Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 9, No. 6, December, 2025

FedBHAD: Energy-Efficient Federated Learning for Black Hole Attack Detection in RPL-Based Low-Power IoT Networks

Senthilkumar Mathi [1*], Gudivada Rohan Lal [1], Lokesh Chowdary Madala [1], Karri Ammi Reddy [1], Putta Jagadhabhiram [1], Ganesh Neelakanta Iyer [2]

[1] *Department of Computer Science and Engineering, Amrita School of Computing, Coimbatore, Amrita Vishwa Vidyapeetham, India.*

[2] *Department of Computer Science, School of Computing, National University of Singapore, Singapore.*

Abstract

The internet of things is a network of connected devices that share and send information over the internet, frequently in resource-constrained situations. These are often built using the routing protocol for low-power and lossy networks (RPL), face significant security problems because of their limited computing power, and have energy constraints. The objective of this study is to design an efficient and lightweight mechanism for detecting black hole attacks on RPL-based internet of things networks. The proposed framework presents a distributed collaborative learning framework to reduce the processing load on central nodes while enhancing real-time threat detection. The novelty of the present work lies in integrating distributed learning with feature-based anomaly detection tailored for RPL environments, thereby improving IoT network security while reducing communication and energy overhead. A customized data retrieval algorithm is developed with the Cooja simulator's configuration and extracts essential network parameters, including rank, expected transmission count, power consumption, forward count, and reception count. The analysis of this dataset allows the detection of black hole attacks. The research analysis indicates that the proposed framework achieves 99.6% detection accuracy, surpassing existing machine learning and deep learning techniques and offering enhanced security, reduced overhead, and lower computational needs.

Keywords:

Black Hole Attack;
Decentralized Training;
Federated Learning;
Internet of Things;
Low-Power and Lossy Networks;
Routing Protocol.

Article History:

Received:	30	August	2025
Revised:	17	November	2025
Accepted:	21	November	2025
Published:	01	December	2025

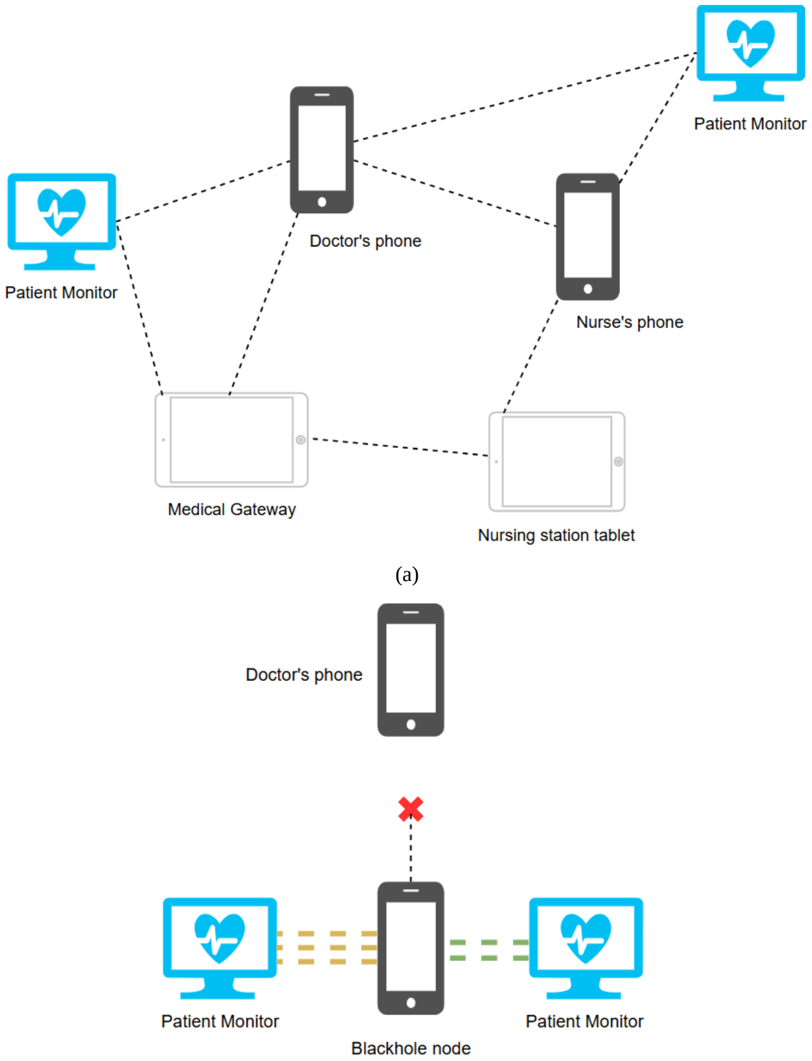
1- Introduction

Recent technological advancements such as faster network speeds, smaller transistor sizes, and high-capacity batteries using silicon–carbon technology have enabled electronic devices to connect to the Internet, communicate, and share data through the Internet of Things (IoT) [1]. The IoT provides the remote control of different devices, paving the way for the development of smarter cities, homes, farms, and industries. One of the fundamental technologies supporting the IoT is the Mobile Ad Hoc Network (MANET), characterized by self-configuration, decentralization, and multi-hop communication across nodes [2]. Low Power and Lossy Networks (LLNs) are designed for devices with unstable communication, limited bandwidth, and severe energy constraints [3]. These networks prioritize energy efficiency and are widely used in healthcare systems, smart farming, and other applications for real-time data collection. The Routing Protocol for LLNs (RPL) is a widely adopted IPv6-based protocol that constructs a rank-oriented network topology to facilitate efficient data transmission in challenging environments. However, RPL of LLNs remains vulnerable to routing-based exploits due to its dependence on rank advertisements [4].

[*] **CONTACT**: m_senthil@cb.amrita.edu

DOI: http://dx.doi.org/10.28991/ESJ-2025-09-06-01

The node with a black hole attack in LLNs is a compromised one that falsely advertises itself as the optimal route and subsequently drops all incoming packets, severely disrupting data transmission and compromising network integrity [5]. Figure 1-a shows a healthcare system using LLNs, where patient monitors, medical gateways, and mobile devices communicate through RPL-based routing. Figure 1-b depicts a black hole attack, where a malicious node intercepts and discards data packets, halting communication. Such attacks are among the most disruptive security threats in LLNs, frequently occurring in IoT, wireless sensor, and smart environment applications [6]. The detection of this behavior is difficult because IoT nodes operate with minimal power, memory, and processing capacity.



(a)

(b)

Figure 1. (a). Healthcare system scenario using LLNs, (b). The scenario involves a black hole attack

Traditional security methods depend on centralized Intrusion Detection Systems (IDS) that analyze large volumes of data externally. These methods require high computational resources, require substantial communication overhead, and may compromise privacy due to continuous data transmission across the network [7]. Researchers have proposed various Machine Learning (ML) and Deep Learning (DL) techniques to reduce security threats in IoT networks [8]; however, such models often necessitate significant computational power [9], rely on centralized models [6], and lack methods to address inherent vulnerabilities in RPL-based LLNs [10].

Many studies have focused on enhancing intrusion detection in RPL-based LLNs. Mayzaud et al. (2016) [4] investigated a comprehensive taxonomy of RPL attacks, emphasizing the ongoing challenge of detecting black hole and rank attacks due to the dynamic and adaptive topology of the protocol. Subsequent methods [6, 9] enhanced the detection accuracy in routing misbehavior; however, these methods depend heavily on centralized models and frequent control message exchanges, resulting in increased energy consumption. Recent investigations, including Krari et al. (2025) [7], presented a DL-based IDS for IoT networks. While these investigations achieve high detection accuracy, they demand extensive data aggregation and fail to preserve node-level privacy. Consequently, a significant research gap remains in designing a lightweight, privacy-preserving, and distributed learning framework capable of accurately detecting black hole attacks in LLNs without the overhead of resource-constrained nodes. To the best of current knowledge, no prior study has applied Federated Learning (FL) using topology-derived features for detecting black hole attacks in RPL-based LLNs. Thus, the present study proposes a collaborative distributed learning (FL) framework that corresponds with the resource-constrained and distributed characteristics of LLNs. Each node trains a local model and only sends its learned parameters to an aggregator instead of sending raw data to a central node. This distributed approach reduces communication overhead, preserves data privacy, and improves the detection of malicious behavior in LLNs.

