1}$$

Each feature is imposed separately on the standard quintile conversion. The goal of the transformation is to disperse marginal values of DAO, DTR, and Trickle timers reset features, which could alter the connection between values [38]. The best nine features for training and testing have been chosen after data normalization and offset, including DIS, DIO, and DAO for transmitted and received 6LoWPAN attributes with high scores as shown in Fig. 3a–c. Finally, all concatenated datasets have different network topologies for each attack.

3.3 Selection of importance features

As the aforementioned result of steps, the importance features were selected by the strongest relationship with the output variable and have been selected by scikit-learn, removing the common and irrelevant features. The features are adjusted based

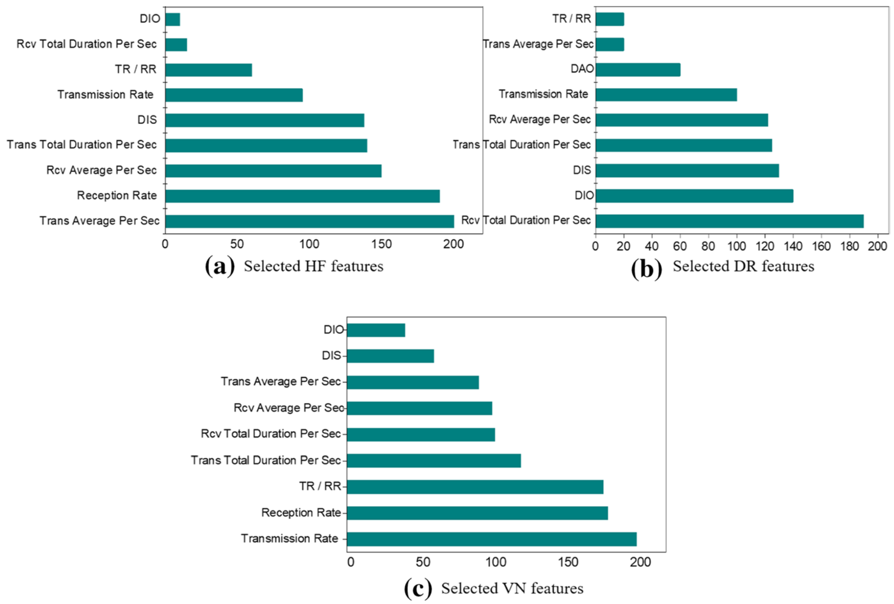

Fig. 3 Feature selection and importance of data preprocessing

on the DIO, DAO, and 6LoWPAN characteristics with comprehensive detection, as listed in Table 3. We addressed the missing values in Pandas DataFrame and use a function that is null () and not null (). Both functions help in checking whether a value is NaN or not. This function can also be used in Pandas series to find null values in a series and then split the dataset into training and test sets. The important features can use to enhance the prediction models. That can apply to selecting these features to keep the highest scores or remove the lowest scores. Figure 3a–c depicts the score of the best nine features after computing its correlation and variance analysis. The ablation experiments have been applied for 18 features using the Pearson coefficient correlation in each phase. Some features were subjected to measuring the

Table 3 Selection of the important features

No.	Features	Detected attack	Min–max (X*)
1	*Reception rate*	HF, VN	0.00668793e−05
2	*Transmission rate*	HF, DR, and VN	0.78469016e−05
3	*Rcv average per sec*	HF, DR, and VN	0.67000000e−01
4	*Rcv total duration Per sec*	HF, DR, and VN	1.00000000e−03
5	*DAO*	DR	0.99997274e−01
6	*DIS*	DR and VN	1.00000000e+00
7	*Trans total duration per sec*	HF, VN	0.03000000e−01
8	*DIO*	HF, DR, and VN	0.01000000e−01
9	*TR/RR*	HF, DR, and VN	0.45000000e−01

correlation with the others, the results indicate that the nine features are considered with a high level of dependence, whereas further experiments indicate to the lack of correlation in other features. The PCC (r) for variables x and y is calculated using Eq. (2)

$$R = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}. \tag{2}$$

The important features selected for the three attacks rely on the packets transmission rate and reception average for 6LoWPAN protocol and the DIO control messages. The important features were selected using a combination of deep neural networks, Pearson correlation coefficients, and histograms [39]. The extracted features' value was assessed using the DNN technique to determine the optimal number of neurons needed in the network, bagging the means to combine unbiased and noisy variables to create a model with a lower variance. Therefore, the analysis has been conducted on selected features using MLP classifier, a feed-forward with at least three-node layers in ANN. For MLP classifier, only one hidden layer multilayer perceptron is utilized, relying on different activation functions.

3.4 Deep neural network (DNN)

The hidden layers have been decided in the DL-ESD model based on our experimental approach. We added the independent variables as input values (X) and dependent variables output values (0, 1) divided by 2. The network has been tuned by adding the extra nodes to reach optimal results with two hidden layers. Therefore, we tested the model's accuracy by varying the number of layers and selecting the one that produces the best result. The neural network consists of 4 layers; the input layer has nine neurons. The output layer has just two neurons as the last layer; this is called a regression model as depicted in Fig. 4. The first hidden layer includes 100 neurons and 100 neurons in the second hidden layer. ReLU activation function is used in hidden's layers. In contrast, the sigmoid function utilizes in the output layer as it is known that network training involves identifying the network model as a structure and then finding the best values from the data to fill in the model. Before starting training, the dataset is split again at a rate of 0.3 as a validation dataset to adjust the model's training performance.

