An intrusion detection system against RPL-based routing attacks for IoT networks

Manjula Hebbaka Shivananjappa[1], Roopa Maidanahalli Seetharamaiah[2], Bharath Viswaraju Sai[3], Arunalatha Jakkanahally Siddegowda[1], Venugopal Kuppanna Rajuk[1]
[1]Department of Computer Science and Engineering, UVCE, Bangalore University Bengaluru, Bengaluru, India
[2]Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, India
[3]Textron, Bengaluru, India

Article Info

Article history:

Received Nov 7, 2023
Revised Dec 16, 2023
Accepted Feb 26, 2024

Keywords:

GNN
IDS
IoT
NIDS
RPL

ABSTRACT

The significant improvements in the internet, internet of things (IoT), communication, and cloud computing have created considerable challenges in providing security for data and devices. In IoT networks, routing protocol for low power and lossy networks (RPL) is a communication protocol that enables devices to exchange information and communicate with limited resources like low processing capabilities, less memory, and less energy. Unauthorized users can access RPL-based IoT networks through the internet, making these networks susceptible to routing attacks. Therefore, designing an intrusion detection system (IDS) is crucial to address attacks from IoT communication devices. In this paper, we proposed graph convolution networks (GCN) Conv, a graph neural network (GNN) method that captures a graph's edge and node features to identify routing attacks. The proposed system has experimented on the RADAR dataset, and experimental findings proved that our approach performs well compared to the state-of-the-art method concerning precision, F1-score, accuracy, and recall.

Corresponding Author:

Manjula Hebbaka Shivananjappa
Department of Computer Science and Engineering, UVCE, Bangalore University Bengaluru
Bengaluru, India
Email: manjulashekhar.2008@gmail.com

1. INTRODUCTION

Internet of things (IoT) is a mechanism that has gained much popularity and has brought enormous capabilities for ubiquitously intelligent connectivity and applications in many domains of human life. It has become a well-known technology in digital communication that connects more devices with the Internet and protocols to enable data transmission and communication between smart devices without requiring human interaction [1]. Intelligent nodes in IoT networks provide active and innovative life for humans by enabling actuating, sensing, and communication capabilities. IoT offers many applications, from simple appliances for smart homes to connected industries and complex intelligent grids. In IoT networks, a node can perform three actions: collecting, transmitting, and processing data. Memory-constrained, small, and less energy-consuming sensors are utilized in the data collection step to gather information from the physical environment in IoT networks. Wi-Fi, IEEE 802.15.4, ZigBee, radio frequency identification (RFID), and wire-based techniques with internet protocol (IP) are used as communication protocols for data transmission. Smart devices in IoT networks process the data to gather valuable data in the data processing stage and make intelligent decisions to send control messages after data collection. Routing protocol for low-power and lossy networks (RPL) routing protocol is used to communicate sensors and actuators in IoT communication environments [2]. Owing to the development and integration of IoT interconnected machines and applications, attackers can perform various routing attacks on IoT devices and against RPL. Various technical controls have been published in the literature for improving authentication, integrity, confidentiality, and access control mechanisms to achieve security for IoT networks.

However, with these techniques, attackers are still able to perform attacks. Designing a mechanism against attacks is challenging due to the number of devices in IoT applications and the enormous amounts of data generated from sensors. It is essential to utilize intrusion detection techniques to provide security for data communications in IoT. Intrusion detection systems (IDSs) are systems referred to as security mechanisms for monitoring behaviour and detecting malicious attacks in a system [3]. IDS should be able to investigate network packets at different layers of IoT networks by applying other security technologies with various protocol stacks [4]. IDSs will be deployed in IoT communication networks to monitor and inspect malicious packets to protect systems. These systems verify all incoming network data and find any sign of malicious packets or intrusion. The deployed security mechanism should take appropriate measures if it identifies the incoming network packet as a threat [5]. Intrusion detection mechanisms should be able to operate under different conditions like less battery backup, low processing capability, massive data processing, and fast response of IoT communication networks. The operations of an IDS can be categorized into three tiers. In the first level, it monitors the incoming network packets with the help of host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS). In the second level, IDS systems are analyzed by applying feature extraction techniques. The third level is the application of anomaly detection methods for detecting malicious traffic in IoT networks. In the subsequent sections, the contents are structured as follows: section 2 describes the background of RPL routing mechanism and attacks against RPL, section 3 provides an overview of established intrusion detection models for identifying routing attacks, section 4 elaborates on the methodology, section 5 delineates the experimental configuration and discoveries, and section 6 presents the conclusions pertaining to the proposed intrusion detection technique.

2. BACKGROUND

RPL is considered a routing mechanism for low-power and lossy networks (LLNs) implemented by the internet engineering task force-(IETF). For IoT networks, it is regarded as the de facto standard protocol. This protocol is designed to provide communications among IoT devices and satisfy constrained devices' requirements. Devices with restricted memory, less computing power, and low battery backup operated resources are included in LLNs [6]. The constrained devices in IoT networks often support only modest data speeds, have lossy connections, are frequently unstable, and have low packet delivery rates. The RPL routing technique has been suggested for various networking settings, including smart grid, urban routing, industrial automation, home automation, and building automation [7]. The RPL protocol is based on distance vector routing methodology, and it constructs a tree-based routing topology named destination oriented directed acyclic graph (DODAG). The graphs DODAG are constructed by utilizing objective function (OF) that uses different techniques to compute the best path between sensor nodes in communication networks [8]. The DODAG is a directed graph that consists of only one route from the leaf node to the root node without loops. Every node in the graph selects a parent node that forwards application packets. All nodes send a DODAG information object (DIO) to announce them as root nodes, find RPL instances, and learn configurations of DODAG. Any new node that is ready to connect to the network topology generates a DODAG information solicitation (DIS) request message and receives DAO Acknowledgment (DAO-ACK) confirming the join [9]. The DIS packet is used to request data from neighbour nodes, and the nodes use the DAO packet to modify the information of their parent nodes in the networks [10].

