A HYBRID APPROACH FOR DETECTION AND MITIGATION OF ROUTING ATTACKS IN RPL USING DEEP LEARNING

Deepak Upadhyay[1], Hiteishi Diwanji[2]

•

[1]Phd Scholar, Gujarat Technological University, Gujarat, India
[2]L. D. Government Engineering College, Gujarat, India
[1]ap_deepak@gtu.edu.in , [2]hiteishi.diwanji@gmail.com

Abstract

The exponential growth of the Internet of Things (IoT) has led to the massive deployment of resource-constrained smart devices in domains such as smart homes, industrial automation, healthcare, and smart cities. To facilitate communication between these devices, the Routing Protocol for Low-Power and Lossy Networks (RPL) has emerged as a widely adopted IPv6-based routing protocol customized for low-power, lossy environments. This study proposes an Intrusion Detection System (IDS) framework that processes raw IoT traffic data to detect and classify RPL routing attacks using a hybrid approach and takes mitigation steps. The IDS framework integrates multiple models, such as a LSTM enhanced with Multi-Head Attention for sequential pattern recognition, gradient boosting models such as LightGBM and CatBoost for efficient tabular data classification, and a Multi-Layer Perceptron (MLP) to replicate ensemble knowledge in a lightweight manner. This hybrid IDS demonstrates the efficacy of combining deep learning and machine learning models to detect and mitigate RPL routing attacks, providing a practical and scalable solution for securing IoT networks with highest accuracy of 93.26 % with mobility and 91.96 % without mobility for CNN+LSTM+Attention, LightGBM, CatBoost, MLP algorithms.

Keywords: Intrusion detection system, data science, machine learning, deep learning, RPL, security, IoT, routing protocols, black hole, rank, Sybil, DIS, selective attack.

I. Introduction

The Internet of Things (IoT) has changed how many sectors work by linking devices that share data over the Internet. IoT systems are now common in smart cities, healthcare, farming, and industry (Hassija et al. [1]). For secure data transfer in these networks, the RPL protocol is widely used. It is designed to handle routing in IoT setups with low power and limited resources. Still, these same limits make RPL weak against several security issues. It can be attacked through blackhole, rank, Sybil, DIS, and selective forwarding (Prajapati et al. [2]). Such attacks affect network stability, cause packet loss, increase power use, and reduce Quality of Service (QoS). This is a serious problem because IoT devices often deal with sensitive data and interact with the physical world. To handle this, researchers have worked on Intrusion Detection Systems (IDS) that can detect routing attacks

DOI: https://doi.org/10.24412/1932-2321-2025-489-479-493

in RPL networks. Many studies rely on datasets such as ROUT-4-2023. But most methods only look at single attack types or use pre-processed data, which does not reflect real-world cases where attacks change and overlap.

This work tries to overcome those limits by using the Contiki/Cooja simulator, which acts as a real-life test setup. The simulator creates a raw packet capture (PCAP) file. Then carried out our own steps like cleaning the data, encoding features, normalizing values, and reducing dimensions. These steps make the detection process more reliable. Furthermore, to address the class imbalance issues inherent in IoT attack datasets, applied the plied the Synthetic Minority Oversampling Technique (SMOTE) for balanced model training.

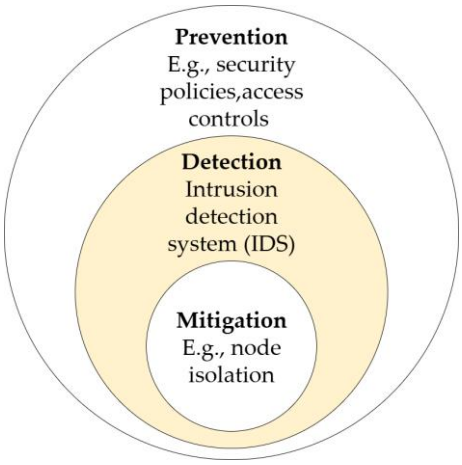

Figure 1: *Network defense*

The proposed detection framework integrates a variety of models, including Long Short-Term Memory (LSTM) networks for identifying sequential traffic behavior, tree-based algorithms such as Light Gradient Boosting Machine (LightGBM) and CatBoost for handling structured tabular data, and a lightweight Multi-Layer Perceptron (MLP) that distills insights from the ensemble. To improve the reliability of the predictions, Test Time Augmentation (TTA) was applied during the inference phase. By employing hybrid machine learning and deep learning techniques, this study facilitates the development of a scalable and accurate IDS for RPL-based IoT networks. The proposed approach not only enhances detection accuracy in mobility environment but also improves generalization across multiple attack types, providing a practical solution for securing IoT environments.

The paper is organized as follows: Section II reviews IDS solutions for RPL-based networks. Section III details the proposed approach, covering simulation setup, dataset, data collection, preprocessing, balancing, feature engineering, model design, training, and evaluation. Section IV presents results, while Section V discusses findings, concludes the study, and highlights future directions.

II. Literature Review

Shahid et al. [3] used the ROUT-4-2023 dataset to develop an IDS with both machine learning and deep learning models, reporting high accuracy with Random Forest (99%) and transformers (97% F1). Yusuf Yavuz et al. [4] proposed a scalable deep learning approach using Cooja-generated data, achieving over 94% accuracy for rank, flooding, and version number attacks. Vatambeti et al. [5] introduced an ensemble model for IIoT routing attacks, where traditional methods such as KNN and SVM underperformed compared to neural models, with their LSTM-based ensemble reaching 94.5% accuracy. Gupta et al. [6] also combined machine learning and deep learning to detect

blackhole attacks by modeling normal network behavior and identifying deviations.
For specific attack scenarios, Çakir et al. [7] applied ML-based detection for DIS flooding, with LR and SVM achieving above 96% accuracy. Neerugatti et al. [8] proposed a distance-based technique to detect rank attacks, while Raghavendra et al. [9] designed a genetic feature selection and fuzzy KNN classifier to detect selective forwarding with improved accuracy and low false positives. Murali et al. [10] developed a lightweight IDS against Sybil attacks inspired by artificial bee colony optimization, reaching around 95% accuracy.
Yu et al. [11] reviewed IoT architecture and attack surfaces at consumer and industrial levels, highlighting vulnerabilities and outlining future research directions for device security.