Let $X = X = \{X_1, X_2, X_3, \ldots, X_n\}$ be the input vector with $n = 9$, the steps are used to enter the product sum activation function "SOP" to calculate the value of "S" of "X" as input values for our features as shown in Eq. (3) and "W" to measure weights. However, the nonlinear activation function is represented by $A(.)$ and w_i and b_i indicate the weights and bias of hidden layers i as Eq. (4) and the activation function ReLU is used in hidden layers, while the Eq. (5) shows the mathematical representation in the neuron that achieved ANN as illustrated in Eq. (6)

$$S = X_1 * W_1 + X_2 * W_2 + b \tag{3}$$

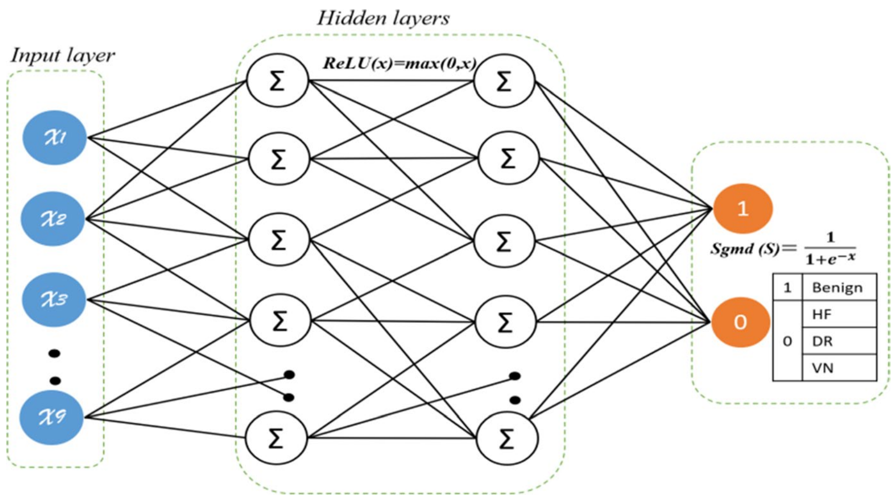

Fig. 4 Structure of our deep neural network

$$H_i(x) = A\left(w_i^T x + b_i\right) \tag{4}$$

$$ReLU(x) = max(0, X) \tag{5}$$

$$Y = \sum (inputs) * (weights) + bias \tag{6}$$

To enhance the measurement accuracy in the output layer, we used the ReLU and Tanh functions, but the accuracy rate is ineffective as it exceeds 65.3% and 55.8%. Besides, the sigmoid function has a distinct "S" curve and a mathematical representation for ReLU and Tanh function in binary classification. After (7) sigmoid function is used in the output layer, our model's accuracy exceeds 98.98%. We also decrease the sharp increase by nearly 62–98% during training phases by applying regularization and dropout. The randomly selected nodes increase additional time and cost to drop out in each stage. As a result, the performance of our deep layers is significantly reduced. In other words, when handling large datasets, interactions between neurons almost always result in overfitting. We also apply dropout and regularization for these reasons. As in [37], Keras is used as a framework for deep learning because it includes several advantages, such as its modularity makes it easy to construct and test complex neural networks. Firstly, Keras is a powerful, easy-to-use Python library. Secondly, it is a high-level API for building and training DL models. Therefore, it makes it possible to create deep neural networks quickly.

$$Y = \frac{1}{1 + e^{-x}} \tag{7}$$

$$Output = f(0, 1) \tag{8}$$

Algorithm: pseudo-code Deep Learning Supervised

Function:
dftrain ⟵ *dataset.csv*
Mixing Dftrain step by step of rows
X , Y ⟵ *dftrain*
Train_Test_split (X, y)
Fitting *(X, Y)* in *MLP classifier* *Used actually to train the model*
Calculate the matrix R correlation coefficiency in features

$$R = \frac{n(\sum xy) - (\sum x)\ (\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2\ ]\ [n\sum y^2 - (\sum y)^2]}}$$

Drop the irrelevant and insignificant features
End of Selection of important features
Definition of X_train, Y_train & X_test, Y_test
DNN Model performance
Model ⟵ *Sequential model*
For input= *X* to end of input **do** change input for each run
for neurons = 1 to n **do** increase neuron for each time
for repeat = 1 to n repeat run every **n**
Train DNN
Adjust neural network layers

$$H_i\,(x) = A(w_i^T x + b_i)$$

Optimizer ⟵ DL-ESD network optimization Adjust Adam Optimizer
Compile MLP classifier
Model ⟵ *X_train* Beginning training
X predicts Y predict ⟵ *df predict*
end for
end for
Saving model and weights as CSV format
end for
Prediction analysis
Load the datasets as dfpredict
X predict Y predict ⟵ *dfpredict*
X predict ⟵ *scaled X predict*
Classification performance & report exports
End function

Tensorflow also includes several implementations for creating a complex DL model. In the training process, the datasets were shuttled to optimize the deep learning model performance and avoid overfitting [40]. The preprocessed dataset is split into *x*_train and *Y*_test. The first, *X*, is the unlabeled portion, and *Y* is the second portion. More specifically, *Y* is the supervised learning portion of our model that makes learning algorithms. X and Y are divided into *X_train*, *X_test*, *Y_train*, and *Y_test*. Train parts are utilized in the training section, while test components measure the training process performance.

