

Routing Layer Attacks in Low-Power IoT Networks: A Topology- and Mobility-Aware Dataset Generation and Machine Learning Benchmarking Study

Mahmoud Mohammed Yousef elbaradie

`mahmoud.elbarady@science.tanta.edu.eg`

Tanta University

afaf Mousa

Menofia University

Gamal Farouk

Menofia University

Article

Keywords: Internet of Things, routing security, machine learning, dataset generation, RPL, anomaly intrusion detection, network topology, mobility

Posted Date: August 25th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10284642/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Routing Layer Attacks in Low-Power IoT Networks: A Topology- and Mobility-Aware Dataset Generation and Machine Learning Benchmarking Study

Mahmoud elbaradie^{1*}, Afaf Mousa², and Gamal Farouk²

1- Computer Science dept., Faculty of Science, Tanta University, Egypt

2- Computer Science dept., Faculty of Computers and Information, Menofia University, Egypt

*Corresponding author: Mahmoud elbaradie

{ mahmoud.elbarady@science.tanta.edu.eg, gamalfaruk@yahoo.com,
afafmousa@yahoo.com }

Abstract—The fast proliferation of Internet of Things (IoT) devices has posed critical security challenges at the routing layer, where resource-constrained nodes are vulnerable to sophisticated attacks that are not detected by traditional detection mechanisms. This paper proposes a systematic study on the generation, characterization, and analysis of a multi-scenario dataset of attacks on the routing layer in Low-Power and Lossy Networks. We systematically investigate five representative RPL routing-layer attacks, categorized by their adversarial objectives: (1) Topology Manipulation: Decrease Rank and Worst Parent Selection; (2) Traffic Interception: Blackhole and Selective Forwarding; and (3) Resource Exhaustion: Flooding. Additionally, we evaluate a multi-attack scenario to reflect the realistic coexistence of adversarial attacks in two types of network topologies: centralized and distributed, and in different working conditions: standard and mobility scenarios. We generated a labeled dataset with network packets and extracted more than 50 network traffic flow and power consumption features using simulation-based experimentation with the Contiki-NG operating system. This dataset addresses an important gap in IoT security research by providing a simulation-based, calibrated, and reproducible benchmark for developing topology-aware anomaly detection systems. The results emphasize the need for scenario-specific detection strategies and provide insight into the complex interaction between network topology, mobility dynamics, and attack manifestation patterns. We make our dataset publicly available with detailed documentation to facilitate reproducible research and comparative studies for IoT security.

Keywords: Internet of Things, routing security, machine learning, dataset generation, RPL, anomaly intrusion detection, network topology, mobility.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

I. Introduction

IoT has been growing at an unprecedented rate, with predictions of more than 29.3 billion connected devices by 2030 [1]. This growth spans multiple application fields, such as smart cities, industrial IoT (IIoT), healthcare monitoring, environmental sensing, and smart buildings, each with distinct communication needs and security challenges. Most of these devices are deployed in Low-Power and Lossy Networks, which are resource-constrained and thus inherently limit the deployment of classical security mechanisms [2].

The routing layer is the foundation of IoT communication, orchestrating packet forwarding across multi-hop networks. The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) has been standardized by the Internet Engineering Task Force (IETF) in RFC 6550 [3], and it has become the de facto routing standard for IoT deployments. However, RPL's architecture favors efficiency over security and offers limited authentication and validation features. This architectural decision increases the attack surface at the routing layer. Malicious or compromised nodes can launch devastating attacks with very limited resources.

Empirical research in recent years has shown that routing-layer attacks account for approximately 22% of all cyber incidents in IoT environments [4]. The consequences of such attacks go beyond the immediate disruption of the network and may include breaches of data integrity, violations of privacy, and deterioration of service availability. The diversity and sophistication of routing attacks like topology manipulation, selective packet dropping, resource exhaustion, and preference falsification require a comprehensive characterization and analysis to guide the development of effective detection mechanisms [5].

Mobility is also a fundamental characteristic of modern IoT deployments, but largely not addressed in attack characterization studies. Routing attacks manifest and are detectable in a fundamentally different way in mobile IoT devices (e.g., autonomous vehicles, wearable health monitors, industrial mobile robots, and personnel tracking systems) subject to continuous topology changes. Attack signatures that are valid in static networks may not be obvious or have completely different behavior patterns under mobility conditions [6]. The variability of such contexts represents an important research gap, as existing datasets do not provide a comprehensive characterization of attack behaviors across realistic deployment scenarios, limiting the generalizability of detection models trained on single-scenario data. Therefore, the performance of anomaly detection systems trained on conventional datasets tends to degrade significantly when they are deployed in diverse real-world environments [7].

The progress of machine learning-based anomaly detection for IoT routing security hinges on the availability of realistic, comprehensive, and well-characterized datasets [8]. However, the landscape of publicly available IoT security datasets shows substantial limitations:

1. Datasets such as KDD Cup 99 and NSL-KDD are introductory datasets for network intrusion detection research, but they are representative of enterprise network traffic that is fundamentally different in scale, protocol composition, and threat landscape from IoT routing communications.
2. Most existing IoT-specific datasets (e.g., N-BaIoT [9], Bot-IoT [10], TON IoT [11]) are mainly centered on application-layer attacks and botnet behaviors, and do not consider routing-layer attacks specific to Low-Power and Lossy Network (LLN) protocols.
3. Existing datasets usually represent single network architectures, missing topology-dependent variations in attack demonstration.
4. Limited availability of publicly incorporated controlled mobility scenarios restricts applicability for nature IoT deployments, and the combination of all attack scenarios.
5. They suffer from inadequate documentation of generation parameters, thwarting reproduction and validation.

Together, these limitations restrict the development and evaluation of generalizable IoT routing security solutions. In this paper, we tackle the challenges by systematically and in a controlled manner generating a comprehensive routing attack dataset specifically designed to capture attack manifestations across the topology and mobility dimensions.

The rest of the paper is organized as follows. Section 2 provides a detailed literature survey on IoT routing protocols, taxonomy of routing attacks, topology implications, and machine learning for anomaly detection. Section 3 presents related work. Section 4 presents our experimental methodology, including the simulation environment setup, topology definitions, mobility models, and the feature extraction process. In Section 5, we present results over our multi-scenario experimental matrix, analyzing attack manifestation patterns, topology and mobility effects, feature characteristics, and machine learning performance. Section 6 points out major gaps in the research and promising avenues for future work.

II. Background

Low-Power and Lossy Networks represent a particular communication paradigm, characterized by three main constraints: (1) limited energy budgets, requiring battery lifetimes of several years on AA-

sized cells; (2) packet loss rates between 1% and 20% due to wireless propagation challenges; and (3) significant variability in link quality due to external interference and environmental factors. These limitations have a significant impact on protocol design, thereby excluding resource-intensive security techniques commonly used in traditional network contexts [12].

The IETF Routing Over Low-Power and Lossy Networks (ROLL) working group standardized the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) in RFC 6550 to address routing constraints. RPL builds a Directed Acyclic Graph (DAG) rooted at one or more DODAG (Destination-Oriented DAG) roots, typically border routers or central gateways. The protocol works with three major types of messages [13]

1. DIS (DODAG Information Solicitation): A node that wants to join a DODAG sends a DIS to request DODAG information from its neighboring nodes.
2. DIO (DODAG Information Object): Periodically propagated by DODAG members to distribute routing information such as rank (distance metric), goal function details and DODAG settings.
3. DAO (DODAG Advertisement Object): Upward messages to inform the parent of the existence of the child nodes, thus allowing two-way communication and downward routing.

In RPL, a routing metric is computed using an Objective Function (OF) and a parent node is selected. The default OF0 [14] uses the hop count as its metric. The Minimum Rank with Hysteresis Objective Function (MRHOF) [15] makes use of the Expected Transmission Count (ETX) which provides a better awareness of the link quality. The hierarchy of nodes is based on the distance measure from the DODAG root to the selection of parents towards the root.

The most remarkable aspect is that RPL provides minimal or no authentication or integrity protection. However, the vast majority of deployed RPL networks are in the unsecured mode, due to the computational overhead and complexity [16]. This design choice exposes several attack vectors at the routing layer. Routing attacks in LLNs are widely studied in literature. Karlof and Wagner [17] classified attacks into two families, as follows: (1) attacks on DAG construction, (2) attacks on data forwarding. Later studies extended this taxonomy to attack mechanisms and impacts.

Sinkhole attacks are among the most significant and extensively studied attacks in RPL networks [18]. In a sinkhole attack, a malicious node advertises better (false) routing metrics, usually a fabricated low rank or claims of direct root connectivity, such that the network traffic is attracted to the malicious node. It creates a “sinkhole” where a lot of traffic is aggregated. The sinkhole attack is based on metric manipulation. The attacker node advertises a rank that is lower (closer to the root) than possible, violating the fundamental RPL constraint that rank must increase monotonically with distance from the root. According to RPL's parent selection algorithm, the parent selection algorithm of RPL is such that the legitimate nodes choose the node with the least advertised rank. This means the traffic is sent to the malicious node.

In a Blackhole attack, the malicious node advertises an attractive routing metric to attract traffic and then drops all received data packets, resulting in total data loss within its subtree. Ghaleb et al. [19] showed that in small networks, a single sinkhole/blackhole node can eliminate 60-90% of network traffic delivery, and the smaller the network, the greater the impact. Attacks can be very easy or very hard. Naive attacks rely on metrics that are clearly false (e.g., rank=0 or negative) and can be easily identified as abnormal traffic patterns. More sophisticated attacks will only slightly distort the metrics, bringing them into the range of normal network variations, and making them much harder to detect [20].

Selective forwarding attacks are carried out by a compromised node that is a participant in routing only in name, but selectively drops packets based on pre-specified criteria [21]. Selective forwarding

drops only selected packets rather than all traffic, as in blackhole attacks, making it more difficult to detect because the node still seems to be partially functional. Selective forwarding can be performed based on several criteria: source address (to block specific source nodes), destination address (to isolate specific targets), packet type (to specifically target DIO messages to disrupt topology formation) or content-based criteria (to drop packets with specific sensor readings).

In the original work by Karlof and Wagner [17], it was shown that selective forwarding is inherently difficult to detect with external observation only because packet drops can be attributed to congestion or link quality rather than malicious activity. The inherent stochastic nature of wireless link quality in LLN environments naturally conceals selective forwarding behavior. Raza et al. [22] empirically assessed the efficacy of this attack and found that selectively forwarding 25–50% of topology management messages could substantially degrade network formation, yet remain below the threshold for common anomalies, underscoring the subtlety of this attack.

Worst parent selection attacks, also called preference falsification attacks, occur when an adversarial node fabricates its parent selection process without directly falsifying its own routing metrics [23]. The attacker induces neighbors to select suboptimal parents or to switch parents frequently. The attack is effective by either (1) spoofing DIO messages pretending to be genuine parents with intentionally degraded ranks, (2) selectively acknowledging some DIOs while ignoring others to create false impressions of parent reliability, or (3) subtly falsifying neighbor link quality indicators to bias the parent selection algorithm.

The insidiousness of worst parent selection attacks is in their plausible deniability: the resulting suboptimal routing may be observed, but explained away by normal network dynamics, wireless link variability, or legitimate objective function behavior. Gradient attacks (a variant of worst parent selection) were shown by Measures et al. [24] to be able to increase the average hop counts by 40% while being undetectable by simple intrusion detection heuristics.

Flooding attacks exploit the periodic broadcast of messages by RPL to always maintain the topology. An attacker generates an excessive number of messages, thus consuming network bandwidth, processing resources, and energy budgets of constrained devices [25]. The flooding attack has several types: (1) DIS flooding: DODAG information solicitation requests are flooded into the network; (2) DIO flooding: fake or excessive DODAG information objects are broadcast; (3) DAO flooding: upstream reachability advertisements are forged. Each variant exploits different aspects of RPL's topology maintenance mechanism.

Flooding attacks directly affect network availability by (1) consuming bandwidth needed for data traffic, (2) causing excessive routing recalculation, and (3) causing repeated parent switches and rank recomputation. The attack is especially effective in resource constrained environments where processing overhead directly translates into measurable service degradation.

Perazzo et al. [26] showed that an attacker node can drain the batteries of neighboring nodes by incessantly flooding DIOs, depleting energy within hours. This is a particularly serious threat for IoT devices permanently deployed and relying on multi-year battery lives.

A dimension that is often overlooked in routing attack characterization is the role of network topology in determining attack effectiveness and detectability. The topological structures of real-world IoT deployments are different due to the application requirements and deployment environments.

Centralized Topologies: Characterized by a single root node (most often a gateway or border router) and other nodes arranged in tree or star configurations. Typical of centralized topologies are: Smart buildings with a central controller, Small-scale sensor networks (nodes < 100), Industrial facilities with a central control room, and Centralized centers of healthcare facilities. Centralized architectures have natural convergence points. They naturally funnel traffic to the single root. Existing

traffic patterns already have single points of concentration, making attacks that concentrate traffic toward the attacker (sinkhole attacks) more effective. However, the existence of a tree topology with few alternative routes can also make attacks more easily detectable – deviations of traffic from the expected paths become more significant [27].

Distributed Topologies Characterized by a set of nodes with intersecting communication ranges and multiple paths between nodes. Distributed topologies are typical for: Extensive environmental monitoring networks, Smart city deployments across large geographic regions, Industrial IoT distributed sensor arrays for factories, and Mobile sensor networks with different topologies. In distributed topologies, traffic naturally spreads across multiple paths, providing inherent redundancy and resilience. By providing alternate paths, the effectiveness of attacks that try to direct traffic to the attacker can be reduced. However, the complexity of multi-path routing and the inherent path diversity may complicate anomaly detection, i.e., attack-induced path diversity becomes hard to distinguish from legitimate variation [28].

Kamhoua et al. [29] empirically evaluated the attack on Contiki-based networks and found that it redirected 67% of traffic in centralized topologies and 41% in distributed topologies (a 39% reduction in attack efficacy).

Mobility is a key feature of emerging IoT deployments, but is largely absent in routing security analysis. Mobile IoT comprises Personnel tracking and emergency systems, Mobile robots and self-driving cars, Body-worn health monitoring devices, Drones for aerial sensing and delivery, and Mobile sensor networks for disaster response. Node mobility causes dynamic topology changes due to the continuous formation and breakage of links. This is shown as:

1. **Frequent topology recomputation.** The DODAG maintenance is initiated to re-establish the connectivity whenever a node leaves or joins the communication range, which leads to 5–10 times increase in DIS/DIO exchanges [30].

2. **Parent Instability:** The quality of the link varies with distance and environmental effects, which can lead to frequent changes in the parent of mobile nodes. As static networks have an average parent change rate of around 0.5–1 change per minute, while mobile scenarios have 5–15 changes per minute [31].

3. **Rank Oscillations:** Node ranks can oscillate as topological changes create and destroy alternative paths, resulting in repeated rank computations.

4. **Packet loss variability:** In mobile scenarios, the baseline packet loss is higher (10-20%) than in static networks (1-5%), which increases the “normal” loss variability in which selective forwarding attacks need to operate [32].

Such dynamics fundamentally alter the patterns of attack manifestation. The natural variability due to mobility may mask an attack signature that is clear in static networks (e.g., a certain rank value or switching pattern to a new parent). Other attacks may be easier to detect, on the other hand. Mobility introduces features (e.g. characteristic parent oscillation patterns) that are not present in static scenarios. Still, there are a few empirical studies on attacks against mobile networks. Thulasiraman et al. [33] studied sinkhole attacks in mobile ad-hoc networks and found that the effectiveness of the attack varies from 20-80% based on mobility patterns, node speed, and pause time, which indicates the importance of mobility in attack characterization.

Consequently, the creation of comprehensive security datasets has become a critical step in evaluating attack behavior and intrusion detection mechanisms. There are two main approaches to generating security datasets: simulation-based and testbed-based.

Simulation-based network simulators (Cooja, ns-3, OMNET++) can be employed to provide fine-grained control over network parameters, attack implementations, and traffic characteristics. Advantages include (1) reproducibility, in that identical experiments can be repeated with identical parameters; (2) ground truth, in that attack occurrence and parameters are precisely known; (3) controllability, in that network conditions can be systematically varied [34]. The main challenge is the simulation-reality gap: how closely behavior in simulation reflects behavior in real deployments. The

main source of errors in simulations is the modeling of wireless propagation. Popular radio propagation models (unit disk graph, log-normal shadowing) are very simplified compared to real environments [35].

Testbed Approach: Physical deployments of real IoT devices show realistic behaviour but are hindered by severe practical problems: (1) reproducibility is difficult because of environmental variability; (2) scalability to large numbers of nodes is expensive; (3) realistic attack scenarios on real devices raise ethical and security problems; (4) the ground truth is difficult to obtain because of the large amount of instrumentation required.

A promising compromise has emerged in the form of hybrid approaches that combine simulation-based initial dataset generation with testbed validation. Reasonable agreement between simulated and actual RPL behavior was observed by Perazzo et al. following Cooja simulation and USNET (IoTlab) testbed validation when using calibrated propagation models.

where special attention is paid to improving the realism of the propagation models and tuning parameters to reduce the difference between simulations and empirical observations. The literature review highlights a number of unresolved challenges that this research aims to address:

1. **Topology-Dependent Attack Characterization:** The impact of network topology on network routing has been studied in general. However, a systematic empirical characterization of the impact of different topologies on the manifestation of attacks is lacking.

2. **Attack Analysis for Mobility:** While mobile IoT deployments are becoming more common, the effects of IoT mobility and routing attacks are not well studied.

3. **Multi-Attack Comprehensive Datasets:** There are several types of attacks that have been studied. However, there is no dataset that captures multiple attacks at the same time with systematic variation of the scenarios.

4. **Reproducibility and tool availability:** Several IoT security research studies do not publish their code, parameters and datasets to the public, limiting the reproducibility and comparison.

5. **Feature Analysis and Selection of Routing Attacks:** Feature engineering has been widely studied in enterprise network IDS, while feature analysis of routing-specific attacks is limited.

Following the establishment of this dataset, the next phase involves examining how machine learning methods can use this data to efficiently identify anomalies. Machine learning for network anomaly detection has been widely developed in the last two decades. Manual creation of attack signatures is required for traditional signature-based intrusion detection systems such as Snort and Suricata. They are effective against known attacks but cannot address new or zero-day variants, which is a critical limitation in the dynamic IoT environment.