2.1. Attacks against RPL

In IoT technology, smart devices are interconnected and communicate via the internet. The RPL protocol provides all communications and connection associations between devices in IoT networks. However, current research indicates that RPL is the target of numerous topologies and cyberattacks. An intruder can alter the configuration information of DODAG packets, send previously received application packets, and change the parameters of packets [11].

The routing attacks associated with RPL protocol in IoT networks are categorized into three groups: (i) resource-based attacks, (ii) topology-based attacks, and (iii) traffic-based attacks, as depicted in Figure 1. They are described as follows:

(i) Resource-based: the attackers will target the resources such as bandwidth, memory, and processing capability of devices to disrupt and degrade the performance of IoT devices. Some of the resource-based attacks are local repair attack, version attack [12], DODAG inconsistency attack, increased rank attack [13], hello flooding attacks.

(ii) Topology-based: the intruders exploit vulnerabilities in network topologies and communication paths. Balckhole attacks [14], wormhole attack [15], sinkhole attack [16], route table falsification attacks, selective forwarding attacks [17], and worst parent attacks [18] are the types of toplogy-based attcaks.

(iii) Traffic-based: the attackers will modify the network traffic flow transferred between IoT devices to compromise the privacy and security of devices. Sybil and clone ID attacks [19] and replay attacks [20] are the examples of traffic-based attacks.

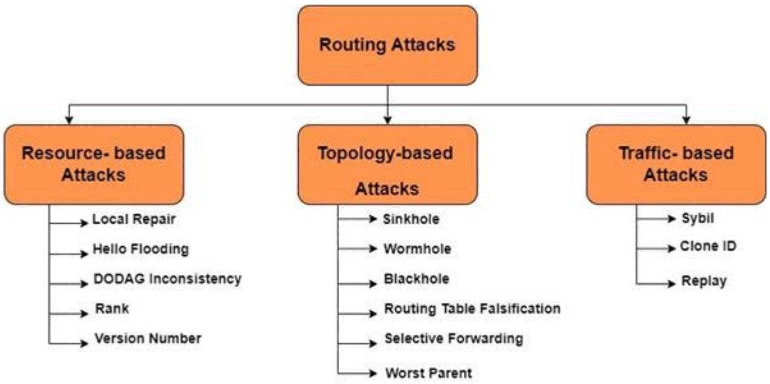


Figure 1. Attacks on RPL-based IoT networks

The key contributions of this proposed work are given below.

- We have designed an intrusion detection model based on GNN to identify routing threats.
- We have adopted the graph convolution networks (GCNConv) layer of graph convolution network on routing attacks dataset for RPL (RADAR) [21].
- The results are compared by evaluating the proposed graph-based intrusion detection model.

3. LITERATURE SURVEY

We provide a quick study of the existing IDS algorithms used in IoT networks to identify routing attacks. Osman *et al.* [22] have introduced a machine learning technique using the light gradient boosting machine (ML-LGBM)for version number threat detection. In IoT data communication, version number threats aim to maliciously increase the version number and bring inconsistency in the DODAG, forcing it to build the DODAG from the start. It depletes the network resources, which has an impact on the network's availability and quality of service (QoS). The work uses the Cooja simulator to simulate the attack. It employs two frameworks: gradient-based one side sampling and exclusive feature bundling (EFB). Precision, F1-score, and accuracy are utilized to evaluate the proposed technique. However, this technique addresses only version number attacks, and other datasets must be considered to analyze the model further. Verma and Ranga [23] have created a NIDS covering major RPL attacks such as selective forwarding, local repair, sybil, blackhole, clone ID, and hello flooding. The work uses the RPL-NIDDS17 dataset for evaluation that contains the packet information of the mentioned threats, created using the NetSim tool. Four different ensemble-based machine learning classifiers are utilized to detect routing attacks. Future work of the paper includes implementing and evaluating the proposed model on smart nodes and building a lightweight security solution for securing the IoT. Başol and Toklu [24] have presented deep learning frameworks, namely gated recurrent unit (GRU) and recurrent neural network (RNN) to identify hello flooding attacks in IoT communication networks. In this proposed model, the authors used Cooja Simulator with Contiki operating system to generate network traffic data. Experimental findings exhibit that the proposed deep learning framework attained more accuracy compared to support vector machine (SVM) and logistic regression (LR) classifiers. However, the proposed model needs to improve scalability issues by adding nodes in network topology. This research can be expanded in the future to take into account different kinds of other attacks. Farzaneh *et al.* [25] have presented an attack identification mechanism with fuzzy logic concept to recognize local repair attacks in RPL-based IoT networks. True positive rate (TPR) and false positive rate (FPR) were used to appraise the proposed intrusion detection scheme. However, this fuzzy logic intrusion detection approach may be used to identify other routing threats in IoT networks.

Yavuz *et al.* [26] aim to address routing attacks, namely hello flooding, version number attacks, and decreased rank attacks. In this proposed model, researchers have generated a dataset known as IoT routing attack dataset (IRAD), which comprises above mentioned attacks. Network-based intrusion detection models are implemented using deep learning and machine learning techniques. Precision, accuracy, and recall are utilized to measure the system's performance. However, this intrusion detection model addressed only three types of attacks. In the future, this system can be used to address various routing attacks and can be extended to consider more number features to detect multiple attacks. Diro and Chilamkurti [27] aim to identify routing attacks in IoT networks by implementing deep learning algorithms. NSL-KDD dataset was utilized for evaluating the proposed system. Detection rate, accuracy, and false alarm rate metrics are applied to measure the efficiency of the intrusion detection mechanism. In the future, the proposed deep learning intrusion detection scheme can be used for other data sets to detect cyber-attacks in IoT networks.