The present research is theoretically based on the concept of distributed intelligence and collaborative learning in limited network settings. It is built upon FL theory, which emphasizes decentralized model training to enhance data privacy and reduce communication overhead. From a network perspective, the approach leverages RPL topology formation theory, wherein rank and parent–child relationships define hierarchical routing structures that can be mathematically modeled to identify abnormal rank deviations typical of black hole attacks. Integrating these theories allows the approach to correlate topological anomalies with learned behavioral patterns, forming a hybrid theoretical foundation that combines statistical detection with rank-based metrics and DL inference (sequential models). This theoretical integration ensures that the proposed model maintains scalability, accuracy, and detection precision under the resource and power constraints inherent in LLNs.

The significant contributions to the proposed research are as follows:

- New data collection method: Features are extracted directly from the constructed RPL network topology rather than through traditional packet-capturing approaches, enabling more accurate and real-time data representation.
- FL integration: Local nodes train on their own, while a central node collects the results, improving both detection accuracy and computational efficiency.
- FedBHAD framework: The proposed Federated Learning for Black Hole Attack Detection (FedBHAD) algorithm integrates local node training with centralized aggregation to achieve precise black hole attack prediction in LLNs.

The rest of the paper is organized as follows. Section 2 discusses a detailed literature review of previous works. Section 3 provides the details of the proposed methodology, which includes a customized algorithm for data extraction. Section 4 details the topology created and the experimental setup used to prepare the dataset. Section 5 explains performance analysis and evaluates how effectively this approach detects black hole attacks. Finally, section 6 sums up the present paper and suggestions for future research to enhance the security of IoT networks.

2- Related Works

A comparison of various ways to detect attacks in IoT networks is presented in Airehrour et al. [11] and Karthikeyan & Revathi [12], including techniques like DL-based Artificial Neural Network (ANN), blockchain, trust-based security protocols, and ML for identifying unusual activity. These methods are recommended to address security issues like jamming, worm infections, DDoS attacks, and intrusion detection, using measurements such as data confidentiality rate, attack detection rate, and attack detection duration. The study finds that while cryptography and DL improve detection accuracy, they require a lot of computing power. These methods are recommended for dealing with security threats like worm infections and DDoS attacks. They use parameters such as the attack detection rate and the attack detection duration. This study indicates that DL techniques increase the detection of accuracy, but they incur a heavy computational cost. This investigation makes it possible to identify novel methods that increase attack detection accuracy with minimum computational cost.

According to the investigation by Shukla et al. [13], hostile nodes that drop packets instead of forwarding them could cause black hole attacks, which are vulnerabilities in MANETs. It suggests a new method to identify and reduce these vulnerabilities using mutation-based ANN and cluster-based Artificial Bee Colony (ABC) optimization. The experimental results indicate that the suggested protocol increases network performance by improving throughput and packet delivery ratio (PDR) compared to the existing techniques. To improve network resistance against such threats, they recommend more investigation with node selection algorithms. They suggested the Optimized Link State Routing (OLSR) protocol in MANETs for safer and more effective communication by detecting vulnerabilities caused by black hole attacks. The aim is to develop an energy-aware method for selecting Multipoint Relays (MPRs), which increases network lifetime with less routing overhead. In addition, using ML methods with the Deep Belief Network (DBN)-OLSR protocol aims to detect black hole attacks. By selecting the MPRs based on real-time indicators like hop count and residual energy, data transmission is optimized [14].

Traditional IDS often uses ML methods such as Support Vector Machines (SVM), Decision Tree (DT), K-Nearest Neighbors (KNN), and Random Forest (RF). Even though these models improve the detection rate, they provide false positives and require significant computational cost. To improve their efficiency, different combined methods have been suggested, like particle swarm optimization IDS and SVM; however, these methods struggle with choosing the best features, take longer to identify solutions, and have slower processing times. This work introduces a new method that uses the Whale Optimization Algorithm (WOA) and the ABC algorithm to select important features, along with a new

Convolutional Neural Network (CNN) to identify intrusions. The WOA-ABC hybrid model improves convergence speed by setting ABC's final cost as WOA's initial cost, which reduces computational overhead [15]. The specifically designed CNN architecture enhances detection accuracy while maintaining low execution time. This hybrid system achieved higher detection accuracy, a lower false-positive rate, and a 76.54% reduction in execution time.

The performance analysis of ML algorithms describes how to detect and mitigate black hole and flooding attacks. The detection uses ML algorithms that are trained on the WSN-DS dataset, which contains a network of 100 nodes divided into 5 groups, and different training-to-testing ratios (from 90:10 to 10:90) are used to determine how well the models work with varying amounts of data. A total of 9 models based on their execution time and detection accuracy are tested and evaluated [16]. The results indicate that AdaBoost achieved the highest accuracy (97.97%) for black hole attacks. The mitigation process requires eliminating the compromised node and selecting a secured path to restore communication among nodes. ML can only detect attacks if the pattern matches the dataset.

The hybrid IDS for RPL serves as an example of how ML and DL approaches can enhance IDS. For this work, they selected the ROUT-4-2023 dataset, which consists of nearly 1.6 million rows of data belonging to four major types of RPL attacks, including black hole [17]. Of the multiple models trained for intrusion detection, the Random Forest model achieved the highest accuracy of 99%, and the transformer-based DL model obtained the highest F1 score of 97% with just 16.8 minutes of training time over five epochs. These results suggest that the use of ML and DL can lead to effective intrusion detection. The authors suggested a hybrid IDS that mixes different methods like setting limits, matching known patterns, and monitoring signals to better detect various RPL attacks, rather than just looking at other studies or starting from scratch. By integrating multiple techniques, the proposed IDS improves detection accuracy while maintaining minimal computational overhead. The IDS uses centralized and distributed detection strategies, which ensures coverage of both localized and network-wide threats. This system is tested in the Cooja simulator and confirms its effectiveness against common RPL attacks, including black hole, hello flood, and insider attacks [18]. The system uses a UDP-based heartbeat protocol for enhancing detection of black hole attacks. This hybrid IDS was evaluated to achieve high detection accuracy while making the CPU 2% less overhead and with negligible power consumption increase (<0.5%).

The work in Choukri et al. [19] investigated an efficient DL platform to detect black hole attacks in RPL networks. It uses a Deep Neural Network (DNN) to monitor network traffic, gathering essential details such as the number of Destination Information Object (DIO) packets and energy levels. The model sets certain limits for locating intrusions, achieving a high detection accuracy of 98.70% and an F1-score of 98, to bring out the difference between normal and harmful traffic behavior. The DNN performance is also being compared with other traditional learners, like random forest, with which higher precision and recall rates of 98% and 99% are achieved. This result indicates that DNN is better at handling complex patterns and adjusting to changes in IoT networks, which helps in detecting routing attacks and improving security in secure areas. The Scale-Hybrid-IDS-AlertNet framework, utilizing DNN for cyberattack detection, has illustrated remarkable accomplishments. Model accuracy ranges from 63.2% to 91.4% across different thresholds, demonstrating the flexibility available to adapt to various network scenarios. The data normalizing techniques, such as the min-max scalar and z-score, were also instrumental in increasing the performance of the model [20].