Anomaly-based detection systems, on the other hand, model normal network behavior, flagging deviations from this model as potential attacks. This method has two major advantages: (1) detection of new kinds of attacks and (2) adaptability to changing network conditions. Machine learning offers a systematic framework for learning models of normal behavior from historical data. Different families of algorithms have been used for network anomaly detection:

Supervised Classification: The training data is labeled with both benign and attack instances, so that supervised classifiers can learn decision boundaries to separate normal traffic from malicious traffic. Good classifiers for network anomaly detection are:

Random Forest [36]: It is an ensemble of decision trees which is very good to capture non-linear relationships and feature interactions. Random Forests are frequently among the best performing Network IDS applications, achieving 95-99% accuracy on standard benchmarks [37].

Support Vector Machines [38]: They are kernel-based classifiers that work well in high-dimensional spaces. SVMs are very good at separating well-defined classes, but they are not very robust to high-dimensional and imbalanced datasets [39].

Gradient Boosting: Sequential optimization of residuals often yields state-of-the-art results (e.g., XGBoost, LightGBM). Recent work on IoT security has shown that gradient boosting approaches can achieve 96-98% accuracy [40].

Neural Networks: Deep learning techniques, including LSTMs and GRUs, can learn the temporal structure of sequential network traffic and have achieved promising results for attack detection using temporal signatures [41].

Unsupervised Anomaly Detection Without labelled attack data, unsupervised methods look for statistical outliers or points outside the learned normal distribution:

Isolation Forest: A Fast Outlier Detection on High-Dimensional Data [42].

Local Outlier Factor: a density-based method that identifies points with lower density than their neighbors [43].

Autoencoders: Neural Networks trained to reconstruct normal traffic and the reconstruction error is used for the detection of anomalies [44].

Supervised learning approaches dominate IoT routing security research because attack research is more controlled and labeled datasets are available. However, the assumption that attack characteristics are the same across different deployment contexts, implicit in supervised learning, may not always hold, thereby reducing the model's transferability [7].

III. Related Work

In fact, although several datasets, such as KDDCUP99 [45], NSL-KDD [46], UNSWNB15 [47], and CICD2017 [48], have been created over the past two decades for evaluation purposes of network-based intrusion detection systems (IDSs), they do not include any specific characteristics of IoT networks, as these datasets do not contain sensors' reading data or IoT network traffic. To respond to this major issue, few efforts focused on the generation of IoT-specific datasets have also been seen in the literature recently. However, they are characterized by some limitations in terms of the IoT-specific information they include.

For instance, the dataset proposed in [45] is an IoT-specific, widely used benchmark dataset in the field of network intrusion detection and cybersecurity, which aims to develop effective methods for detecting network attacks. It contains various features, such as protocol information, source and destination IP addresses, port numbers, and connection durations.

The KDDCUP99 dataset suffers from an imbalance between normal and attack instances. It has a much larger number of normal instances than the relatively small number of attack instances. Such class imbalance can negatively affect the performance of machine learning models trained on the dataset as they tend to be biased towards the majority class. Developed in 1999 Since then the network landscape has changed drastically. New types of attacks and network behaviors have emerged, which may not be adequately represented in it. Also, the dataset does not include any information about the time sequence of network events or mobility scenarios.

The NSL-KDD dataset proposed in [46] is an improved version of the KDDCup99 dataset, designed to address some of its limitations. including new attack scenarios that were not present in the original KDDCup99 dataset. It also addresses the issue of class imbalance by providing a more balanced distribution of normal and attack instances, which helps in training and evaluating machine learning models more effectively. contain events reflecting attack scenarios; however, they do not cover a diverse set of attack scenarios and do not include sensors' reading data or information related to the behavior of the IoT devices (e.g., sensors/actuators) within the network. Therefore, these IoT datasets can mainly be used for detecting only a limited number of network-based attacks against IoT networks, as they do not contain adequate information for detecting a wide range of network-based attacks and/or attacks that manipulate sensor measurement data or compromise IoT devices within the IoT network.

The UNSW-NB15 dataset proposed in [47] is derived from real-world network traffic captured in a controlled environment at the University of New South Wales (UNSW), Australia. The dataset aims to provide a diverse and up-to-date representation of network traffic, including both normal and

malicious activities. Although UNSW-NB15 is based on network traffic captured in a specific environment, it might not fully reflect the variety and complexity of real-world network scenarios. The dataset's applicability to different network settings and configurations should be considered. Secondly, the dataset was collected over a specific time period and may not account for the evolution of network attacks and techniques since its collection. It is important to be aware of the merging attack strategies and mobility scenarios.

The authors in [48] study the behavior of IoT devices when they function under normal operation conditions, as well as when they are under attack, calling it the Irad dataset. The proposed datasets include information related to the behavior of the IoT devices based on a simulated normal scenario and a simulated malicious scenario. The classes of the IoT device behavior datasets are two (i.e., normal behavior and abnormal behavior), corresponding to the case of a binary classification problem. Also, the authors utilized the Cooja simulator of the open-source Contiki operating system (OS) in order to simulate a normal IoT network scenario and a malicious IoT network scenario and generate corresponding normal and malicious datasets.

The Irad consider the latest and widespread dataset that recommended industrial IDS (IIDS), which provides power trace behavior. but the data suffers from a lack of routing and traffic information that presents node behavior and also doesn't include attacks that represent all attack categories and static scenarios.

The author in [49] used the Cooja IoT simulator to generate high-fidelity attack data in IoT networks with 10 to 1000 nodes. The proposed dataset has been produced by real-world equivalent simulations using the open-source Contiki-Cooja simulator, which generates raw packet capture (PCAP) files that are subsequently converted to Comma-Separated Values (CSV) files for text-based analysis. The CSV files are subsequently input into the feature pre-processing module of our system. The features are derived from the traffic flow data contained in the CSV files. The proposed dataset are generated depending on the behavior of the network, providing some attacks, namely: Hello flooding, Decrease rank and version number, then test the presented dataset via a machine learning intrusion detection system, achieving an accuracy ratio of 94.9 to 99 % in some cases.

A Novel, Realistic Dataset for Intrusion Detection in IoT, as presented in [50], is explicitly crafted to fulfill the demand for authentic IoT routing attack data. This dataset records network traffic produced by simulated IoT settings utilizing the Contiki-Cooja platform, emphasizing the RPL (Routing Protocol for Low-Power and Lossy Networks) protocol. It considers both legitimate and malicious traffic and attack scenarios such as Blackhole, Flooding, DODAG Version Number and Decreased Rank. The dataset contains detailed packet-level features, including transmission and reception rates, packet counts, and protocol-specific details. This robust framework enables the successful development and assessment of intrusion detection systems (IDS) through artificial intelligence and machine learning techniques.

The proposed ROUT Dataset in [51] is a good base for research on IoT routing security by replicating four major RPL-based routing threats. ROUT consists of large network traffic data obtained from IoT environments under normal and attack conditions. The collection contains different packet properties including statistics on transmission and reception. It is well annotated to distinguish between benign and malicious operations. The structured nature of ROUT and the focus on RPL protocol vulnerabilities are well-suited for machine learning-based anomaly detection, benchmarking, and comparative analysis of security solutions in IoT networks.

Reference Paper	Network Strategy		Dataset Category		Mobility	Attack Category				Featured Number	Simulation Type
	Distribut ed	Centralized	Anomaly based	Signature based		Resource	Topology	Traffic	Combination		
KDDCUP99. [45]	X		X			X		X		32 – TCP and UDP Data Traffic	physical LAN simulating
NSL-KDD. [46]	X		X	X		X				41- same structure as KDD, refined	physical LAN simulating
UNSWNB15. [47]	X			X			X			47-Basic network traffic, Content, Time.	IXIA Perfect Storm tool

Irad. [48]	X		X			X	X			40- RPL-based IoT context, reduced to ~30 via feature selection	Cooja -Contiki
Dhifallah[49]	X		X				X			20 -RPL traffic, Rank, Version.	Cooja -Contiki / Contiki-NG
IoT-RPL [50]	X		X			X	X			16- TIME, SRC/DST, LENGTH, INFO, TR/RR, TAT/RAT, TPC/RPC, TTT/TRT, DAO/DIS/DIO.	Cooja -Contiki / Contiki-NG
ROUT [51]	X			X	X	X				15-protocol-aware: DIO/DIS/DAO counts, packet fwd/drop, convergence time, control overhead, mobility context.	Cooja -Contiki
Proposed Anomaly dataset	X	X	X		X	X	X	X	X	51-traffic flow -power consumption -time -node position-Mobility context.	Cooja -Contiki / Contiki-NG

Table 1: Comparison between the proposed dataset and related work.

IV. Methodology

4.1 Research Design and Experimental Framework

In this paper, we present a controlled experimental methodology for generating a structured dataset and systematically evaluating routing attacks across different network topologies and operational mobility conditions. The proposed dataset design is structured along three dimensions: (1) Attack type based on different categories” Target- Impact”: Decrease rank” Network Traffic-Misappropriation “, Blackhole, Selective Forwarding “Network Topology-Isolation “, Worst Parent “Network Topology Suboptimization”, and Flooding “Network Resource -Direct” as well as a combined attack scenario. (2) Network topology: two topology types are examined namely distributed and centralized configuration (3) Operating condition: two operational settings are implemented: a static scenario, in which all nodes remain fixed in their locations, and a mobility scenario, in which nodes may change location, move out of their original neighborhood, reconfigure their Destination-Oriented Directed Acyclic Graph (DODAG), and forward messages through alternative parent nodes.

Based on this design, the total number of experimental configurations is 24 attack scenarios (6×2 “**topologies**” $\times$ 2 “**static vs mobility**”), **plus 4 benign baseline scenarios, for a total of 28 distinct configurations**. Each configuration was repeated three times using different random seeds to account for stochastic variability, yielding a total of 84 simulation runs; each run lasted 30 minutes.

The experiments were conducted using the Contiki-NG real-time operating system in conjunction with the Cooja network simulator [52]. Contiki-NG is the modern successor to the widely adopted Contiki operating system and is specifically designed for Internet of Things (IoT) and Low-Power and Lossy Network environments. Within this framework, Cooja enables cycle-accurate emulation of widely used IoT hardware platforms, such as TelosB “T-Mote” and Zolertia “Z-Mote”, while offering comprehensive support for the simulation and analysis of IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) behavior.

The choice of the Contiki-NG/Cooja platform was motivated by several factors relevant to this study's objectives. First, it offers a native implementation of RPL with support for several objective functions, enabling realistic evaluation of routing behavior in both benign and adversarial environments. Second, the emulation fidelity of Cooja has been validated in previous studies through comparisons with physical testbeds, indicating that it is suitable for rigorous experimental research [53]. Third, the platform enables reproducibility: simulation runs are deterministically reproducible by using predefined random seeds. Finally, its open source and extensible architecture allows for custom attack logic, protocol modifications and packet level monitoring to be implemented. 4. Finally, Contiki-NG and Cooja are widely used in the IoT and LLN research community, which gives additional credibility and comparability to the current work with the existing literature [54].

Finally, build the data packet based on node traffic behavior and power consumption information to enhance the strength of the detection system. The dataset contains about 50 different detection features that represent overall information about each node. Table 2 explores the proposed dataset features and the abbreviations for each. Figures 1 and 2 show the real data from the simulator and the Excel sheet.

NO	Feature	Description
1	Sim_time	simulation time
2	Clock_time()	clock time (i.e., by default, 128 ticks/second)
3	ID	Mote ID
4	Seqno	sequence number
5	P	Label
6	Idle_listen	idle listen energy consumption for this cycle
7	Idle_transmit	idle transmission energy consumption for this cycle
8	Listen	Listen to the energy consumption for this cycle
9	Transmit	transmission energy consumption for this cycle
10	Lpm	LPM energy consumption for this cycle
11	Cpu	CPU energy consumption for this cycle
12	all_idle_listen	accumulated idle listening energy consumption
13	all_idle_transmit	accumulated idle transmission energy consumption
14	all_listen	accumulated listening energy consumption
15	all_transmit	accumulated transmission energy consumption
16	all_lpm	accumulated Low Power Mode energy consumption
17	all_cpu	accumulated CPU energy consumption
18	Rimeaddr	rime address
19	Pos_x	Position of node (Coordinates x)
20	Pos_y	Position of node (Coordinates y)
21	Destination	The ID of the target node
22	Node_Rank	The rank of node
23	Parent_id	The parent address
24	Parent_rank	The rank of the parent node
25	Ins_id	Instance ID of the node
26	Num_of_mess	The number of the forward message
27	Num_of_neighbor	Number of neighbors
28	Sum_of_neighbor_ranks	The sum of neighbor rank
29	Node_V_num	Node Version number
30	Min_hop_increase	Minimum hop increase (the lowest rank)
31	Message_type	The type of the forward message
32	Buf_size	The size in bytes of data packet
33	Dio.l.time	DODAG lifetime
34	Dag id	Direct Acyclic Graph identification
35	Storing	Type of network model
36	DODAG V N	DODAG Version Number
37	Intero	Interoperability
38	ETX	expected transmission count
39	Loc R	Local Repair

40	Net_cat	The network type (centralized-distributed)
41	Node_type	Number explore the type of node 1-mean server,2- for the parent nodes 3- for leaf nodes
41	Child_num	The number of children for the node
43	Received_count	The number of received messages
44	Forward_count	The number of forward messages to their parent
45	Packet_ratio	The ratio between the packets sent and received
46	Overall_Radio_Duty_Cycle	The cumulative percentage of time that the node's radio remains active (either transmitting or listening) relative to the total elapsed simulation time.
47	Interval_Radio_Duty_Cycle	Measures the percentage of time the radio is active (transmitting or listening) during the current observation interval.
48	Overall_TX_Duty_Cycle	The cumulative percentage of total simulation time during which the node spends transmitting packets.
49	Interval_TX_Duty_Cycle	The percentage of time spent transmitting during the current monitoring interval.
50	Overall_RX_Duty_Cycle	The cumulative percentage of total simulation time during which the node's radio is in listening or receiving mode.
51	Interval_RX_Duty_Cycle	Measures the percentage of time the node spends listening or receiving during the current observation interval.

Table 2: The Proposed Dataset Feature.

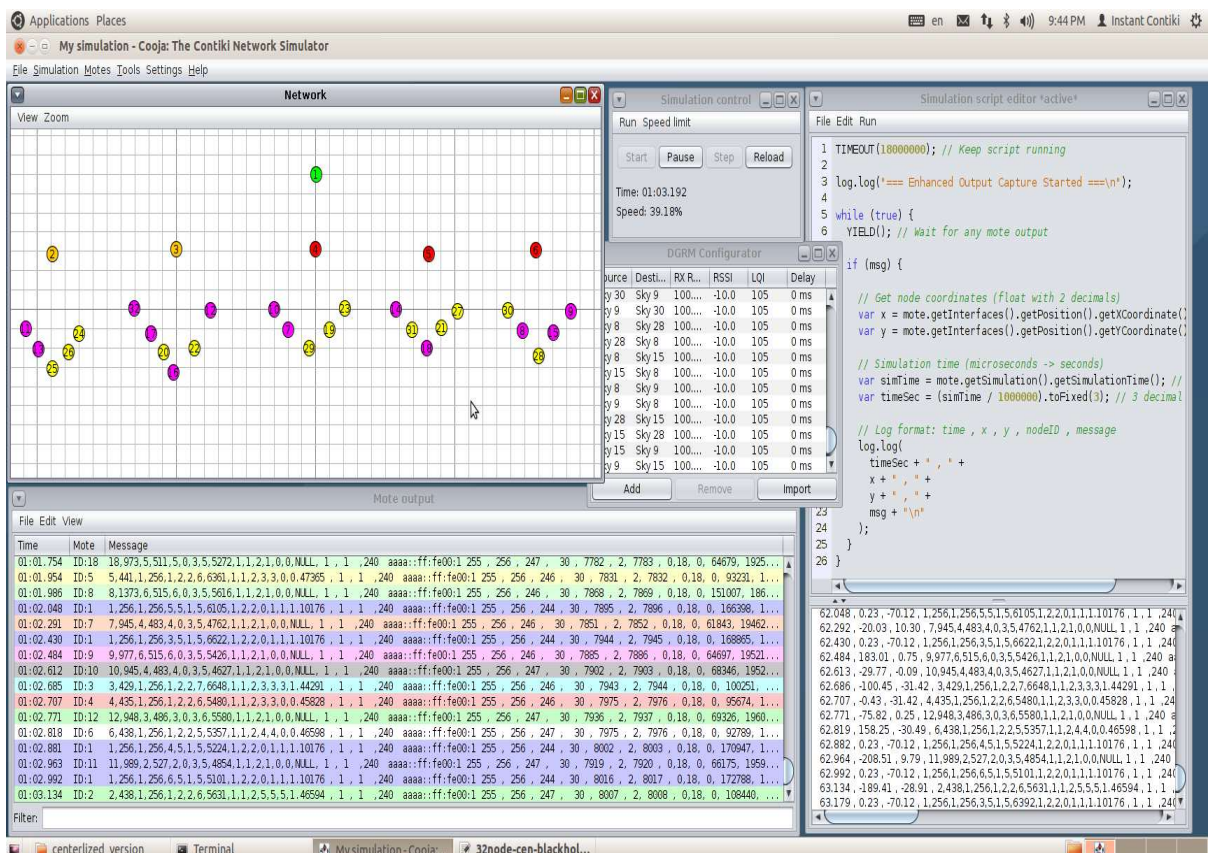


Figure 1: Cooja simulation Mote Captured Data Packet.

where T is the total number of trees, V_t is the set of decision nodes in tree t , and $\Delta I(v, t)$ is the impurity reduction at node v . The impurity criterion can be computed using either Gini impurity or entropy-based measures, depending on the implementation used.

The resulting feature importance scores are normalized to obtain a probabilistic weight distribution:

$$\sum_{i=1}^M w'_i = 1 \quad (2)$$

where M represents the total number of features and w'_i represents the normalized importance weight of feature i .

In more formal language, the information gain for a feature is the reduction in entropy before and after splitting the dataset:

$$IG = Entropy(Parent) - \sum_i \left(\frac{|S_i|}{|S|} \times Entropy(S_i) \right) \quad (3)$$

where S_i is the subset of samples obtained after splitting on a specific feature and $|S|$ is the total number of samples in the parent node. The importance score of each feature is calculated by summing the information gain values for that feature across all trees in the ensemble.

To ensure robust feature selection, a dual-threshold strategy is used. First, we select features with normalized importance scores above the 80th percentile based on an absolute importance criterion. Second, a cumulative importance threshold is applied that corresponds to 95% of the total ensemble importance. The final feature subset is determined on the basis of the intersection of both criteria, so that only the features with consistent and context-aware discriminative ability are preserved. The multi-stage selection mechanism improves computational efficiency and robustness of the method against simulation artifacts and variations in topology-dependent behavior.