III. Methodology

This study followed standard data science steps: data collection, preprocessing, and dataset exploration to find patterns in features. These steps supported machine learning and deep learning models to test and compare methods. The work was done in Python because of its open-source tools.

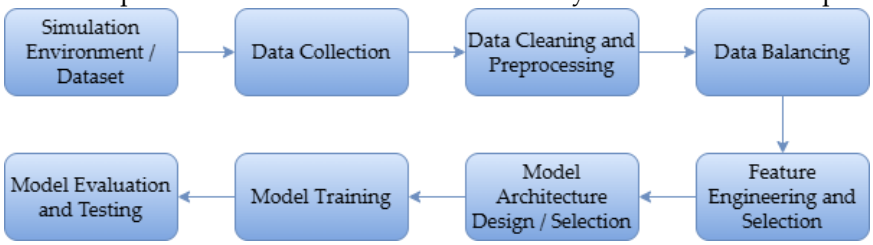

Figure 2: *Methodology diagram*

I. Simulation Environment / Dataset

The foundation of this study begins with the simulation of realistic RPL-based IoT network environments to capture the behaviors of both normal and malicious routing operations. The simulations were conducted using the Cooja network simulator, which operates atop the Contiki OS.

Table 1: *Simulation parameters of reliability measure against attack*

Simulation Parameters	Value
Simulation tool	Contiki /Cooja 3.0
Simulation coverage area	100 m * 100 m
Deployment Type	Random position (based on smart home)
Total number of nodes	30
Emulated nodes	T-mote Sky
Malicious nodes	1:10
TX range	50 m
TX ratio	100%
RX ratio	30-100%
Link failure model	UDGM
Interference range	50 m
Routing protocols	THC-RPL
Mobility speed	0-6.23 km/h
Simulation time	60 min

In this study, five routing attacks were implemented in Contiki: Blackhole, Sybil, DIS Flooding, Rank, and Selective Forwarding. Each attack was programmed into selected nodes and compiled through the node firmware.

I. Blackhole Attack

In a blackhole attack, a malicious node pretends to have the best route to the destination. Other nodes forward packets to it, and the data is dropped, creating a "black hole."

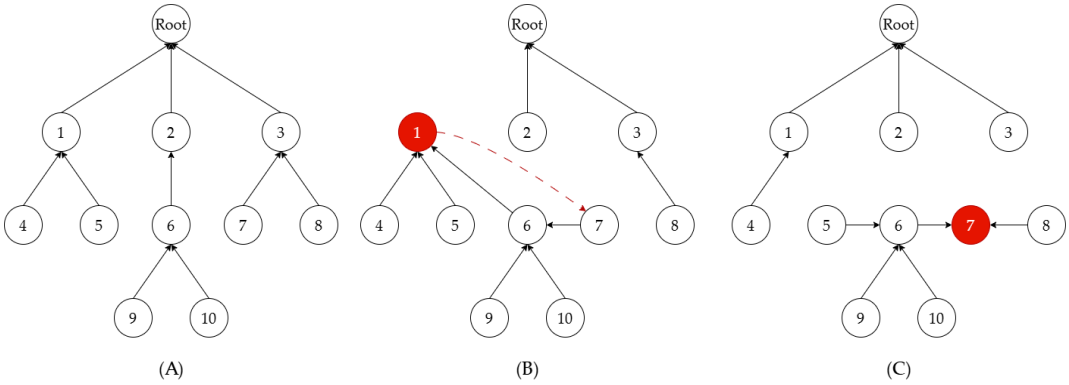

Figure 3: *Blackhole Attack*

Figure 3(A) shows a normal IoT network with 10 nodes forming a DODAG. In Figure 3(B), node 1 disrupts routing by acting as a preferred parent and absorbing packets. Later, in Figure 3(C), the malicious node shifts to another location (e.g., node 7), continuing the attack (Ahmed et al. [12]).

II. Sybil Attack

A Sybil attack occurs when a malicious node uses multiple fake identities in an RPL network. The attacker gains extra influence, leading to instability, data loss, or denial-of-service.

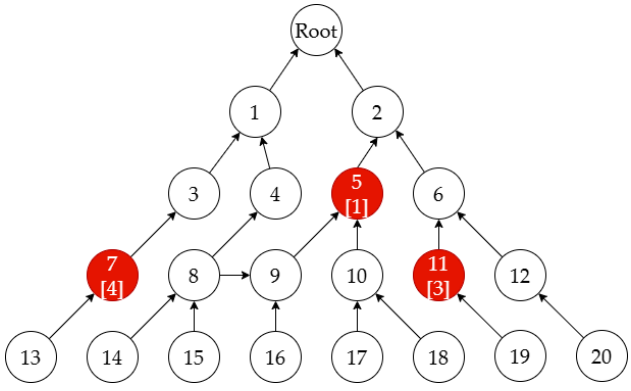

Figure 4: *Sybil Attack*

Figure 4 shows nodes 2, 3, and 10 as Sybil nodes, each using false identities (e.g., [9], [13], [19]) to mislead their neighbors.
Detection of Sybil behavior focuses on three aspects: routing metrics, identities, and packet handling.

First, ETX and rank are monitored; sudden, unexplained rank drops may indicate malicious nodes. Second, nodes are checked for multiple IPv6 identities mapped to the same link-layer address. Third, packet logs are compared across nodes to detect altered or missing packets.

III. DIS Attack

In this attack, a malicious node floods the network with DODAG Information Solicitation (DIS) messages, disrupting routing and creating heavy overhead.

