

Tailored machine learning-based IDS for RPL-based IoT networks with feature optimization and attack profiling using a novel dataset

Mehdi Rouissat^{1,2}, Hichem Sid Ahmed Belkhira³, Mohammed Belkheir^{2,4},
Allel Mokaddem^{2,4}, Djamila Ziani^{2,4}, Ibrahim S. Alsukayti⁵

Securing RPL-based Internet of Things (IoT) networks remains a critical challenge due to the limited computational and energy resources of sensor nodes, making them vulnerable to routing attacks. This paper proposes a light-weight Intrusion Detection System (IDS) for the detection of five prominent RPL-specific attacks, namely Version Number Attack, DIS Flooding, Reduced Rank, Worse Parent, and Local Repair Attack. Unlike previous work that employed traditional datasets, a novel dataset is generated by simulating actual real-world RPL-based IoT scenarios, encompassing both normal and malicious behaviors. The dataset contains 13 features representing network behavior that were carefully selected to reflect the characteristics of RPL communications. Five machine learning models consisting of K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), XGBoost, and LightGBM were trained and evaluated using both the entire feature set and a subset of the top 7 most informative features. LightGBM performed the best among them, with 99.47% accuracy, 99.58% precision, 99.37% recall, and 99.47% F1-score. Even with fewer features, it still yielded robust results, confirming its robustness and viability in constrained environments. The proposed architecture offers a feasible and efficient solution to enhancing RPL security in real-world IoT networks. The proposed framework highlights the potential of tailored machine learning models and original datasets to detect protocol-specific threats while respecting the limitations of low-power networks.

Keywords: IoT, security, IDS, machine learning, edge computing, RPL, Cooja

1 Introduction

The proliferation of the Internet of Things (IoT) has significantly reshaped the digital landscape by enabling a vast number of devices to communicate, share data, and interact autonomously. From smart cities and intelligent healthcare systems to industrial automation and agriculture, IoT networks are now deeply embedded in critical infrastructures. However, this ubiquity comes with an escalating security cost. IoT networks are especially vulnerable at the edge layer, where devices with constrained resources limited in memory, processing power, and energy are often deployed in open and untrusted environments. These constraints hinder the implementation of conventional security mechanisms such as strong encryption, complex authentication schemes, and real-time anomaly detection, leaving edge nodes exposed to a wide array of cyber threats [1]. The use of IPv6-based communication protocols like RPL (Routing Protocol for Low-Power and Lossy Networks) in such environments further introduces protocol-specific vulnerabilities that attackers can exploit with minimal effort [2]. Intrusion Detection

Systems (IDS) play a vital role in protecting IoT networks by continuously monitoring traffic and device behavior to identify anomalous patterns indicative of cyberattacks [3]. Unlike preventive solutions such as firewalls, IDSs offer the advantage of detecting zero-day threats and previously unknown attack vectors. To address the precited concerns, the proposed work introduces a Machine Learning (ML)-based IDS tailored for RPL-based IoT networks, with a particular focus on the edge. This IDS is designed to detect five of the most prominent RPL-specific attacks: Version Number Attack (VNA), DIS Flooding Attack, Decreased Rank Attack, Worse Parent Attack, and Local Repair Attack. These attacks can severely disrupt the topology and functionality of IoT networks by manipulating routing information, exhausting network resources, and degrading Quality of Service (QoS) [4].

In this study, we built a comprehensive dataset from a controlled simulation environment, capturing the behavioral patterns of the network under both normal and attack scenarios. A total of 13 distinct network behavior features were extracted, encompassing metrics related

¹ STIC Laboratory, University Aboubekr Belkaid, Tlemcen 13000, Algeria

² Nour Bachir University Center, El-Bayadh, Algeria

³ ENS Saida, Algeria

⁴ LIMA Laboratory, Nour Bachir University Center El-Bayadh, Algeria

⁵ Department of Computer Science, College of Computer, Qassim University, Buraydah, 51452, Saudi Arabia
m.rouissat@cu-elbayadh.dz

to routing control messages, energy consumption, parent selection, and packet transmission rates. These features serve as inputs to five well-known machine learning classifiers: K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), XGBoost, and LightGBM. Each classifier was trained using both the full feature set and a reduced subset consisting of the 7 most informative features, identified through feature importance analysis. By leveraging the power of machine learning and focusing on attack detection at the edge, this work aims to enhance the security and resilience of IoT networks against evolving threats. The findings of this study contribute to the development of intelligent and scalable IDS solutions that can be effectively integrated into real-world IoT deployments.