After that, Min-Max normalization is applied to the remaining attributes to make the numerical scales of the attributes consistent and let them contribute equally to the training of classifiers:

$$Z_{\text{norm}} = \frac{(z_{\text{act}} - z_{\text{min}})}{(z_{\text{max}} - z_{\text{min}})} \quad (4)$$

where Z_{norm} is the normalized feature value, Z_{act} is the original feature value, while Z_{min} and Z_{max} correspond to the minimum and maximum value of the feature in the dataset, respectively. This normalization process avoids large numerical ranges of features from dominating the learning process and improves convergence stability across classifiers.

Subsequently, various supervised machine learning classifiers were trained on the normalised dataset to identify multi-class routing attacks. The dataset was partitioned using a stratified hold-out split of 70% for training and 30% for final testing. To optimize hyperparameters and prevent data leakage, stratified 5-fold cross-validation was applied exclusively on the training partition.

4.3 Machine Learning evaluation

Following feature selection and data pre-processing, several supervised machine learning algorithms were trained and evaluated to assess their performance for detecting routing-layer attacks in IoT and Low-Power and Lossy Network (LLN) environments. The selected classifiers belong to different learning paradigms and were selected due to their proven performance in security applications, their ability to handle multi-class classification problems, their computational efficiency, interpretability and robustness to noisy and topology-dependent network traffic. The models subject to evaluation are Decision Tree (DT), Random Forest (RF), Extremely Randomised Trees (Extra Trees), k-Nearest Neighbours (KNN), Logistic Regression (LR), and Multi-Layer Perceptron (MLP) as a shallow neural network baseline.

The selected algorithms provide a fair comparison between light and heavy approaches, so that the trade-off between classification performance and deployability in resource-limited IoT environments can be assessed. Furthermore, this comparison allows us to assess the generalisation ability of each model in different network topologies, various mobility conditions, and multiple routing attack scenarios.

Decision Tree (DT) was added due to its low computational complexity and high interpretability which enabled attack classification through hierarchical rule-based decision boundaries. However, single decision tree models are prone to overfitting when trained on noisy or highly dynamic network traffic. DT therefore mainly serves as an interpretable baseline for more robust ensemble methods.

To overcome this limitation, ensemble learning techniques, namely, Random Forest (RF) and Extremely Randomized Trees (Extra Trees), were used. The random forest constructs many decision tree models by bootstrapping samples and random selection of features, which reduces the variance and improves the generalisation ability. Extra Trees randomly choose split thresholds, making the models more diverse and lowering computational cost while preserving good predictive power. Both ensemble methods are very suitable for large scale network telemetry data and show robustness to noisy observations and frequent changes of topology typical for IoT and LLN environments.

Besides their classification performance, ensemble methods provide feature importance estimates which enhance model interpretability by showing which routing and network characteristics are most important in discriminating attacks. They are especially effective in distinguishing advanced routing attacks like Blackhole, Selective Forwarding and Decrease Rank attacks, since they are able to model complex nonlinear relationships between traffic behaviour and routing metrics.

The k-Nearest Neighbors (KNN) algorithm was explored as a distance-based learning method that classifies network traffic based on similarity between feature vectors. While KNN is computationally expensive at prediction time, it can produce competitive results with proper feature selection and normalization. For this study, the Euclidean distance metric was used to measure the similarity between traffic instances:

$$d(x,y)=\sqrt{\sum_{i=1}^n(x_i - y_i)^2} \quad (5)$$

where x_i and y_i denote feature values of two network traffic instances.

KNN is especially effective when feature normalization is applied correctly, because distance calculations are highly sensitive to feature-scale variations. However, the algorithm has increased computational complexity during inference due to the need to store and compare all training instances. Therefore, KNN is an important baseline method for the trade-off between classification accuracy and real-time deployability.

For comparison purposes, the evaluation framework included a shallow Multi-Layer Perceptron (MLP) architecture as a Neural Network baseline. The MLP is composed of an input layer, a hidden layer and an output layer and is connected through non-linear activation functions. MLP models are capable of learning very complex and non-linear feature interactions that could be overlooked by traditional machine learning classifiers. Hidden neurons iteratively optimize weight parameters using backpropagation and gradient descent optimization methods. The neuron activation process is defined as :

$$a_j=f(\sum_{i=1}^n w_{ij} x_i + b_i) \quad (6)$$

where a_j denotes neuron activation, w_{ij} represents connection weights, and f denotes the activation function.

Though more sophisticated neural architectures could offer better predictive power, a lightweight MLP was deliberately chosen to keep it fair in the comparison of computational feasibility in resource-constrained IoT settings.

Hyperparameters were optimized for each classifier using stratified 5-fold cross-validation on the training data. Performance evaluation was done with different metrics such as Accuracy, Precision, Recall, F1-Score, Detection Rate (DR), False Positive Rate (FPR) and Area Under Receiver Operating Characteristic Curve (AUC-ROC). The metrics provide an overall measure of how well the classifier can distinguish between normal network traffic and different kinds of routing attacks.

All experiments were performed in Python due to its rich ecosystem of scientific computing and machine learning libraries. The main software libraries used in this work are: Machine learning models implementation and evaluation using scikit-learn • Pandas for data preprocessing and manipulation. • NumPy for numeric calculations. • Matplotlib and Seaborn for graph analysis and data visualization. The experiments were carried out in an environment of Jupyter Notebook, which allows iterative experiments and reproducible workflow management.

In the experimental evaluation phase, stratified k-fold cross-validation was used to avoid sampling bias and to obtain statistically reliable performance estimates. Specifically, 5-fold cross-validation was used by partitioning the dataset into 5 mutually exclusive subsets. The process was repeated 10 times, and the final performance metrics were calculated as the mean over all folds. Stratified partitioning ensures that all types of routing attacks are accurately represented, avoiding evaluation bias due to class imbalance. This validation strategy significantly improves the robustness and generalizability of the experimental results, while reducing variance associated with random train-test partitioning. Several statistical and operational evaluation metrics have been used to fully evaluate the effectiveness of the machine learning classifier.

Accuracy is the ratio of the number of correctly classified instances to the total number of samples:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

where TP, TN, FP, and FN are True Positives, True Negatives, False Positives, and False Negatives, respectively.

Accuracy is a good overall performance measure, but it can be misleading for imbalanced intrusion detection datasets.

Correctness Precision is the percentage of predicted attacks that are malicious:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (9)$$

High precision indicates a low false alarm rate, which is very important for the practical deployment of IDS.

Recall, also known as Detection Rate (DR) or Sensitivity, measures the proportion of actual attacks that are correctly identified by the classifier:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (10)$$

A high recall value is needed to reduce the number of malicious activities that are not detected in the network.

F1-Score This is the harmonic mean between Precision and Recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (11)$$

The F1-Score is especially useful for evaluating the performance of a machine learning classifier with imbalanced class distributions.

False Positive Rate (FPR) is the percentage of benign traffic that is incorrectly identified as malicious:

$$FPR = \frac{FP}{FP + TN} \quad (12)$$

In operational classifier systems, minimizing FPR is of paramount importance in order to reduce unnecessary security alerts and administrative overhead.

The Receiver Operating Characteristic (ROC) curve shows the relationship between the Detection Rate and False Positive Rate for different classification thresholds. The Area Under the ROC Curve (AUC-ROC) is a threshold-independent measure of the classifier's separability capability. The higher the value of AUC, the better the discrimination between benign and malicious network traffic patterns.

V. Result

5.1 Feature Selection and Ranking Results

The feature selection phase employed the Extremely Randomized Trees (Extra Trees) algorithm to evaluate the discriminative power of $M = 51$ network and contextual features across four dynamic LLN scenarios. The Mean Decrease in Impurity (MDI) analysis revealed that routing-layer metrics varied in importance for attack discrimination. Table 3 shows the globally top-ranked features for each routing-attack-combined scenario, which combines data from all four scenarios. To better assess the attack behaviour under various network conditions. The integration reflects the variation in the network topology and node mobility, resulting in a varied dataset with common attack characteristics under different conditions. The Combined dataset allows to develop strong intrusion detection models by training on different environments, learning features that differentiate between normal and malicious behaviour and providing an evaluation framework that resembles real-world deployment scenarios for IoT where conditions can change. The following subsections discuss variations for specific scenarios.

Rank	Decrease Rank Feature - ratio	Blackhole Feature -ratio	Selective Forward Feature -ratio	Worst Parent Feature -ratio	Flooding Feature -ratio	Combined Feature -ratio
1	Life time 0.1596	Forward ratio 0.1273	Time in Second 0.2716	parent rank 0.4293	all idle listen 0.1191	Forward ratio 0.1448
2	Num of neighbour 0.0781	Time in Second 0.1165	Forward ratio 0.211	all idle listen 0.0839	Destination 0.0987	num send Message 0.0854
3	all cpu 0.0724	Num Child 0.0762	pos of y 0.1407	Time in Second 0.0736	Num of neighbour 0.086	all idle listen 0.0814
4	parent rank 0.0616	Life time 0.0724	Num of neighbour 0.0706	pos of y 0.0678	Time in Second 0.0776	pos of y 0.0732

5	all idle transmit 0.0586	pos of y 0.0627	Forward Count 0.0429	Num of nieghbour 0.0401	Listen 0.0771	Message type 0.0707
6	Sum of niegh rank 0.0506	Sum of niegh rank 0.062	Num Child 0.0402	Destination 0.033	pos of y 0.0771	Listen 0.0497
7	Rank 0.0503	Node type 0.058	Node type 0.0338	Listen 0.0317	all idle transmit 0.0589	Time in Second 0.0468
8	all idle listen 0.0471	all cpu 0.0448	Num recive message 0.025	Sum of niegh rank 0.0276	Node type 0.0575	Node type 0.045
9	Time in Second 0.0452	Num of nieghbour 0.0413	Sum of niegh rank 0.0216	all idle transmit 0.0215	Sum of niegh rank 0.0479	Forward Count 0.0392
10	Num recive message 0.0416	Num recive message 0.0377	num send Message 0.0166	Node type 0.0174	num send Message 0.0423	parent rank 0.035

Table 3: Top-Ranked Features by Attack Category.

5.1.1 Decrease Rank Attack

For the Decrease Rank attack, the adversary's objective is to manipulate the RPL objective function by advertising an artificially low rank value, thereby attracting child nodes and subverting the DODAG topology. The feature importance analysis reveals that, in all four cases, node lifetime (life time) is generally the most important differentiating factor and consistently ranks first. With the increased forwarding load, legitimate nodes that select the attacker as a parent will experience a higher rate of packet forwarding. This increases transmit/receive duty cycles and idle-listen energy expenditure, accelerating battery depletion and directly decreasing Life_time. False rank also causes frequent rank inconsistencies and local repair procedures, which further increase control-message overhead (DIO/DAO) and aggravate energy drain. Life_time shows the largest difference between the benign and attacked nodes because it is mostly manifested as an increased energy consumption, not obvious packet loss or latency spikes in the early stage.

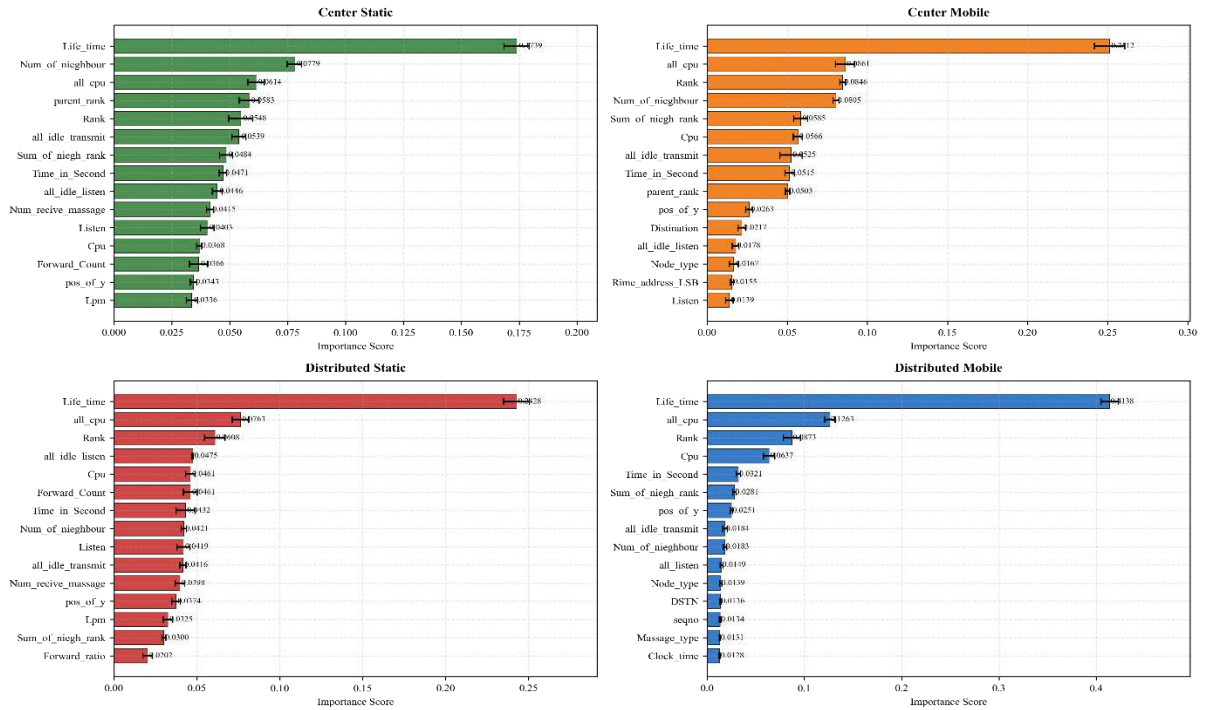


Figure 3: The Feature Scenario- Based comparison.

Figure 3 shows that `life_time` has high feature-importance scores (in particular, 0.445 in the distributed with mobility), thus reflecting its ability to capture the cumulative energetic consequence of the attack. Secondary features like `Rank` or `Num_of_neighbors` provide context, but Lifetime captures these effects in a single, highly predictive scalar. However, the extent of its dominance is highly scenario-dependent, indicating that the data-generating process is strongly moderated by both topological structure and node mobility. Interestingly, the most skewed feature importance is in the Distributed topology with mobility, where lifetime dominates the total predictive power. In comparison, the importance of features in the Center without mobility is more evenly distributed, suggesting a more complex, multifactorial decision boundary. The interplay between topology and mobility fundamentally changes the relative importance of routing metrics (e.g., `Rank`, `parent rank`), neighbor density (`Num_of_neighbors`), energy state indicators (all idle transmit, all idle listen), and traffic-related variables (`Time in Served`, `Forward Count`).

5.1.2 Blackhole Attack

The Blackhole attack is characterized by the adversarial node dropping all received packets while continuing to participate in routing control traffic. Consequently, as shown in Figure 4, at the overall level, no individual feature is overwhelmingly dominant across all four configurations. Instead, the importance profiles are moderated by the interaction of architectural topology and mobility dynamics. The dominant classifier in centralized architectures is traffic-forwarding efficiency (`Forward_ratio`) and temporal session metrics (`Time_in_Second`), while in distributed topologies, node structural classification (`Node type`, `Num Child`) is more relevant, especially in static modes. Mobility tends to emphasize the importance of a small set of dynamic features in centrally located deployments, whereas in distributed networks, it reconfigures the hierarchy from tree-structural properties toward forwarding behavior and node taxonomy variables. The spatial coordinate `pos_of_y` has only moderate predictive relevance in centralized settings; i.e., the geometric placement of nodes is meaningful only when routing is performed through a centered node.

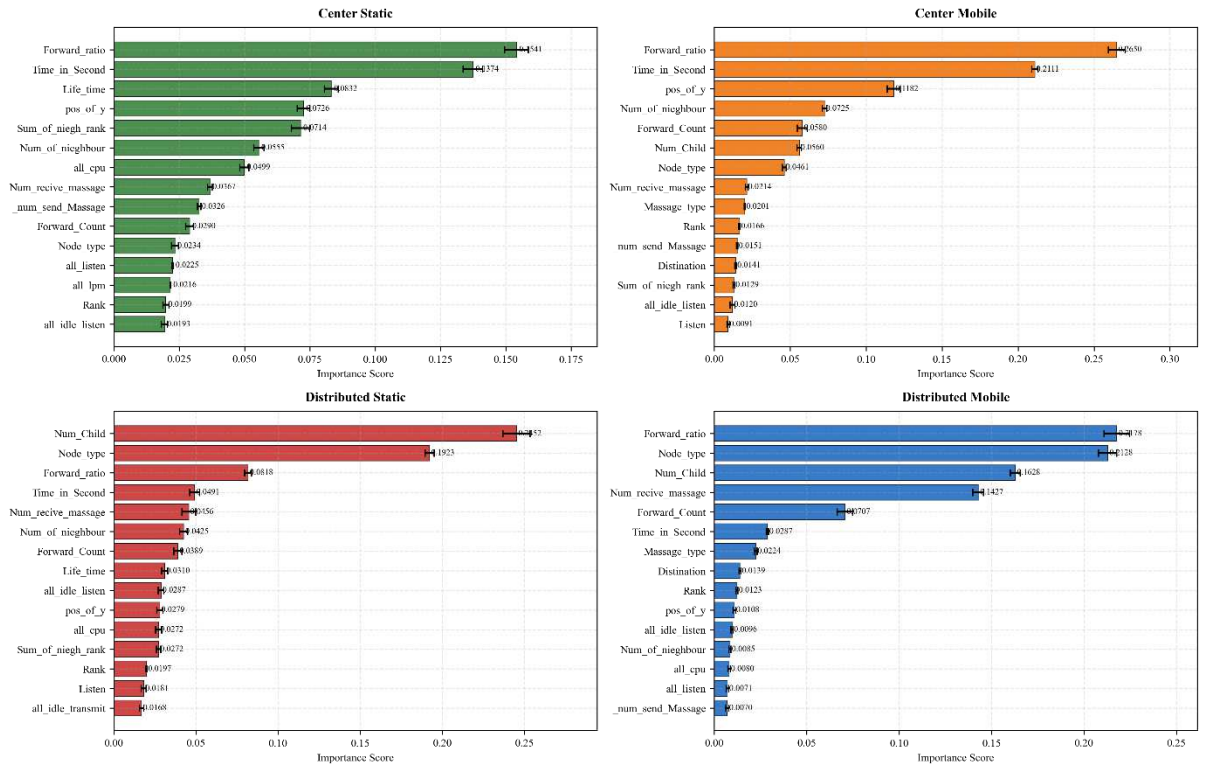


Figure 4: The Feature Scenario- Based comparison.