4 Experimental results and evaluation

The primary objectives of this research are to develop methods for RPL attack detection to improve prediction accuracy rates in IoT networks with low error. As the outcome is actual, the system prediction is true; otherwise, it is all false. This case is called positive if the forecast is related to the attack. Otherwise, it is negative. Therefore, there are four logical possibilities: true and safe prediction, correct and attack, false-negative and safe attack, respectively, where: *True Positive* "*TP*," *True Negative* "*TN*," *False Positive* "*FP*," *and False Negative* "*FN*." The classification error is the ratio of incorrect predictions to totality prediction numbers [19, 41].

Accuracy (ACC): is the percentage of true detection over total data instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

Precision: represents how many of the returned attacks are correct.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

Recall: measures the ratio between a true positive and a total of both a true positive and a false negative.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{11}$$

F1 score: is the weighted harmonic mean of the precision and recall and reflects the balance between P and R.

$$F1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{12}$$

Upon nature and amount of the dataset, we applied different ratios and picked the 75–25 ratio to give the best performance result. Also, the picked ratio is proportional to the midsize of our data into 75% for the training set and 25% for the testing set to evaluate the provided model with a biased evaluation fit on the training dataset. Figure 5a shows that training epochs rate is very appropriate and it also can be inferred that the DL-ESD model has performed best in training and testing accuracy.

Table 7 represents routing attack has various methods of feature selection. The features have been applied to each problem in the dataset and fed to the neural network. It can be seen that the MLP technique has high accuracy, precision, recall, and $F1$ score values performed well. Also, Fig. 5b depicts the loss rate values to the model parameters by adjusting the weight vector values through various optimization approaches that have reduced the training time. It means how efficiently our

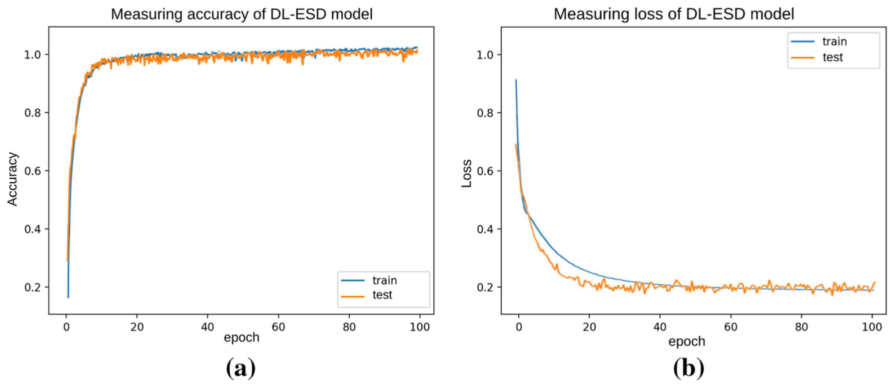

Fig. 5 Measuring performance of training and testing epochs: **a** accuracy rate of DL-ESD, **b** loss error of DL-ESD

model behaves after every optimization iteration, in which the weights change in each iteration of 10 iterations. In training phase, the higher number of neurons and epoch, the higher accuracy, and the lower the loss rate. In Fig. 6, the ROC curve close observation of performance matrices shows the ability of sensitivity to correctly predict malicious nodes as harmful nodes while specificity ability to predict normal nodes as malicious nodes correctly.

4.1 Confusion matrix

The detection accuracy rate of the DL-ESD scheme success should be high, but the false-positive rate should be lower [42]. The misclassification rate is directly proportional to the false alarm rate, as presented in Tables 4 and 5. The classification results illustrate high performance in our scheme. Figure 7a shows that the bias rate of the safety packets is increasing, and the false-negative rate of the total hostile