2 Related work

Intrusion Detection Systems (IDS) in IoT environments have garnered extensive attention, particularly with the proliferation of edge-based architectures and RPL (Routing Protocol for Low-power and Lossy Networks) deployments. In recent years, various approaches have been proposed to detect network threats, either based on general IoT behavior or focusing specifically on routing-layer vulnerabilities in RPL-based systems. The authors in [5] introduced an ensemble-based IDS that combines auto encoders and deep learning classifiers, including Transformer, CNN, and LSTM models. Their system was designed to operate within fog-edge layers and was tested across three benchmark datasets: NSL-KDD, CICIDS2017, and TON-IoT. Although they achieved high accuracy exceeding 99%, their proposed IDS was effective for typical IoT security events but did not consider the unique constraints or routing-specific threats that characterize RPL-based networks. In [6] the authors specifically addressed this gap by proposing a lightweight cooperative IDS tailored for RPL-based networks. Their system extracted 13 network-layer features such as rank change frequency, DIO interval variance, and packet drop rates. They implemented a lightweight artificial neural network (ANN) model and achieved 99.99% accuracy in detecting RPL-specific threats. Another recent work in [7] where the authors developed FLsec-RPL, a fuzzy logic-based IDS specifically targeting the DIO suppression attack, a subtle but disruptive RPL vulnerability. Their IDS used only four lightweight features derived from simulation and showed accuracy between 96% and 99% depending on the network scenario. However, the narrow focus on a single attack type and limited dataset scope posed challenges for generalization to more complex or composite threat environments. In [8], the authors proposed SecEdge, a novel deep learning-based cybersecurity framework tailored for mobile IoT environments, aiming to detect and mitigate cyber-

attacks in real time at the network edge. The approach integrates Transformer models, Graph Neural Networks (GNNs), and Federated Learning to provide a lightweight yet effective IDS capable of adapting to dynamic IoT environments. The authors in [9] introduced DTL-IDS, an optimized intrusion detection framework designed for Industrial IoT (IIoT) networks. It employs a deep transfer learning (DTL) approach, leveraging seven pre-trained CNN architectures (Xception, EfficientNet). Evaluated against 14 distinct cyber-attacks, DTL-IDS achieves 100% detection accuracy, demonstrating exceptional precision in a complex IoT context. However, its reliance on image transformation and ensemble-based deep learning demands significant computational resources. Liu et al. [10] adopted a different approach by developing an edge-based IDS focused on system-level behavioral profiling. Their edge-optimized solution achieved 91% accuracy but was tailored to host-based threats and required continuous device monitoring, making it less adaptable to large-scale multi-hop IoT networks using RPL. In [11], the authors proposed a hybrid IDS model combining Linear Discriminant Analysis (LDA) with Logistic Regression. Tested on the NSL-KDD dataset with 12 features, the system demonstrated 96.56% accuracy. While the results were promising, the approach remained within the realm of general IoT anomaly detection and did not consider routing-based attacks or constrained RPL topologies. In [12], the authors propose a real-time intrusion detection framework for IoT environments using deep learning techniques. The study leverages the CIC IoT-DIAD 2024 dataset, a comprehensive flow-based network traffic dataset encompassing both benign and malicious scenarios. However, the work remains generalized and does not address protocol-specific attacks such as RPL threats, which are more common in constrained IoT environments. Finally, in [13] the authors tackled IDS for smart metering systems using Isolation Forest and Decision Tree classifiers. Their study focused on real-time detection of false data injection attacks and cyber anomalies. Achieving an accuracy of around 95%, their model was validated using real-world smart grid datasets. However, the application domain was very specific to energy systems and didn't generalize well to broader IoT routing infrastructures or RPL-based deployments.

A comparison of these works is provided in Tab. 1. As illustrated, most of the existing IDS proposals rely on public datasets such as NSL-KDD, UNSW-NB15, IoT-23, and CICIDS, which do not accurately capture the unique features and traffic patterns of RPL-based IoT networks. Furthermore, few studies attempt to simulate RPL-specific threats or generate contextually rich datasets that reflect the nuanced behavior of routing-layer attacks.

Table 1. Comparative summary of recent IDS approaches for IoT networks

Ref.	Dataset used	Features	Attacks detected	Algorithms used	Accuracy (%)
[5]	NSL-KDD, CICIDS, TON-IoT	Not stated	General IoT attacks (DDoS, spoofing)	CNN, LSTM, Transformer ensemble	>99%
[6]	UNSW-NB15, TON-IoT	Not stated	General network intrusions	Modified Random Forest	>99%
[7]	NSL-KDD	12	General anomalies	LDA, Logistic Regression	96.56%
[8]	CIC IoT-DIAD 2022	Not stated	General IoT attacks	Deep Learning (e.g., CNN, LSTM)	~99%
[9]	Custom Cooja RPL Simulations	13	6 RPL-specific attacks	DT, RF, SVM, KNN, etc.	99.99%
[10]	Simulated federated RPL network	4	RPL attacks (e.g., Rank, DIS)	Federated learning classifier	96-99%
[11]	Contiki/Cooja network logs	Not stated	RPL routing and firewall rule evasion	Routing logic + local firewall rules	Not reported
[12]	CIC IoT-DIAD 2024	Flow-based	General IoT anomalies	CNN, LSTM, RNN, MLP	Binary: 99.53%, Multi: 99.12%
[13]	Real AMI smart meter deployment	Not stated	False data injection, cyber anomalies	Isolation Forest + Decision Tree	~95%
This work	Custom RPL-based Simulated Dataset	13 → 7	VNA, DIS Flooding, Decreased Rank, Worse Parent, Local Repair	KNN, DT, RF, XGBoost, LightGBM	Up to 99.47%