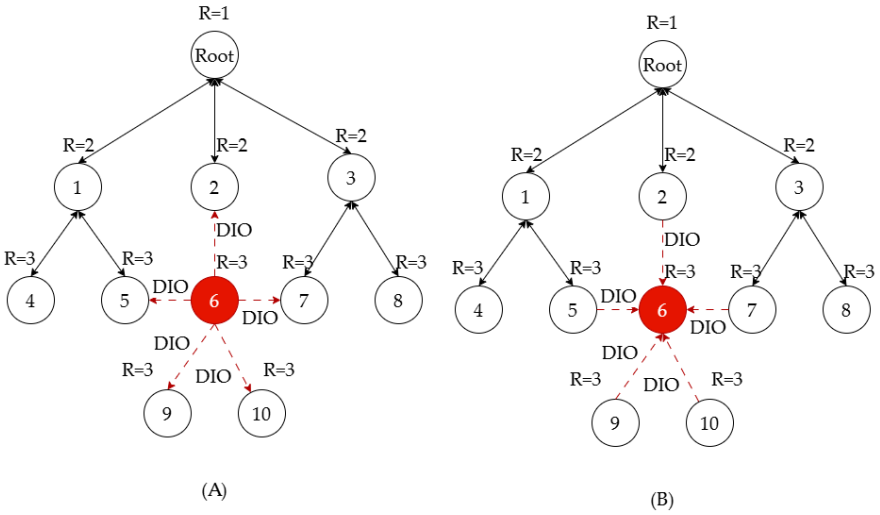

Figure 5: *DIS Attack*

Figure 5 shows a node repeatedly sending DIS messages. Normally, DIS is used by a node to discover the topology and join a DODAG. In a flooding attack, thousands of DIS requests are sent, causing a traffic surge that can bring down the network within minutes.
The attack increases radio activity and power use on the malicious node, since continuous DIS transmission keeps the radio busy. Mobility has little effect, as the attacker still floods the network while changing location. A common defense is to limit DIS requests per node and apply rate-limiting protocols (Verma et al. [13]).

IV. Rank Attack

A Rank Attack targets the RPL topology by broadcasting false rank values. Each node in RPL has a rank showing its distance from the root in the DODAG. Malicious nodes advertise artificially low ranks to mislead neighbors into choosing them as parents. This distorts the topology and causes non-optimal paths, routing loops, and packet loss.

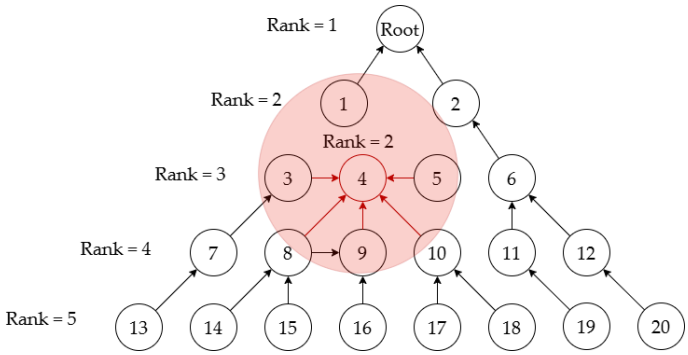

Figure 6: *Rank Attack*

Figure 6 shows node 4 broadcasting a fake rank (Rank = 2) to appear closer to the root. Neighboring nodes (3, 5, 8, 9, 10) then select it as their parent.
Detection involves tracking rank changes over time. Sudden or repeated drops in rank may reveal malicious behavior. Rank values are checked in RPL packets during analysis to identify such nodes (Almusaylim et al. [14]).

V. Selective Forwarding Attack

In this attack, a malicious node drops some packets while forwarding others. Unlike a blackhole, where all traffic is lost, selective forwarding avoids detection by forwarding part of the traffic. This lets the attacker target specific data flow and cause intermittent failures.

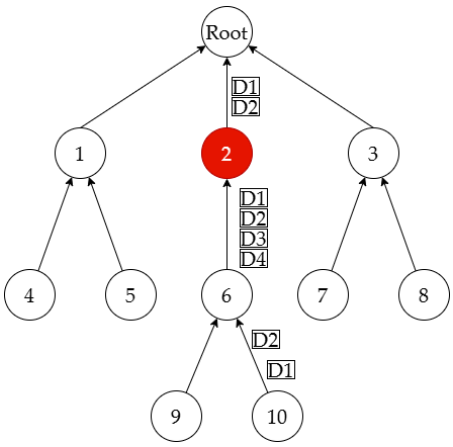

Figure 7: *Selective Forwarding Attack*

Figure 7 shows node 2 forwarding packets D1 and D2 but dropping D3 and D4, disrupting the path between node 6 and the root. Because RPL relies on intermediate nodes to forward traffic, a compromised node can appear normal while discarding selected packets. This degrades reliability, increases retransmissions, and reduces trust in routing paths.
Detection uses metrics such as Packet Delivery Ratio, RSSI, LQI, packet loss, retransmission rate, sequence numbers, ACK rate, hop count, and trust scores. Forwarding paths are analyzed for inconsistencies. Mitigation includes redundant routing and validating forwarding behavior with cryptographic checks.

II. Data collection

During simulation, traffic was captured in real time with Cooja's packet sniffer and stored as PCAP files. These files logged node communication, including normal packets and attack traffic. Wireshark and PyShark were used to extract protocol fields from 6LoWPAN, IPv6, ICMPv6, and RPL headers. Each PCAP file was then converted into a CSV format, where rows represent packets and columns represent features such as icmpv6.rpl.dao.sequence, wpan.seq_no, ipv6.hlim, and data.len. Finally, datasets were labeled by scenario: "normal" for regular traffic and attack-specific tags for malicious packets.

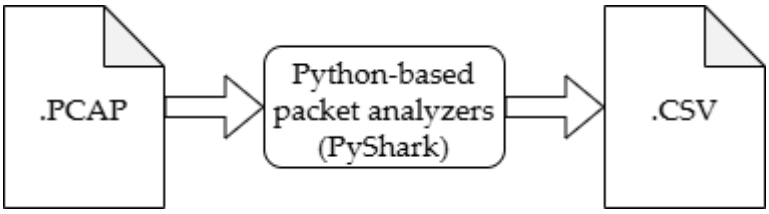

Figure 8: *extract feature from. PCAP and save in .CSV*

III. Data cleaning and preprocessing

After converting the raw PCAP traffic data into structured CSV format, a comprehensive data cleaning and preprocessing phase was undertaken to prepare the dataset for machine learning and deep learning model training.