Fig. 6 Receiver operating characteristic

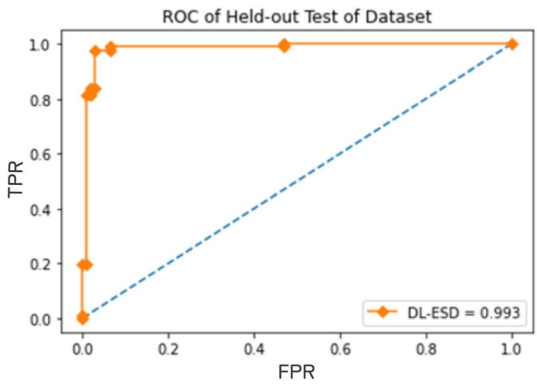

Table 4 Prediction rate of training stage

	Predicted class	
Actual class	718,861	67
	70	2671

Table 5 Prediction rate of testing stage

	Predicted class	
Actual class	308,032	81
	75	1099

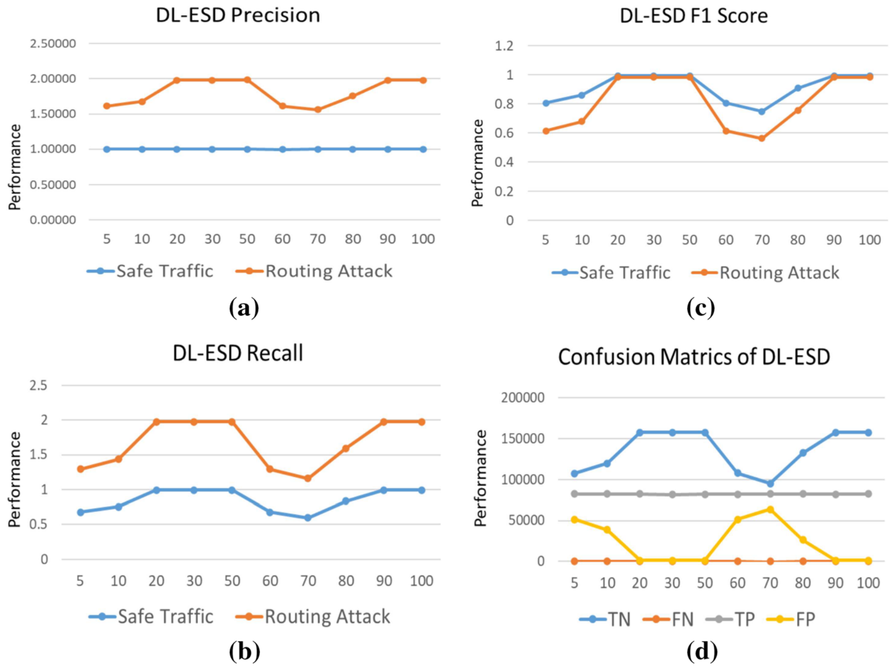

Fig. 7 Overall performance of measure the detection ratio of model: **a** precision, **b** recall, **c** $F1$ score, and **d** confusion matrix

packets is decreasing; it can be summarized that the classifier understands the positive value of one class from another.

Figure 7b illustrates the sensitive rate of the classifier and the simple raise in the bias ratio to the positive values. In contrast, the classifier decreases the sensitive rate of false negative for a package rate. It can be summarized that the classifier can understand the positive value of a class from another.

Table 6 Training classification from confusion matrices for our model

Class	Precision	Recall	$F1$ score	Support
Benign	1.00	1.00	1.00	718,925
Attack	0.96	0.97	0.97	2744
Avg	1.00	1.00	1.00	721,669

Table 7 Testing classification from confusion metrics for our model

Class	Precision	Recall	$F1$ score	Support
Benign	1.00	1.00	1.00	308,116
Attack	0.97	0.97	0.99	1171
Avg	1.00	1.00	1.00	1171

Figure 7c exhibits the F1 score as the weighted average of the sensitivity and precision rate for positive and negative values. $F1$ is usually more effective than precision, especially if the class distribution is uneven. But this model reflects the balance between P and R classes, which the confusion matrix indicates to uneven class distribution. Thus, that will be contributed much to classifier performance.

The proposed detection method effectively achieves the highest TPs, TNs, and lowest instances of FNs. Figure 7d indicates that the prediction ratio for TP's input values is the classification result, and TN is largely high. At the same time, the false committed by the classifier for (FN, FP) is low. It also indicates that the classifier performance and the expected ratios are satisfactory. The detection rate with training confusion metrics and multiple datasets are listed in Table 6. The binary classification approach is obtained for training and testing stages based on DNN technique. Due to the inability to measure the bias ratio among the classes in the classifier, it is necessary to rely on classification reports to obtain a deeper concept of the strength and performance of the classifier more than the accuracy. The experimental results in class 0 and class 1 classification report indicate that the MLP classifier is more biased toward class 0 in training and testing, as listed in Tables 6 and 7.

4.2 Training and testing analysis

The ablation experiments drive the enhancement of the neural network performance based on the correlation coefficient to considered features. The features are split into different levels accordingly to the node's behavior. We have applied the incremental training stage and conducted several tests over IRAD preprocessed features, including the multiclass categories of three attacks and normal nodes for binary classification based on MLP classifier; we also compared MLP with shallow machine learning, KNN, SVM, NB, LR, and MLP techniques and state-of-the-art routing attacks. The test is applied using the weights learned during the training stage. This section reports the average of running the training model 10 iterations. The comparison results show the effectiveness of early detection and identify the best parent in

Table 8 Calculating scoring time during the training and testing stages of our approach based on binary classification compared to different ML techniques in our work

Attack	Dataset rows	Classifier	Training (ms)	Testing (ms)
Hello flood (HF)	1,048,576	KNN	94.05	7.30
		SVM	205	22.05
		NB	120.40	33.02
		LR	8033	120.84
		MLP	78.32	5.22
Decreased rank (DR)	1,047,821	KNN	56.45	10.03
		SVM	42.128	14.81
		NB	42.043	11.70
		LR	6140	168.14
		MLP	33.02	8.05
		KNN	0	21.04
Version number (VN)	1,048,576	SVM	311.52	33.07
		NB	4065.10	78.91
		LR	2601.76	42.41
		MLP	46.40	15.10

the DODAG construction to keep the IoT network in service. Table 8 and Fig. 8a summarize the training and testing accuracy results compared to supervised DL algorithms.