Unlike the majority of related works that rely on general-purpose datasets with limited relevance to routing-layer threats, our proposed IDS is trained on a custom-generated dataset derived from detailed simulations of RPL-based IoT networks. The dataset includes 13 carefully selected features that capture the nuanced behavior of IoT nodes and reflect real-world traffic dynamics. These features were chosen to ensure high relevance to routing-layer anomalies while remaining computationally feasible for edge deployment. To further enhance the effectiveness of the detection process, we employed a diverse set of machine learning classifiers, including K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), XGBoost, and LightGBM. This ensemble of algorithms not only ensures robust detection across different attack types but also allows comparative analysis to identify trade-offs between performance and computational efficiency. As such, the proposed IDS offers a practical and accurate solution tailored to the security needs of RPL-based IoT environments.

The main contributions of this work are threefold. First, we provide a rich and labeled dataset that captures fine-grained IoT routing behavior under multiple RPL attack scenarios. Second, we evaluate and compare the performance of multiple ML classifiers, analyzing their accuracy, precision, recall, and computational efficiency under full and reduced feature settings. Third, we emphasize the feasibility of deploying ML-based IDSs at the edge, even with limited resources, by selecting lightweight models and optimizing feature selection to reduce overhead without compromising detection accuracy.

Additionally, in contrast to many existing IDS frameworks that require distributed deployment or introduce computational overhead on resource-constrained nodes, the proposed IDS architecture is designed to operate exclusively at the sink node of the RPL-based IoT network. The sink node, being significantly more capable in terms of computational power and memory, is well-suited to handle the processing demands of machine learning-based intrusion detection

without impacting the performance of other nodes. This centralized deployment strategy ensures that the sensor and intermediate nodes remain unaffected by IDS-related tasks, preserving their limited resources and prolonging network lifetime. As a result, the proposed approach offers a highly efficient and lightweight security solution, particularly suitable for low-power and lossy networks (LLNs) commonly found in IoT applications relying on RPL routing protocols.

3 The proposed IDS system

With the advent of edge computing, the devices at the edge of the network, thus devices closest to the data source, are becoming smart devices that first process data locally before sending it to the centralized cloud servers. Thus, edge computing can reduce communication latency, lessen network burdens, and improve data privacy and security by processing data locally and in accordance with data protection regulations. However, the main challenge of edge computing is the limited resources and power of edge devices. Thus, equipping one intrusion detection agent in each IoT device is not feasible due to their limited resources.

As a solution to this issue, in this paper we adopt a centralized IDS based framework where the IDS is deployed on the sink node, which is the most powerful and resourceful node in the network. The proposed IDS is powered by machine learning algorithms that leverage the ability to learn complex patterns in network behaviors to identify malicious behavior from normal behavior. A machine learning based IDS can exploit network data as well as real-time network data flow to allow it to detect RPL specific attacks. This enhances both the performance and agility of security as it constantly adapts in dynamic IoT settings.

Figure 1 depicts the activity diagram of the proposed IDS, showing the overall flow from the collection of the dataset to the training and testing of ML models. The goal of the IDS is to detect intrusions based on differences between normal and attack behavior of the nodes. Once an intrusion is identified, the system broadcasts the malicious node's ID to the network in DIO messages. This alert allows any child nodes of the attacker to disconnect and choose a new other parent node.

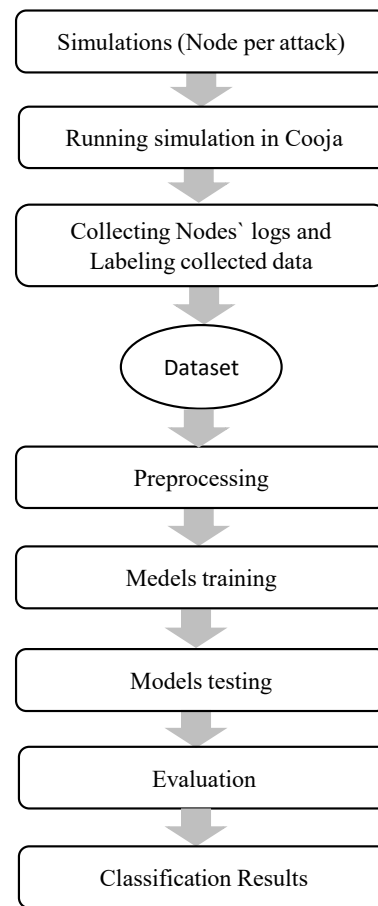


Fig. 1. The activity diagram of the proposed IDS

The IDS process can be divided into three stages:

- **Behavior monitoring:** In this stage the IDS collects data on IoT nodes behavior over the network. This data is then processed to create a feature vector that represents the current status of the system.
- **Intrusion detection:** The stored data on feature vectors is then used to identify outliers or anomalies from normal behavior. This is the main detection part of the IDS. This phase is described in more detail in the following section.
- **Alerting:** When an anomaly is verified as an attack, the system raises an alert and broadcasts the ID of the malicious node to the network in the DIO messages. This alert allows any child nodes of the attacker to disconnect and choose a new parent node that is trustworthy.