Sharma and Verma [28] have implemented a machine learning multiclass classification algorithm to locate rank and wormhole intrusions from IoT network connections. Researchers have generated a dataset with the help of the Cooja simulator with the Contki operating system. The precision, detection rate, and accuracy metrics are used for measuring the effectiveness of the intrusion detection method. However, the proposed dataset comprises of only rank, wormhole, and benign data. In the future, the proposed system may be used to consider more cyber-attacks. Zahra *et al.* [29] have used a machine learning algorithm with artificial neural networks (ANNs) to detect network threats, namely hello flooding, version, and decreased rank attacks. The proposed method employs precision, F1-score, and recall as assessment metrics. However in the future, the suggested model can be enhanced to use machine learning algorithms to address other threats in RPL-based network topologies. Table 1 summarizes the literature works with the dataset, methods used, and limitations.

Table 1. Related work and limitations

Author	Method	Dataset	Results	Limitations

Osman *et al.* [22]	Machine learning method to detect version number attack	Own dataset	Precision= 0.990%
			Recall=0.993%
			F1-score=0.993%
			Accuracy=99.60%
				Authors considered only version number attack.

Verma and Ranga [23]	ML classifiers to address routing attacks	RPL-NIDDS17	Accuracy=94.5% Receiver operating characteristic curve (ROC)=0.98%	Evaluated the performance of IDS.
Başol and Toklu [24]	Deep learning frameworks to detect hello flooding attcaks.	Own dataset	Accuracy=99.50%	Needs to improve on scalability issue.
Farzaneh *et al.* [25]	Fuzzy logic to address local repair attack.	Own dataset	True positive Rate=95.75% False positive rate=0.89%	Authors have considered only local repair attack routing attack.

Yavuz *et al. [*26]	NIDS to detect routing attacks.	Own dataset	ROC=95.6%
			F1-score=96.2
				Addressed only three types of attacks.

Diro and Chilamkurti [27]	Deep learning model to detect hello flooding attcaks	NSL-KDD	Accuracy=99.27%
			Recall=97.50%
				The proposed model can be extended to address other routing attacks.

Sharma and Verma [28]	ML methods to address routing attacks	Own dataset	Precision= 99.36%
			Recall=99.15%
			F1-score=99.26%
				The proposed dataset contains rank, wormhole, and benign data.

Zahra *et al.* [29]	ML methods to address routing attacks.	Own dataset	Accuracy=100%	Addressed only few attacks.
Proposed method	GCN based model to detect routing attacks.	RADAR dataset	Detection accuracy =98%	The proposed model has low detection rate for clone ID, hello flooding, rank, replay, selective forwarding, sybil, and sink hole attacks in contrast to state-of-the-art technique [21].

4. METHOD

In this section, we adopted a deep learning technique called GNNs to recognize routing attacks in IoT networks. A high-level overview of our suggested GNN model is shown in Figure 2. The network flow data set is initially used to create the graphs, which are then sent to the GCNConv model's training phase in the following step. By collecting the edge and node embeddings, network data is classified into attack and normal classes in the final step. We first introduce a summary of the dataset followed by a suggested intrusion detection model and results and discussion. The explanation of these steps is given in the following sub-sections.

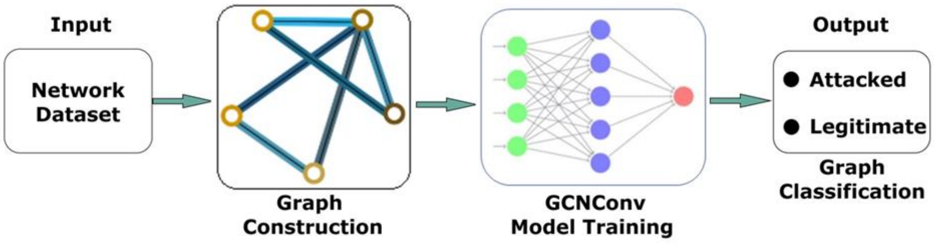


Figure 2. Proposed GCN-based intrusion detection system

4.1. Dataset

We use RADAR [21] to evaluate the model. Network simulator (NetSim) is utilized to generate the dataset. This RADAR dataset contains five simulations with 16 nodes for developing the simulations, with one border router for a single DODAG information. NetSim saves the packet trace file comprising packets communicated during the simulation. The RADAR dataset contains fourteen different attack scenarios: version, continuous sinkhole, legitimate, worst parent, clone ID, selective forward, blackhole, sinkhole, hello flooding, sybil attack, wormhole, replay, rank, DIS, and local repair attacks. This dataset contains various features for each transmitted packet in the network: application name, sender, destination, identities of communicating nodes, start time, size of the payload, network physical and datalink layers, and arrival time. It also contains the transmitter and receiver gateways and the following hop node address, rank value, and version of RPL packets. The experiment was conducted using Windows 10 with 64 GB RAM capacity and an Intel Xeon v3 processor.

4.2. Intrusion detection based on GNN

An IoT network contains a more significant number of heterogeneous systems communicating via the internet. Millions of users are potentially using IoT applications. Devices in IoT networks are vulnerable to cyber-attacks at any time. Therefore, intrusion detection methods are needed to secure devices and information in IoT network infrastructure.

The GCN algorithm is a well-known GNN that can be applied to graphs for image classification, link prediction, node classification, and graph classification [30] and take advantage of their structural data. The GCN Conv algorithm works on graph data by transforming the network data by representing a set of objects and the connections between them. A graph G can be designated as G=(V, E) where V is a set of vertices, E is the edges between them, and the graph can either be directed or undirected. Nodes denote the objects, and their relationship is represented as edges. An adjacency matrix can be used to describe the information in a graph. In computer networks, the devices are denoted as nodes, and the network data communications among them are denoted as edges in the graph. Node properties are called node features, and edge properties are known as edge features.