The training accuracy of HF demonstrates that MLP technique achieves a performance reach of 0.992% higher than the four techniques within 78 ms of training duration and up to 0.5% of testing period. In Fig. 8b, the DR sample indicates that among both MLP and KNN a relative parity of training accuracy with an approximate increase of MLP with 0.98% within 33 ms as an optimal time is observed. In Fig. 8c, the MLP technique in VN shows high detection accuracy of 0.97% compared to other methods within 46% ms of training duration. In order to evaluate the robustness and effectiveness, Fig. 8d compares the test performance accuracy of our classifier with four other classifiers K-nearest neighbors (KNN), logistic regression (LR), support vector machine (SVM), and naïve Bayes (NB). HF in SVM has a lower accuracy than the other classifier. Therefore, it cannot be recommended because its learning time is also high, while the MLP classifier has higher accuracy and lowest time, as listed in Table 8.

Figure 8e shows the test performance accuracy compared with the other four classifiers; the test accuracy of NB and LR is low compared to the other classifier. Therefore, they cannot be recommended because its learning time is high, while the MLP classifier is higher accuracy and lowest time, as listed in Table 8.

Figure 8f shows the test performance accuracy compared with four other classifiers; the test accuracy of five classifiers is uneven and has a slight improvement from each other. Therefore, they can be recommended due to good learning time, as shown in Table 8'..

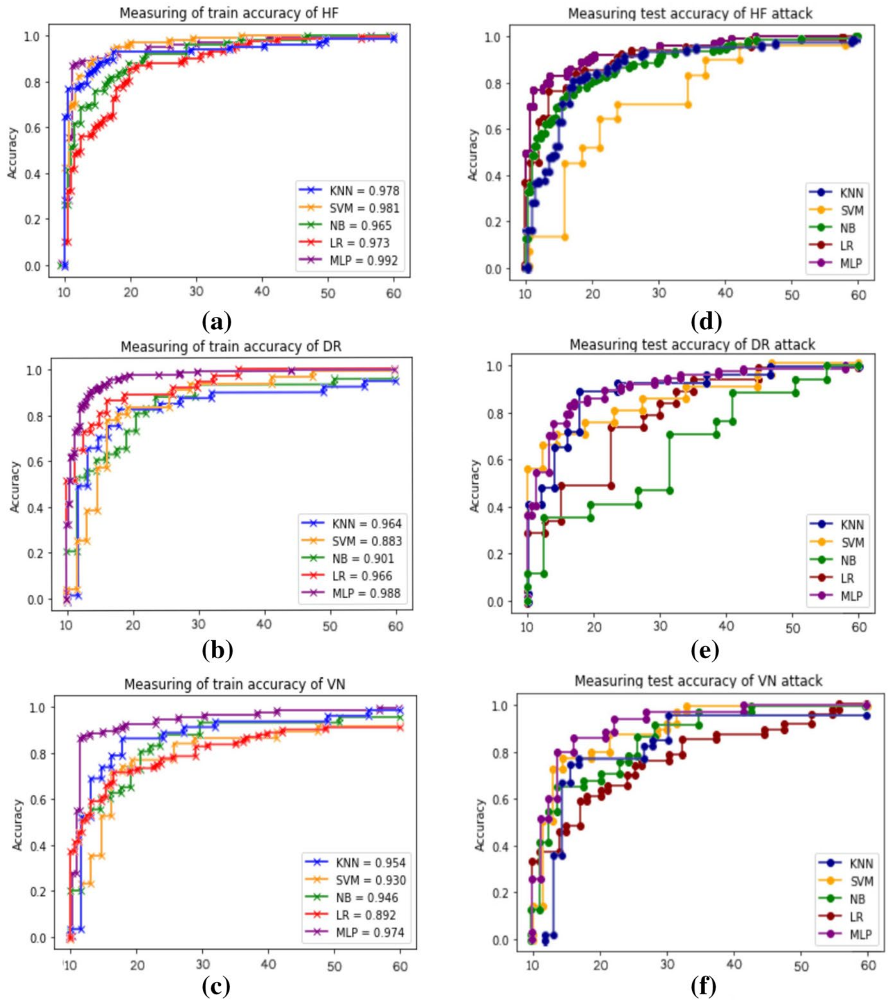

Fig. 8 Comparison of training and testing accuracy of our classifier with other techniques: training accuracy of HF (**a**), training accuracy of DR (**b**), training accuracy of VN (**c**), as well as the testing accuracy of HF (**d**), training accuracy of DR (**e**), and testing accuracy of VN (**f**)

4.3 Performance discussion

Upon the analysis methods in Sect. 4, the performance results show a speed and decrease in epoch time, which is reflected in the perfect performance of the model. Table 9 compares the performance metrics for our model to recent studies that generated IRAD dataset, and other studies that used the same dataset. DNN model obtained the best performance of HF attack through training the detection model using five features, the DR and VN attacks were trained using ten features, and the performance accuracy of DR and VN models is higher with 0.94 and 0.95 $F1$ score.