4 Dataset preparation

One major contribution of this paper is the creation and introduction of a new dataset, which is available at the following link:

<https://github.com/MehdiRouissat/TailoredIDS>

This dataset aims to address the lack of open access resources supporting empirical studies on security in RPL-based IoT networks. The dataset is based on five various on/off attacks: VNA, DIS flooding, Decreased Rank Attack, Worse Parent and local repair Attack. Based on the proposed implementation, all the malicious nodes conducting the different 5 attacks adopt a time-switching approach, switching between being a legitimate node during a given time interval and between being a malicious node during another given time interval, respecting a defined pattern. Following these behaviors, the malicious nodes follow a less detectable threat profile, which allows to test our IDS in different networks behaviors. This cyclical behavior is respected by all the malicious nodes. Each attacking node possesses its own individual attack strategy but respects the same general principle of periodically halting its malicious behavior, which allow to reduce the detectability of the attacks by limiting the time period of malicious behavior.

The simulation thus encompasses ten recursive phases: Normal Operation, VNA-Active, Normal Operation, Decreased Rank Attack-Active, Normal Operation, DIS-Active, Normal Operation, Worst Parent-Active, Normal Operation, Local Repair-Active and Normal Operation, then the same pattern will start over, as illustrated in Fig. 2. Specifically, the VNA node switches between passive and active states every 6, 66, and 126 minutes (formula 1), maintaining this cycle throughout the simulation. The Decreased Rank Attack begins at minute 18, terminates after 6 minutes, and recommences every 60 minutes (formula 2). The DIS attack is initiated at 30 minutes, following a similar on/off strategy every 60 minutes. Similarly, the Worst Parent attack starts at 42 minutes, while local repair attack is initiated at 54 minutes.

$$T_{\text{attackVNA}} = 6 + (n \times 6), n = 1, 2 \dots N \quad (1)$$

$$T_{\text{attackDRA}} = 18 + (n \times 6), n = 1, 2 \dots N \quad (2)$$

where

$T_{\text{attackVNA}}$ are the instants when the VNA is scheduled,

$T_{\text{attackDRA}}$ are the instants when the DRA is scheduled.

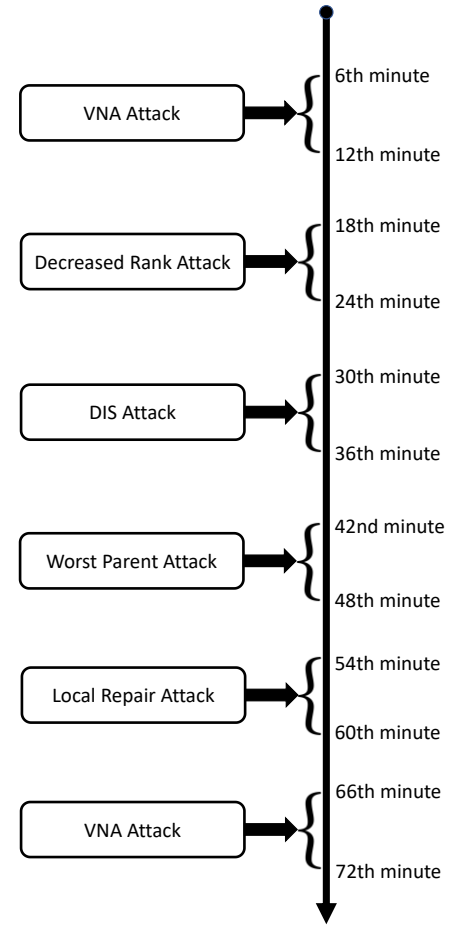


Fig. 2. The proposed cyclical behavior of the attacks

As shown in Fig. 2, the simulation is intended to delay the first attack by six minutes after the network start-up. The intended delay permits the network to settle into operation and stabilize, allowing it to be in a steady-state before conducting the first attack, and avoid collecting data during the warm-up period. When the six minutes of start-up concludes, the malicious node responsible for the VNA attack operates for a six-minute period. Every attack period is followed by an attack free period of 6 minutes. This cyclical attack pattern not only emulates realistic adversarial behavior but also ensures that system metrics are not noised by transient initialization effects.

The studied network consists of the 19 sensor nodes, plus a sink node, so there are 20 total nodes generating 11400 messages during the 20hours of simulation based on a transmission period of 120 seconds by each node. Statistics were recorded using the script editor of the Cooja simulator, which was configured to output messages, allowing for a streamlined extraction of the data [14]. The extraction resulted in a raw output that was saved as a file and converted to a CSV format as a MLplatform input file.