The model in Ahmadi & Javidan [21] suggested a Long Short-Term Memory (LSTM)-based trust mechanism to identify RPL attacks. The Packet Forwarding Ratio (PFR) is extracted in sliding windows and utilized to forecast expected behavior through an LSTM (window size = 3, 4 hidden units). The differences between the forecasted and actual PFR are calculated using a hybrid Euclidean-Magnitude metric to obtain trust scores. Using DETONAR datasets, the system is tested for black holes and selective forwarding attacks, successfully identifying harmful nodes at certain times when the attacks started (like at 500 seconds and 666 seconds). It showed significant spikes in deviation reaching up to 1.0 and steep reductions in trust levels, while honest nodes maintained static scores, demonstrating high detection accuracy for the model.

The methods used to detect black holes in RPL-based IoT networks are filled with shortcomings in the form of increased communication overhead, non-global coverage, and limited usage. In contrast to these techniques, INSULATE in Prajisha & Vasudevan [22] uses an ANN classifier to achieve high accuracy and an optimal detection rate of 99.23%. INSULATE also outshines other methods with a higher PDR of up to 97.8% and an end-to-end delay lowered by 0.21 s - 0.7 s. INSULATE also applies Dempster-Shafer theory to enhance decision-making over voting based on majority.

The work (ELG-IDS) in Osman et al. [23] uses a genetic algorithm to pick out important features and choose the best ones for finding RPL attacks in IoT networks. This method uses ensemble techniques such as stacking and extreme parameter optimization for higher accuracy. It detects version numbers and DIS flooding attacks with remarkable accuracy rates of 99.18%, 99.38%, and 99.66%, respectively, with an overall accuracy of 97.90% in multi-classification mode.

The ABGF-AODV approach, described in Gurung & Mankotia [24], adds new checks to AODV to stop black hole and flooding attacks in MANET. It ensures minimal overhead while differentiating genuine nodes from adversarial ones by using specialized route request and reply sequences. The results show that ABGF-AODV reliably delivers up to 92% of packets, even when there are many attacks, and it performs better than regular AODV. The work in Reshi et al. [25] investigated the consequences of black hole attacks on IoT networks, analyzing their disruptive impact on overall performance. It achieves a PDR of 98.21% for a network under black hole infiltration. The solution of this work helps bring back the performance of the network to normal levels by dealing with harmful node actions using a mix of checking routes again and finding new paths, showing it could work well in real-life IoT situations.

In the earlier studies, a common idea is the increasing need for using both ML and DL, or simpler mixed defenses, to tackle serious threats like black holes and flooding in wireless sensor networks, mobile ad hoc networks, and RPL-based IoT. The suggestions include improved classifiers, FL, and mixed trust or heartbeat detection methods. Collectively, they consistently report improved detection accuracy, reduced latency, and minimal overhead, signs that smart intrusion detection is essential for real-world constrained networks. By bringing together different methods, better detection techniques, and new ways to handle problems, these studies show that using smart design is a key to strong and flexible security for RPL based IoT networks. Table 1 lists the summarization of the related works.

Table 1. Overview of related works in black hole attack detection

Related works	Dataset used	Attack-focused	Methodology
Shukla et al. [13]	Simulated data in NS2 with parameters for MANET nodes	Black hole	Cluster-based ABC optimization and mutation-based neural network
Shahid et al. [17]	ROUT-4-2023 dataset	Black hole, flooding, decreased rank	ML and transformers for intrusion detection
Garcia et al. [18]	Contiki-NG and Cooja simulations	Black hole, hello flood	Hybrid IDS with heartbeat protocols and signature matching
Karthikeyan & Revathi [12]	-	Black hole, wormhole	Multiple ML models and bio-inspired algorithms for anomaly detection
Shafi & Venkata Ratnam [14]	Simulated MANET data with NS2	Black hole	DBN-based OLSR protocol with energy-aware MPR selection
Hussain et al. [15]	NSL-KDD dataset	General intrusions (including black hole)	Hybrid whale optimization-ABC with CNN
Choukri et al. [19]	Cooja simulations for RPL scenarios	Black hole	DNNs for network traffic analysis and attack detection
Abdiyeva et al. [20]	CICIDS2019 dataset	Black hole, rushing, flooding, wormhole	DNNs for intrusion detection
Kurtkoti et al. [16]	WSN-DS dataset generated with 100 nodes and 5 clusters	Black hole, flooding	Comparative analysis of ML models (Adaboost, Random Forest, etc.) and mitigation strategies
Ahmadi and Javidan [21]	DETONAR dataset, black hole & selective forwarding scenarios	Black hole, selective forwarding	LSTM-based trust approach: sliding windows to predict PFR and generate trust scores
Prajisha & Vasudevan [22]	-	Black hole	INSULATE approach with ANN classifier, Dempster-Shafer for combining evidence
Osman et al. [23]	A large self-constructed dataset for RPL	Version number attack, decreased rank, DIS flooding	Ensemble learning (stacking), GA-based feature selection, extreme parameter optimization
Karimy and Reddy [26]	Nine different IoT nodes, a custom RPL-Attacks dataset created in Contiki-OS	RPL-based routing attacks	FL approach, local models aggregated for intrusion detection, up to 99.9% improvements
Belkheir et al. [27]	Implementation with Z1 nodes in Contiki's Cooja simulator	Black hole	Distributed detection with collaborative packet-based approach, achieving a PDR of 88.86%
Yazdanypoor et al. [28]	Extensive simulations	Black hole	Hybrid detection: anomaly detection + cryptographic verification, achieving 88% PDR in black hole
Abdallah et al. [29]	Logs from intermediate nodes in MANET scenarios	Black hole	SVM-based anomaly scores, scanning node traffic for maliciousness threshold, 99.96% detection accuracy
Gurung & Mankotia [24]	NS-2 simulation environment	Black hole, flooding	Integrates advanced path validation checks in AODV, achieving up to 92% PDR

3- Proposed Methodology

Figure 2 presents a proposed framework that efficiently detects network threats by combining intelligent data extraction with a privacy-preserving learning approach. It begins by capturing real-time radio traffic from a simulated RPL network, where the simulator is carefully modified to log critical parameters that reflect the network's behavior and possible anomalies. This tailored data extraction results in a dataset that mirrors real-world conditions. Following data collection, FL is applied to each node, training a local model using its data without sharing raw information. A central server receives only the model updates, which it aggregates to construct a refined global model. This process improves detection accuracy and maintains data privacy, reduces communication overhead, and is well suited for the resource-constrained nature of IoT and LLN environments.

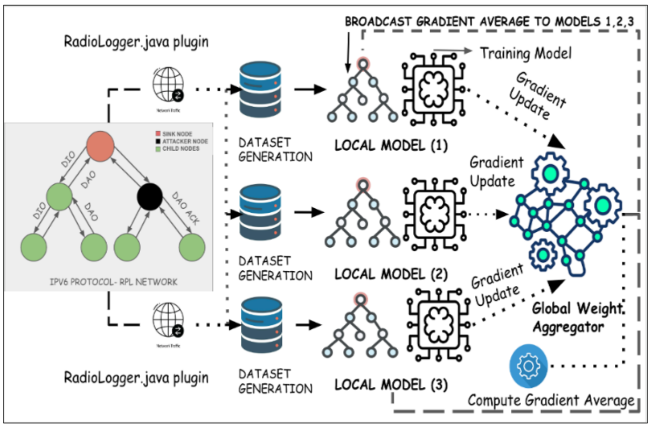


Figure 2. Proposed framework

Algorithm 1 shows the RPL network traffic capture and parameter computation algorithm that analyzes network traffic and detects black hole attacks. The algorithm process starts with capturing key parameters from the network, including node ID, number of destinations, and network addresses (source, destination, and edge node) with an IPv6-based format. These parameters are used for mapping the communication between nodes and recognizing attack patterns. Nodes are given a rank between 256 and 2048, which tells them where they fit in the network hierarchy. The root node is assigned a base rank of 256, and the attacker node is assigned a rank of 257 to make itself more visible to the root node and change the way the network routes traffic through it. The rank of the client nodes is computed by the number of destinations they have within the network. The minimum and maximum rank values are computed to observe any irregularities or malicious behavior in the network.