5.1.3 Selective Forwarding Attack

Selective Forwarding represents the most insidious attack class, as the adversary intermittently drops packets while maintaining plausible forwarding behavior. Figure 5 highlights the importance of features, which exhibit the similarity of features captured from the blackhole attack that happens due to the inheritance routing behavior between these two. This agreement is not surprising, as both attacks are essentially aimed at the packet forwarding process. In a Blackhole attack, the malicious node announces itself as an attractive route and then drops almost all incoming packets. Selective Forwarding attacks do the same forwarding manipulation but drop only a subset of packets according to a predefined strategy. Hence, these two attacks cause similar disruptions in traffic propagation, routing continuity, and communication efficiency; therefore, the Extra Trees algorithm identifies common sets of discriminative features. In all cases, metrics related to forward, such as Forward ratio and Forward Count, indicators of time such as Time in Second and Life_time, descriptors of neighborhood such as Num_of_neighbor and Num Child, and variables of resource-consumption, such as all cpu and Cpu, are repeatedly among the top-ranked attributes.

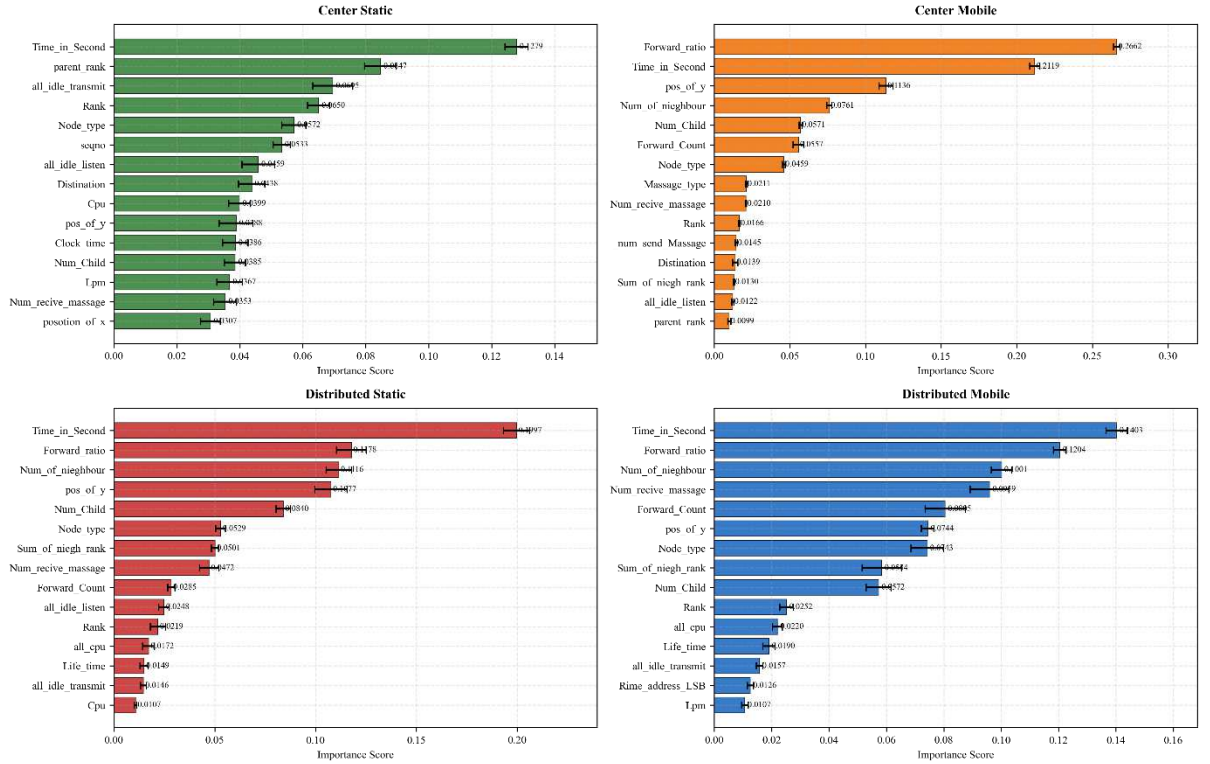


Figure 5: The Feature Scenario- Based comparison.

5.1.4 Worst Parent Attack

The Worst Parent attack (a sinkhole variant) manipulates the RPL parent selection process by advertising an attractive rank while possessing poor link quality or malicious intent. In the centralized configurations explored in Figure 6, the importance distributions exhibit a remarkable similarity across all deployment scenarios, namely the combined dataset, centralised static, centralised mobile, distributed static and distributed mobile settings.

In all the scenarios we observe a clear dominance of packet_ratio and this feature gets always the highest importance score by a considerable margin over the remaining features. This result implies that the Worst Parent attack is uniquely characterised by disruptions in packet forwarding behaviour, irrespective of network topology or node mobility. The consistency of this feature across all deployment configurations further suggests that packet forwarding

dynamics are an intrinsic behavioural signature of malicious parent selection, rather than a topology-dependent artefact.

The importance values decline gradually with no abrupt drops after the dominant feature, indicating that the detection model relies on a complementary set of routing and traffic related features rather than a single discriminative variable. The relative order of some secondary features changes slightly between centralised and distributed deployments, but the overall contribution is unchanged. It means that the network topology and mobility mainly affect the relative importance of supporting features while keeping the overall structure of the feature space intact.

Figure 6 also shows that the feature importance profiles are very similar for all scenarios, even when the network configuration differs. Such a degree of consistency indicates that the proposed feature selection strategy is robust to the variations in deployment and can successfully identify features that generalise across heterogeneous IoT environments. Hence, the selected feature subset is a robust representation of the Worst Parent attack behaviour, reducing the chances of scenario-specific feature bias and helping to achieve the strong generalisation performance of the machine learning classifiers in the subsequent experiments.

Overall, the results show that the proposed feature engineering approach is able to effectively capture the essential routing characteristics of the Worst Parent attack and is robust to topology changes and node mobility, thus providing a reliable basis for accurate and deployment-independent intrusion detection.

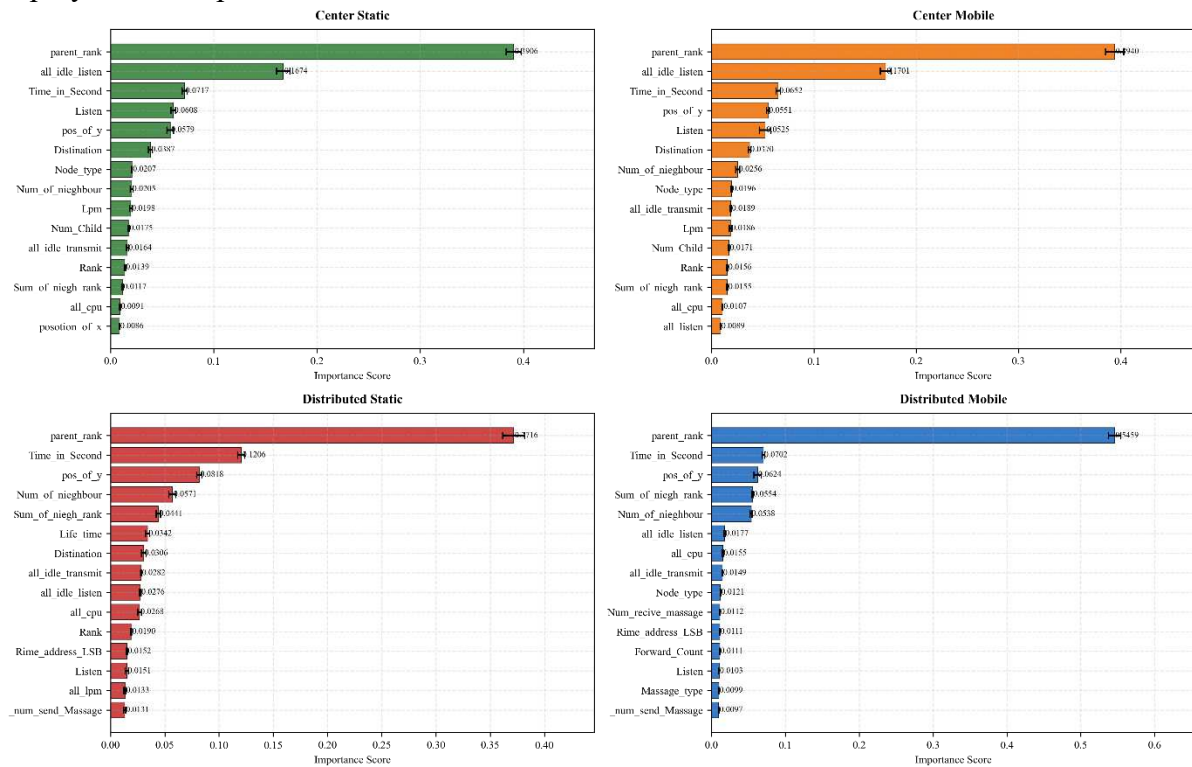


Figure 6: The Feature Scenario- Based comparison.

5.1.5 Flooding Attack

The feature importance analysis for Flooding attack detection for the four network topology scenarios (Figure 7) shows different discrimination patterns that are specific to the Hello Flood and DIO flooding attacks. In the centralized topologies with and without mobility, Destination is the most important feature with importance scores of 0.205 and 0.204, respectively. This finding implies that abnormal destination-address distributions are a good

indicator of flooding behaviour when the border router has a global view of the network traffic. The `all_idle_listen` feature is ranked second in both cases (0.113 and 0.122), showing that radio is listening all the time is a distinctive feature of flooding attacks. Listen and Time_in_Second (both 0.101) are more discriminative in the centralized topology without mobility, showing that the stable timing of legitimate DIO transmissions in static environments helps distinguish abnormal periodic flooding traffic. However, node mobility results in transient listening behaviour that partially hides these temporal patterns.

The feature ranking is quite different in distributed topologies because each node has only local observations and lacks information about the global traffic. In the distributed topology with mobility, the most prominent features are `all_idle_listen` (0.164) and `num_send_Message` (0.155), indicating that the detection of flooding attacks mainly by means of continuous channel occupancy and increased transmission activity. The relatively high importance of Listen (0.127) and `all_idle_transmit` (0.093) further show that local radio behaviour gives the strongest evidence for attack detection in mobile distributed environments.

The importance profile of the distributed topology without mobility is very different. Here, the most important feature is `Node_Type` with an importance score of 0.194 which is much higher than the other cases. This implies that the node roles information can be utilized as a robust behavioral baseline for anomaly detection induced by flooding, for static distributed networks. Also, `all_cpu` (0.115), `Dag_ID` (0.112) and `Num_Child` (0.072) are important features indicating the existence of flooding attacks, which create a measurable overhead in computation and structural changes to the DODAG (e.g., parent-child relationships) that can be detected. These structural indicators are less informative in mobile settings because valid topology changes produce similar variation.

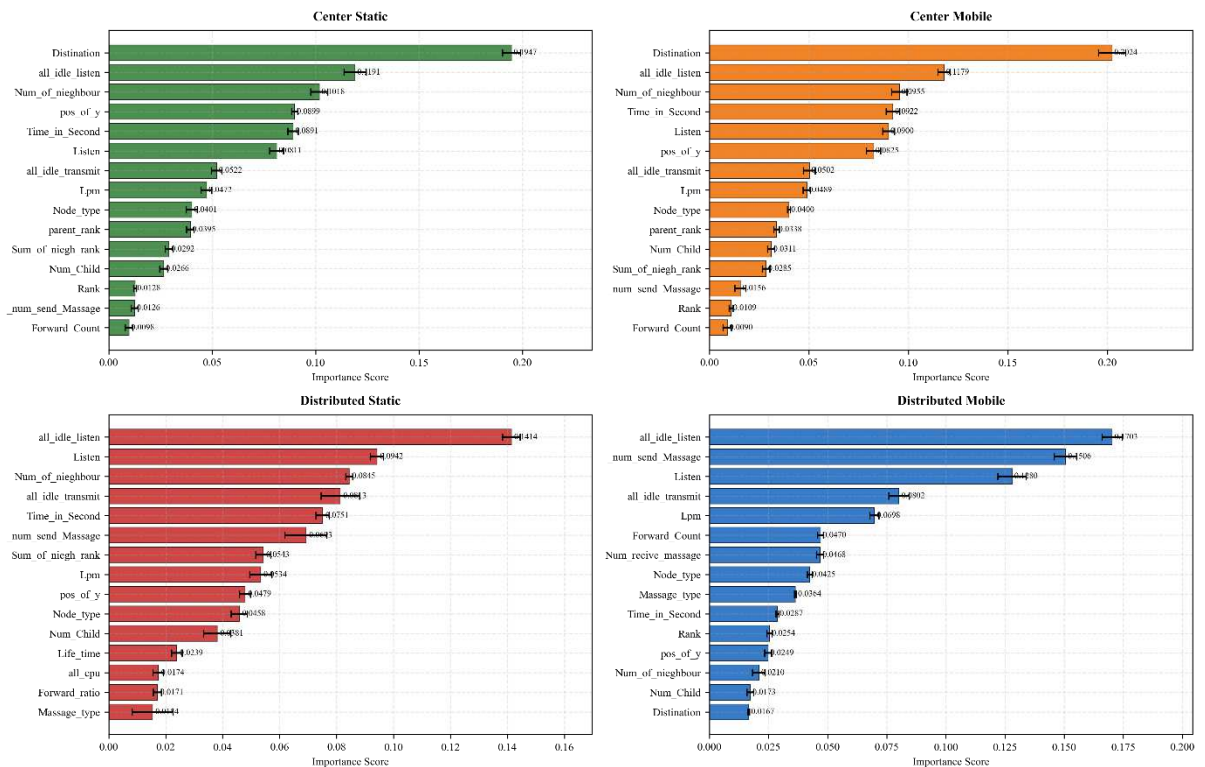


Figure 7: The Feature Scenario- Based comparison.

5.1.6 Combined Attacks

Figure 8 presents the feature importance analysis for the combined multi-attack scenario, which clearly shows a significant difference between the centralised and distributed network topologies. This highlights the ability of the ensemble model to dynamically adapt to the context when facing multiple routing attacks. For both mobile and non-mobile centralised topologies the importance scores are strongly concentrated in a few dominant features. In the static scenario the most influencing feature has a score of 0.250 and in the mobile scenario it has a score of 0.221. Then, the other feature scores fall fast. This distribution shows that the model is able to distinguish between different attack classes using a small number of highly informative traffic features, thanks to the border router providing global traffic visibility. The increasing importance of the leading feature in the centralised static topology further suggests that the discriminative power of the former is enhanced under stable network conditions, while node mobility introduces legitimate behavioural changes, which decrease the predictive capability of individual features and increase the contribution of secondary attributes.

The distributed topologies, however, have a much flatter distribution of feature importance. The highest importance scores are 0.111 in the static scenario and 0.079 in the mobile scenario, and the remaining features show a smooth decline and similar contribution. This trend indicates that without centralised traffic visibility, attack detection relies on the fusion of information from a larger set of complementary features, rather than a few dominant predictors.

The distributed topology with mobility has the most consistent importance profile, with the top fifteen ranked features ranging from 0.079 to 0.028. This result demonstrates that the simultaneous occurrence of multiple attacks, frequent parent switching and dynamic topology changes increase the complexity of the detection task significantly. Therefore, the model gives more importance to the structural descriptors (Network_Category and Node_Type), and distributes the remaining weights to several traffic and topology related features, to capture subtle attack specific behavioural patterns that are not directly inferable from local network information.

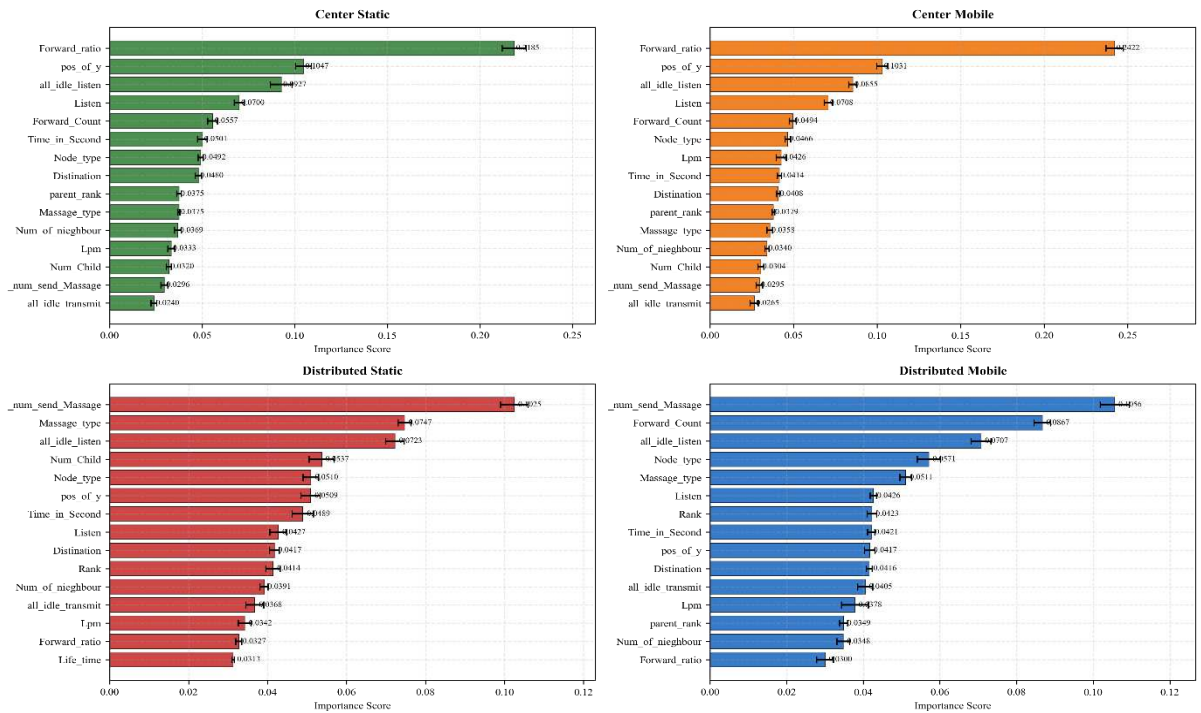


Figure 8: The Feature Scenario- Based comparison.