Table 2. Features collected from nodes, reported every 120 seconds

	Attributes	Description
1	Time	Time of sending the data vector
2	Id	Nodes`ID (number)
3	Sentvn	Advertised version number
4	mypparent	Parent of the node
5	pparentrank	Rank of the node`s parent
6	children	Number of the node`s children
7	sentDIS	Number of sent DIS messages
8	sentDIO	Number of sent DIO messages
9	generatedDAO	Number of generated DAO messages
10	forwardedDAO	Number of forwarded DAO messages
11	generatedNPDAO	Number of generated no-path DAO messages
12	forwardedNPDAO	Number of forwarded no-path DAO messages
13	Ttlsent	Total sent messages
14	receivedDIS	Number of received DIS messages
15	receivedDIO	Number of received DIO messages
16	receivedDAO	Number of received DAO messages
17	Ttlrcvd	Total received messages
18	Energy	Consumed energy

The output has 16 separate attributes that each detail some facet of network activity, as seen in Tab. 2, where the time, ID and mypparent, pparentrank and children are neglected when preparing the dataset for ML operation, which make 13 features in actual use. The dataset presented in this study was specifically generated for the application of machine learning techniques to improve security for RPL-based IoT networks. Most existing datasets (to the best of our knowledge) mostly limit their feature set to only a small number of features, our dataset provides a more comprehensive and rich representation of network behavior. In a large feature space, the dataset affords reflection of normal behavior and malicious behavior in more detail and depth and increases the robustness of machine learning models to better detect threats. Overall, this dataset strives for granularity and maintains a closer reflection of the operational reality of IoT deployments and is a substantial contribution to those researchers who wish to develop better performing and safer intrusion detection systems in the space of IoT security.

5 Obtained results

5.1 Performance metrics comparison

Table 3 compares the five classifiers KNN, DT, RF, XGB and LGBM in terms of their performance on the original dataset of 13 features and the performance based on only 7 features chosen by feature importance. For 13 features, all the models did well, LightGBM is the most excellent with an F1-score of 99.47%, followed closely by XGBoost at 99.43%, and Random Forest at 99.14%. KNN and Decision Tree follow with F1-scores of 98.00% and 97.33%, respectively. These attest to the strength of ensemble-based models for intrusion detection. When reducing to the top 7 features, LightGBM remains highly stable at 99.20% F1-score with a drop of just 0.27%. Decision Tree modestly enhances F1-score from 97.33% to 97.46%, which may suggest better generalization due to reduced overfitting. Random Forest also remains highly performing at 98.47% F1-score, falling only by 0.67%, suggesting immunity to dimensionality changes.

Table 3. Obtained results regarding the performance metrics

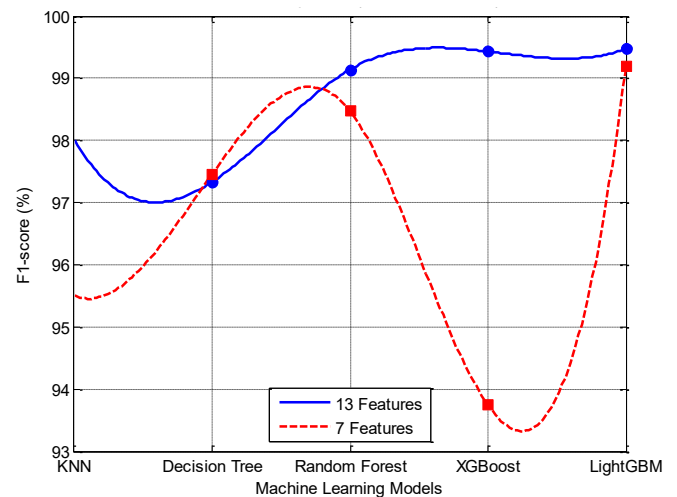
Model	Features	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	13	97.97	96.92	99.10	98.00
	7	95.51	95.63	95.38	95.50
XGBClassifier	13	99.43	99.49	99.37	99.43
	7	93.93	96.83	90.84	93.74
Random Forest	13	99.14	99.07	99.22	99.14
	7	98.47	98.27	98.68	98.47
Decision Tree	13	97.34	97.85	96.82	97.33
	7	97.46	97.53	97.39	97.46
LightGBM	13	99.47	99.58	99.37	99.47
	7	99.20	99.13	99.28	99.20

KNN meanwhile, presented a more noticeable drop, from 98.00% to 95.50%. Although still acceptable, this drop indicates KNN's sensitivity to feature space completeness. XGBoost presented the most drastic performance drop, with its F1-score falling from 99.43% to 93.74%, and recall from 99.37% to 90.84%. This indicates the likelihood that important feature interactions were lost, affecting its recall-based detection of attacks.

Despite the feature reduction, LightGBM continues to possess superior performance in all metrics: 99.20% accuracy, 99.13% precision, 99.28% recall, and 99.20% F1-score, proving its good generalization and tolerance to less input. This stability makes it very suitable for light-weighted time IDS deployment. Overall, reducing the number of features causes some degradation in performance in every model, but the severity of the impact varies between models, but LightGBM and Random Forest continue to show great, stable, and robust performance, and therefore are the most suitable for both low- and high-dimensional detection problems in constrained environments. Their ability to preserve performance irrespective of fewer inputs militates for their deployment in edge-computing setups typical of today's IoT security architectures.