Algorithm 1. RPL network traffic capture and feature extraction

```
Input: Network nodes, RadioLogger.java parameters
Output: Captured traffic, ranks, ETX, energy consumption, packet flow analysis
Step 1: Capture radio traffic
    Modify the required file with customized parameters
    Locate data fields and map to corresponding row and column numbers
    for time t = $t_0$ to $t_n$ do (// Specify column IDs)
Extract and compute essential parameters:
Get the node ID and number of destinations from an existing file.
Capture network address: Source, Destination, and Edge node in IPv6 format "fe80::212"
Step 2: Rank computation
   base_rank ← 256      // for server node
    attacker_rank ← 257 // for attacker node
    Rank ← base_rank + root_rank + n × increment
    Compute minimum and maximum rank:
min_rank ← root.rank
max_rank ← max(n) × root_rank
Step 3: Transmission parameter computation
    Extract sending time from existing file
Compute:
Elapsed_Time ← End_Time - Start_Time
Sending_Rate ← Packet_Length / Elapsed_Time
Delta_Time ← Time - Elapsed_Time
Step 4: Packet flow computation
    Packet_Count ← Packet_Count + Hidden_Packets
    Drop_Count ← Forward_Count - Received_Count
Step 5: Energy computation
    Energy ← (cpu.time + lpm.time + rx.time + tx.time)
    Assign abnormal energy consumption value to attacker nodes
Step 6: ETX computation   // Estimated Transmission Count
    ETX ← ETX + (recorded.etx × ETX_Alpha) +
           ((packet_etx × (ETX_Scale - ETX_Alpha) / ETX_Scale) + dests.length + i)
    end for
Step 7: Other network computations
    Compute hop_count based on number of destinations
    Compute distance of the node to the sink node
    Compute node honesty level:
Node_Honesty ← fw.count - rw.count + dests.length +
                         (fw.count × $w_1$) + ETX + (rank × $w_2$)
Step 8: Node labeling and data capture.
    Assign labels to nodes:
Edge_Node
Root_Node
Black hole_Node
Child_Node
End Algorithm
```

In the transmission parameter computation, the metric sending rate and delta time are computed to analyze packet movement. The rate at which packets of fixed length are transmitted is computed using the sending rate and transmission delays, which are measured by delta time. In the preprocessing, these parameters are filtered out to reduce unnecessary noise. The packet flow analysis tracks the forward count, reception count, and drop count of each node within the network. The attacker nodes have a forward count set to zero, exhibiting black hole activity. The energy consumption of the node is measured using CPU time, low power mode time, receive time, and transmit time. The attacker node consumes more energy due to its malicious nature. Hence, it is a vital parameter during intrusion detection. Additionally, the expected transmission count (ETX) is computed to determine the efficiency of data delivery within the network. The algorithm measures the hop count, where a node with a smaller number of destinations is expected to have a low hop count and vice versa. The distance of the node is measured using RSSI values to detect false rank claims by a malicious node in the network. The node's honesty level ensures the reliability of each node based on packet transmissions. Lastly, the nodes are classified into different types of nodes—edge, root, black hole, and child. The unwanted parameters, like sending time and packet data, are removed to refine and optimize the dataset for FL-based anomaly detection. This approach detects black hole attacks better, thereby improving the security of RPL-based IoT networks.

Algorithm 2. **FL algorithm for black hole attack detection in RPL**

```
Step 1: Distributed model initialization
  Define RPL network topology:
▷ DODAG root node (central node): Aggregates model updates
▷ Parent nodes (relay nodes): Forward routing data
▷ Child nodes (IoT nodes): Send messages
  Initialize global model $G_0$ at DODAG root node
Step 2: Collaborative training rounds
  for round r = 1 to R do
Step 2-(i): Data collection and preprocessing
For each node j = 1 to N in parallel, do
Initialize local parameters $L_{\square} \leftarrow G_e$
        for epoch i = 1 to $E_{local}$ do
          for each data point $D_{i\square} \in D_{\square}$ do
            Extract features
            Preprocess:
              ▷ IP numerical encoding
              ▷ Time-based normalization
          end for
        end for
Step 2-(ii): Edge-level model training
Compute loss:
$L = -\ (1 / |D_{\square}|)\ {}_{i=1}\Sigma^{|D_{\square}|}\ [\ y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)\ ]$
▷ Where $y_i$ is the actual label (1 = normal, 0 = attack)
▷ $\hat{y}_i$ is the predicted probability
▷ $|D_{\square}|$ is the number of samples
        Update via Adam:
          $m_{\square} = \beta_1\ m_{t-1} + (1 - \beta_1)\ \nabla L$
          $v_{\square} = \beta_2\ v_{t-1} + (1 - \beta_2)\ \nabla L^2$
$\theta_{\square} = \theta_{t-1} - (\alpha / \sqrt{}(v_{\square} + \varepsilon))\ m_{\square}$
▷ Gradient update ($\nabla L$): Detect anomalies
▷ First moment estimate ($m_{\square}$): Tracks attack trends
▷ Second moment estimate ($v_{\square}$): Stabilizes training
▷ Model weights update ($\theta_{\square}$): Updates at IoT nodes
▷ Historical memory ($\beta_1$, $\beta_2$): Maintains long-term stability
▷ Learning rate ($\alpha$): Controls responsiveness
        Apply dropout (20%)
Save weights $w_{\square} \leftarrow L_{\square}$
      end for
Send $w_{\square}$ to DODAG root node
    end for
Step 2-(iii): Federated aggregation
Compute the federated average:
$G_{e+1} \leftarrow (1 / n)\ {}_{j=1}\Sigma^{n}\ (L_{\square} \cdot m_{\square})$
      ▷ ($L_{\square}$: Local model weights of node j, $m_{\square}$: Fraction of total training data)
Broadcast $G_{e+1}$ to all nodes
Step 2-(iv): Centralized model training
if validation accuracy for all 10 rounds, then
Trigger early stopping
      end if
  end for
 End Algorithm
```

Algorithm 2 illustrates the FL algorithm for detecting black hole attacks in RPL networks. The process begins with the global model initialization at the Destination-Oriented Directed Acyclic Graph (DODAG) root node (G_0) and defines the network with parent and child nodes. During the collaborative learning rounds (r), there are two steps: the first is the data collection and preprocessing stage, and the second is local training. Every node (j) trains the local model (L_j) using the locally extracted data (D_j) with the lightweight DNN architecture for LLN devices. The data extraction and preprocessing stage includes the extraction of many features and the standardization of the inputs, and in the edge-level training phase, there are several computations done for binary cross entropy for loss calculations. Anomalies such as black hole attacks are detected using the gradient updates (∇L), while the Adam Optimizer updates the parameters based on gradients with the first and second moment estimates (m_t and v_t) for stable updates. Once the models are trained locally, the weights (w_j) are sent to the root node, and federated averaging is done where the global model is updated to (G_{e+1}). Based on weight contributions from each node's local model (L_j) and data proportion (m_j). The updated global model is broadcast to all nodes. Furthermore, during the centralized training, the global model has certain refinements with early stopping triggered if there is no improvement in the accuracy over 10 rounds. The algorithm prioritizes security by adjusting to new attack patterns through parameter updates and optimizing resource use with dropout regularization.