The GCN architecture combines node and edge features with multi-layer perception (MLP). The primary goal of GCN is to learn node embeddings by collecting the information from neighbors' nodes, updating the node states, and finally performing message passing to aggregate node features and connections and for learning structural and feature-based information that is used for predictions. The message-passing layer aggregates edge and node features through several iterations. This technique will be repeated to test model accuracy. For categorization, the neural network receives all pooled messages. Figure 3 demonstrates the GCN framework's end-to-end categorization process. A graph containing edge and node embeddings is used by GCN. This GCN model computes predictions using node and neighbor features. Fix aggregate methods to mix all the node's neighbors' information after gathering all the features. Each iteration samples node information and combines it with node embeddings from previous levels, then applies the activation function. This technique will be repeated to test model accuracy. Finally, the neural network classifies all pooled messages.

GCNs are used to handle graph-related data. The networks are designed to process graph-structured data by considering the relationship between nodes and edges, and they aggregate all information from neighboring nodes. GCNs effectively utilize graph information to perform node, edge, and graph-level predictions. The GCN model is found in many applications, such as classifying links, nodes, and graphs, identifying cluster nodes in community and recommendation systems, and graph attention networks.

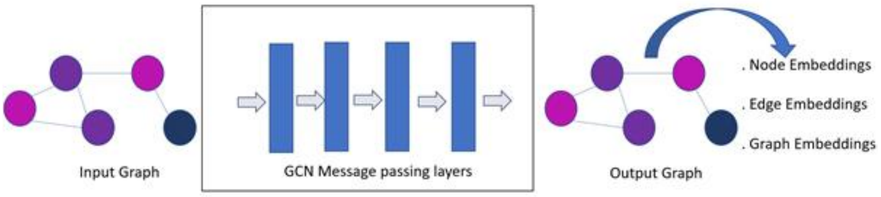


Figure 3. Working principle of graph convolution network

4.3. Algorithm: intrusion detection model aiming at routing attacks

The steps involved in detecting routing attacks using GCN Conv model is depicted algorithm 1:

- Phase 1: extraction of features using the proposed GCN Conv model: in this phase, we have used the data from the RADAR data set. From the simulation logs of the chosen data set, the features are selected to obtain a feature pool. The features were selected from the desired data set, as illustrated in Table 2. The features presented in Table 2 are helpful to detect various attacks. The resulting features are then given as input for constructing a graph.

Table 2. Selected features for proposed GCN model

Serial number	Name of the feature
F1	Application packets received.
F2	Transferred DIO packets.
F3	Version number.
F4	DAO packets sent.
F5	Transmitted DIS packets.
F6	DIO packets accepted.
F7	Transmitted application control packets.
F8	Application rate (sent and received).
F9	Rank.
F10	DAO packets received.
F11	NextHop IP.

- Phase 2: graph construction: the network graph is built using the features that were collected during phase 1. The recording of data communication in a network relies heavily on network flows. Nodes' actions are detailed in the network data, which includes things like the amount of data packets sent and received, the time it took for data to flow, and the source and destination nodes' data transfer rates. Network flow data is better represented using graphs. Graphs describe the communication and connection between nodes and edges. In our graph development process, the attributes chosen in phase 1 are utilized to build the network graph. Using a 10-second interval that represents a snapshot of the network that RPL nodes have constructed, we generate a graph for every simulation. The nodes and three edge features indicated in Table 2 are DAO packets, application packets, and DIO packets exchanged. At 10-second intervals during the simulation, GCNs and embedded nodes and edges are used to build graphs. Taking the attack time into account, labels are applied to each graph.
- Phase 3: binary classification using the GCN Conv algorithm: in this step, GCN Conv uses a two-hop neighborhood to calculate its own information as well as that of its neighbors. After that, the values that were obtained are passed to the mean aggregator function. Neural networks use the mean aggregator function to compile characteristics from nearby nodes in a graph. A 32-hidden-layer graph convolutional neural network was trained by us. The 32 neurons that make up the coupled neural network make up the categorization head. After the dropout layer, which has a probability of 0.5, the output of the graph convolutional layers is transmitted via the global average pooling layer. For regularization, this technique employs the rectified linear unit (ReLU) activation function. To alleviate the issue of the vanishing gradient, this linear function is employed. When the input is positive and the negative dimension is zero, it creates the output. One way to measure a model's accuracy in data classification is by looking at its binary cross-entropy. Our Adam optimizer is configured with a binary cross-entropy loss function and a learning rate of 0.01. The suggested method can detect assaults by processing values transmitted through learned GCN Conv layers after the model's parameters have been adjusted during training. Then, the embeddings of the nodes and edges are computed by these layers. After the data is transformed into graphs, the trained GCN Conv layers are applied to find the embeddings of the nodes and edges. The SoftMax layer converts all embeddings into class probabilities. The next step is to evaluate the classification procedure's efficacy by comparing actual class labels using performance measures.

Algorithm 1. Intrusion detection model aiming at routing attacks

```
Input: Network traffic data
Output: Binary-class labelling of attacks.
Begin
Phase 1: Extract the features from network traffic data
Phase 2: Graph creation
Create a graph G= (V, E) for each simulation from time window of 10 seconds
Phase 3: Binary classification using GCN Conv algorithm.
Input G (V, E) to 2 layers of GCN Conv
Outputs label for every G
End
```

5. EXPERIMENTAL RESULTS

The proposed work is carried out on a machine with a Tesla T4 GPU configuration and 6 GB of RAM. The Google Colab platform was used to train and test the model, and the code was developed in Python. Graph creation was done with PyTorch geometric. Four of the five simulations from the data set were utilized for training the model in the suggested work, and one simulation was utilized to evaluate the system. 80% of the data is utilized for training, while 20% of the data is utilized for testing as part of the experiment. The selection of hyperparameters has a compelling impact on the outcome of the suggested model. Hyperparameters settings used for the proposed intrusion detection model to detect routing attacks are shown in Table 3.