Figure 3 highlights how each model manages the decreasing features space. LightGBM did not change much and its F1-score slightly dropped, showing its robustness even with limited features. Random Forest also showed good stability and reduced only a little bit. However, XGBoost showed a steeper drop in performance, suggesting that it relies on additional features to perform really well. KNN struggles as well, but again that was kind of expected as it rarely has any good results without the benefit of all the features. The only surprising one was the Decision Tree, which actually improved when features reduced. Overall,

models like LightGBM and Random Forest are not only robust (good performing) but they are very robust (stable performance) even when removing features.

**Fig. 3.** F1 Score comparison using 13 features and top 7 features**Table 4.** Confusion matrices

Model	TN	FP	FN	TP
KNN	3225	105	30	3300
XGBClassifier	3313	17	21	3309
Random Forest	3299	31	26	3304
Decision Tree	3259	71	106	3224
LightGBM	3316	14	21	3309

Table 4 summarizes the obtained results regarding the confusion matrix that has four key metrics True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) which allows to compare the performance of machine learning models for binary classification purposes. The TP (3309), TN (3313), FP (17), and FN (21) shows that the XGB Classifier has the most correct predictions, and the least mistakes compared to the other models, while also having great precision and recall. LightGBM comes in a close second with high TP and TN values but less FP and FN, indicative of equally good performance. On the other hand, Decision Tree is not good with a lot more false predictions (FP=71, FN=106) that may lead to extensive misclassifications. KNN, although balanced in TP and TN, has a very high number of FP (105), so it is less reliable for scenarios where false alarms are costly. Overall, the confusion matrix identifies the improved performance of ensemble-based models like XGBClassifier and LightGBM in this classification problem.

5.2 Feature sensitivity and impact

The best three used classifiers responded differently to feature reduction, which may be dependent on how much they base their assumptions on the data they had available.

5.2.1 XGBoost classifier

Figure 4 depicts Feature Sensitivity for the case of XGBoost classifier, this latter shows a noticeable reliance on certain features compared to other classifiers. Out of 13 features, it heavily relies on sent DIS (38.75%) and received DIS (20.42%). This means that nearly 60% of its predictive power comes from just these two. The other features, even those that are usually important like generated DAO and total energy, play a minor role.

The presence of only 7 features amplifies this trend even further – received DIS increases to 39.77%, while sent DIS does not lag behind at 24.82%. This might indicate that XGBoost wields great capabilities but becomes too reliant on a single narrow deep feature, this could account for the steep drop in performance when those patterns fail to generalize. The accuracy plummets from 99.43% to 93.93%, alongside recall suffering more than an eight percent dip. Despite being considered a robust gradient boosting algorithm, XGBoost turns a bit fragile in the absence of wider context.

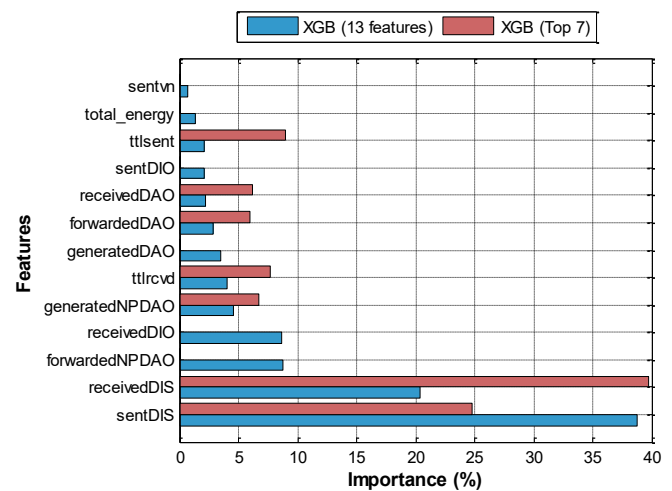


Fig. 4. Features impact in XGBoost classifier case

5.2.2 RF classifier

As Fig. 5 illustrates, the importance of Random Forest's top six features range from 7% to 14%, is distributed fairly evenly among its 13 features: ttlsvnt, received DIO, generated DAO, ttlrcvd, generated NPDAO, and sent DIO. Based on a comprehensive analysis of the dataset, this diversity suggests a healthy ensemble.

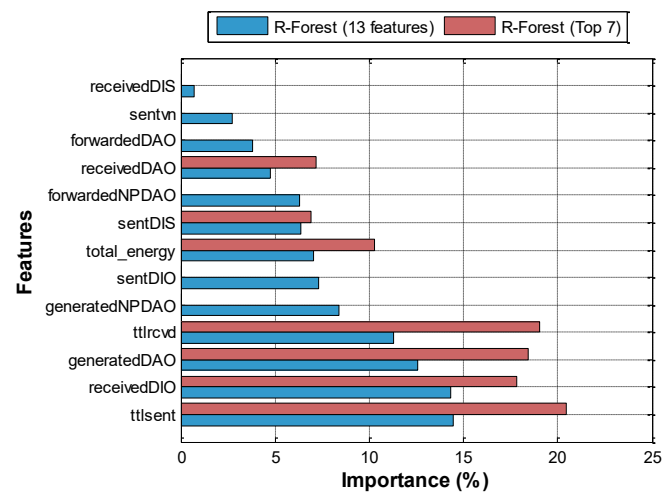


Fig. 5. Features impact in RF classifier case

The model focuses more on ttlsvnt, ttlrcvd, and generated DAO when it is reduced to seven features. It is interesting to note that the full set already placed a high value on these features. The reason for RF's performance only slightly declined (accuracy increased from 99.14% to 98.47%) is due to this consistency in key features.