4- Experimental Setup

For the experiment, Contiki OS is installed on a virtual machine using VMware Workstation 17 Player. The virtual machine is allocated 6 GB of RAM and runs on a Dell Inspiron 15 laptop with the following configuration: 16 GB RAM, 512 GB internal storage, and a 64-bit Windows operating system. The Contiki OS utilized the Contiki NextGen version. Table 2 details the environmental setup configurations used for the dataset generation. The network simulation uses a 100 m x 100 m area to construct a network topology with 30 nodes. This setup incorporates various types of nodes, each with a specific role: a single border router node managing network traffic, a server node acting as the central data repository, 3 attacker nodes simulating malicious entities, and 25 client nodes representing typical network users. Sky motes were used to build the network because they work well with RPL and are suitable for use in resource-limited IoT environments. The motes have 10 KB RAM, 48 KB ROM, and a maximum communication range of 100 meters, which allows us to emulate a resource-constrained IoT scenario and provide suitable lightweight security mechanisms. Each node in the network topology can transmit data across a 50-meter range, with a 100-meter interference range. This capability enables us to manage communications during the simulation and to replicate interference scenarios. The CC2420 radio model can capture radio traffic for a specific period and allows for the analysis of the captured traffic.

Table 2. Parameter settings of the simulator

Parameters	Details
Virtual OS	UBUNTU
Simulator	Contiki OS/Contiki NG—Cooja Simulator
Virtual RAM of Contiki NG	6 GB
Radio environment	UDGM (distance loss)
Simulation time	900000 ms
Topology	Random
Size of network	10, 20, 30
Objective function	MRHOF
Mote type	Sky Mote
Simulation area	100 m * 100 m
Sky Mote memory specifications	RAM: 10 KB, ROM: 48 KB, EEPROM: 1024 KB
Plugins	RadioLogger.java
Transmission range	50 m
Interference range	100 m
Dynamic traffic rate	Min: 1 ppm and Max: 50 ppm
Constant traffic rate	1 ppm
Radio model	CC2420
DIO_MIN	12
DIO_MAX	20
MAX_RANK	2048
MIN_RANK	256

The IoT-based RPL topology is structured with a base rank of 256, assigned to the server node, and extends up to a maximum rank of 2048. This hierarchical ranking facilitates efficient routing within the network. To establish the prediction of black hole attacks, the RadioLogger.java plugin file is modified to capture specific radio message fields and log the required data for more analysis. To ensure the scalability of the solution and enable predictive analysis across different network configurations, network topologies were constructed with 10, 20, and 30 motes. The 30-node topology is determined to be the optimal solution. This decision is based on the observation that scaling beyond 30 motes resulted in network overload. This overload is triggered by using both constant and dynamic traffic rates, along with the physical system's limits, which do not handle the extra network load when going beyond 30 motes. The dataset is published and available in Mathi et al. [30]. The simulated environment comprises a border router, a server mote, attacker motes (strategically assigned node IDs 7, 17, and 27), and several client motes. The simulation script editor is then activated, and a timer is set to 900,000 milliseconds. This duration allows for the capture of a substantial amount of radio traffic equivalent to 15 minutes of network activity, which is crucial for subsequent analysis.

The ‘tcpip.c’ file is modified to enable logging of specific metrics. It contains energy consumption, rank and hop values (exploring network routing dynamics), and ICMPv6 RPL control messages. This log information is directed to both the mote output window and the simulation script log console, providing a comprehensive record of network behavior over time. To align with the requirements of black hole analysis, the RadioLogger.java plugin is customized. The fields within this plugin were adjusted to capture the required data for this investigation, and the ICMPv6 packet analyzer is changed to ensure the synchronization of packet flow across the network. Several tools within the simulation environment assist in monitoring and data collection. The radio message window shows all radio traffic in real time and provides a graphical display of the network communication. The buffer listener window tracks individual nodes and determines packet routing in an experimental environment. Additionally, the timeline window presents a visual chronology of network events for each mote, highlighting transmission, reception, and overall radio activity. The system converts the captured data from the radio message into a dataset. The dataset is used to run ML, DL, and FL algorithms and make predictive models for black hole attacks within RPL-based IoT networks.

5- Results Analysis and Performance Evaluation

Table 3 shows the ML models with varying performance across different investigations, with some excelling while others struggle. Shahid et al. (2024) [17] reported that RF and DT models achieved 99% accuracy, showing their strength in decision-making. On the other hand, Stochastic Gradient Descent (SGD) and Gaussian Naïve Bayes (GNB) performed poorly, with accuracies of 75% and 52%, which shows they have problems while dealing with complex patterns and imbalanced data. Similarly, Kurtkoti et al. (2022) [16] reported that the models RF (98.6%), J48 (96.72%), Bagging Classifier (98.45%), and AdaBoost (98.49%) achieved results, while models like KNN (89.42%), GNB (64.24%), SGD (64.2%), and Multilayer Perceptron (MLP) (64.24%) performed poorly, likely because they depend on specific feature engineering methods.

Table 3. Analysis of the accuracies of related works of ML algorithms

Research works	Model	Accuracy in %
Shahid et al. (2024) [17]	RF	99
	DT	99
	SGD	75
	GNB	52
Kurtkoti et al. (2022) [16]	Random Tree	92.6
	Random Forest	98.6
	J48	96.72
	Bagging Classifier	98.45
	AdaBoost	98.49
	KNN	89.42
	GNB	64.24
	SGD	64.2
	MLP	64.24

One key limitation of traditional ML models is their dependence on preprocessing and manual feature selection, making them less effective in handling high-dimensional and complicated data. The ML algorithms depend on predefined statistical measures and correlations, which restrict their adaptability. The DL models can automatically extract features and identify hidden patterns, thereby enhancing the algorithms for any problem type. As a result, there is a growing development toward the DL approaches, which provide enhanced accuracy and flexibility in solving more sophisticated challenges.

The evaluation metrics in Table 4 for different ML models highlight significant variations in their performance. Shahid et al. (2024) [17] found that random forest and decision tree models scored perfectly on all measures, with 1.0 for precision, recall, and F1-score, showing they are very effective at correctly identifying both normal and attack cases. These models perform well in decision-making and can generalize well when trained on good-quality data. However, SGD and GNB struggled significantly, with SGD showing a precision of 0.43, a recall of only 0.04, and an F1-score of 0.07. SGD achieved a precision of 0.43, a recall of just 0.04, and an F1-score of 0.07, showing that while it could correctly identify a few positive cases, it failed to detect most attacks, making it unreliable in this scenario. Similarly, GNB achieved a precision of 0.26, a recall of 0.17, and an F1-score of 0.22, showing its weaker ability to differentiate between normal and attack traffic effectively. The poor recall values for these models show a high number of false negatives, meaning many attacks remain undetected. Since ML algorithms essentially depend on correlations and predefined parameters, they struggle to adapt when new and unknown threats emerge. This limitation makes them less effective in real-world RPL security cases, where attack patterns are constantly evolving. The DL models are better suited for securing RPL networks against dynamic and sophisticated threats because they can learn hierarchical features and adapt to unseen attack patterns.

Table 4. Analysis of precision, recall, and F1 scores

Research works	Model	Precision	Recall	F1 score
Shahid et al. (2024) [17]	RF	1	1	1
	DT	1	1	1
	SGD	0.43	0.04	0.07
	GNB	0.26	0.17	0.22

Figure 3 shows an accuracy comparison of scores across different DL models for identifying anomalies in RPL networks. The accuracy of each model is plotted to highlight its performance across various contributions. The trend shows a steady enhancement in accuracy as models evolve from Feedforward Neural Networks (FFNN) to more advanced models like transformers and DNNs. The proposed model achieves the highest accuracy and shows that as models grow in complexity and learning capacity, their ability to detect attacks becomes better.