Table 3. Training parameters for intrusion detection task

Hyperparameters	Value
Number of layers	2
Loss function	Binary cross entropy
Learning rate	0.01
Epoch	50
Batch size	64
Neurons on each layer	32
Activation function	ReLU
Gradient optimizer	Adam
Dropout	0.5

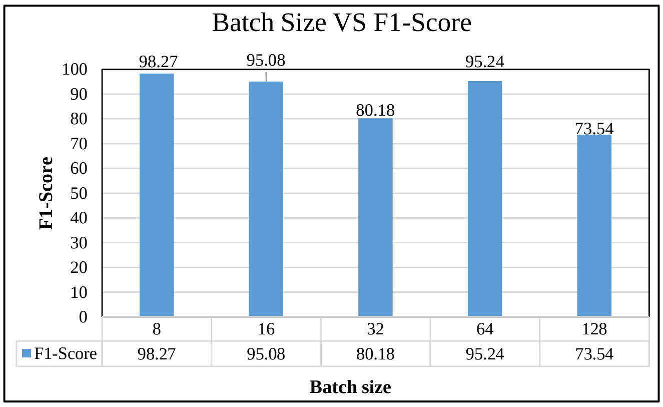
5.1. Results

In this section, we discuss the experimental findings of the suggested method to identify routing threats in IoT networks. The GCN-based intrusion detection model is evaluated using different performance metrics, namely F1-score, precision, recall, and accuracy. The results of the proposed intrusion detection technique are presented in Table 4 and Figure 4. The accuracy is the critical evaluation metric employed to compute system performance by dividing the successfully classified values by the total number of predicted values generated from the system. The precision metric predicts whether the given data belongs to a particular class (attack or normal data), and recall is used to measure the quality of the predictions by determining the ratio of positive values predicted correctly as positive to the total number of positive values. Combining the model’s recall and precision values, F1-score gives the harmonic mean to measure the system performance.

Table 4. Comparison results of the developed intrusion detection model

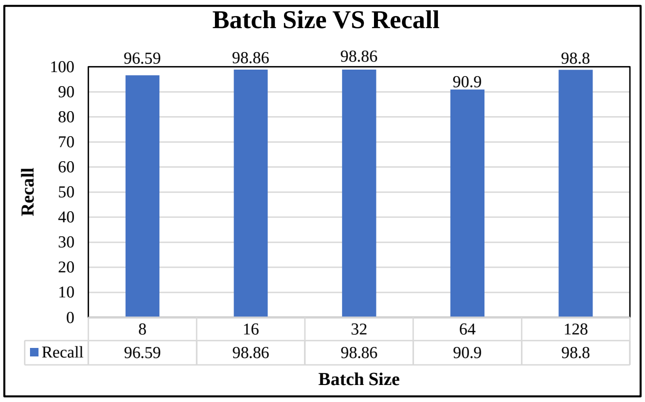
Method name	Attack type	Detection accuracy	Method name	Attack type	Detection accuracy
Proposed model	Blackhole	96	DETONAR [21]	Blackhole	60
	Clone ID	98		Clone ID	100
	Continuous sinkhole	81		Continuous sinkhole	60
	DIS	100		DIS	100
	Hello flood	98		Hello flood	100
	Local repair	98		Local repair	40
	Rank	92		Rank	100
	Replay	92		Replay	100
	Selective forwarding	98		Selective forwarding	100
	Sinkhole	72		Sinkhole	100
	Sybil	91		Sybil	100
	Version	98		Version	80
	Wormhole	82		Wormhole	80
	Worst parent	59		Worst parent	80

Table 3 compares the proposed intrusion detection method and the existing one. The proposed technique achieves a high detection rate for black holes, continuous sinkholes, local repair, version, and wormhole attacks. Experimental results show that the model is robust in detecting the various types of routing attacks. We evaluated the suggested intrusion detection model with distinct batch sizes like 8, 16, 32, 64, 128, and epochs of 50 and found optimal results with batch size 8. Figures 4 to 7 depict comparison graphs of the suggested model F1-score, recall, precision, and accuracy against distinct batch sizes for the blackhole attack using the RADAR dataset. In deep learning, the batch size represents the number of training examples used in one iteration, and it is used as a hyperparameter in the training phase.



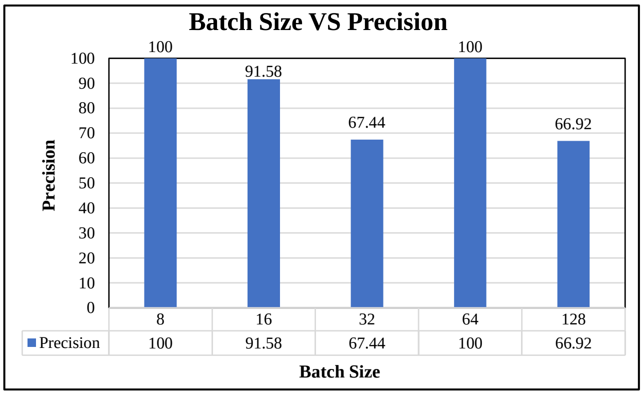
	8	16	32	64	128
■ F1-Score	98.27	95.08	80.18	95.24	73.54

Figure 4. F1-score results at different batch size aiming at detection of blackhole attack



	8	16	32	64	128
■ Recall	96.59	98.86	98.86	90.9	98.8

Figure 5. Recall results at different batch size aiming at detection of blackhole attack



	8	16	32	64	128
■ Precision	100	91.58	67.44	100	66.92

Figure 6. Precision results at different batch size aiming at detection of blackhole attack

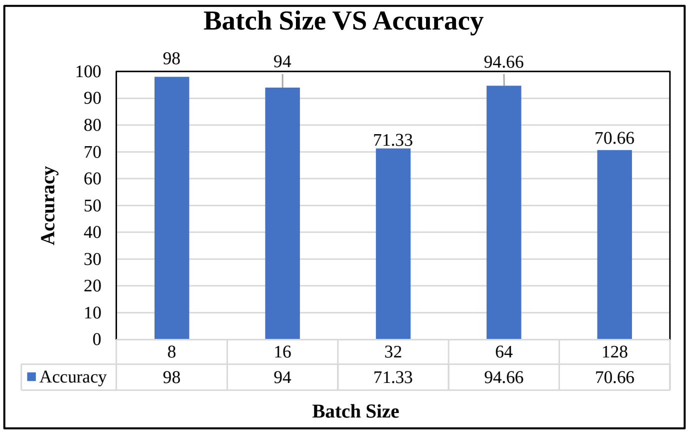


Figure 7. Accuracy results at different batch size aiming at detection of blackhole attack