Table 9 Comparison of the quantitative measure of performance metrics with other Techniques

Rank	Method	HF			DR			VN		
		Precision	Recall	*F*1 score	Precision	Recall	*F*1 score	Precision	Recall	*F*1 score
4	*ML-RPL*	–	–	–	0.97	0.97	0.97	–	–	–
3	*DNN*	0.98	0.97	0.98	0.95	0.96	0.94	0.94	0.94	0.95
2	*GAN-C*	0.93	0.92	0.92	0.84	0.82	0.83	0.73	0.68	0.70
1	*DL-ESD*	0.96	0.98	0.97	0.98	0.98	0.96	0.95	0.96	0.97

Whereas, our model used nine features for each attack during training with uneven correlation level. ML-RPL used binary classification and multi-classification for a sample of IRAD dataset is DR attack. In the binary classification, the training accuracy reached 97.17% and 97.01% for testing accuracy, while in multi-classification used the same parameters with SoftMax as the activation function, the training and testing accuracy of the model obtained 96.59% and 96.39% for testing phase. GAN is used to detect any fake samples that could confuse the learning cycle of the detection model. The performance measuring is compared between GAN-C and an independent SVM classifier to select the proper model in IoT.

The training results showed slight improvement, although the proposed model evaluated the performance compared to one classifier. It took a lower number of epochs (about 50) to reach an accuracy of 91%. In contrast, our proposed classifier evaluated the performance accuracy with four algorithms. It leaves no doubt that our model is more efficient. In the same context, Table 10 states that the DL-ESD scheme has the highest accuracy compared with the DNN [24], the author of the IRAD dataset used in our study.

Almusaylim [10] proposed a security routing protocol (SRPL-RP) for RPL rank and version number attacks. The proposed protocol detects and isolates attacks and adds them to the blocklist. The detection is based on a comparison of the ranking mechanism. The analysis results indicate that the PDR packet delivery rate of (98.48%) and SRPL-RP achieved an accuracy rate of (99.92%) under routing attacks. A recent study [29] suggested a machine learning model consisting of three steps: data collection, feature extraction, and two classification methods. The decreased rank IRAD dataset has been used to train ML-RPL model for new features that have been added manually. MLRP indicates that the accuracy rate is up (97%). The authors depend on actual sensor code through the data generated in the simulation scenarios, and the performance accuracy reached 96%. CCN method [19] predicts suspicious traffic on IoT networks, and the authors generated an IoT dataset consisting of five datasets. Due to the lack of studies that use the IRAD dataset, we chose two subsets to compare with the used datasets in our research. The results indicate a relative decrease in the detection of version number motes; the detection accuracy rate in both HF and VN attacks reached 93.63%. In iIoT, [21] is based on Industrial IoT networks that detect hello flood,

Table 10 Comparison classification results of our model with state-of-the-art studies

Rank	Methods	DL type	Accuracy (%)	Loss (%)	Ref. no.
7	*GAN-C*	Unsupervised	91	9.08	[20]
6	*iIoT*	Unsupervised	92.00	7.35	[21]
5	*CNN*	Supervised	93.63	6.02	[19]
4	*DNN*	Supervised	96.53	4.11	[24]
3	*ML-RPL*	Supervised	97.01	3	[29]
2	*SRPL-RP*	Hybrid	98.30	1.70	[10]
1	*DL-ESD*	Supervised	98.85	2.5	–

version number, black hole, and sinkhole attacks. The performance accuracy among the Interval rate (200–1000 s) of hello flood indicates 92%, and version number reaches 93%, respectively.

GAN-C model [20] takes adversarial training into account and has created a generative adversarial network classifier (GAN-C) with support vector machine (SVM). The study adopts DL parallel learning, and the results show a relatively much lower level of training to achieve an appreciable detection accuracy of 91%. What gives our adopted mechanism a preference over the proposed mechanisms is that DL-ESD can find easy arithmetic solutions at a high rate of efficiency, as the detection accuracy has reached (98.85%), the precision rate of (97.50%), recall rate of (98.33%), and F1 score rate of (97.01%), model performance values are evidence of the model scalability.

5 Conclusion and future work

This study proves that deep learning techniques are more efficient in complex security issues in IoT security. A new scheme called DL-ESD has been performed to detect routing attacks early. The LDA proved a potential to maximize the distances between the mean classes (between classes) and reduce the distance between the mean of the same class (intraclass), which produced more distinct features, it was implemented with the MLP classification algorithm. At the same time, the data was normalized using min–max scaling, which eliminated the worst overfittings of fewer data points in training samples. The important features are based on the highest correlation level features. The introduced approach applies the binary classification method in lightweight deep learning techniques. It can classify the behavior as a normal node or routing attack as available in the processed dataset. Therefore, we observe a high enhancement in MLP classifier performance, it shows high accuracy in testing and training and a low runtime compared to other classifiers. The results of DL-ESD model performance also show better detection efficiency. This scheme requires firmware adjustment on IoT objects, and its computational complexity is still low. In future work, we plan to enhance the detection range using a better technique based on edge computing environment of widely comprehensive routing attacks in RPL protocol.