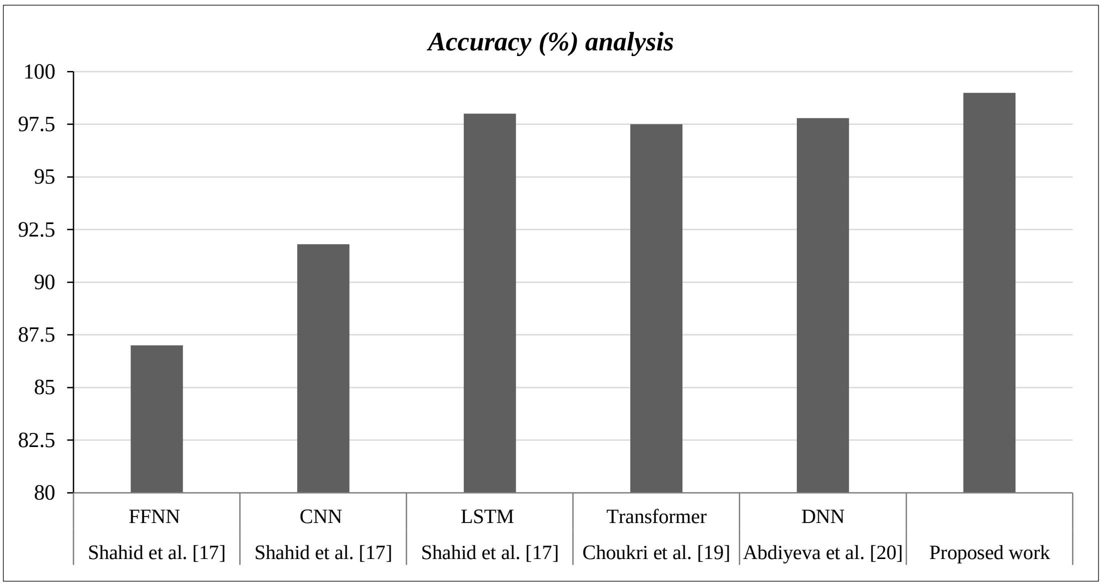
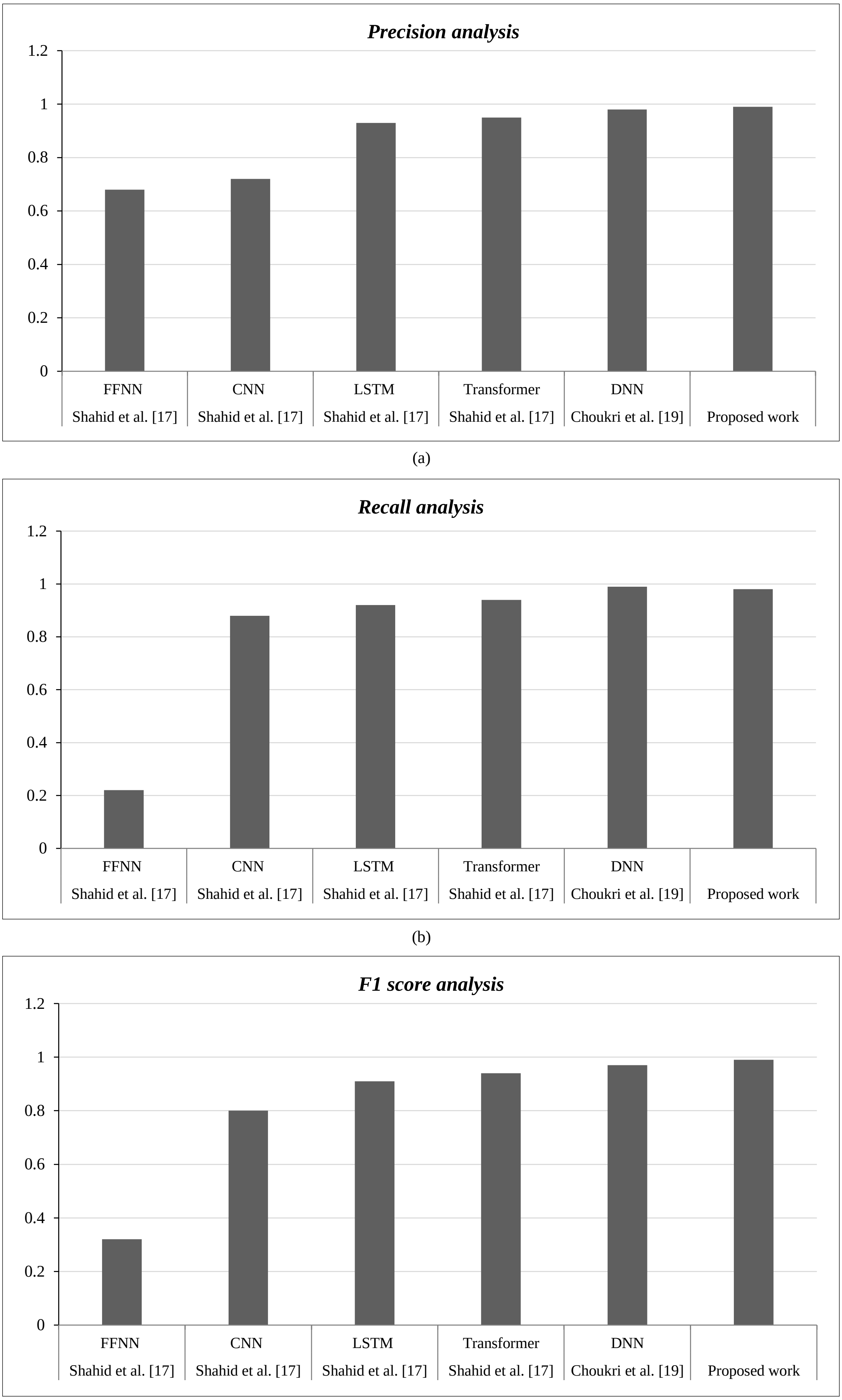


Figure 3. Experimental results of the proposed methodology vs. DL models in other works

Among the examined models, CNN, LSTM, and transformers demonstrate strong performance, with accuracy increasing as model depth and architecture complexity improve. The DNN model also performs well, aligning closely with the highest-performing models. The proposed model enhances accuracy further, likely due to a more effective training strategy tailored to real-world network situations. The general trend in this comparison shows that more sophisticated DL models are better suited for detecting attacks in RPL networks, surpassing traditional ML approaches. The increasing accuracy across models shows the significance of incorporating more advanced architectures for securing modern networks.

Figures 4-a to 4-c exhibit the precision, the recall, and the F1-score for each model, respectively. These figures show how well each model works based on various measures. These metrics are essential when it comes to understanding how accurately and reliably a model detects black hole attacks—especially in real-world scenarios where both missed detections and false alarms can be costly. Across all these figures, as the models become more advanced, the overall performance improves. The results show that the proposed model performs significantly better than other related works.



Precision analysis

(a)

Recall analysis

(b)

F1 score analysis

(c)

Figure 4. (a). Comparison of precision of proposed methodology vs. other related works, (b). Comparison of recall of proposed methodology vs other related works, (c). Comparison of F1-score of proposed methodology vs. other related works

Table 5 provides a comparison of the proposed FedBHAD model with representative prior works and baseline algorithms. The reported values are directly taken from the existing works—Shahid et al. (2024) [17] and Kurtkoti et al. (2022) [16]. In this study, the baseline experiments were conducted with the experimental setup. As listed in Table 5, traditional ML approaches such as GNB and SGD achieve only 52% and 75% accuracy, revealing difficulties in capturing nonlinear attack behavior and managing class imbalance. In contrast, ensemble methods (RF, DT) perform better but still depend on centralized learning. The proposed FedBHAD framework achieves 99.6% accuracy with better precision and recall, confirming its ability to deliver reliable detection under constrained IoT environments while preserving data privacy.