5.2.3 LightGBM classifier

As Fig. 6 depicts, LightGBM shows the most evenly distributed reliance on features. All features are weighted moderately. It has 13 features. The most influential are `total_energy` (18.24%), `sentvn` (13.54%), `sentDIO` (10.87%) and `receivedDIO` (9.78%), features make up 5-6% for `ttlrcvd` and `received DAO` respectively. This suggests it is a well-regularized model.

When reduced to 7 features, the model exhibits the same role across features: `sent DIO` showed a 12% contribution, `total energy` (10%), `sentvn` (9%) and `ttlrcvd` (9%). The model did well not to have too much reliance from one feature. Although 6 features were removed, the drop in performance was small, ranging from 99.47% (all 13 features) to 99.20% accuracy (7 features, with 6 removed); this indicates LightGBM's ability to learn the importance of distributed patterns relatively quickly and efficiently. Given its consistency, LightGBM is an optimal algorithm for some use cases, which require the selection of features, such as real time IoT or streaming analytics.

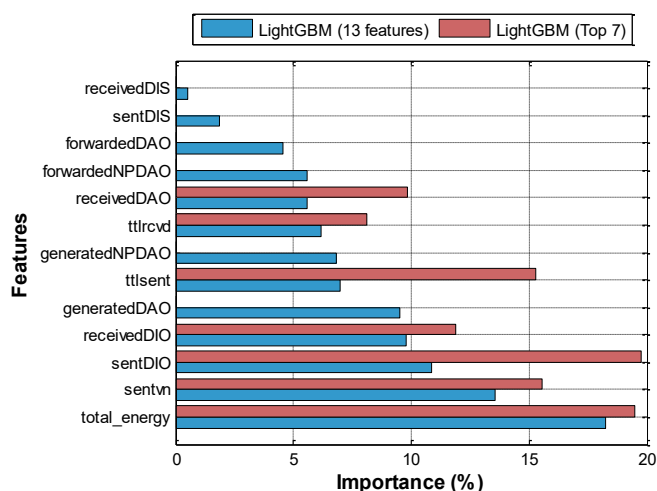


Fig. 6. Features impact in LightGBM classifier case

Overall, the comparison revealed that classifiers responded to feature reduction differently, which may be dependent on how much they base their assumptions on the data they had available. LightGBM (as previously noted), not only was more stable but turned in the most efficient results, in terms of consistent performance but more balanced use of features, maintaining a great value even when reduced to 7 features, making it an ideal classifier for a simplified, real-time detection approach. Similarly, Random Forest provided consistent results with every feature count. This is largely due to its ensemble-based learning and ability to generalize regardless of reduced or pruned features. Without much surprise, Decision Tree performed slightly better

as features were trimmed, but that value probably had little to do with information being lost and everything to do with disallowing low-impact information from creating definable overfitting. KNN, and XGBoost performance declined more noticeably. In the case of KNN, the closest point decision relies heavily on choice richness. It will likely perform poorly when representing data sub-dimensions and removes feature weighting as the document type occurs in more typical classification contexts. Conversely, even though XGBoost leveraged all four domains well, it was of little use when compared generally due to its order-dependence with lower features; we could say that it simply uses lower feature weights from all four documents too heavily. Thus, when working with fewer features may have less worth in determining meaningfulness all seekers will come to rely more heavily on a reasonable amount of scaled importance – LightGBM, and Random Forest – but LightGBM distributed importance so much better.

6 Conclusions

This work presented a machine learning-based Intrusion Detection System for RPL-based IoT networks, targeting five critical routing attacks: Version Number Attack, DIS Flooding, Decreased Rank, Worse Parent, and Local Repair. A primary contribution of this study is the development of a novel, custom dataset tailored to the specific behavior of IoT-RPL environments, capturing realistic and protocol-specific attack signatures. Another significant contribution is the comparative analysis of five classifiers KNN, DT, RF, XGBoost, and LightGBM trained and evaluated on both a full set of 13 features and a reduced set of the 7 most informative features. Feature importance analysis revealed that metrics such as `sentDIO`, `generatedDAO`, and `total_energy` were highly influential in detecting abnormal behavior. LightGBM consistently outperformed other models across all metrics accuracy, precision, recall, and F1-score even when using fewer features, making it ideal for lightweight, real-time IDS in constrained IoT environments. Random Forest also showed robust generalization, while Decision Tree benefitted from reduced overfitting under feature selection. In contrast, KNN and XGBoost were more sensitive to feature trimming, leading to performance degradation.