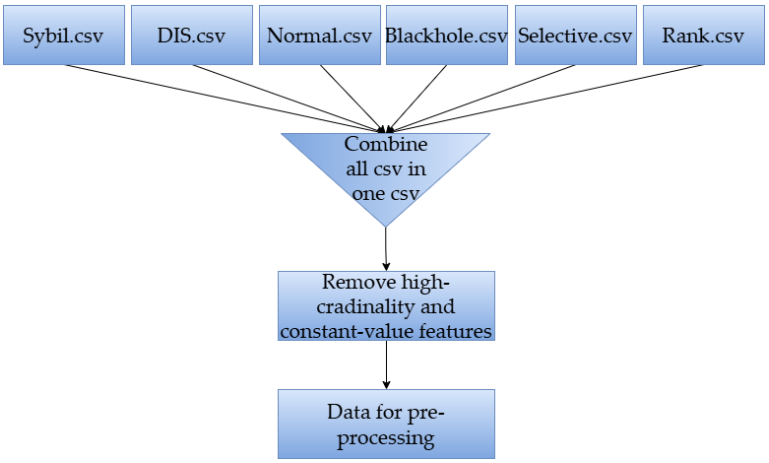

Figure 9: *Illustrated the removed high-cardinality and constant-value*

Figure 9 shows the merging of multiple CSV files, each containing a different attack type, into a single dataset. High-cardinality and constant-value features, which add little to classification and risk overfitting, were removed. The resulting cleaned dataset was then used for further processing.

IV. Data balancing

Figure 10 shows the imbalance between normal and attack classes in the dataset. Sybil and DIS dominate with nearly 100,000 and 80,000 samples, while Black Hole, Selective Forward, and Rank each have fewer than 30,000. Such disparity biases models toward majority attacks, reducing detection accuracy for underrepresented ones like Rank or Black Hole.

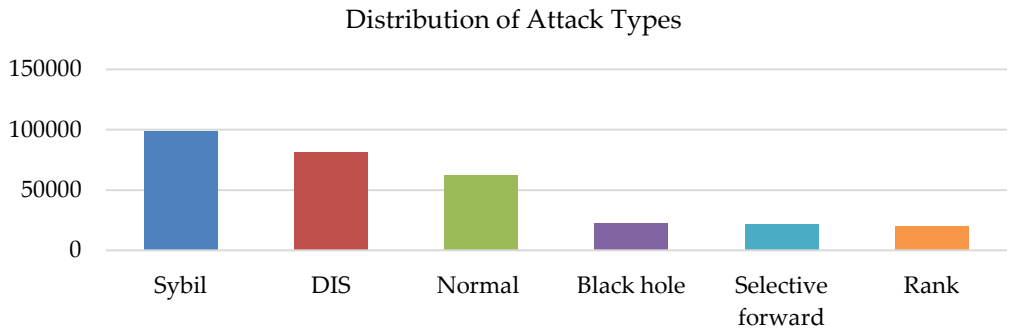

Figure 10: *Distribution of network records by attack type*

To overcome class imbalance, SMOTE (Synthetic Minority Oversampling Technique) was used in preprocessing. Instead of simply duplicating data, SMOTE creates new samples for minority classes by interpolating between nearby points. This adds variety, balances the dataset, and helps the IDS recognize subtle attack patterns that appear less often. With SMOTE, detection accuracy and fairness across different attack types improve (Dablain et al. [15]).

V. Feature Engineering and Selection

Feature engineering is key to building an IDS in RPL-based IoT networks. After converting PCAP files to CSV, categorical fields (e.g., address modes, flags, message types) were encoded as numbers, and booleans converted to binary values. This allowed features to be processed consistently by ML and DL models.

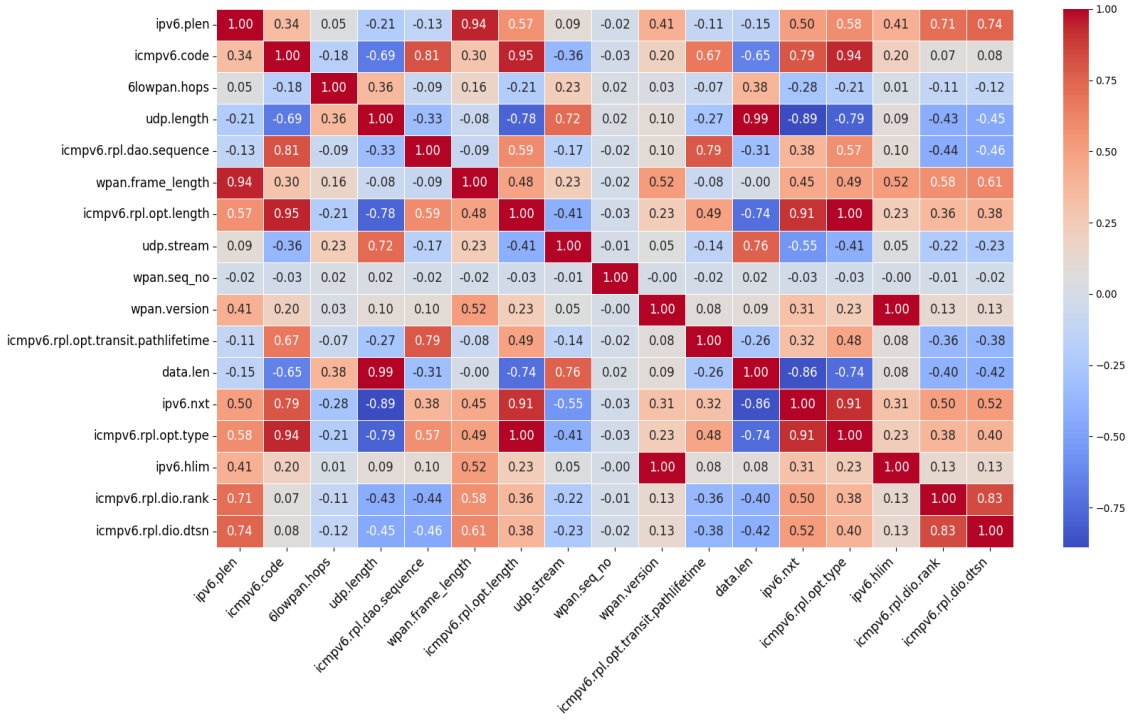

Figure 11: *Relationship depiction between network features.*

Figure 11 shows that some fields, such as icmpv6.code, icmpv6.rpl.opt.length, and icmpv6.rpl.opt.type, are highly correlated, while others like udp.stream and wpan.seq_no are independent. This helps remove redundant attributes and improve model efficiency.