Table 5. Comparative summary of previous studies and the proposed FedBHAD framework

Research works	Dataset used	Method/Model	Accuracy (%)	Key findings
Shahid et al. (2024) [17]	Simulated IoT network (Cooja, Contiki)	RF	99.0	High accuracy but high energy cost; lacks distributed learning support.
Kurtkoti et al. (2022) [16]	Custom RPL dataset	DT	99.0	Good detection rate; sensitive to overfitting and network noise.
Existing ML Baseline	RPL-based simulated dataset	KNN	89.4	Performs moderately; limited scalability in large networks.
		GNB	52.0	Fails to capture nonlinear attack patterns; poor recall.
		SGD	75.0	Sensitive to learning-rate tuning; unstable on imbalanced data.
		MLP	64.2	Moderate performance; convergence is slower under constrained devices.
Proposed FedBHAD	Cooja-generated LLN dataset	Collaborative learning	99.6	Balanced performance across accuracy, precision, recall, and F1; reduced communication overhead; privacy-preserving decentralized training.

Although Karimy & Reddy [26] and the proposed FedBHAD employ FL for IoT security, the proposed FedBHAD differs from Karimy & Reddy [26] (who applied a generic FL framework with Differential Privacy (DP) for broad IoT intrusion detection) by focusing on the detection of black hole attacks in RPL-based LLNs rather than generic intrusion detection. FedBHAD introduces semantic-aware routing features, a lightweight and adaptive federated structure optimized for resource-constrained Sky motes, and achieves higher accuracy (99.6%) with lower communication overhead. In contrast, Karimy et al. used a conventional FedAvg + DP setup (with an additional mechanism for DP to protect model updates) on general datasets without addressing LLN constraints. Hence, FedBHAD provides a more targeted, efficient, and practically deployable FL framework for constrained IoT environments.

It is important to note that the results are based on the existing reported findings rather than reimplemented experiments. The direct comparison is therefore influenced by differences in datasets, feature selection, preprocessing methods, and evaluation protocols. Nevertheless, the comparison analysis shows consistent trends: Traditional models such as SGD and GNB show reduced accuracies (75% and 52%), indicating limited ability to capture complex and nonlinear attack patterns in RPL-based networks. In contrast, ensemble and collaborative learning approaches demonstrate improved performance by handling class imbalance and nonlinear relationships more effectively. The proposed FedBHAD framework outperforms these earlier investigations with multiple metrics such as accuracy, precision, recall, and F1-score that demonstrate its robustness in detecting malicious nodes while maintaining low communication overhead. Although methodological differences restrict a one-to-one quantitative comparison, the observed improvements strongly suggest that the proposed model generalizes better under constrained IoT conditions.

To quantitatively evaluate the energy efficiency of the proposed FedBHAD framework, a comparative analysis is performed between the centralized model and the FedBHAD framework using Contiki's built-in energy tracking metrics (CPU, LPM, RX, and TX times). The simulation is executed for 900,000 ms (15 minutes) under network conditions with 30 nodes. The results listed in Table 6 demonstrate that FedBHAD significantly reduces node-level energy consumption by minimizing radio transmissions and receptions. The decentralized training model allows nodes to update their models without having to send raw data to the central node, thereby minimizing communication energy cost. On average, the proposed FedBHAD model achieved an approximately 28% reduction in total energy consumption compared to the centralized approach, confirming its suitability for LLN environments.

Table 6. Energy efficiency analysis of the proposed FedBHAD framework

Energy metric	Centralized model (mJ)	FedBHAD framework (mJ)	Energy reduction (%)
CPU time energy	3.45	2.67	22.6
Low power mode (lpm) energy	1.92	1.48	22.9
Transmission (tx) energy	2.31	1.59	31.2
Reception (rx) energy	2.85	1.93	32.3
Total energy consumption	10.53	7.67	27.8

6- Conclusion

The present paper proposes a new algorithm, FedBHAD, to detect black hole attacks in LLNs using a FL framework. By combining local training at individual nodes with centralized model aggregation and analysis, the proposed work identifies black hole attack behavior without compromising data privacy or increasing network overhead. A customized dataset was generated using the Cooja simulator with essential network parameters such as rank, ETX, power consumption, and packet forwarding behavior. The extracted features of the dataset generated enable the proposed framework to distinguish between normal and black hole node activities with better precision. The proposed method achieved a detection accuracy of 99.6%, outperforming traditional ML and DL models while preserving the energy efficiency of the RPL-based IoT networks. The results show that integrating a collaborative framework can enhance detection of the black hole attacks.

Beyond addressing black hole attacks, the proposed FedBHAD approach can also be adapted to handle other threats in RPL networks, including sinkhole, wormhole, and Sybil attacks. The decentralized approach of the suggested framework allows IoT nodes to continuously learn and adapt to evolving attack patterns, providing protection in dynamic environments. Furthermore, the privacy-preserving feature of FL aligns strongly with modern data control and ethical AI principles, making the model suitable for critical applications such as healthcare, smart cities, and industrial IoT. Future research could concentrate on integrating the proposed framework with blockchain approaches, specialized edge hardware, and real-time analytics to enhance transparency, response time, and energy efficiency. Additionally, exploring the potential of resource-aware graph neural networks and federated reinforcement learning is a topic for further research.

7- Declarations

7-1- Author Contributions

Conceptualization, S.M., G.R.L., L.C.M., K.A.R., and P.J.; methodology, S.M., G.R.L., L.C.M., K.A.R., and P.J.; software, S.M., G.R.L., L.C.M., K.A.R., and P.J.; validation, S.M. and G.R.L.; formal analysis, S.M. and G.R.L.; investigation, S.M. and G.R.L.; resources, S.M., G.R.L., L.C.M., K.A.R., and P.J.; data curation, S.M., G.R.L., L.C.M., K.A.R., and P.J.; writing—original draft preparation, S.M., G.R.L., L.C.M., K.A.R., and P.J.; writing—review and editing, S.M., G.R.L., and G.N.I.; visualization, S.M. and G.R.L.; supervision, S.M.; project administration, S.M. and G.N.I.; funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

7-2- Data Availability Statement

The data presented in this study are openly available in IEEE Dataport: RPLGuard_Dataset-v1-2025: RPL Dataset for Guarding Against Attacks. doi:10.21227/mhgp-zg08.

7-3- Funding and Acknowledgements

The authors from Amrita Vishwa Vidyapeetham gratefully acknowledge Sri Mata Amritanandamayi Devi (Amma), Chancellor, Amrita Vishwa Vidyapeetham, for her inspiration and for providing financial support for the Article Processing Charges (APC) of this publication.

7-4- Institutional Review Board Statement

Not applicable.

7-5- Informed Consent Statement

Not applicable.

7-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

8- References

[1] Mathi, S., Akshaya, R., & Sreejith, K. (2022). An Internet of Things-based Efficient Solution for Smart Farming. Procedia Computer Science, 218, 2806–2819. doi:10.1016/j.procs.2023.01.252.

[2] Vidhya, S. S., Mathi, S., Ananthanarayanan, V., & Iyer, G. N. (2024). IP-RPL: An Intelligent Power-aware Routing Protocol for Next Generation Low Power Networks. IEEE Sensors Journal. doi:10.1109/JSEN.2024.3506816.

[3] Anjali, A. N., Sreelakshmi, C. B., & Remya Nair, T. (2023). Blackhole Attack Detection in Wireless Sensor Network Using Backpropagation Algorithm. 2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023, 1–6. doi:10.1109/ICCCNT56998.2023.10307586.

[4] Mayzaud, A., Badonnel, R., & Chrisment, I. (2016). A taxonomy of attacks in RPL-based internet of things. International Journal of Network Security, 18(3), 459–473. doi:10.6633/IJNS.201605.18(3).07.