Acknowledgements This work is supported by National Natural Science Foundation of China under Grant Nos: 61572095 and 61877007.

Data availability Internet routing attacks data used in this study is available at link: https://www.github.com/iot-attacks/irad.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Chen S, Xu H, Liu D et al (2014) A vision of IoT: applications, challenges, and opportunities with China perspective. IEEE Internet Things J 1:349–359. https://doi.org/10.1109/JIOT.2014.2337336
2. Ye J, Cheng X, Zhu J et al (2018) A DDoS attack detection method based on SVM in software defined network. Secur Commun Netw. https://doi.org/10.1155/2018/9804061
3. Li Y, Zuo Y, Song H et al (2021) Deep learning in security of Internet of Things. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3106898
4. Liu D, Yan Z, Ding W et al (2019) A survey on secure data analytics in edge computing. IEEE Internet Things J 6:4946–4967. https://doi.org/10.1109/JIOT.2019.2897619
5. Mahmoud R, Yousuf T, Aloul F et al (2016) Internet of things (IoT) security: current status, challenges and prospective measures. In: 2015 10th Int Conf Internet Technol Secur Trans ICITST 2015, pp 336–41. https://doi.org/10.1109/ICITST.2015.7412116
6. Mazhar MS, Saleem Y, Almogren A et al (2022) Forensic analysis on Internet of Things (IoT) device using machine-to-machine (M2M) framework. Electron 11:1126. https://doi.org/10.3390/ELECTRONICS11071126
7. Chiang M, Zhang T (2016) Fog and IoT: an overview of research opportunities. IEEE Internet Things J 3:854–864. https://doi.org/10.1109/JIOT.2016.2584538
8. Srivastava A, Gupta BB, Tyagi A et al (2011) A recent survey on DDoS attacks and defense mechanisms. Commun Comput Inf Sci (CCIS) 203:570–580. https://doi.org/10.1007/978-3-642-24037-9_57
9. Chang TY, Hsieh CJ (2018) Detection and analysis of distributed denial-of-service in internet of things-employing artificial neural network and apache spark platform. Sens Mater 30:857–867. https://doi.org/10.18494/SAM.2018.1789
10. Almusaylim ZA, Jhanjhi NZ, Alhumam A (2020) Detection and mitigation of RPL rank and version number attacks in the internet of things: SRPL-RP. Sensors (Switzerland) 20:1–25. https://doi.org/10.3390/s20215997
11. Musaddiq A, Zikria YB, Zulqarnain et al (2020) Routing protocol for Low-Power and Lossy Networks for heterogeneous traffic network. EURASIP J Wirel Commun Netw 2020:1–23. https://doi.org/10.1186/S13638-020-1645-4/TABLES/11
12. Butun I, Osterberg P, Song H (2020) Security of the Internet of Things: vulnerabilities, attacks, and countermeasures. IEEE Commun Surv Tutorials 22:616–644. https://doi.org/10.1109/COMST.2019.2953364
13. Harbi Y, Aliouat Z, Refoufi A et al (2021) Recent security trends in internet of things: a comprehensive survey. IEEE Access 9:113292–113314. https://doi.org/10.1109/ACCESS.2021.3103725
14. Raoof A, Matrawy A, Lung CH (2019) Routing attacks and mitigation methods for RPL-based Internet of Things. IEEE Commun Surv Tutorials 21:1582–1606. https://doi.org/10.1109/COMST.2018.2885894
15. Kim J, Kim J, Thu HLT et al (2016) Long short term memory recurrent neural network classifier for intrusion detection. In: 2016 Int Conf Platf Technol Serv PlatCon 2016—Proc Published Online First: 19 April 2016. https://doi.org/10.1109/PLATCON.2016.7456805
16. Saeed A, Ahmadinia A, Javed A et al (2016) Intelligent intrusion detection in low-power IoTs. ACM Trans Internet Technol. https://doi.org/10.1145/2990499
17. Diro AA, Chilamkurti N (2018) Distributed attack detection scheme using deep learning approach for Internet of Things. Future Gener Comput Syst 82:761–768. https://doi.org/10.1016/j.future.2017.08.043
18. Samy A, Yu H, Zhang H (2020) Fog-based attack detection framework for Internet of Things using deep learning. IEEE Access 8:74571–74585. https://doi.org/10.1109/ACCESS.2020.2988854
19. Kamel SOM (2020) Mitigating the impact of IoT routing attacks on power consumption in IoT healthcare environment using convolutional neural network. IJCNIS. https://doi.org/10.5815/ijcnis.2020.04.02
20. Nayak S, Ahmed N, Misra S (2021) Deep learning-based reliable routing attack detection mechanism for industrial Internet of Things. Ad Hoc Netw 123:1570–8705. https://doi.org/10.1016/J.ADHOC.2021.102661
21. Qureshi KN, Rana SS, Ahmed A et al (2020) A novel and secure attacks detection framework for smart cities industrial internet of things. Sustain Cities Soc 61:102343. https://doi.org/10.1016/J.SCS.2020.102343