5.2 Machine Learning Performance by Attack and Scenario

Second, we checked the discriminative power and generalization ability of the proposed dataset using several supervised machine learning classifiers. The proposed dataset was evaluated with the selected feature set using eight commonly used classifiers in five representative deployment scenarios, including the all_combined dataset and centralized and distributed topologies under static and mobile conditions.

This evaluation aimed to test if the generated dataset contains enough informative and topology-independent behavioral characteristics that can be consistently learned by different machine learning paradigms. Experimental results showed high classification performance in all evaluation scenarios and ensemble-based classifiers showed better robustness and stability. The high values of Accuracy, Precision, Recall, Specificity, F1-score, and AUC demonstrate the proposed dataset's ability to capture the unique behavioral signatures of routing attacks, with competitive generalization across diverse network topologies and mobility conditions. The detailed classification results for each routing attack are given in Tables 4–10 and Figures 9–14. All reported performance measures are macro-averaged values of a stratified five-fold cross-validation.

5.2.1 Decrease Rank Attack Detection

The Decrease Rank attack achieved the best detection performance of all evaluated routing attacks, suggesting that its routing manipulation behaviour results in very distinguishable traffic patterns under diverse deployment conditions. The ensemble-based classifiers performed better than all the other classifiers, and the Extra Trees had the best mean F1-score value (0.9996), with the Random Forest, K-Nearest Neighbors, and Decision Tree following closely with a value above 0.998. In particular, Random Forest, Extra Trees, KNN and Logistic Regression achieved comparatively higher detection performance ($F1 = 1.0000$) in the all_combined scenario, which indicates the fusion of multiple network topologies and mobility conditions provides more comprehensive characterisation of the routing behaviour, thus improving the separability between legitimate and malicious traffic. Conversely, MLP and SVM exhibited relatively lower performance, with Logistic Regression having the greatest variation across deployment scenarios, suggestive of the fact that linear decision boundaries are less capable of modelling the complex nonlinear interactions introduced by routing-layer attacks.

Scenario	Classifier	Accuracy	Precision	Recall	F1-Score	AUC-ROC	FPR
All Combined	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.99957	0.99977	0.9975	0.9986	0.999207	0
	DecisionTree	0.99994	0.99969	0.99997	0.99983	0.99997	0.0008
	GradientBoosting	0.99994	0.99969	0.99997	0.9998	1	0.00102
	KNN	1	1	1	1	1	0
	SVM	1	1	1	1	1	0
	LogisticRegression	1	1	1	1	1	0
Distributed Without Mobility	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.99854	0.9973	0.97549	0.985709	0.99995	0.048898
	DecisionTree	0.99977	0.9979	0.9979	0.9979018	0.9979	0.004082
	GradientBoosting	0.99988	0.999943	0.99795	0.9989	1	0.004082
	KNN	1	1	1	1	1	0
	SVM	0.97923	0.78522	0.989325	0.8576	0.99736	0
	LogisticRegression	0.973320	0.75486	0.9862	0.8302	0.998648	0
	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.988394	0.98805	0.918015	0.9481	0.998792	0.163102
	DecisionTree	0.99973	0.999856	0.997959	0.99889	0.99795	0.004082

Distributed With Mobility	GradientBoosting	0.99973	0.998	0.999855	0.9989	1	0
	KNN	0.99946	0.996078	0.99971	0.99785	1	0
	SVM	0.990013	0.93506	0.99465	0.96245	0.99797	0
	LogisticRegression	0.97354	0.857710	0.9858	0.90983	0.99868	0
Center With Mobility	RandomForest	0.99990	0.99995	0.99795	0.998945	1	0.004082
	ExtraTrees	1	1	1	1	1	0
	MLP	0.9975	0.99464	0.94913	0.969967	0.9994	0.10155
	DecisionTree	0.9999	0.99995	0.99795	0.99894	0.99795	0.004082
	GradientBoosting	0.99963	0.99223	0.99981	0.995925	0.9998	0
	KNN	1	1	1	1	1	0
	SVM	0.99344	0.89009	0.996649	0.935926	0.99917	0
	LogisticRegression	0.98902	0.8394	0.99438	0.900754	0.99963	0
Center Without Mobility	RandomForest	0.99986	0.99993	0.99387	0.99685	0.9999	0.0122
	ExtraTrees	0.99991	0.99995	0.99591	0.99789	1	0.008163
	MLP	0.999461	0.99775	0.97773	0.987252	0.999901	0.04448
	DecisionTree	0.99995	0.99997	0.99795	0.9989577	0.99795	0.004082
	GradientBoosting	0.999776	0.9979	0.99181	0.9947665	1	0.01632
	KNN	0.99991	0.99995	0.99591	0.99789	0.9959	0.008163
	SVM	0.9910	0.7771	0.9914	0.853197	0.9986	0.008163
	LogisticRegression	0.9836	0.7021	0.99172	0.78344	0.9993	0

Table 4: ML Performance for Decrease Rank Attack by Scenario.

In all deployment scenarios, the classifier performance was consistently high. However, significant differences were observed depending on the network configuration. The all_combined scenario provided the best overall performance, with most classifiers achieving near-perfect classification results, demonstrating the effectiveness of training under heterogeneous deployment conditions. In contrast, the center_static scenario was the hardest scenario for several conventional classifiers, especially Logistic Regression, MLP and Gradient Boosting, which had a measurable drop in F1-score. Similar behaviour was observed in the distributed_static scenario, but with less pronounced degradation. In contrast, the mobile deployment scenarios mostly improved the classification performance with Random Forest and Extra Trees maintaining strong detection, and the other classifiers with moderate performance gains. These results demonstrate that mobility adds further routing variations which improve the discriminative properties of the selected feature set, especially for the case of ensemble learning algorithms.

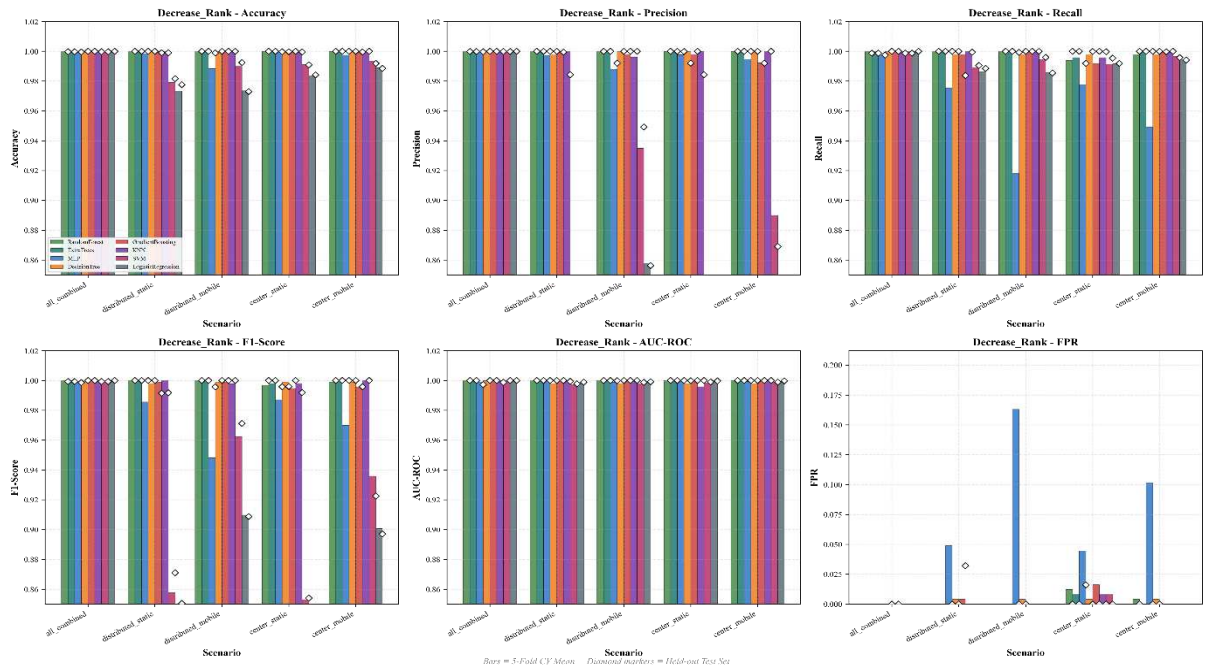


Figure 9: classifier comparison.

The selected feature set has shown consistently high robustness to different network topologies for both centralised and distributed deployments. The ensemble-based classifiers showed almost equal performance in both deployment settings, indicating a high resilience to topological variations. Interestingly, the all_combined dataset performed consistently better than many scenario-specific datasets, particularly for classifiers with relatively simple decision boundaries such as Logistic Regression. The result implies that the models can learn more generalised decision boundaries rather than topology-specific patterns, when they are exposed to a wider variety of routing behaviours.

In addition, the distributed deployment scenarios generally achieved similar or slightly better performance than the centralised scenarios. Among the evaluated configurations, the center_mobile case performed more consistently across all classifiers, showing that moderate topology dynamics can improve the generalisation of the models without sacrificing classification accuracy. In general, the results show that the proposed approach is suitable for heterogeneous IoT and LLN environments, since it includes various deployment conditions in the model development, which enhances robustness and generalisation, while maintaining high classification performance.

5.2.2 Blackhole Attack Detection

The Blackhole attack showed consistently high detection performance across all tested deployment scenarios, indicating the effectiveness of the proposed feature representation in identifying routing anomalies related to malicious packet dropping behaviour.

Scenario	Classifier	Accuracy	Precision	Recall	F1-Score	AUC-ROC	FPR
all combined	RandomForest	0.99849	0.9937	0.98811	0.990816	0.99996	0.02324
	ExtraTrees	0.99879	0.99439	0.99105	0.99267	0.999964	0.01743
	MLP	0.99394	0.9725	0.95302	0.96244	0.99913	0.09174
	DecisionTree	0.99879	0.99124	0.99437	0.99276	0.99437	0.01045
	GradientBoosting	0.99974	0.9987	0.998204	0.9984	0.99997	0.00348
	KNN	0.99214	0.9467	0.960392	0.95323	0.995056	0.07435
	SVM	0.96752	0.785346	0.98137	0.85426	0.992347	0.00346
	LogisticRegression	0.94706	0.72375	0.965692	0.79387	0.98571	0.01392
Distributed With Mobility	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.99534	0.99744	0.976	0.98636	0.9977	0.048
	DecisionTree	1	1	1	1	1	0
	GradientBoosting	1	1	1	1	1	0
	KNN	1	1	1	1	1	0
	SVM	0.985663	0.93846	0.98849	0.961505	0.9997	0.008
	LogisticRegression	0.98798	0.95513	0.980852	0.967002	0.999682	0.028
Distributed Without Mobility	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.99595	0.976011	0.959258	0.96734	0.99789	0.08
	DecisionTree	1	1	1	1	1	0
	GradientBoosting	1	1	1	1	1	0
	KNN	0.99973	0.99615	0.999865	0.9979	0.99984	0
	SVM	0.98501	0.845719	0.992248	0.903902	0.997308	0
	LogisticRegression	0.98096	0.81906	0.980495	0.88106	0.997705	0.02
Center With Mobility	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.999203	0.99956	0.995901	0.9977	0.999835	0.008197
	DecisionTree	1	1	1	1	1	0
	GradientBoosting	1	1	1	1	1	0
	KNN	0.999840	0.99918	0.999911	0.9995	1	0
	SVM	0.99968	0.99837	0.999823	0.9991	1	0
	LogisticRegression	1	1	1	1	1	0
	RandomForest	0.99764	0.99633	0.978560	0.98713	0.999728	0.0426
	ExtraTrees	0.99732	0.99008	0.981497	0.98555	0.9998431	0.0361
	MLP	0.9929	0.96605	0.958212	0.96204	0.998673	0.0803
	DecisionTree	0.997483	0.9875	0.98624	0.98662	0.986244	0.0262
	GradientBoosting	0.998619	0.99616	0.989173	0.9925	0.999736	0.0213

Center Without Mobility	KNN	0.987577	0.9289	0.94296	0.93517	0.9916	0.1065
	SVM	0.96086	0.7799	0.97941	0.84821	0.9936	0
	LogisticRegression	0.9608	0.77990	0.97941	0.84821	0.98941	0

Table 5: ML Performance for Blackhole Attack by Scenario.

The ensemble-based classifiers proved to be the most robust in all deployment scenarios. Gradient Boosting had the highest average F1-score of 0.9982, followed by Decision Tree, Extra Trees and Random Forest, all of which had mean F1-scores greater than 0.995. These tree-based methods attained perfect classification performance ($F1 = 1.000$) for both distributed deployment scenarios, demonstrating their capability of capturing the complex nonlinear relationships among routing, traffic, and topology-related features. The traditional classifiers, in contrast, performed less well and less consistently. MLP and KNN still demonstrated competitive performance but the average performance of ensemble methods was slightly better. SVM and Logistic Regression had the lowest average F1-scores, indicating that linear or kernel-based decision boundaries may not be as effective in modelling the complex behavioural patterns associated with Blackhole attacks.

High classification performance was achieved for all deployment scenarios, but differences between network configurations were observed. The most challenging setting was the centralised static scenario, where several classifiers showed a moderate drop in detection performance. The largest decrease was observed for KNN while the ensemble-based methods maintained F1-scores close to unity with only marginal performance degradation as compared to the distributed scenarios. In contrast, the centralised mobile scenario resulted in higher classification performance for almost all examined models. The Random Forest, Extra Trees, Decision Tree, Gradient Boosting and Logistic Regression achieved almost perfect detection at zero false positive rate. The KNN and SVMs also showed great improvements from their results of the centralised static configuration. These results indicate that the mobility of the nodes adds an extra routing dynamics to increase the behavioural separation between legitimate and malicious traffic to improve the discriminative ability of the standard ML models

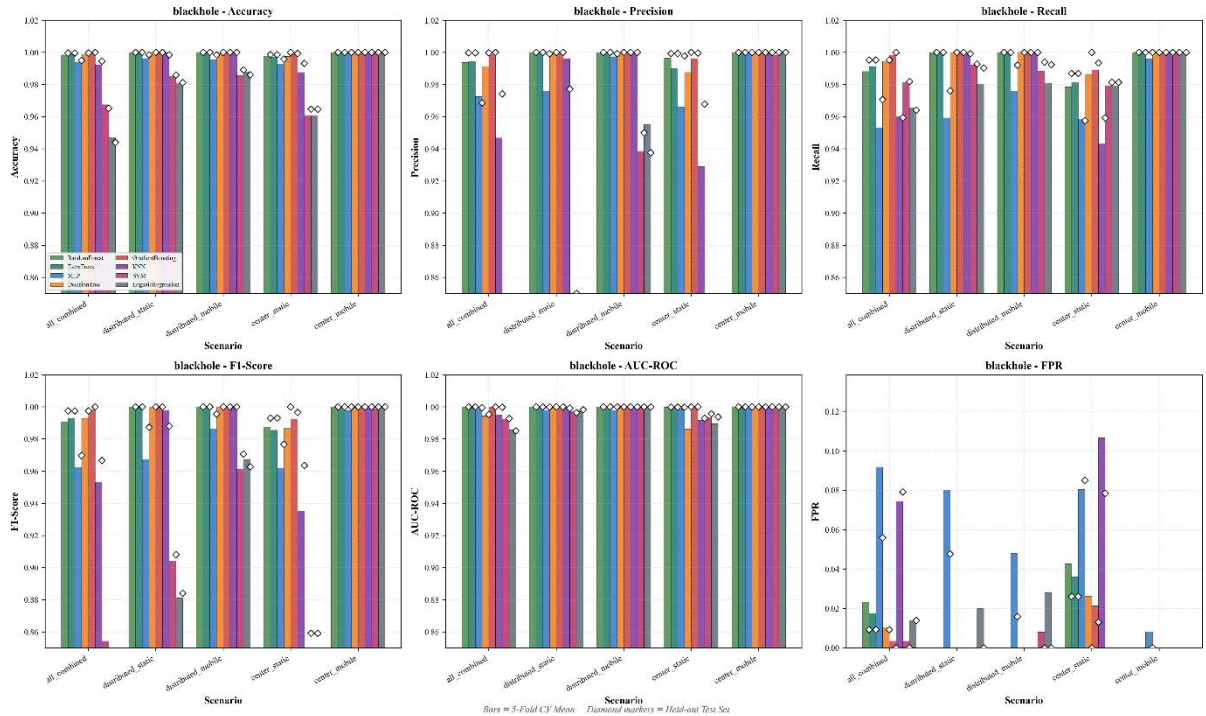


Figure 10: classifier comparison.

By comparison of centralised and distributed deployments we see a clear influence of the network topology on the detection performance. The best classification results were achieved for both distributed static and distributed mobile scenarios. The top ensemble classifiers achieved the highest results in all evaluation metrics. The all_combined dataset, in contrast, achieved slightly lower but highly competitive results, because it considers the heterogeneous routing behaviours originating from multiple deployment conditions. Interestingly, the increased classification complexity did not affect the overall best performance of the Gradient Boosting, which was robust to topology and mobility variations. The relatively poor performance of SVM and Logistic Regression also suggests that simple linear decision boundaries are not enough to adequately model Blackhole attack behaviour. Instead, the improved performance of tree-based ensemble models suggests that the attack causes complex non-linear interactions between routing, traffic and energy related features that are better modelled by ensemble learning. In conclusion, the findings show that distributed implementations result in more separable feature spaces than centralised static networks, enabling a more robust detection of Blackhole attacks and highlighting the importance of considering deployment diversity when building machine learning-based intrusion detection systems in IoT settings.

5.2.3 Selective Forwarding Attack Detection

The Selective Forwarding attack showed the highest variation in terms of classification performance among all the tested routing attacks, highlighting the higher difficulty in detecting a selective packet dropping behaviour under different deployment scenarios. In brief, the best average performance was Extra Trees with a consistent mean F1-score (1.0000). K-Nearest Neighbours, Random Forest, Decision Tree, and Gradient Boosting had all great detection ability with mean F1-scores close to one. In contrast, the traditional classifiers showed much more variation in performance. Although SVM and Logistic Regression achieved good overall results, MLP had the lowest average performance across all considered attacks (mean F1 = 0.9117), indicating that neural network-based classification was considerably more sensitive to changes in network topology and deployment conditions than ensemble-based approaches.