[5] Panda, N., Supriya, M. (2023). Blackhole Attack Prediction in Wireless Sensor Networks Using Support Vector Machine. Advances in Signal Processing, Embedded Systems and IoT, Lecture Notes in Electrical Engineering, Springer, Singapore. doi:10.1007/978-981-19-8865-3_30.

[6] Wakili, A., Bakkali, S., & Alaoui, A. E. H. (2024). Machine learning for QoS and security enhancement of RPL in IoT-Enabled wireless sensors. Sensors International, 5, 100289. doi:10.1016/j.sintl.2024.100289.

[7] Krari, A., Hajami, A., Toubi, A., Errakha, K. (2025). A Deep Learning Approach to Strengthening IoT RPL Protocol Security Against Black Hole Attacks Advances in Intelligent Systems and Digital Applications, ISDA 2025, Lecture Notes in Networks and Systems, vol 1485. Springer, Cham, Switzerland. doi:10.1007/978-3-031-95326-2_4.

[8] Sharma, N., & Dhiman, P. (2025). A survey on IoT security: challenges and their solutions using machine learning and blockchain technology. Cluster Computing, 28(5), 313. doi:10.1007/s10586-025-05208-0.

[9] Raghavendra, T., Anand, M., Selvi, M., Thangaramya, K., Santhosh Kumar, S. V. N., & Kannan, A. (2022). An Intelligent RPL attack detection using Machine Learning-Based Intrusion Detection System for Internet of Things. Procedia Computer Science, 215, 61–70. doi:10.1016/j.procs.2022.12.007.

[10] Sharma, D. K., Dhurandher, S. K., Kumaram, S., Datta Gupta, K., & Sharma, P. K. (2022). Mitigation of black hole attacks in 6LoWPAN RPL-based Wireless sensor network for cyber physical systems. Computer Communications, 189, 182–192. doi:10.1016/j.comcom.2022.04.003.

[11] Airehrour, D., Gutierrez, J., & Ray, S. K. (2016). Securing RPL routing protocol from blackhole attacks using a trust-based mechanism. 2016 26th International Telecommunication Networks and Applications Conference (ITNAC), 115–120. doi:10.1109/atnac.2016.7878793.

[12] Karthikeyan, M., & Revathi, S. T. (2024). Approaches to Detecting Threats in Wireless Sensor Networks for Data Transmission Security. 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), 1–7. doi:10.1109/accai61061.2024.10601841.

[13] Shukla, M., Brijendra Kumar Joshi, & Singh, U. (2024). A Novel Machine Learning Algorithm for MANET Attack: Black Hole and Gray Hole. Wireless Personal Communications, 138(1), 41–66. doi:10.1007/s11277-024-11360-4.

[14] Shafi, S., & Venkata Ratnam, D. (2023). New Energy Aware MPR Selection for Securing OLSR Routing Scheme under Black Hole Attack: A Machine Learning Approach. Wireless Personal Communications, 132(3), 1917–1931. doi:10.1007/s11277-023-10690-z.

[15] Hussain, K., Xia, Y., Onaizah, A. N., Manzoor, T., & Jalil, K. (2022). Hybrid of WOA-ABC and proposed CNN for intrusion detection system in wireless sensor networks. Optik, 271, 170145. doi:10.1016/j.ijleo.2022.170145.

[16] Kurtkoti, M., Premananda, B.S., Vishwavardhan Reddy, K. (2022). Performance Analysis of Machine Learning Algorithms in Detecting and Mitigating Black and Gray Hole Attacks. Innovative Data Communication Technologies and Application. Lecture Notes on Data Engineering and Communications Technologies, Springer, Singapore. doi:10.1007/978-981-16-7167-8_69.

[17] Shahid, U., Zunnurain Hussain, M., Zulkifl Hasan, M., Haider, A., Ali, J., & Altaf, J. (2024). Hybrid Intrusion Detection System for RPL IoT Networks Using Machine Learning and Deep Learning. IEEE Access, 12, 113099–113112. doi:10.1109/ACCESS.2024.3442529.

[18] Garcia Ribera, E., Martinez Alvarez, B., Samuel, C., Ioulianou, P. P., & Vassilakis, V. G. (2022). An Intrusion Detection System for RPL-Based IoT Networks. Electronics (Switzerland), 11(23), 4041. doi:10.3390/electronics11234041.

[19] Choukri, W., Lamaazi, H., & Benamar, N. (2022). A Novel Deep Learning-based Framework for Blackhole Attack Detection in Unsecured RPL Networks. 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT). doi:10.1109/3ict56508.2022.9990664.

[20] Abdiyeva-Aliyeva, G., Hematyar, M., & Bakan, S. (2021). Development of System for Detection and Prevention of Cyber Attacks Using Artificial Intelligence Methods. 2021 2nd Global Conference for Advancement in Technology (GCAT), 1–5. doi:10.1109/gcat52182.2021.9587584.

[21] Ahmadi, K., & Javidan, R. (2024). A novel RPL defense mechanism based on trust and deep learning for internet of things. Journal of Supercomputing, 80(12), 16979–17003. doi:10.1007/s11227-024-06118-5.

[22] Prajisha, C., & Vasudevan, A. R. (2022). An Intrusion Detection System for Blackhole Attack Detection and Isolation in RPL Based IoT Using ANN. Advanced Computing, 332–347. doi:10.1007/978-3-030-95502-1_26.

[23] Osman, M., He, J., Zhu, N., & Mokbal, F. M. M. (2024). An ensemble learning framework for the detection of RPL attacks in IoT networks based on the genetic feature selection approach. Ad Hoc Networks, 152, 103331. doi:10.1016/j.adhoc.2023.103331.

[24] Gurung, S., & Mankotia, V. (2024). ABGF-AODV protocol to prevent black-hole, gray-hole and flooding attacks in MANET. Telecommunication Systems, 86(4), 811–827. doi:10.1007/s11235-024-01154-1.

[25] Reshi, I. A., Sholla, S., & Najar, Z. A. (2024). Safeguarding IoT networks: Mitigating black hole attacks with an innovative defense algorithm. Journal of Engineering Research (Kuwait), 12(1), 133–139. doi:10.1016/j.jer.2024.01.014.

[26] Karimy, A. U., & Reddy, P. C. (2024). Analyzing Federated Learning as a novel approach for enhancing security and privacy in the Internet of Things (IoT). 2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 1–7. doi:10.1109/icaect60202.2024.10468686.

[27] Belkheir, M., Rouissat, M., Mokaddem, A., Ziani, D., & Lorenz, P. (2025). Lightweight Novel Approach for Collaborative Packet-Based Mitigation of Blackhole Attacks in RPL-Based IoT. Journal of Network and Systems Management, 33(2), 34. doi:10.1007/s10922-025-09908-1.

[28] Yazdanypoor, M., Cirillo, S., & Solimando, G. (2024). Developing a Hybrid Detection Approach to Mitigating Black Hole and Gray Hole Attacks in Mobile Ad Hoc Networks. Applied Sciences (Switzerland), 14(17), 7982. doi:10.3390/app14177982.

[29] Abdallah, A. A., El Sayed Abdallah, M. S., Aslan, H., Azer, M. A., Cho, Y.-I., & Abdallah, M. S. (2024). Enhancing Mobile Ad Hoc Network Security: An Anomaly Detection Approach Using Support Vector Machine for Black-Hole Attack Detection. International Journal of Safety and Security Engineering, 14(4), 1015–1028. doi:10.18280/ijsse.140401.

[30] Mathi, S., Gudivada, R. L., Madala, L. C., Reddy, K. A., & Putta, J. (2025). RPLGuard_Dataset-v1-2025: RPL Dataset for Guarding against Attacks. IEEE Dataport. doi:10.21227/mhgp-zg08.