Figures 8 and 9 represent a comparison graph of the proposed intrusion detection model's precision, recall, F1-score, and accuracy parameters for different epochs during training. In machine learning, it's crucial to determine the ideal number of epochs to use when assessing the proposed model during the training phase. An epoch is a hyperparameter that signifies the number of runs in which the training dataset is processed. The proposed model performs well in detecting routing attacks with chosen hyperparameters, viz, hidden size of 32, 50 epochs, batch size of 8, and learning rate of 0.01. The observed results indicate that the proposed method effectively detects routing threats. Figure 10 shows a schematic illustration of the results of the proposed graph-based intrusion detection model in terms of accuracy, F1-score, precision, and recall evaluation parameters.

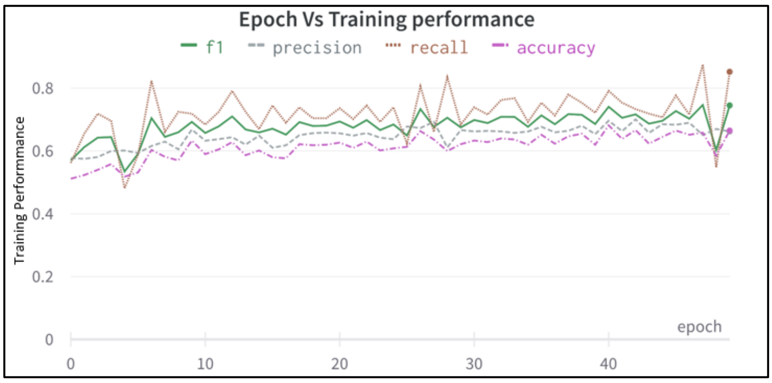


Figure 8. F1-score, precision, recall, and accuracy at different epochs for blackhole attack

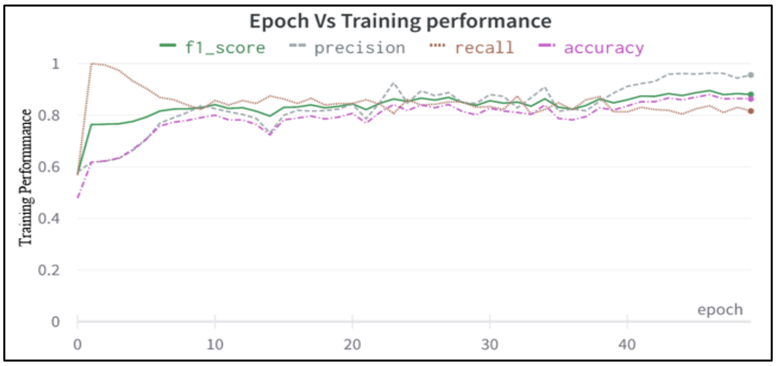
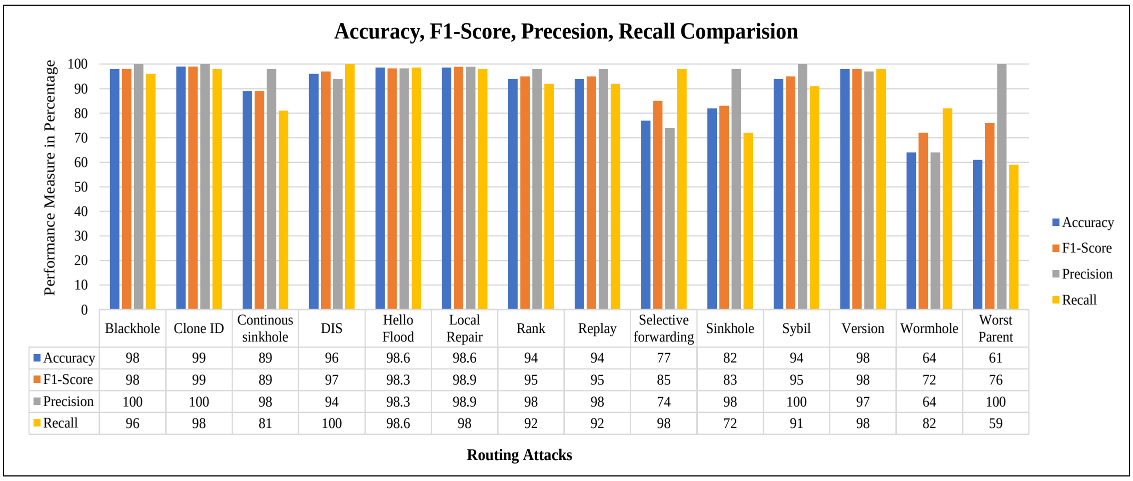


Figure 9. F1-score, precision, recall, and accuracy at different epochs for continuous sinkhole attack



	Blackhole	Clone ID	Continous sinkhole	DIS	Hello Flood	Local Repair	Rank	Replay	Selective forwarding	Sinkhole	Sybil	Version	Wormhole	Worst Parent
■ Accuracy	98	99	89	96	98.6	98.6	94	94	77	82	94	98	64	61
■ F1-Score	98	99	89	97	98.3	98.9	95	95	85	83	95	98	72	76
■ Precision	100	100	98	94	98.3	98.9	98	98	74	98	100	97	64	100
■ Recall	96	98	81	100	98.6	98	92	92	98	72	91	98	82	59

Figure 10. Intrusion detection results aiming at routing attacks in terms of accuracy, F1-score, precision, and recall

In the intrusion detection mechanism, the GCN-based model that we have proposed performs well in detecting routing attacks in contrast to the existing method [21]. Our method achieved a high detection rate for multiple types of routing attacks. However, the proposed model has a low detection rate for clone ID, hello flooding, rank, replay, selective forwarding, Sybil, and sinkhole attacks in contrast to state-of-the-art techniques. This proposed intrusion detection method can be extended to investigate other datasets that contain various cyber-attacks and can be integrated with deep learning models like GANs and graph attention networks.