Scenario	Classifier	Accuracy	Precision	Recall	F1-Score	AUC-ROC	FPR
all combined	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.9991	0.9995	0.9881	0.993745	0.9997	0.023
	DecisionTree	1	1	1	1	1	0
	GradientBoosting	1	1	1	1	1	0
	KNN	1	1	1	1	1	0
	SVM	1	1	1	1	1	0
	LogisticRegression	0.9993	0.99396	0.9967	0.99537	0.9999	0.0059
Distributed With Mobility	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.99769	0.9988	0.9285	0.96095	1	0.1428
	DecisionTree	1	1	1	1	1	0
	GradientBoosting	1	1	1	1	1	0
	KNN	1	1	1	1	1	0
	SVM	0.9976	0.9375	0.9988	0.9661	1	0
	LogisticRegression	0.9965	0.91176	0.9982	0.950733	1	0
	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.9987	0.99938	0.9285	0.96123	1	0.1428
	DecisionTree	1	1	1	1	1	0
	GradientBoosting	1	1	1	1	1	0
	KNN	1	1	1	1	1	0

Distributed Without Mobility	SVM	0.9987	0.9375	0.9993	0.9664	1	0
	LogisticRegression	0.99023	0.7333	0.9951	0.81570	1	0
Center With Mobility	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	1	1	1	1	1	0
	DecisionTree	1	1	1	1	1	0
	GradientBoosting	1	1	1	1	1	0
	KNN	1	1	1	1	1	0
	SVM	1	1	1	1	0.999	0
Center Without Mobility	LogisticRegression	1	1	1	1	1	0
	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.99845	0.4992	0.5	0.4996	1	0
	DecisionTree	1	1	1	1	1	0
	GradientBoosting	1	1	1	1	1	0
	KNN	1	1	1	1	1	0
	SVM	1	1	1	1	1	0
	LogisticRegression	1	1	1	1	1	0

Table 6: ML Performance for Selective Forward Attack by Scenario.

The overall performance of the all_combined scenario was competitive, but the detailed scenario-wise analysis revealed that the performance varies significantly across different deployment configurations. The aggregated dataset resulted in consistently high classification performance for most classifiers, with Extra Trees and KNN achieving high detection, and Random Forest close to perfect classification. However, this overall performance masked significant degradation in some deployment scenarios. The most difficult environment for Selective Forwarding detection was the static centralised scenario. Ensemble-based classifiers experienced only a moderate decrease in performance, whereas the MLP classifier showed a severe decrease in both Precision and Recall, resulting in the lowest F1-score observed throughout the experimental study. Similarly, Logistic Regression also suffered a significant performance drop under this setup, indicating that the feature distribution generated by selective packet dropping in static centralized networks is harder to model using linear or shallow neural architectures. In contrast, the classification performance of SVM and Extra Trees remained high, indicating greater robustness to this harsh deployment condition. The centralized mobile scenario presented a significant performance improvement, with almost all the evaluated classifiers achieving near-perfect classification results. In the distributed deployment scenarios, the improvements were similar, but still MLP showed more sensitivity than the other classifiers. These results suggest that mobility adds additional routing dynamics that make the Selective Forwarding behaviour more distinguishable, thus allowing for more accurate discrimination between legitimate forwarding actions and malicious packet-dropping activities.

A comparison of deployment scenarios shows that the performance of Selective Forwarding is highly sensitive to network topology and node mobility. Unlike the other routing attacks, the performance of the classifiers changed significantly across different deployment scenarios, suggesting that the behavioral patterns of selective packet dropping are heavily dependent on the stability of the routing process and the dynamics of the network. The centralized static topology was consistently the worst in terms of detection performance, particularly for neural network and linear classifiers, while the centralized mobile configuration achieved the highest and most consistent classification results for almost all models tested. Interestingly, tree-based ensemble methods performed consistently high across different deployment configurations, showing their higher capability to capture nonlinear interactions between routing, traffic, and topology-related features due to Selective Forwarding attacks.

Conversely, the large degradation observed for the MLP classifier in the centralized static scenario suggests that static routing conditions lead to feature distributions that are hard to distinguish for common neural architectures. The subsequent improvement in detectability for mobile deployments further supports the hypothesis that topology variation and routing dynamics offer more discriminative information that improves attack detectability. In conclusion, these results show the importance of considering deployment diversity for the correct evaluation of intrusion detection systems, and at the same time, they highlight the better robustness and generalization ability of ensemble learning methods for detecting Selective Forwarding attacks in a heterogeneous IoT environment.

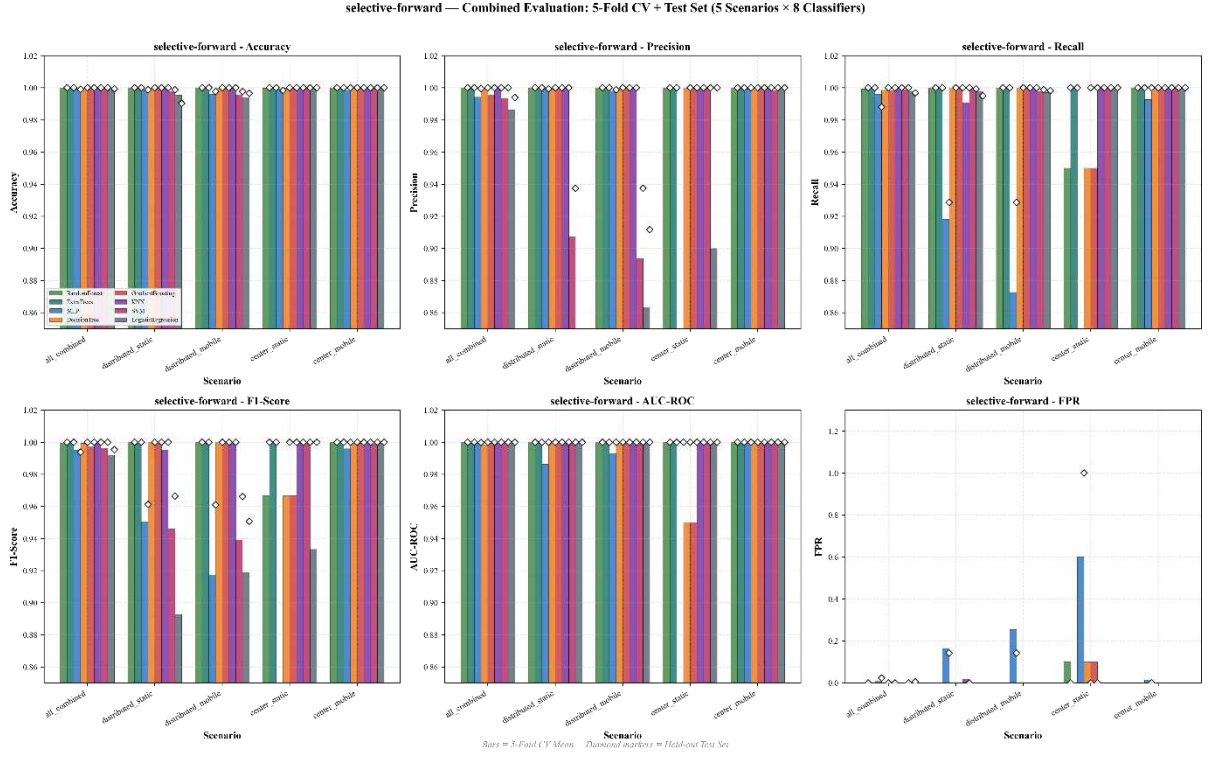


Figure 11: classifier comparison.

5.2.4 Worst Parent Attack Detection

The Worst Parent attack showed consistently high detection performance in all the deployment scenarios considered in the evaluation, indicating that the proposed feature representation is able to capture the routing anomalies associated with malicious parent selection. Overall, Extra Trees performed the best on average (mean F1 = 0.9995) followed by Random Forest, K-Nearest Neighbours, Gradient Boosting and Decision Tree, which all scored the mean F1-scores higher than 0.9998. The minor performance variations across the top ensemble classifiers suggest that the behavioural features of Worst Parent attacks result in very distinct routing patterns, irrespective of the underlying learning algorithm. In contrast, MLP, SVM and Logistic Regression showed relatively lower average performance, though still kept a satisfactory detection capability. Moreover, consistently low false-positive rates for all evaluated models further confirm that normal routing behaviour can be effectively distinguished from malicious parent selection, demonstrating the reliability of the proposed feature set for intrusion detection.

In all deployment scenarios, competitive classification performance was achieved, however, a few differences dependent on the topology were observed. The classification results

obtained in the all_combined case were very robust. Random Forest and Extra Trees presented an almost perfect detection performance with very low false-positive rates. Slight performance degradation was observed under the distributed static deployment, especially for Gradient Boosting, Decision Tree, MLP and Logistic Regression. However, the corresponding decreases were still relatively small for the ensemble-based classifiers confirming their robustness under heterogeneous routing conditions .

In the distributed mobile scenario, a very different behavior was observed, where all the classifiers evaluated reached competitive classification performance and all the evaluation metrics. This is the best overall detection performance observed in the entire experimental study and it shows that the routing dynamics induced by node mobility significantly improve the distinguishability of malicious parent selection behaviour. The centralized static and centralized mobile scenarios resulted in intermediate performance, with ensemble methods consistently achieving near-perfect classification, while SVM and Logistic Regression were relatively more sensitive to deployment-specific variations.

Scenario	Classifier	Accuracy	Precision	Recall	F1-Score	AUC-ROC	FPR
all combined	RandomForest	0.99995	0.99997	0.9995	0.99974	0.9995	0.00095
	ExtraTrees	0.99990	0.999504	0.9995	0.9995	0.99951	0.00095
	MLP	0.99918	0.99427	0.9968	0.9955	0.9993	0.00570
	DecisionTree	0.99954	0.99578	0.9993	0.99752	0.9993	0.00095
	GradientBoosting	0.99986	0.99903	0.9994	0.9992	0.9995	0.00095
	KNN	0.99968	0.99805	0.99847	0.99825	0.9989	0.00285
	SVM	0.9927	0.9344	0.9957	0.96275	0.99873	0.00095
	LogisticRegression	0.9868	0.89221	0.9926	0.93573	0.996474	0.00095
Distributed With Mobility	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.9971	0.998389	0.9802	0.989006	0.99902	0.03952
	DecisionTree	1	1	1	1	1	0
	GradientBoosting	1	1	1	1	1	0
	KNN	1	1	1	1	1	0
	SVM	1	1	1	1	1	0
	LogisticRegression	1	1	1	1	1	0
Distributed Without Mobility	RandomForest	0.99979	0.99989	0.9961	0.997948	0.9979	0.00792
	ExtraTrees	0.99979	0.99802	0.99794	0.997967	0.9979	0.004
	MLP	0.99807	0.9837	0.9778	0.98039	0.9973	0.04337
	DecisionTree	0.99969	0.99617	0.99789	0.99698	0.9978	0.004
	GradientBoosting	0.9996	0.997972	0.9959	0.99695	0.998	0.00792
	KNN	0.9998	0.99994	0.998	0.99896	0.998	0.004
	SVM	0.99573	0.931858	0.99586	0.9611	0.9975	0.004
	LogisticRegression	0.9873	0.838951	0.9915	0.899102	0.9962	0.004
Center With Mobility	RandomForest	0.9998	0.999914	0.9987	0.99932	1	0.00253
	ExtraTrees	1	1	1	1	1	0
	MLP	0.99791	0.989601	0.9929	0.991267	0.99984	0.0126
	DecisionTree	0.99983	0.99991	0.9987	0.99932	0.99873	0.0025
	GradientBoosting	0.9998	0.999914	0.99873	0.99932	0.99873	0.0025
	KNN	0.99983	0.99991	0.99873	0.99932	1	0.0025
	SVM	0.99069	0.9364	0.9951	0.963439	0.99998	0
	LogisticRegression	0.99262	0.94783	0.9961	0.970476	0.9995	0
Center Without Mobility	RandomForest	0.99967	0.9998	0.9975	0.99866	0.9999	0.0049
	ExtraTrees	1	1	1	1	1	0
	MLP	0.9976	0.9917	0.9895	0.990577	0.9995	0.0199
	DecisionTree	0.99967	0.99982	0.9975	0.99866	0.99751	0.0049
	GradientBoosting	0.9996	0.99982	0.9975	0.99867	0.9975	0.0049
	KNN	0.9996	0.99867	0.9986	0.99867	0.9999	0.0024
	SVM	0.9925	0.95007	0.9961	0.9716	0.9994	0
	LogisticRegression	0.9929	0.95268	0.9962	0.97303	0.9993	0

Table 7: ML Performance for Worst Parent Attack.

The comparison of the different deployment scenarios reveals that Worst Parent attacks are highly affected by the network topology and node mobility. While the performance differences were relatively small for static deployments, adding mobility consistently improved the performance of all evaluated classifiers. In the distributed mobile scenario, the detection performance was consistently high across all evaluated learning algorithms used. This observation suggests that dynamic parent reselection may introduce richer temporal and routing features that can be learned by the standard ML models more easily than the relatively stationary routing behaviour in static networks.

Interestingly, the topology effect was different for conventional and ensemble-based classifiers. Random Forest, Extra Trees, Gradient Boosting and KNN showed a stable high performance in all deployment scenarios, while SVM and Logistic Regression were more sensitive to topology changes, meaning that linear decision boundaries are less able to capture the complex interactions introduced by the malicious parent selection. The better performance obtained with the all_combined scenario suggests that exposure to more in the classifier, when trained under heterogeneous deployment conditions, while improving the model generalization in diverse IoT environments. Overall, these results confirm the robustness of ensemble methods and their independence from topologies for detecting Worst Parent attacks, but also highlight the need for testing intrusion detection systems in both static and mobile network environments.

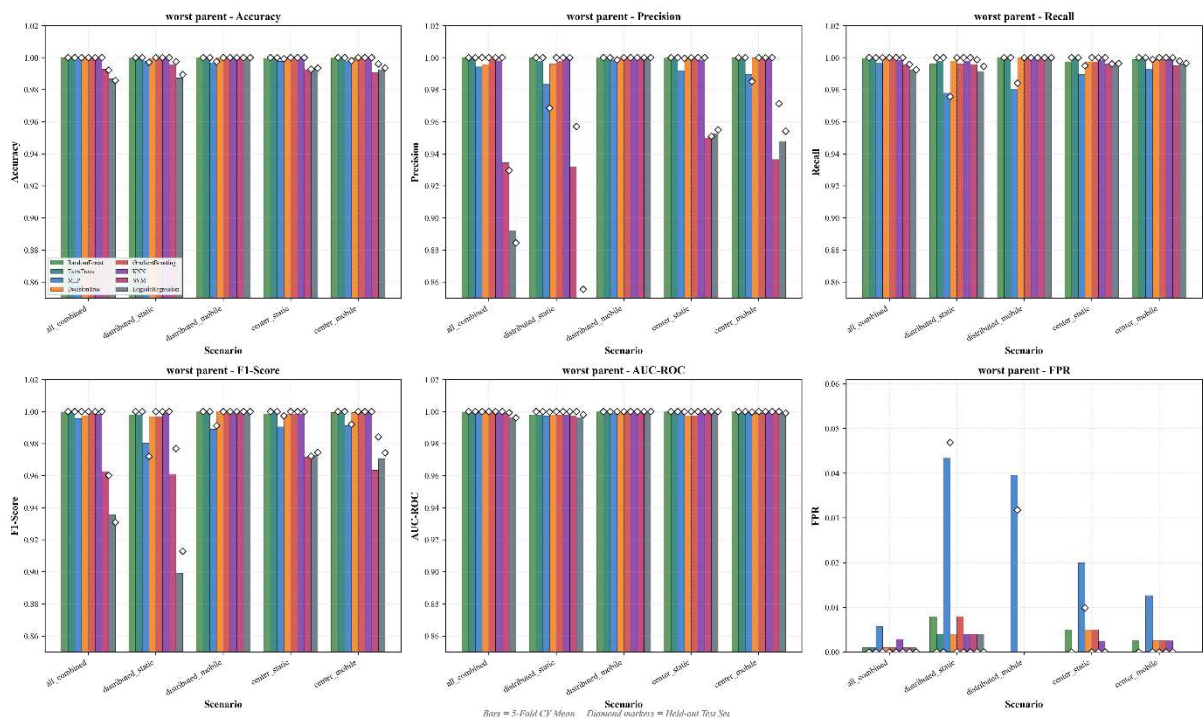


Figure 12: classifier comparison.

5.2.5 Flooding Attack Detection

The Flooding attack exhibited high detection performance for all the evaluated classifiers. This implies that the selected feature set can effectively capture the excessive routing traffic generated during flooding attacks. The overall best performance was shown by ensemble based classifiers with Extra Trees, Gradient Boosting and Random Forest all achieving the highest mean F1-score (0.9993). Meanwhile, Decision Tree and k-Nearest Neighbours both also

achieved almost perfect classification performance. SVM and Logistic Regression had relatively lower average performances. But all the classifiers tested got mean F1-scores higher than 0.93. The corresponding AUC values remained consistently close to one, indicating excellent class separability for whatever learning algorithm. Contrary to the other routing attacks studied in this work, flooding showed relatively small performance variations across classifiers, indicating that this attack produces characteristic and consistently distinguishable behavioural patterns.

All deployment scenarios achieved high classification performance, although network configurations showed moderate differences. In most classifiers the all_combined dataset gave the best overall performance, while Decision Tree had the best F1-score. In this heterogeneous setting, Random Forest, Extra Trees and Gradient Boosting also resulted in near perfect classification performance. Conversely, Logistic Regression and SVM showed more severe performance drops, indicating that they are more sensitive to the diversity of routing behaviours in the combined dataset. Overall, these findings suggest that the selected feature set has enough discriminative information to reliably detect Flooding attacks in heterogeneous IoT and LLN deployment scenarios, whereas ensemble learning methods consistently show the highest robustness and generalisation performance.