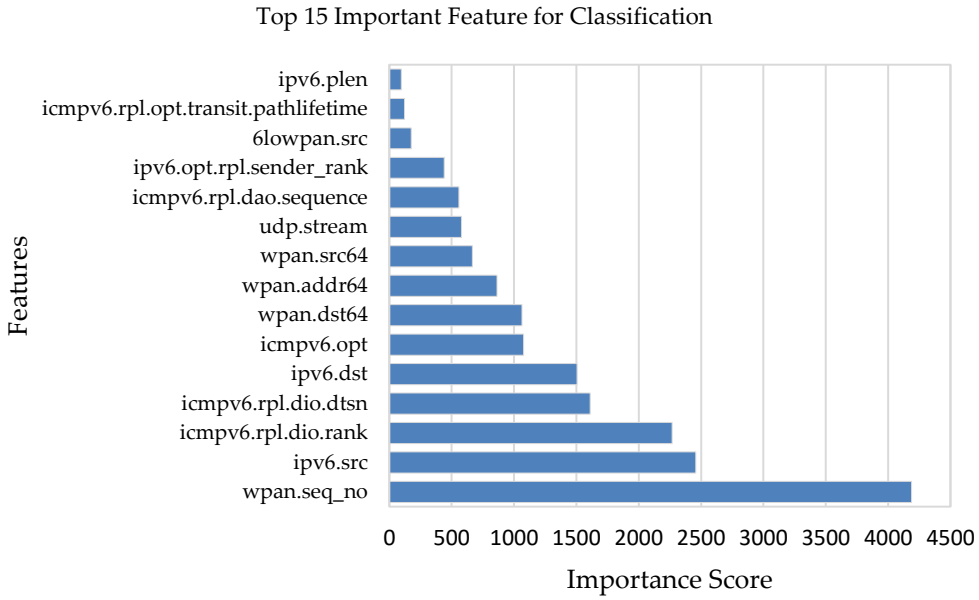

Figure 12: *Illustration of top 15 features*

As shown in Figure 12, features such as wpan.seq_no, ipv6.src, and icmpv6.rpl.dio.rank strongly influence classification. Address fields (wpan.addr64, wpan.src64) and routing parameters (icmpv6.rpl.dao.sequence, ipv6.opt.rpl.sender_rank) also help capture node activity and routing errors.

Outliers were detected using the Interquartile Range (IQR) method, where values beyond 1.5×IQR from Q1 or Q3 are flagged. Figure 13 shows that fields like wpan.seq_no, icmpv6.rpl.dio.rank, and udp.stream display outliers, which may indicate abnormal network behavior.

$$Q_1 = \left(\frac{n+1}{4}\right)^{th} \quad (1)$$

$$Q_3 = \left(\frac{3(n+1)}{4}\right)^{th} \quad (2)$$

Lower bound: Q1 - (1.5 * IQR)

Upper bound: Q3 + (1.5 * IQR)

In Figure 13, some features such as wpan.seq_no, icmpv6.rpl.dio.rank, icmpv6.rpl.dao.sequence, icmpv6.rpl.opt.transit.pathlifetime, and udp.stream show clear outliers. These patterns suggest unusual network behavior, which can help in spotting attack traffic in IoT routing