6. CONCLUSION

In this study, a graph-based intrusion detection model for IoT networks is suggested to identify routing attacks. The proposed work involves the application of GCN Conv for intrusion detection by extracting the features of the network data set, constructing graphs, and, finally, graph-level predictions by considering the node and edge features. We have evaluated the proposed system on the RADAR dataset, and from the experimentation findings, it performs well in detecting routing attacks. The suggested method dramatically accomplished a high detection rate in noticing blackhole, continuous sinkhole, wormhole, version, worst parent, and local repair attacks in contrast to the state-of-the-art technique. The proposed method can be further extended to investigate other graph neural network models like graph SAGE and graph attention networks.

REFERENCES

[1] F. Hussain *et al.*, "A two-fold machine learning approach to prevent and detect iot botnet attacks," *IEEE Access*, vol. 9, pp. 163412–163430, 2021, doi: 10.1109/ACCESS.2021.3131014.

[2] A. Jamalipour and S. Murali, "A taxonomy of machine-learning-based intrusion detection systems for the internet of things: a survey," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9444–9466, Jun. 2022, doi: 10.1109/JIOT.2021.3126811.

[3] O. Sbai and M. Elboukhari, "Mobile Ad Hoc networks intrusion detection system against packet dropping attacks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 2, p. 819, May 2022, doi: 10.11591/ijeecs.v26.i2.pp819-825.

[4] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey," *Journal of Cloud Computing*, vol. 7, no. 1, 2018, doi: 10.1186/s13677-018-0123-6.

[5] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. P. C. Rodrigues, and Y. Park, "Intrusion detection protocols in wireless sensor networks integrated to internet of things deployment: survey and future challenges," *IEEE Access*, vol. 8, pp. 3343–3363, 2020, doi: 10.1109/ACCESS.2019.2962829.

[6] M. A. Saare, S. A. Lashari, A. Khalil, M. A. Al-Shareeda, and S. Manickam, "Review of routing protocol for low power and lossy network in the internet of things," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, no. 2, p. 865, Nov. 2023, doi: 10.11591/ijeecs.v32.i2.pp865-876.

[7] A. Musaddiq, Y. Bin Zikria, Zulqarnain, and S. W. Kim, "Routing protocol for low-power and lossy networks for heterogeneous traffic network," *Eurasip Journal on Wireless Communications and Networking*, vol. 2020, no. 1, p. 21, Dec. 2020, doi: 10.1186/s13638-020-1645-4.

[8] M. C. Belavagi and B. Muniyal, "Multiple intrusion detection in RPL based networks," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 467–476, 2020, doi: 10.11591/ijece.v10i1.pp467-476.

[9] A. J. H. Witwit and A. K. Idrees, "A comprehensive review for rpl routing protocol in low power and lossy networks," *Communications in Computer and Information Science*, vol. 938, pp. 50–66, 2018, doi: 10.1007/978-3-030-01653-1_4.

[10] V. R. J. Manne and S. Sreekanth, “Detection and mitigation of RPL routing attacks in internet of things,” in *Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development, INDIACom 2022*, Mar. 2022, pp. 481–485, doi: 10.23919/INDIACom54597.2022.9763140.

[11] A. Raoof, A. Matrawy, and C. H. Lung, “Routing attacks and mitigation methods for RPL-based internet of things,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1582–1606, 2019, doi: 10.1109/COMST.2018.2885894.

[12] A. Aris and S. F. Oktug, “Analysis of the RPL version number attack with multiple attackers,” *2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, Cyber SA 2020*, 2020, doi: 10.1109/CyberSA49311.2020.9139695.

[13] F. Zahra, N. Z. Jhanjhi, S. N. Brohi, N. A. Khan, M. Masud, and M. A. AlZain, “Rank and wormhole attack detection model for RPL-based internet of things using machine learning,” *Sensors*, vol. 22, no. 18, p. 6765, Sep. 2022, doi: 10.3390/s22186765.

[14] R. Sahay, G. Geethakumari, B. Mitra, and V. Thejas, “Exponential smoothing based approach for detection of blackhole attacks in IoT,” in *International Symposium on Advanced Networks and Telecommunication Systems, ANTS*, Dec. 2018, vol. 2018-December, pp. 1–6, doi: 10.1109/ANTS.2018.8710073.

[15] P. Perazzo, C. Vallati, D. Varano, G. Anastasi, and G. Dini, “Implementation of a wormhole attack against a rpl network: Challenges and effects,” in *2018 14th Annual Conference on Wireless On-Demand Network Systems and Services, WONS 2018 - Proceedings*, Feb. 2018, vol. 2018-January, pp. 95–102, doi: 10.23919/WONS.2018.8311669.

[16] C. Ioannou and V. Vassiliou, “Accurate detection of sinkhole attacks in iot networks using local agents,” in *2020 Mediterranean Communication and Computer Networking Conference, MedComNet 2020*, Jun. 2020, pp. 1–8, doi: 10.1109/MedComNet49392.2020.9191503.

[17] J. Jiang and Y. Liu, “Secure IoT routing: selective forwarding attacks and trust-based defenses in RPL network,” *arXiv preprint*, 2022, [Online]. Available: http://arxiv.org/abs/2201.06937.

[18] U. Kiran, “IDS to detect worst parent selection attack in RPL-based IoT network,” *2022 14th International Conference on COMmunication Systems and NETworkS, COMSNETS 2022*, pp. 769–773, 2022, doi: 10.1109/COMSNETS53615.2022.9668340.

[19] C. D. Morales-Molina *et al.*, “A dense neural network approach for detecting clone id attacks on the rpl protocol of the iot,” *Sensors*, vol. 21, no. 9, p. 3173, May 2021, doi: 10.3390/s21093173.