Scenario	Classifier	Accuracy	Precision	Recall	F1-Score	AUC-ROC	FPR
all combined	RandomForest	0.99992	0.99996	0.99955	0.9998	1	0
	ExtraTrees	0.99992	0.99996	0.9995	0.9997	0.9999	0
	MLP	0.9988	0.9951	0.9967	0.9959	0.9994	0.000802
	DecisionTree	0.99996	0.99998	0.9997	0.9998	0.9997	0
	GradientBoosting	0.9999	0.99996	0.9995	0.9997	0.9996	0
	KNN	0.99936	0.9966	0.9991	0.9978	0.9994	0.000573
	SVM	0.9654	0.8503	0.9806	0.9016	0.9968	0.037391
	LogisticRegression	0.9227	0.7539	0.9564	0.8144	0.9891	0.083566
Distributed With Mobility	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.999796	0.9994	0.999874	0.9996	1	0.000253
	DecisionTree	0.99959	0.9997	0.99895	0.9993	0.9989	0
	GradientBoosting	1	1	1	1	1	0
	KNN	1	1	1	1	1	0
	SVM	0.98798	0.9712	0.9925	0.9812	0.9987	0.01492
	LogisticRegression	0.98818	0.97149	0.9926	0.9816	0.9994	0.01466
Distributed Without Mobility	RandomForest	0.99986	0.99993	0.9949	0.9974	1	0
	ExtraTrees	0.99986	0.99993	0.99499	0.9974	1	0
	MLP	0.99973	0.99986	0.99023	0.9949	0.9899	0
	DecisionTree	0.99986	0.99993	0.99499	0.9974	0.9949	0
	GradientBoosting	0.99987	0.99993	0.99499	0.9974	1	0
	KNN	0.99973	0.99986	0.9902	0.9949	1	0
	SVM	0.99986	0.99993	0.9949	0.9974	0.9999	0
	LogisticRegression	1	1	1	1	1	0
Center With Mobility	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.9987	0.99481	0.9962	0.99549	0.9999	0.000814
	DecisionTree	1	1	1	1	1	0
	GradientBoosting	1	1	1	1	1	0
	KNN	0.99987	0.99993	0.99915	0.9995	1	0
	SVM	0.99887	0.9926	0.99938	0.9959	1	0.001221
	LogisticRegression	0.9821	0.90476	0.99036	0.9423	0.9998	0.01926
Center Without Mobility	RandomForest	0.99987	0.99993	0.99916	0.9995	0.9991	0
	ExtraTrees	0.99987	0.99993	0.99916	0.99954	1	0
	MLP	0.99863	0.99546	0.99466	0.99503	0.999865	0.000673
	DecisionTree	0.99987	0.99993	0.99916	0.9995	0.9991	0
	GradientBoosting	0.99987	0.99993	0.9992	0.9995	0.9989	0
	KNN	0.99962	0.99979	0.9975	0.9986	0.99749	0
	SVM	0.99887	0.9933	0.9986	0.9959	0.99999	0.001076
	LogisticRegression	0.97921	0.8917	0.9881	0.9332	0.9996	0.022332

Table 8: ML Performance for Flooding Attack.

In general, the distributed deployment scenarios showed better classification performance, with most of the ensemble based classifiers showing almost perfect detection. The best performance of the Logistic Regression was obtained in the distributed static scenario, which is the best performance among all the experiments evaluated. Also, the distributed mobile scenario consistently yields high results for almost all classifiers. The central mobile deployment also maintained high classification performance, with only MLP and Logistic Regression showing small decreases compared to the best-performing ensemble methods. Results show that the selected feature set is still effective in heterogeneous deployment environment and significant differences of classifier sensitivities to topology and node mobility are observed.

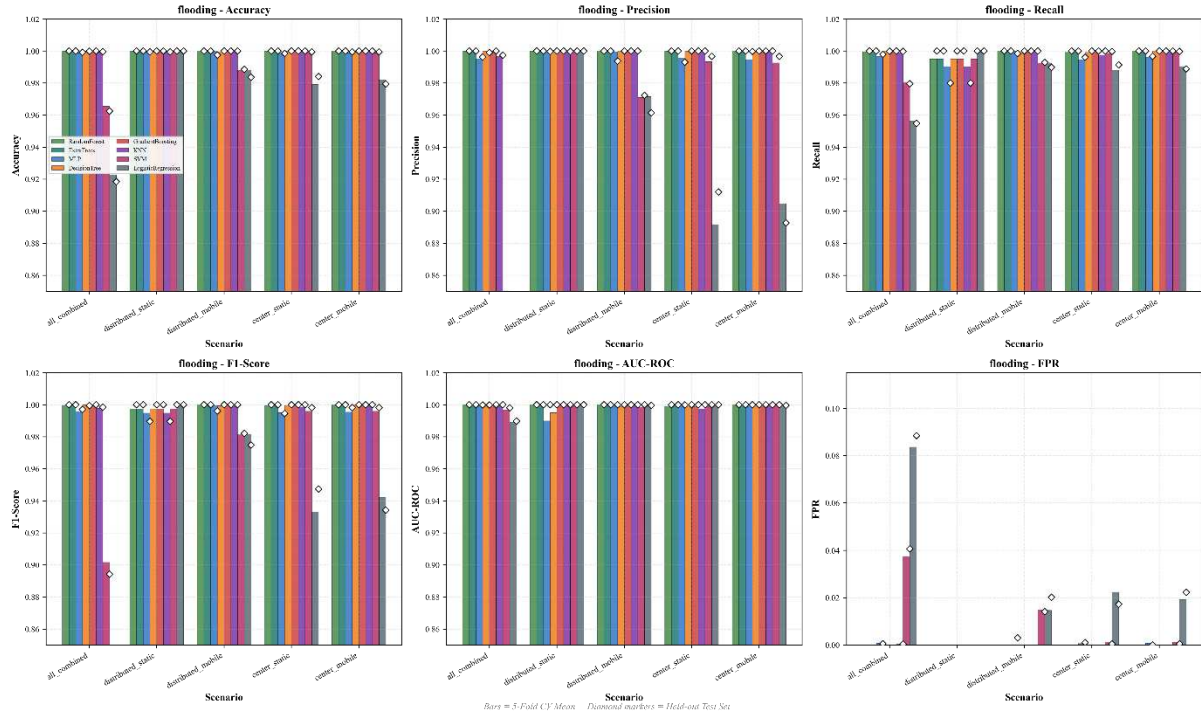


Figure 13: classifier comparison.

5.2.6 Combined Attack Detection

The combined attack scenario gives the most complete evaluation of the dataset as it includes multiple routing attacks in different network topologies and mobility conditions. Therefore, it is the most challenging benchmark to evaluate the robustness and generalisation of the machine learning algorithms under test. The increased complexity of the classification did not adversely affect the detection performance of the classifiers evaluated. Gradient Boosting showed the best overall performance with a mean F1-score of 0.9997 and consistently achieved near perfect classifications in all deployment scenarios. Extra Trees and Random Forest followed closely behind with mean F1-scores above 0.998, while Decision Tree also remained highly competitive. The consistently better performance of the ensemble-based methods demonstrates their ability to model the heterogeneous routing behaviours introduced by multiple concurrent attacks. MLP and KNN achieved high detection performance, while SVM and Logistic Regression performed relatively poorer, indicating that linear and kernel-based decision boundaries are less appropriate for the complex feature distributions caused by mixed attack scenarios.

All deployment scenarios yielded consistently high classification performance, although differences between the centralised and distributed environments were noticeable. The best overall performance was achieved in the centralised static and the centralised mobile scenario

where most of the classifiers achieved near-perfect detection. These results demonstrate that centralised monitoring retains richer and more discriminative routing information in the presence of multiple routing attacks, thus enabling more accurate classification. Although the distributed scenarios still proved to be highly effective, the reduced global visibility led to a more balanced distribution of discriminative information over multiple features, thus making the classification task relatively more difficult for conventional models based on machine learning.

Scenario	Classifier	Accuracy	Precision	Recall	F1-Score	AUC-ROC	FPR
all combined	RandomForest	0.9992	0.9994	0.9983	0.9988	0.9999	0.003138
	ExtraTrees	0.99918	0.9994	0.99816	0.9987	0.9998	0.003634
	MLP	0.9944	0.9928	0.99079	0.9918	0.9991	0.015692
	DecisionTree	0.9981	0.9971	0.99722	0.9971	0.9972	0.004294
	GradientBoosting	0.9997	0.9998	0.99942	0.9996	0.9997	0.001156
	KNN	0.9966	0.9956	0.99465	0.9951	0.9982	0.008919
	SVM	0.98259	0.9704	0.9786	0.9744	0.9949	0.0282
	LogisticRegression	0.9347	0.8929	0.92429	0.9071	0.9815	0.09399
Distributed With Mobility	RandomForest	0.9983	0.9984	0.9968	0.9976	0.99998	0.005738
	ExtraTrees	0.9986	0.9992	0.9971	0.9981	0.99998	0.005738
	MLP	0.9891	0.9861	0.9831	0.9845	0.99859	0.02787
	DecisionTree	0.9981	0.9971	0.99763	0.997	0.997633	0.003282
	GradientBoosting	0.99981	0.9998	0.99959	0.9997	1	0.00082
	KNN	0.9953	0.9924	0.9943	0.9934	0.99886	0.007377
	SVM	0.9628	0.9373	0.9614	0.9486	0.99212	0.041011
	LogisticRegression	0.8496	0.7912	0.8656	0.8132	0.96398	0.10498
Distributed Without Mobility	RandomForest	0.9985	0.9988	0.9964	0.9976	0.99997	0.006864
	ExtraTrees	0.9985	0.998	0.99648	0.9976	0.99997	0.006864
	MLP	0.9920	0.9872	0.9866	0.986	0.9974	0.0221
	DecisionTree	0.99717	0.9947	0.9959	0.99532	0.9959	0.006098
	GradientBoosting	0.99985	0.999	0.999618	0.9997	0.9992	0.000763
	KNN	0.99548	0.9911	0.99398	0.9925	0.9988	0.008388
	SVM	0.93462	0.8739	0.94633	0.90269	0.98754	0.03508
	LogisticRegression	0.87546	0.7975	0.90147	0.8285	0.9681	0.057214
Center With Mobility	RandomForest	1	1	1	1	1	0
	ExtraTrees	1	1	1	1	1	0
	MLP	0.99782	0.9981	0.99552	0.9968	0.9999	0.008597
	DecisionTree	0.9998	0.9999	0.99969	0.9998	0.9996	0.000613
	GradientBoosting	0.9997	0.9996	0.99961	0.9996	0.9999	0.000613
	KNN	1	1	1	1	1	0
	SVM	1	1	1	1	1	0
	LogisticRegression	0.99959	0.9993	0.999519	0.9994	0.99998	0.000613
Center Without Mobility	RandomForest	0.9998	0.99992	0.9997	0.9998	1	0.000528
	ExtraTrees	1	1	1	1	1	0
	MLP	0.99904	0.9992	0.9981	0.9986	0.99999	0.003694
	DecisionTree	0.9996	0.9995	0.9993	0.9994	0.99939	0.001055
	GradientBoosting	0.99976	0.9998	0.9994	0.9996	0.9994	0.001055
	KNN	1	1	1	1	1	0
	SVM	1	1	1	1	1	0
	LogisticRegression	0.99928	0.9989	0.99897	0.9989	1	0.001583

Table 9: ML Performance for Combined Attacks.

In contrast, the distributed deployment scenarios turned out to be considerably more difficult. Ensemble-based classifiers continued to show near perfect performance while conventional classifiers showed more significant performance degradation especially Logistic Regression, SVM and to a lesser extent MLP. The distributed mobile scenario was the most demanding evaluation condition as the interaction between heterogeneous attack behaviours, decentralised routing and dynamic topology changes increased the complexity. Nevertheless, Gradient Boosting was very stable in all cases and proved to be very robust even when the network environment was more variable.

The comparative analysis shows that the performance of attack detection is significantly affected by the network topology and node mobility in the presence of multiple routing attacks. Results show that centralised deployment always outperforms the distributed setting in terms of detection performance, implying that the aggregation of traffic at a single monitoring point retains higher coherent routing patterns and thus improves the feature separability. In contrast, the traffic behaviour in distributed environments has more diversity and heterogeneity, which makes it more difficult to obtain accurate decision boundaries, especially for classifiers with weak ability of modelling nonlinearities.

These challenges notwithstanding, the ensemble-based classifiers showed a remarkably stable performance across all deployment scenarios, demonstrating a strong resilience to topology and mobility variations. The comparatively larger performance degradation for Logistic Regression and SVM also shows that linear decision boundaries are not enough to model the complex interactions that are introduced by concurrent routing attacks. The overall high Accuracy, Precision, Recall, Specificity, F1-score and AUC values achieved by best ensemble classifiers validate that the proposed feature representation is effective and generalises to heterogeneous IoT environments with multiple simultaneous routing attacks.

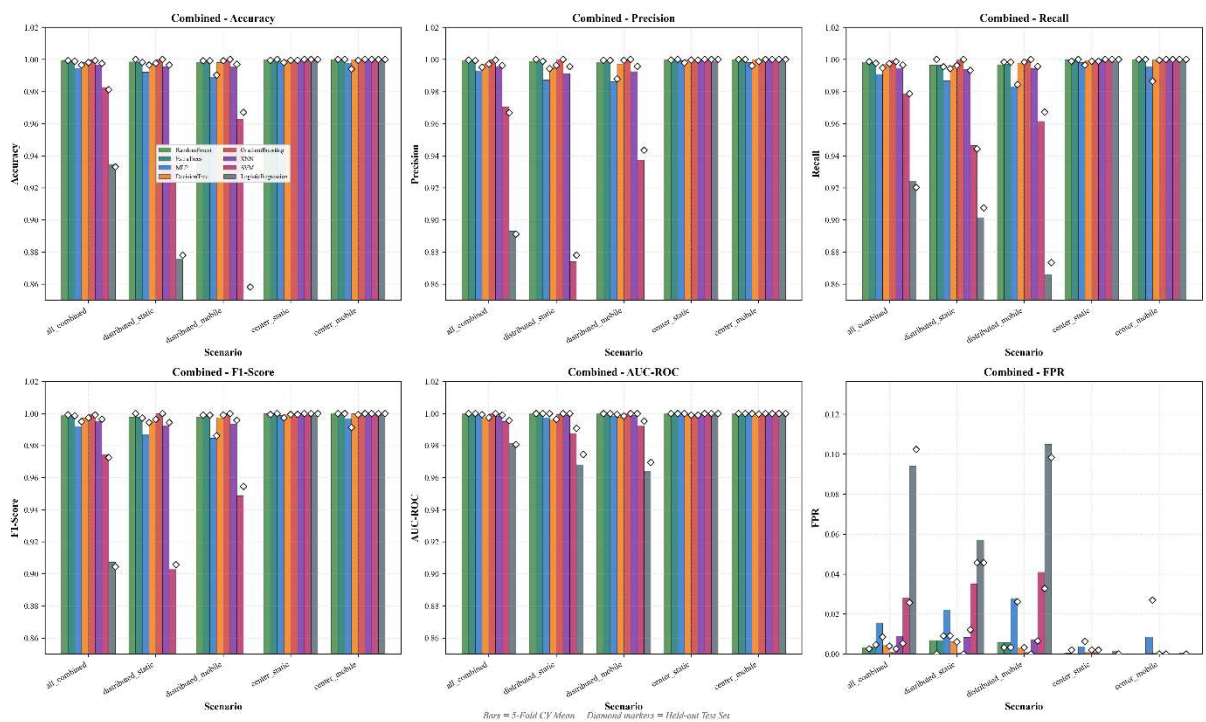


Figure 14: classifier comparison.

5.3 Comparative Analysis and Deployment Recommendations

A comparative study on all six routing attack datasets shows that ensemble learning methods are more robust under various deployment scenarios. Gradient Boosting, Extra Trees, and Random Forest consistently performed best in classification, with almost perfect Accuracy, Precision, Recall, Specificity, F1-score, and AUC for all the evaluated attacks. Their steady performance shows their resilient ability to capture the nonlinear relations between routing behavior, network topology, and traffic dynamics that characterize RPL routing attacks. The Decision Tree also showed a competitive performance, but slightly lower than that of the ensemble methods in the more heterogeneous scenarios.

K-Nearest Neighbors (KNN), one of the traditional classifiers, has surprisingly performed well with competitive detection accuracy for most attack types, despite its relatively simple

learning strategy. In contrast, MLP, SVM and Logistic Regression were more sensitive to changes of deployment conditions. In particular, MLP suffered from significant performance degradation in certain scenarios, while the linear classifiers always resulted with lower detection performance. This reveals that simple linear decision boundaries are not enough to model the complex interactions caused by routing-layer attacks.

The experimental results also demonstrate that the network topology and the node mobility have significant impacts on attack detectability. Although no single deployment scenario consistently provided the best performance across all attack types, several clear trends were identified. Static centralized deployments typically presented the most difficult operating conditions, especially for attacks whose behavioral characteristics resemble legitimate routing dynamics. In contrast, mobile and distributed deployments often improved classification performance by introducing additional routing variability, thereby increasing the separability between normal and malicious traffic.

Our observations suggest that the detectability of an attack depends not only on the attack mechanism itself, but also on the relationship between routing behavior, network topology, and node mobility. Thus, assessing intrusion detection systems in a single deployment configuration could result in overly optimistic performance estimates and overlook the diversity of real-world IoT environments.

Attack Class	Centralized Without Mobility	Centralized With Mobility	Distributed Without Mobility	Distributed With Mobility	Overall Best Performing
Decrease Rank	99.86% (Gradient Boosting)	100.00% (Random Forest)	100.00% (Random Forest)	100.00% (Random Forest)	Random Forest
Blackhole	100.00% (Extra Trees)	100.00% (Random Forest)	99.99% (Gradient Boosting)	99.98% (Gradient Boosting)	Gradient Boosting / Random Forest / Extra Trees
Selective Forwarding	100.00% (Decision Tree)	100.00% (Extra Trees)	100.00% (Random Forest)	100.00% (Random Forest)	Random Forest
Worst Parent	99.99% (Random Forest)	100.00% (Random Forest)	100.00% (Logistic Regression)	100.00% (Random Forest)	Random Forest
Flooding	100.00% (Extra Trees)	100.00% (Extra Trees)	99.99% (KNN)	100.00% (Random Forest)	Extra Trees / Random Forest
Combined	100.00% (Extra Trees)	100.00% (Random Forest)	100.00% (Random Forest)	100.00% (Random Forest)	Random Forest

Table 10: Summary of Best-Performing Classifiers by Attack .

A comparison of the evaluated routing attacks shows measurable differences in their detection difficulty. Consistent with the detection results, Decrease Rank has the best overall detection performance, indicating that malicious rank manipulation has very unique routing characteristics easily captured by the chosen feature set. Flooding, Selective Forwarding, and Worst Parent also obtained consistently high detection performance, although their sensitivity to topology and mobility was different across deployment scenarios. Conversely, the detection of the Blackhole attack was the most difficult, likely because packet dropping behavior may partially overlap with legitimate packet loss due to congestion or unstable wireless links.

Interestingly, the Combined Attack dataset did not show the worst overall detection performance even when it included multiple routing attacks at the same time. Instead, the heterogeneous character of the aggregated data set appears to provide higher behavioral diversity, allowing ensemble learning methods to build more generalized decision boundaries while still maintaining competitive classification performance in diverse deployment conditions.