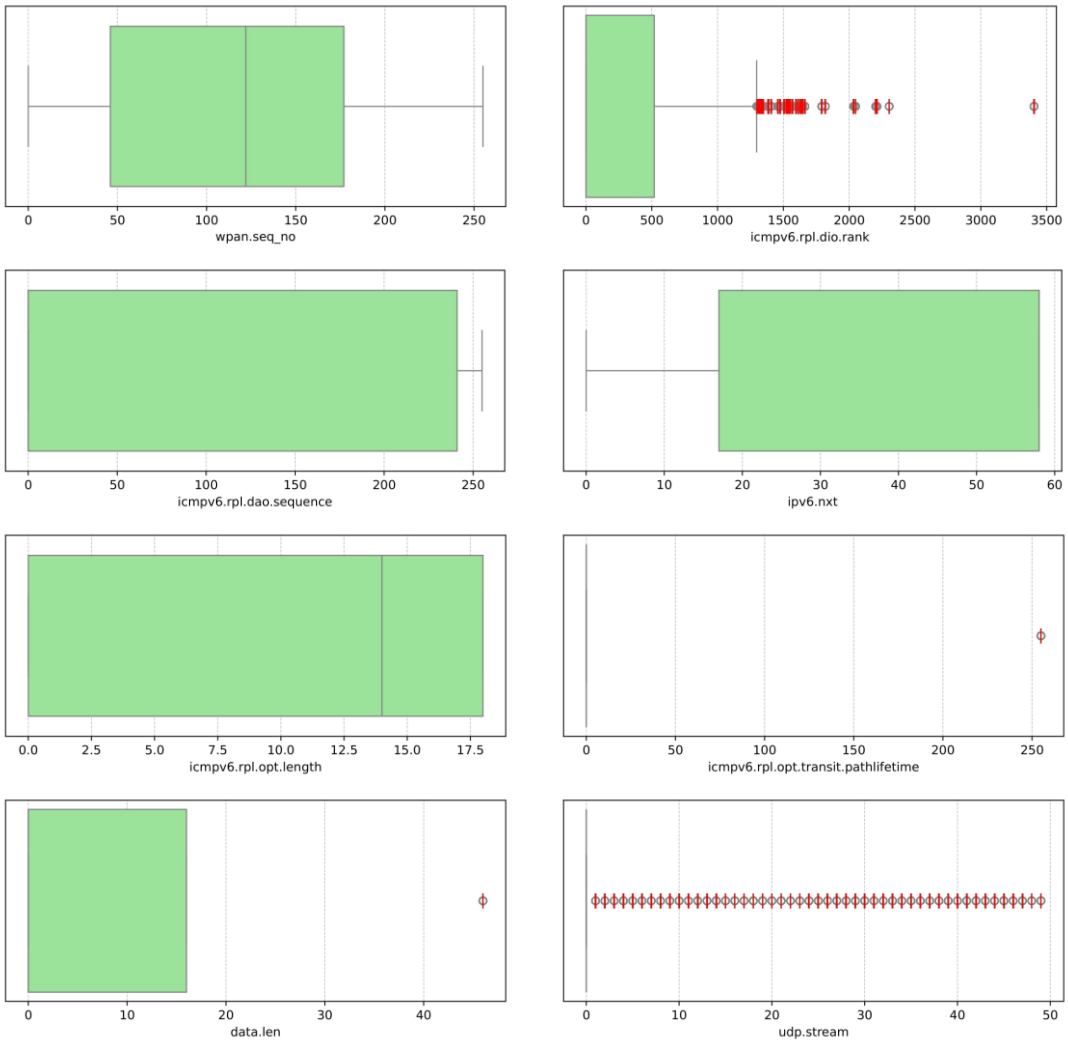

Figure 13: *Illustration of outliers and quartiles in features*

The violin plot in Figure 14 shows how packet lengths are spread under both normal and attack traffic in the IoT network. In the Attack category, the packet lengths display a multi-modal distribution, with peaks around 0.4, 0.7, and 0.9, suggesting a high variability in packet sizes during attack scenarios. In contrast, the Normal category shows a more concentrated distribution, especially around the 0.85–0.95 range, indicating more consistent and uniform packet sizes under normal network conditions.

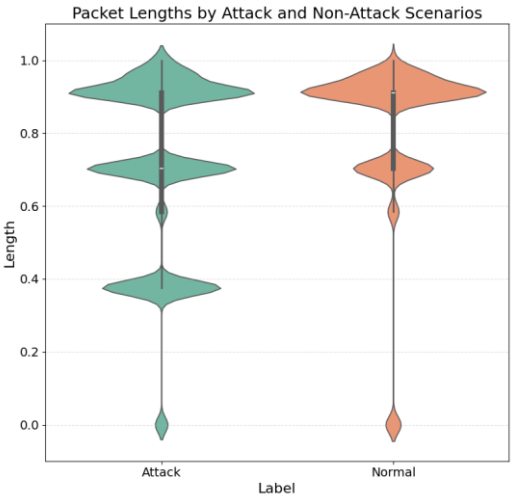

Figure 14: *Comparative illustration of packet lengths in attack and non-attack scenarios.*

VI. Model architecture design/selection

Table 2: *Classification of Detection Models*

Category	Model	Purpose
Tree-Based Models	- LightGBM - Decision Tree - Random Forest	Capture non-linear relationships and provide interpretable feature importance; efficient for tabular IoT data.
Linear / Optimization-Based Models	- Stochastic Gradient Descent (SGD) - Gaussian Naive Bayes(GNB)	Provide baseline comparison; fast and simple models for probabilistic and linear separation.
Neural Network Models	- Recurrent Neural Network (RNN) - Feedforward Neural Network (FFNN) - Convolutional Neural Network (CNN) - Long Short-Term Memory (LSTM)	Learn temporal and spatial patterns in sequential packet data; enhance learning from complex traffic behaviors.
Hybrid / Composite Models	- CNN + LSTM Hybrid - CNN + LSTM + Attention + LightGBM + CatBoost + MLP	Combine strengths of deep learning and machine learning; improve detection accuracy, generalization, and deployment efficiency.

VII. Model training

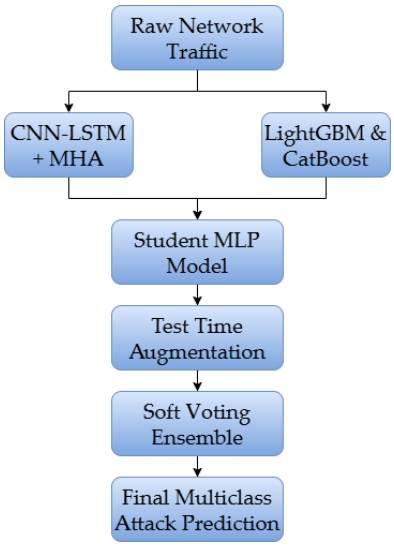

Figure 15: *Illustration of top 15 features*

Figure 15 shows the training and inference workflow. Raw traffic was processed in parallel by CNN-LSTM with Multi-Head Attention, LightGBM, and CatBoost. Predictions from these models acted as teachers for knowledge distillation into a lightweight MLP. During inference, Test Time Augmentation was applied, and final outputs were obtained with soft voting, combining model

strengths for multiclass attack detection.

Table 3: *Deep leaning models training time*

Model	Epochs	Ave. time pre-Epoch (Sec)	Total time (mins)
FFNN	50	4	3.33
CNN	50	11	9.1667
LSTM	50	17	14.1667
RNN	50	10	8.333
CNN+LSTM	50	9	7.5
CNN+LSTM+ Attention, LightGBM, CatBoost, MLP	35	75	43.75