[20] A. Mayzaud, R. Badonnel, and I. Chrisment, “A taxonomy of attacks in RPL-based internet of things,” *International Journal of Network Security*, vol. 18, no. 3, pp. 459–473, 2016, doi: 10.6633/IJNS.201605.18(3).07.

[21] A. Agiollo, M. Conti, P. Kaliyar, T. N. Lin, and L. Pajola, “DETONAR: detection of routing attacks in RPL-based IoT,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1178–1190, 2021, doi: 10.1109/TNSM.2021.3075496.

[22] M. Osman, J. He, F. M. M. Mokbal, N. Zhu, and S. Qureshi, “ML-LGBM: a machine learning model based on light gradient boosting machine for the detection of version number attacks in RPL-based networks,” *IEEE Access*, vol. 9, pp. 83654–83665, 2021, doi: 10.1109/ACCESS.2021.3087175.

[23] A. Verma and V. Ranga, “ELNIDS: ensemble learning based network intrusion detection system for RPL based internet of things,” in *Proceedings - 2019 4th International Conference on Internet of Things: Smart Innovation and Usages, IoT-SIU 2019*, Apr. 2019, pp. 1–6, doi: 10.1109/IoT-SIU.2019.8777504.

[24] Y. Başol and S. Toklu, “A deep learning-based seed classification with mobile application,” *Turkish Journal of Mathematics and Computer Science*, vol. 13, no. 1, pp. 192–203, Jun. 2021, doi: 10.47000/tjmcs.897631.

[25] B. Farzaneh, M. Koosha, E. Boochanpour, and E. Alizadeh, “A new method for intrusion detection on RPL routing protocol using fuzzy logic,” in *2020 6th International Conference on Web Research, ICWR 2020*, Apr. 2020, pp. 245–250, doi: 10.1109/ICWR49608.2020.9122278.

[26] F. Y. Yavuz, D. Ünal, and E. Gül, “Deep learning for detection of routing attacks in the internet of things,” *International Journal of Computational Intelligence Systems*, vol. 12, no. 1, pp. 39–58, 2018, doi: 10.2991/ijcis.2018.25905181.

[27] A. A. Diro and N. Chilamkurti, “Distributed attack detection scheme using deep learning approach for internet of things,” *Future Generation Computer Systems*, vol. 82, pp. 761–768, May 2018, doi: 10.1016/j.future.2017.08.043.

[28] S. Sharma and V. K. Verma, “AIEMLA: artificial intelligence enabled machine learning approach for routing attacks on internet of things,” *Journal of Supercomputing*, vol. 77, no. 12, pp. 13757–13787, Dec. 2021, doi: 10.1007/s11227-021-03833-1.

[29] F. T. Zahra, N. Z. Jhanjhi, S. N. Brohi, and N. A. Malik, “Proposing a rank and wormhole attack detection framework using machine learning,” in *MACS 2019 - 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics, Proceedings*, Dec. 2019, pp. 1–9, doi: 10.1109/MACS48846.2019.9024821.

[30] J. Zhou *et al.*, “Graph neural networks: a review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020, doi: 10.1016/j.aiopen.2021.01.001.

BIOGRAPHIES OF AUTHORS

Manjula Hebbaka Shivananjappa is a full time Research Scholar in the Department of Computer Science and Engineering at University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India. She received her Master’s degree in Computer Networks and Engineering from RVCE, Bangalore, India. Her research interests are in the field of IoT, cyber security, and computer networks. She can be contacted at email: manjulashekhar.2008@gmail.com.

Roopa Maidanahalli Seetharamaiah is currently working as an Associate Professor in the Department of Computer Science and Engineering at Nitte Meenakshi Institute of Technology, Bengaluru, India. She completed her full-time Ph.D. in Computer Science and Engineering at University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India. She received her Master's in Information Technology from Bangalore University, India. She has over 25 research publications in international journals and conference proceedings. Her research interests are data mining, deep learning, multivariate time series analysis, cyberbullying, the IoT, and computer networks. She can be contacted at email: roopams22@gmail.com.

Bharath Viswaraju Sai has completed his Master of Technology in Software Engineering at University Visvesvaraya College of Engineering (UVCE). He is currently working as Security Engineer in an Aerospace and Defense company. He has an industry experience of about four years and his work specializes on threat detection and response. He is very passionate in Cybersecurity and has worked in various fields of Cybersecurity including EdTech startups, Aerospace and defense, and Finance. He has completed various certifications like CEH, ECIH, and CRTP. He can be contacted at email: bharath.vs@outlook.com.

Arunalatha Jakkanahally Siddegowda is currently the Professor in the Department of Computer Science and Engineering at University of Visvesvaraya College of Engineering, Bangalore University, Bangalore, India. She obtained her Bachelor of Engineering from P E S College of Engineering, Mandya, in Computer Science and Engineering, from Mysore University and she received both her Master degree and Ph.D. in Computer Science and Engineering from University Visvesvaraya College of Engineering, Bangalore University, Bangalore. She has over 32 research publications in international journals and conference proceedings. Her research interest is in the area of biometrics, image processing, big data, IoT, and cyber security. She can be contacted at email: arunajs99@gmail.com.

Venugopal Kuppanna Rajuk is the former Vice-Chancellor of Bangalore University. He served UVCE and Bangalore University for over the last four decades. He has eleven degrees, with Master of Engineering in Computer Science and Engineering (CSE) from IISc Bangalore including two Ph.D., one in Economics from Bangalore University and another in CSE from IITM, Chennai. He has authored and edited 86 books published more than 1,200 papers in international conferences and journals and has 40 patents to his credit. He has awarded Ph.D. to 30 students and supervised more than 700 Post Graduate dissertations in CSE. He received IEEE Fellow and ACM Distinguished Educator award from USA for his outstanding contributions to CSE. He can be contacted at email: venugopalkr@gmail.com.