The overall results show that the proposed topology and mobility-awareness-based evaluation framework provide an in-depth assessment of machine learning-based intrusion detection systems for low-power IoT networks. Ensemble learning methods were more robust,

accurate, and generalizable across all routing attacks than conventional linear models, which were more sensitive to deployment-specific variations. These results underline the importance of testing intrusion detection models on heterogeneous network topologies and mobility conditions, thus providing a more realistic assessment of their applicability to real-world RPL-based IoT environments.

5.4 Cross-Topology and Cross-Condition Generalization

To further evaluate the generalization capability of the proposed intrusion detection framework, an additional set of experiments was conducted using a cross-topology and cross-condition evaluation strategy. Unlike the previous experiments, where training and testing were performed within the same deployment scenario, the classifiers in this experiment were trained exclusively using the All Combined dataset and subsequently evaluated on each individual deployment scenario, namely Centralized Static, Centralized Mobile, Distributed Static, and Distributed Mobile.

This evaluation protocol represents a significantly more realistic deployment setting because the classifier is required to recognize attack patterns under network conditions that were not used as independent training environments. Consequently, the obtained results provide a direct assessment of the model's ability to generalize across different network topologies and mobility conditions.

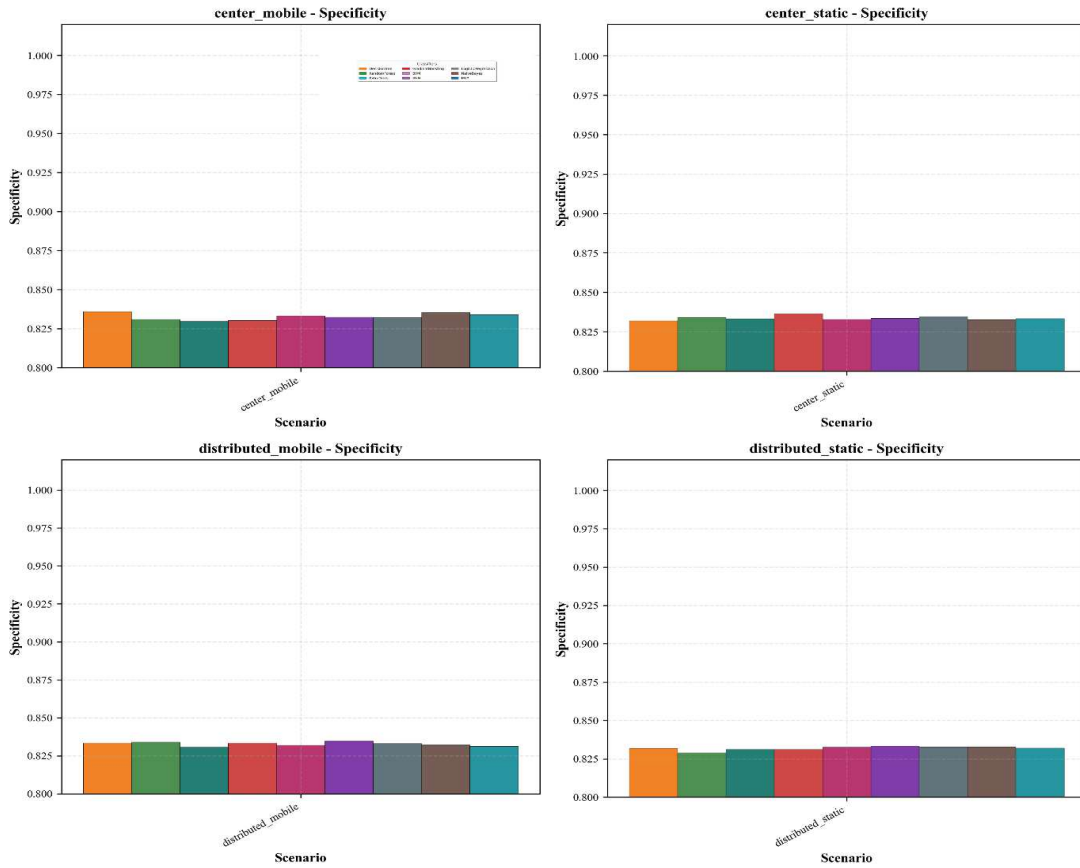


Figure 15: Specificity across deployment scenario.

In general, the results shown in Figure 15 show high Specificity for all the deployment scenarios, which means that the tested models can differentiate between the legitimate network traffic and malicious behavior effectively, even if they are tested under different network conditions from the ones used during training. The small range of Specificity across evaluated classifiers also indicates that the chosen features set is able to capture attack features that are

informative across different network topologies and mobility scenarios, reducing the impact of distribution shifts due to topology and mobility.

Although some small differences were seen in individual classifier performance, overall classification performance was very consistent across all four deployment scenarios. The best values of Specificity were achieved with tree-based ensemble methods, and the reductions in the other classifiers were negligible. These results suggest that ensemble learning is more robust when the models are evaluated under conditions that were not seen during deployment. Furthermore, the transfer of models trained on the heterogeneous has not caused significant performance degradation. All Combined dataset to individual deployment scenarios together imply that exposure to different network conditions enhances the model generalisation, while decreasing the dependence on topology-specific features.

To summarise, these results show that training under heterogeneous network conditions enables standard ML models to learn generalised representations of routing attacks that are effective on different topological structures and mobility configurations. This improved generalisation shows the applicability of the evaluated models to real-world RPL-based IoT and Low-Power and Lossy Network (LLN) deployments, in which the network topology and node mobility evolve over time dynamically.

5.5 Statistical Validation

To evaluate the dataset reliability, all machine learning classifiers were evaluated by using stratified five-fold cross-validation under the same experimental conditions. The goal of this evaluation was not to optimise a particular classification model, but to evaluate whether the dataset provides sufficiently informative and discriminative features to accurately distinguish between normal and malicious routing behaviours under heterogeneous deployment conditions.

The experimental results demonstrated a stable high level of performance with low variation across the five validation folds, which suggests that the extracted routing and traffic features are stable for different training and validation partitions. The consistency demonstrates that the dataset is not dependent on a particular data split, thereby facilitating reproducible supervised learning performance.

To further assess its representativeness, the dataset was evaluated using classifiers based on various learning paradigms, including decision tree, ensemble, distance-based, probabilistic, linear and neural network methods. Most classifiers performed well in classifying all routing attacks studied, despite the fundamentally different learning mechanisms. This consistency shows that the discriminative information mainly comes from the dataset rather than the characteristics of a specific machine learning algorithm.

The dataset was also evaluated in cross-topology and cross-condition experiments, where models trained on the all_combined dataset were evaluated on previously unseen centralised, distributed, static and mobile deployment scenarios. The consistent good prediction performance on these unseen environments suggests that the dataset captures attack characteristics that generalise across different deployment conditions, and does not simply learn topology-specific traffic patterns.

The experimental results in general show that the dataset is internally consistent, reproducible and sufficiently informative to reliably classify routing-layer attacks under different network conditions. It has a variety of attack types, deployment topologies and mobility scenarios, making it a suitable benchmark for evaluating machine learning techniques for security in RPL-based IoT and Low-Power and Lossy Network (LLN) environments.

VI. Conclusion

This work presents a comprehensive evaluation of a topology- and mobility-aware IoT intrusion detection dataset for the routing layer attacks in Low-Power and Lossy Networks (LLNs). The dataset was widely tested in different deployment scenarios, including centralised and distributed topologies with and without mobility, to determine its adequacy to build reliable machine learning-based Intrusion Detection Systems (IDSs). Experimental results show that the dataset contains enough discriminative information to perform accurate multi-class classification of the major RPL routing attacks, i.e., Decrease Rank, Blackhole, Selective Forwarding, Worst Parent, Flooding, and the combined attack scenarios.

The comparison of eight supervised machine learning algorithms demonstrated that ensemble learning methods consistently provide the best overall performance. The results of Random Forest, Extra Trees and Gradient Boosting were almost perfect in the detection of most attack categories and topology scenarios, exhibiting robustness to the changes in network architecture and node mobility. Especially, the multi-attack combined scenario, which is closer to real operational environments, has achieved excellent detection performance, validating the capability of the dataset to describe complex interactions of attacks. Conversely, classifiers such as Logistic Regression and SVM performed relatively worse in the multi-attack scenario. Thus, the choice of the algorithm is crucial when the decision boundaries become highly non-linear.

Moreover, an analysis of feature importance revealed that the most discriminative features are dependent on the deployment architecture. Centralised topologies were mostly based on the global traffic characteristics while distributed environments relied more on local radio activity, node-role information and topology-aware features. The findings emphasise the significance of incorporating contextual and topology-aware information into the design of IDSs for resource-constrained IoT networks.

In conclusion, the results show that the proposed dataset can be a good benchmark for testing machine learning-based intrusion detection methods in RPL-based IoT networks. Its wide range of attack types, network topologies and mobility conditions enables a broad and reproducible evaluation of the machine learning classifier performance under realistic deployment scenarios. The entire dataset including raw PCAP traces, processed CSV data, feature extraction scripts and simulation configurations is publicly available, which further supports transparency and reproducible research.

Future work will consider extending the benchmarking framework to other publicly available IoT intrusion detection datasets, exploring deep learning and graph-based detection models, and assessing transfer learning approaches to enhance cross-dataset generalisation and establish common evaluation practices for IoT intrusion detection research.

To ensure reproducibility, the complete dataset (raw PCAP traces, processed CSV files, feature extraction scripts, and simulation configurations) is publicly available via DOI 10.5281/zenodo.21250137

VII. Reference

- [1] Statista. Internet of Things—Number of Connected Devices Worldwide. Statista (2023). Available at: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>
- [2] Shelby, Z., Hartke, K. & Bormann, C. The Constrained Application Protocol (CoAP). RFC 7252 (2014).

- [3] Winter, T., Thubert, P., Brandt, A. et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550 (2012).
- [4] Gartner. IoT Security: Attack Vectors and Incident Response. 2023 Gartner Security & Risk Management Summit (2023).
- [5] Said, O. A comprehensive taxonomy of the attacks on wireless sensor networks. *J. Commun. Netw.* 16, 670–680 (2014).
- [6] Conti, M., Di Pietro, R., Mancini, L. V. & Mei, A. A survey of attacks on IEEE 802.15.4 ZigBee networks. *ACM Comput. Surv.* 43, 1–36 (2011).
- [7] Cardenas, A. A., Rubio-Hernan, J. E., Garcia-Alfaro, J. & Jajodia, S. Transferability and security of machine learning-based intrusion detection models. *J. Cybersecur.* 3, 49–61 (2017).
- [8] Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).
- [9] Sivanathan, M. et al. Classifying IoT devices in the wild. In *Proc. 2019 ACM SIGSAC Conference on Internet of Things Design and Implementation (IoTDI)* 1–13 (2019).
- [10] Koroniotis, N., Moustafa, N., Sitnikova, E. & Turnbull, B. Towards the development of realistic botnet dataset in the Internet of Things for smart city services. *Future Gener. Comput. Syst.* 100, 779–796 (2019).
- [11] Moustafa, N. et al. TON IoT datasets. Available at: <https://cloudstor.aarnet.edu.au/plus/s/fnrNovdkYq2PvrJ> (2020).
- [12] Vasseur, J. P. & Dunkels, A. *Interconnecting Smart Objects with IP: The Next Internet* (Morgan Kaufmann, 2010).
- [13] Yaro, S. D., Uhomobhi, J. N. N. & Sodiya, E. Packet loss and congestion prediction in wireless sensor networks. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* 6, 1–12 (2016).
- [14] Thubert, P. Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL). RFC 6552 (2012).
- [15] Levis, P. et al. The Trickle algorithm. RFC 6206 (2011).
- [16] Amsüss, C., Krüger, C. & Rietmann, S. RPL deployment challenges and opportunities. *Internet-Draft, IETF* (2021).
- [17] Karlof, C. & Wagner, D. Routing attacks on wireless sensor networks. In *Proc. 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys)* 366–372 (2006).
- [18] Dvir, A., Holczer, A. & Buttyán, L. VeRA—Version number and rank authentication in RPL. In *Proc. 8th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)* 1–8 (2012).
- [19] Ghaleb, A. et al. A survey of limitations and enhancements of the Internet Protocol Version 6 Routing Protocol for Low-Power and Lossy Networks: a focus on the objective function. *IEEE Commun. Surv. Tutor.* 21, 2946–2976 (2019).
- [20] Le, A. et al. Topology attack on RPL-based Internet of Things. In *Proc. IEEE Conference on Communications and Network Security (CNS)* 195–203 (2015).

- [21] Rahbari, H. & Khayyambashi, M. A trust-based selective forwarding attacks mitigation in wireless sensor networks. In Proc. 2013 5th International Conference on Computer Science and Information Technology (CSIT) 98–104 (2013).
- [22] Raza, S. et al. Securing communication in the Internet of Things: a survey. *J. Netw. Comput. Appl.* 67, 173–187 (2016).
- [23] Shin, B. et al. RPL specification and security assessment. Tech. Rep., NIST (2012).
- [24] Measures, S. & Heidemann, J. Performance and security of the RPL routing protocol. In Proc. 6th ACM Workshop on Real World Wireless Sensor Networks (REALWSN) 55–62 (2012).
- [25] Iova, A. et al. On the stability and optimality of qualitative routing in low-power and lossy networks. In Proc. IEEE 34th Conference on Local Computer Networks (LCN) 801–808 (2009).
- [26] Perazzo, G. et al. Low-cost denial of service attacks on the RPL routing protocol. In Proc. 6th International Conference on Communications and Electronics (ICCE) 380–385 (2016).
- [27] Rangwala, A., Kulkarni, S. S. & Sevcik, K. C. The impact of path lengths on packet-switching network dynamics. *IEEE/ACM Trans. Netw.* 17, 1118–1130 (2009).
- [28] Raychaudhuri, D. et al. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In Proc. First IEEE Workshop on Wireless Network Measurements (WinMee) 1–9 (2005).
- [29] Kamhoua, C. A. et al. Cyber-resilience assessment of interdependent infrastructure systems from excessive interdependencies. Resilience Week (RWS) (2017).
- [30] Song, L., Willke, T. L. & Kunz, A. A survey on route recomputation in dynamic ad hoc networks. In Proc. 2005 IEEE Wireless Communications and Networking Conference (WCNC) 1069–1074 (2005).
- [31] Sikri, G. & Garg, D. P. Analysis of parent switching frequency in RPL networks under mobility conditions. In Proc. IEEE 27th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) 1–7 (2016).
- [32] Hassan, A. A., Ouma, W. Y. & Wanjiru, M. A. Impact of node mobility on packet delivery in wireless sensor networks. *Int. J. Adv. Comput. Sci. Appl.* 8, 1–8 (2017).
- [33] Thulasiraman, P. et al. Characterizing sinkhole attacks in wireless ad hoc sensor networks. In Proc. 2nd International Workshop on Ubiquitous Computing and Networking (IWUCN) 1–8 (2008).
- [34] Moustafa, N. & Slay, J. The evaluation of network anomaly detection systems: statistical analysis of the unsupervised learning results. *Inf. Secur. J. Glob. Perspect.* 25, 24–36 (2016).
- [35] Cohen, W. W. Fast effective rule induction. In Proc. 12th International Conference on Machine Learning (ICML) 115–123 (1995).
- [36] Breiman, L. Random forests. *Mach. Learn.* 45, 5–32 (2001).

- [37] Liu, Y. et al. Network intrusion detection using random forests. In Proc. 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA) 831–836 (2014).
- [38] Boser, B. E., Guyon, I. M. & Vapnik, V. N. A training algorithm for optimal margin classifiers. In Proc. Fifth Annual Workshop on Computational Learning Theory 144–152 (1992).
- [39] Moustafa, N. & Slay, J. UNSW-NB15: a comprehensive data set for network intrusion detection systems. In Proc. 2015 Military Communications and Information Systems Conference (MilCIS) 1–6 (2015).
- [40] Chen, T. & Guestrin, C. XGBoost: a scalable tree boosting system. In Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) 785–794 (2016).
- [41] Le, A. et al. The impacts of internal threats on RPL-based IoT networks. In Proc. IEEE CNS (2016).
- [42] Liu, F. T., Ting, K. M. & Zhou, Z. Isolation forest. In Proc. 2008 Eighth IEEE International Conference on Data Mining 413–422 (2008).
- [43] Breunig, M. M. et al. LOF: identifying density-based local outliers. In Proc. 2000 ACM SIGMOD International Conference on Management of Data 93–104 (2000).
- [44] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015).
- [45] Tavallaee, M. et al. A detailed analysis of the KDD CUP 99 data set. In Proc. 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA) 1–6 (2009).
- [46] Tavallaee, M., Bagheri, E., Lu, W. & Ghorbani, A. A. NSL-KDD: A data set for network-based intrusion detection systems. In *Proc. 2012 Conference on Communications and Network Security (CNS)*, 1–6 (IEEE, 2012).
- [47] Moustafa, N. & Slay, J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proc. 2015 Military Communications and Information Systems Conference, MilCIS 2015, Canberra, ACT, Australia, 10–12 November 2015 1–6 (2015).
- [48] Al-Hamadi, H., Gawanmeh, A. & Al-Qutayri, M. Irad: Anomaly detection in IoT networks using machine learning. In *Proc. 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, 135–140 (IEEE, 2020).
- [49] Dhifallah, W., Tarhouni, M., Moulahi, T. & Zidi, S. A novel realistic dataset for intrusion detection in IoT based on machine learning. In *Proceedings of the 2021 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–6 (IEEE, Dubai, United Arab Emirates, 2021).
- [50] Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *Proceedings of the ICISSP2018, Funchal, Madeira, Portugal, 22–24 January 2018*; pp. 108–116.

- [51] Emeç, M. & Özcanhan, M. H. ROUT-4-2023: RPL based routing attack dataset for IoT. IEEE DataPort (2023).
- [52] Tuz-Zahra, F., Zaman, N. & Jhanjhi, M. Cooja simulator step by step manual. DOI:10.13140/RG.2.2.33168.15360 (2020).
- [53] Perazzo, G. et al. Experimental validation of a flooding-based routing protocol for low-power and lossy networks. In Proc. IEEE 16th International Conference on Emerging Technologies & Factory Automation (ETFA) 1–8 (2011).
- [54] Le, A. et al. Specification and security assessment of the RPL routing protocol. In *Proc. 7th International Conference on Networked Systems (NETYS)*, 1–15 (Springer, 2019).