IV. Results

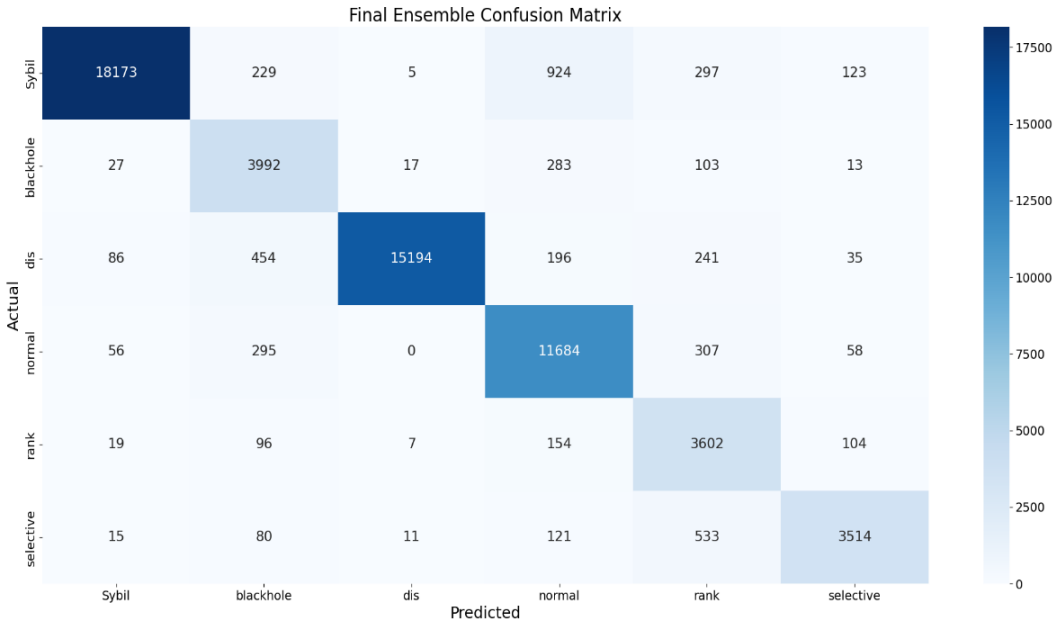

Figure 16: *Illustration of Confusion matrix*

The confusion matrix presented above evaluates the performance of the final ensemble model used for classifying various RPL routing attacks and normal traffic in IoT networks. In figure 16, the diagonal elements represent the number of correct predictions for each class, while the off-diagonal elements indicate misclassifications.

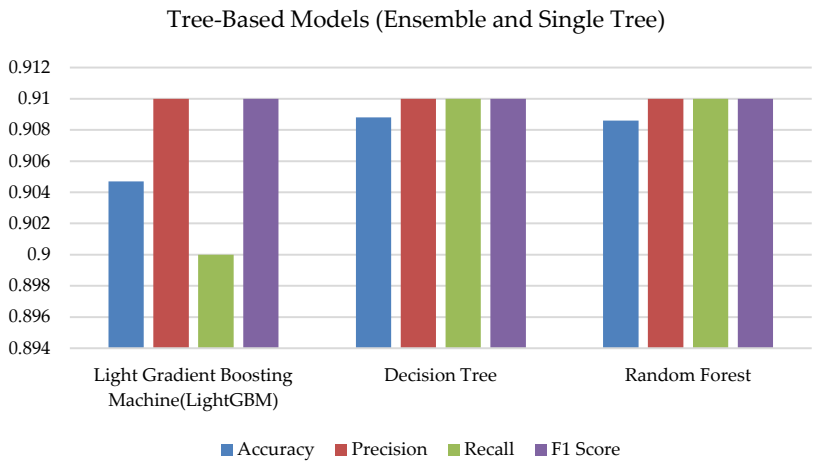

Figure 17: *Chart of accuracy, precision, recall, F1-score of Tree-based models*

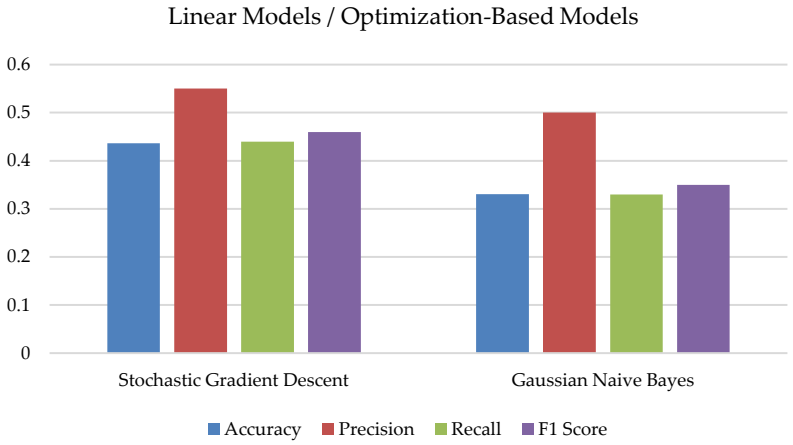

Figure 18: *Chart of accuracy, precision, recall, F1-score of Linear models*

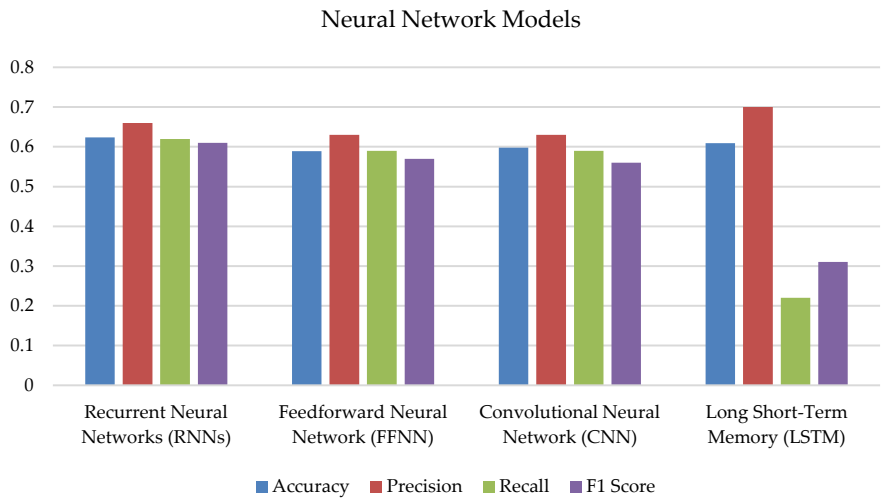

Figure 19: *Chart of accuracy, precision, recall, F1-score of Neural network models*