22. Qasem ZAH, Esmaiel H, Sun H et al (2019) Enhanced fully generalized spatial modulation for the internet of underwater things. Sensors (Switzerland) 19:1–16. https://doi.org/10.3390/s19071519
23. Thamilarasu G, Chawla S (2019) Towards deep-learning-driven intrusion detection for the internet of things. Sensors (Switzerland). https://doi.org/10.3390/s19091977
24. Yavuz FY, Ünal D, Gül E (2018) Deep learning for detection of routing attacks in the internet of things. Int J Comput Intell Syst 12:39–58. https://doi.org/10.2991/ijcis.2018.25905181
25. Seth AD, Biswas S, Dhar AK (2020) Detection and verification of decreased rank attack using round-trip times in RPL-based 6LoWPAN networks. In: Int Symp Adv Networks Telecommun Syst ANTS 2020, December 2020, pp 3–8. https://doi.org/10.1109/ANTS50601.2020.9342754
26. Sharma S, Kumar VV (2021) AIEMLA: artificial intelligence enabled machine learning approach for routing attacks on internet of things. J Supercomput. https://doi.org/10.1007/s11227-021-03833-1
27. Prakash PJ, Lalitha B, Prakash PJ et al (2022) Optimized ensemble classifier based network intrusion detection system for RPL based Internet of Things keywords Internet of Things · RPL based IoT · Intrusion detection system · Voting ensemble classifier · Feature selection. Wirel Pers Commun. https://doi.org/10.1007/s11277-022-09726-7
28. Sharma S, Verma VK (2021) Security explorations for routing attacks in low power networks on internet of things. J Supercomput 77:4778–4812. https://doi.org/10.1007/s11227-020-03471-z
29. Osman M, He J, Mahiuob F et al (2021) Artificial neural network model for decreased rank attack detection in RPL based on IoT networks. Int J Netw Secur. https://doi.org/10.6633/IJNS.202105
30. Manne VRJ, Sreekanth S (2022) Detection and mitigation of RPL routing attacks in Internet of Things. In: Proc 2022 9th Int Conf Comput Sustain Glob Dev INDIACom 2022 2022, pp 481–5. https://doi.org/10.23919/INDIACOM54597.2022.9763140
31. Zannone N, Alaa Al-Amiedy T, Anbar M et al (2022) A systematic literature review on machine and deep learning approaches for detecting attacks in RPL-based 6LoWPAN of Internet of Things. Sensors. https://doi.org/10.3390/s22093400
32. Palattella MR, Accettura N, Vilajosana X et al (2013) Standardized protocol stack for the internet of (important) things. IEEE Commun Surv Tutorials 15:1389–1406. https://doi.org/10.1109/SURV.2012.111412.00158
33. Agiollo A, Conti M, Member S et al (2021) DETONAR: detection of routing attacks in RPL-based IoT. IEEE Trans Netw Serv Manag 18:1178–1190
34. Mayzaud A, Badonnel R, Chrisment I (2017) A distributed monitoring strategy for detecting version number attacks in RPL-based networks. IEEE Trans Netw Serv Manag 14:472–486. https://doi.org/10.1109/TNSM.2017.2705290
35. Llns I (2021) A holistic framework for prediction of routing attacks. J Supercomput. https://doi.org/10.1007/s11227-021-03922-1
36. Sarhan M, Layeghy S, Moustafa N et al (2021) Feature extraction for machine learning-based intrusion detection in IoT networks. http://arxiv.org/abs/2108.12722
37. Moustafa N, Member S, Slay J et al (2017) Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. IEEE Trans Big Data. https://doi.org/10.1109/TBDATA.2017.2715166
38. Ghaleb B, Al-Dubai AY, Ekonomou E et al (2019) A survey of limitations and enhancements of the IPv6 routing protocol for low-power and lossy networks: a focus on core operations. IEEE Commun Surv Tutorials 21:1607–1635. https://doi.org/10.1109/COMST.2018.2874356
39. Roy D, Murty KSR, Mohan CK (2015) Feature selection using Deep Neural Networks. In: Proc Int Jt Conf Neural Networks 2015, 2015 September. https://doi.org/10.1109/IJCNN.2015.7280626
40. Bhuyan MH, Bhattacharyya DK, Kalita JK (2015) An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection R. Pattern Recognit Lett 51:1–7
41. Singh K, Dhindsa KS, Nehra D (2020) T-CAD: a threshold based collaborative DDoS attack detection in multiple autonomous systems. J Inf Secur Appl 51:102457. https://doi.org/10.1016/j.jisa.2020.102457
42. Ingre B, Yadav A (2015) Performance analysis of NSL-KDD dataset using ANN. In: Int Conf Signal Process Commun Eng Syst—Proc SPACES 2015, Assoc with IEEE 2015, pp 92–6. https://doi.org/10.1109/SPACES.2015.7058223

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Mohammed Albishari[1] · Mingchu Li[1] · Runfa Zhang[1] · Esmail Almosharea[1]

Mohammed Albishari
malbeshari@mail.dlut.edu.cn; mohmmdalbishari@gmail.com

Runfa Zhang
zhangrf@mail.dlut.edu.cn

Esmail Almosharea
es.mosharea@gmail.com

[1] School of Software Technology, Dalian University of Technology, Dalian 116620, Liaoning, China