An important architectural design choice is that the IDS is deployed solely at the sink node, which is computationally more capable than other devices in the network. This ensures no computational or memory overhead on resource-constrained IoT nodes, preserving network efficiency. The high accuracy achieved is also supported by the well-labeled and diverse nature

of the dataset, effective feature selection aligned with RPL protocol behavior, and rigorous cross-validation that prevents overfitting or data leakage.

Future work will investigate dynamic feature selection, adaptive learning mechanisms, and hybrid ensemble models to improve the detection of complex and evolving RPL-based attacks.

References

- [1] M. Belkheir, M. Rouissat, A. Mokaddem, D. Ziani, and P. Lorenz, "Lightweight novel approach for collaborative packet-based mitigation of blackhole attacks in RPL-Based IoT," *Journal of Network and Systems Management*, vol. 33, no. 2, Feb. 2025, doi: 10.1007/s10922-025-09908-1.
- [2] A. Bang and U. P. Rao, "Impact analysis of rank attack on RPL-Based 6LOWPAN networks in internet of things and aftermaths," *Arabian Journal for Science and Engineering*, vol. 48, no. 2, pp. 2489–2505, Oct. 2022, doi: 10.1007/s13369-022-07342-y.
- [3] A. Heidari and M. A. J. Jamali, "Internet of things intrusion detection systems: a comprehensive review and future directions," *Cluster Computing*, vol. 26, no. 6, pp. 3753–3780, Oct. 2022, doi: 10.1007/s10586-022-03776-z.
- [4] M. Salayma, "Risk and threat mitigation techniques in internet of things (IoT) environments: a survey," *Frontiers in the Internet of Things*, vol. 2, Jan. 2024, doi: 10.3389/friot.2023.1306018.
- [5] M. Tawfik, "Optimized intrusion detection in IoT and fog computing using ensemble learning and advanced feature selection," *PLoS ONE*, vol. 19, no. 8, p. e0304082, Aug. 2024, doi: 10.1371/journal.pone.0304082.
- [6] H. Azzaoui, A. Z. E. Boukhamla, P. Perazzo, M. Alazab, and V. Ravi, "A lightweight cooperative intrusion detection system for RPL-based IoT," *Wireless Personal Communications*, vol. 134, no. 4, pp. 2235–2258, Feb. 2024, doi: 10.1007/s11277-024-11009-2.
- [7] C. Kim, C. So-In, Y. Kongsorot, and P. Aimtongkham, "FLSec-RPL: a fuzzy logic-based intrusion detection scheme for securing RPL-based IoT networks against DIO neighbor suppression attacks," *Cybersecurity*, vol. 7, no. 1, Sep. 2024, doi: 10.1186/s42400-024-00223-x.
- [8] K. A. Awan, I. U. Din, A. Almogren, A. Nawaz, M. Y. Khan, and A. Altameem, "SecEdge: A novel deep learning framework for real-time cybersecurity in mobile IoT environments," *Heliyon*, vol. 11, no. 1, p. e40874, Dec. 2024, doi: 10.1016/j.heliyon.2024.e40874.
- [9] S. Latif, W. Boulila, A. Koubaa, Z. Zou, and J. Ahmad, "DTL-IDS: An optimized Intrusion Detection Framework using Deep Transfer Learning and Genetic Algorithm," *Journal of Network and Computer Applications*, vol. 221, p. 103784, Nov. 2023, doi: 10.1016/j.jnca.2023.103784.
- [10] A. Mudgerikar, P. Sharma, and E. Bertino, "Edge-based intrusion detection for IoT devices," *ACM Transactions on Management Information Systems*, vol. 11, no. 4, pp. 1–21, Aug. 2020, doi: 10.1145/3382159.
- [11] P. Mahadevappa, R. K. Murugesan, R. Al-Amri, R. Thabit, A. H. Al-Ghushami, and G. Alkaws, "A secure edge computing model using machine learning and IDS to detect and isolate intruders," *MethodsX*, vol. 12, p. 102597, Feb. 2024, doi: 10.1016/j.mex.2024.102597.
- [12] Md. A. Hossain, "Deep learning-based intrusion detection for IoT networks: a scalable and efficient approach," *Research Square*, Mar. 2025, doi: 10.21203/rs.3.rs-6042512/v1.
- [13] K. Naveeda and S. M. H. S. S. Fathima, "Real-time implementation of IoT-enabled cyberattack detection system in advanced metering infrastructure using machine learning technique," *Electrical Engineering*, vol. 107, no. 1, pp. 909–928, Jul. 2024, doi: 10.1007/s00202-024-02552-z.
- [14] M. Rouissat, M. Belkheir, and H. S. A. Belkhira, "A potential flooding version number attack against RPL based IOT networks," *Journal of Electrical Engineering*, vol. 73, no. 4, pp. 267–275, Aug. 2022, doi: 10.2478/jee-2022-0035.

Received 28 April 2026