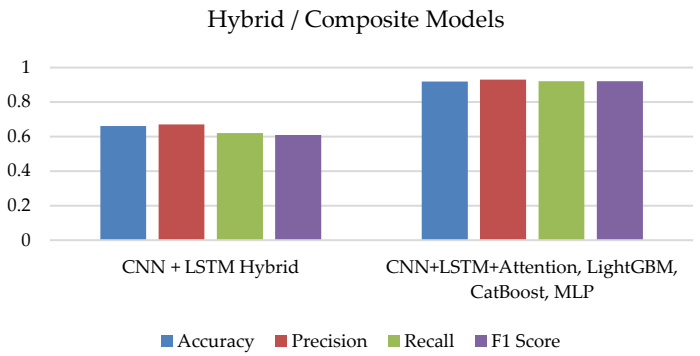

Figure 20: *Chart of accuracy, precision, recall, F1-score of hybrid/composite model*

The figures compare four model categories: tree-based, linear, neural networks, and hybrid models, using accuracy, precision, recall, and F1 score. Decision Tree and Random Forest perform well across all metrics, while LightGBM trails slightly in recall. Linear models such as Stochastic Gradient Descent and Gaussian Naive Bayes perform poorly, showing they are less suited for this task. Neural networks (RNN, FFNN, CNN, LSTM) give moderate results, with LSTM reaching high precision but low F1 due to weak recall. The strongest results come from hybrid models, especially CNN+LSTM+Attention with LightGBM, CatBoost, and MLP, which achieve near-perfect balance. This shows that combining deep learning with boosting methods gives the best accuracy for RPL attack detection.

Table 4: *Illustrate the accuracy between mobility and without mobility attack*

Model	Accuracy with mobility	Accuracy without mobility
Decision Tree	0.9088	0.913
Random Forest	0.9086	0.9132
LightGBM	0.9047	0.9155
RNN	0.6236	0.6316
LSTM	0.6094	0.6152
CNN	0.5979	0.6172
FFNN	0.5894	0.5923
Stochastic Gradient Descent	0.4363	0.597
Gaussian Naive Bayes	0.3304	0.3784
CNN+LSTM+Attention, LightGBM, CatBoost, MLP	0.9196	0.9326
CNN + LSTM Hybrid	0.6604	0.6616

V. Conclusion

This work presented an Intrusion Detection System (IDS) for finding and mitigating routing attacks in RPL-based IoT networks. The system was built with both deep learning and machine learning models, making it practical and able to scale for IoT setups. Authors fabricated a realistic RPL IoT environment in the Cooja simulator to record normal and attack traffic.To solve the problem of class imbalance in IoT datasets, SMOTE was used to generate extra samples for minority classes. The detection setup tested different models: LSTM for sequential traffic patterns, LightGBM and CatBoost for tabular data, and a simple MLP for combining outputs. Authors also used Test Time Augmentation (TTA) during prediction to improve stability. Results show that this mixed approach

improves accuracy and works across multiple attack types, including blackhole, Sybil, DIS flooding, rank, and selective forwarding. With the highest accuracy of 93.26% with mobility and 91.96% without mobility for CNN+LSTM+Attention, LightGBM, CatBoost, and MLP algorithms, this hybrid intrusion detection system (IDS) shows how well deep learning and machine learning models can be combined to detect and mitigate RPL routing attacks. It is a scalable and useful solution for protecting IoT networks.

References

[1] Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. In IEEE Access (Vol. 7). doi: 10.1109/ACCESS.2019.2924045

[2] Prajapati, A. K., Pilli, E. S., Battula, R. B., Varadharajan, V., Verma, A., & Joshi, R. C. (2025). A comprehensive survey on RPL routing-based attacks, defences and future directions in Internet of Things. *Computers and Electrical Engineering*, *123*. doi: 10.1016/j.compeleceng.2025.110071

[3] 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

[4] Yusuf YAVUZ, F., Ünal, D., & Gül, E. (2018). *Deep Learning for Detection of Routing Attacks in the Internet of Things*.

[5] Vatambeti, R., & Mamidisetti, G. (2023). Routing Attack Detection Using Ensemble Deep Learning Model for IIoT. *Information Dynamics and Applications*, *2*(1), 31–41. doi: 10.56578/ida020104

[6] Gupta, M., Vashishth, T. K., & Verma, P. K. (2024). Machine learning and deep learning based intrusion detection for blackhole attacks in mobile ad-hoc networks. *Multidisciplinary Science Journal*, *6*(11). doi: 10.31893/multiscience.2024209

[7] ÇAKIR, S., & YALÇIN, N. (2021). Detection of DIS Flooding Attacks in IoT Networks Using Machine Learning Methods. *European Journal of Science and Technology*. doi: 10.31590/ejosat.1014917

[8] Neerugatti, V., & Reddy, A. R. M. (2019). Machine learning based technique for detection of rank attack in RPL based internet of things networks. *International Journal of Innovative Technology and Exploring Engineering*, *8*(9 Special Issue 3), 244–248. doi: 10.35940/ijitee.I3044.0789S319

[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] Murali, S., & Jamalipour, A. (2020). A Lightweight Intrusion Detection for Sybil Attack under Mobile RPL in the Internet of Things. *IEEE Internet of Things Journal*, *7*(1), 379–388. doi: 10.1109/JIOT.2019.2948149

[11] Yu, M., Zhuge, J., Cao, M., Shi, Z., & Jiang, L. (2020). A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices. *Future Internet*, *12*(2). doi: 10.3390/fi12020027

[12] Ahmed, F., & Ko, Y. B. (2016). Mitigation of black hole attacks in Routing Protocol for Low Power and Lossy Networks. *Security and Communication Networks*, *9*(18), 5143–5154. doi: 10.1002/sec.1684

[13] Verma, A., & Ranga, V. (2020). Mitigation of DIS flooding attacks in RPL-based 6LoWPAN networks. *Transactions on Emerging Telecommunications Technologies*, *31*(2). doi: 10.1002/ett.3802

[14] Almusaylim, Z. A., Jhanjhi, N. Z., & Alhumam, A. (n.d.). *Detection and Mitigation of RPL Rank and Version Number Attacks in the Internet of Things: SRPL-RP*. doi: 10.3390/s20215997

[15] Dablain, D., Krawczyk, B., & Chawla, N. V. (2023). DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(9). doi: 10.1109/TNNLS.2021.3136503