

MID OCEAN UNIVERSITY

Research Methodology in Computer Science (AI800)

RESEARCH PROPOSAL

**Machine Learning-Based Detection of Version Number
Attacks in RPL-Based Internet of Things Networks**

Student: Fatemah Waleed

Lecturer: Dr. Mohammed Anbar

Course Code: AI800

Final Research Project

2026

Abstract

The Routing Protocol for Low-Power and Lossy Networks (RPL) supports IPv6 communication among resource-constrained Internet of Things (IoT) devices, but its DODAG versioning mechanism can be exploited through a Version Number Attack (VNA). A compromised node can advertise an illegitimate higher version number, trigger unnecessary global repairs, increase control traffic and energy use, and reduce network stability. This proposal investigates a lightweight machine-learning detector using a centralized monitoring architecture at the RPL root/border router. The study will generate controlled normal and VNA traffic in Contiki-NG/Cooja, construct node-level time-window observations, and compare Logistic Regression, Decision Tree, Random Forest, and LightGBM against a protocol-aware rule-based baseline. To test whether high accuracy is driven mainly by an obvious attack signature, each machine-learning model will be evaluated both with and without DODAG version-related features. The evaluation will combine node-level detection metrics with deployment cost, communication overhead, network performance, and an unseen-network-size test in which models trained on 20- and 40-node scenarios are evaluated on 60-node scenarios. The intended contribution is a reproducible evaluation protocol that distinguishes statistical detection performance from practical deployment feasibility rather than relying on accuracy alone.

Keywords: Internet of Things, RPL, Version Number Attack, Machine Learning, Intrusion Detection, 6LoWPAN, Lightweight Security.

Chapter 1: Introduction and Research Context

1.1 Introduction

The Internet of Things connects large numbers of sensing, monitoring, and control devices, many of which operate with limited memory, processing capability, bandwidth, and battery power. These constraints make conventional security mechanisms difficult to deploy without affecting network lifetime or quality of service. For IPv6-based Low-Power and Lossy Networks, RPL organizes devices into a Destination-Oriented Directed Acyclic Graph (DODAG) and provides routing while accounting for unstable links and constrained nodes.

RPL uses a DODAG version number to coordinate global repairs. Under legitimate operation, the root increments the version number when a global topology reconstruction is required. In a Version Number Attack, a malicious or compromised non-root node advertises an illegitimate higher version number in DIO control messages. Neighboring nodes may accept the update, reset routing state, and propagate it, causing repeated repairs and unnecessary control traffic. Prior work shows that this behavior can reduce network stability and lifetime and can degrade packet delivery, delay, and energy performance [1], [2], [4].

1.2 Background

1.2.1 RPL and DODAG Operation

RPL was designed for Low-Power and Lossy Networks where links may be unreliable and nodes are constrained. A DODAG is rooted at a sink or border router. Nodes select parents according to an objective function and a rank value representing their relative position toward the root. RPL control traffic includes DIO messages for topology information, DIS messages for soliciting DIOs, and DAO messages for downward routing information. The Trickle mechanism regulates the transmission of control messages so that stable networks generate less control traffic.

1.2.2 Version Number Mechanism and Attack

The version number is a DODAG-wide value intended to distinguish one global topology version from another. A legitimate increment should originate from the DODAG root. A malicious node can exploit insufficient integrity protection of this mechanism by advertising a false increment. The resulting global repair can cause nodes to recalculate routes, reset Trickle timers, and transmit additional control packets. Repeated manipulation may therefore act as a denial-of-service mechanism against constrained devices [1], [4], [5].

1.2.3 Why Machine Learning?

Rule- and specification-based defenses can be lightweight and fast, but their decisions depend on predefined assumptions about valid behavior. Machine-learning models can learn combinations of routing and traffic characteristics that separate normal from malicious behavior. Osman et al. demonstrated strong results for a Light Gradient Boosting Machine detector for VNA traffic, while later studies explored ensemble, hybrid deep-learning, and reinforcement-learning approaches [3], [7], [8], [9]. The research challenge is to preserve this detection capability without creating an IDS whose computation, memory, or communication cost is unsuitable for constrained IoT environments.

1.3 Problem Statement

Version Number Attacks (VNAs) remain a serious internal threat in RPL-based IoT networks because a compromised node can advertise an illegitimate higher DODAG version number and trigger unnecessary global repairs [1], [4], [11]. These repairs can increase control-message activity and energy consumption and can

degrade packet delivery, delay, and network stability [2], [5], [6]. Existing defenses include distributed monitoring [1], secure or collaborative RPL mechanisms such as SRPL-RP and CDRPL [2], [4], supervised machine-learning detection such as ML-LGBM [3], and reinforcement-learning approaches such as QSec-RPL [6].

Although these approaches report promising results, their limitations make direct comparison difficult. Protocol-based methods can be lightweight but depend on predefined routing rules [1], [2], [4], while intelligent IDS approaches may introduce training, computation, memory, or model-complexity costs [3], [7]-[10]. In addition, published studies use different datasets, topologies, attack settings, and evaluation measures, and simulation remains the dominant source of evidence [3]-[6], [11]. Therefore, the specific gap addressed by this proposal is the lack of a common experimental evaluation that identifies which lightweight supervised classifier offers the most useful balance of VNA detection performance and deployment cost across defined RPL network sizes and attack intensities.

1.4 Research Goal

To design and evaluate a lightweight machine-learning-based model for detecting Version Number Attacks in RPL-based IoT networks while balancing detection effectiveness, resource efficiency, and scalability.

1.5 Research Objectives

1. To quantify the impact of Version Number Attacks on RPL network performance using packet delivery ratio, average end-to-end delay, control-message overhead, and energy-related indicators.
2. To generate and preprocess a labeled dataset of normal and VNA traffic from controlled RPL simulation scenarios with multiple network sizes and attack intensities.
3. To train and compare lightweight supervised machine-learning classifiers and evaluate them using detection and resource-cost metrics, including F1-score, false-positive rate, inference time, and model size.
4. To select the most suitable model and assess the stability of its detection and resource performance across the defined network sizes and attack intensities.

1.6 Research Questions

- RQ1: How does a Version Number Attack affect packet delivery ratio, average end-to-end delay, control-message overhead, and energy-related indicators in the defined RPL scenarios?
- RQ2: Which lightweight machine-learning classifier provides the best detection performance and resource-cost trade-off when evaluated against the same dataset and a simple rule-based baseline?
- RQ3: How well does the selected detector generalize to an unseen larger network size when trained on 20- and 40-node scenarios and tested on a 60-node scenario?

1.6.1 Research Hypothesis

H0: There is no statistically significant difference in F1-score between the evaluated machine-learning detectors under matched RPL test conditions.

H1: At least one evaluated machine-learning detector has a statistically significantly different F1-score from the comparison detector(s) under matched RPL test conditions ($\alpha = 0.05$).

1.7 Research Scope

The study is limited to RPL-based IPv6/6LoWPAN IoT networks and focuses on Version Number Attacks launched by compromised internal nodes. Experiments will be performed in a controlled Contiki-NG/Cooja

environment. Network size and attack intensity will be treated as separate experimental factors: network size will vary while the attacker proportion is held approximately constant, and attack intensity will be varied independently through the frequency/rate of forged version advertisements. Mobility is intentionally excluded so that VNA behavior and network-size generalization can be isolated without adding another major source of routing variation; it remains an external-validity limitation and future-work direction. The study compares supervised machine-learning models with a protocol-rule baseline and does not claim generalization to every RPL attack type or to physical deployments not tested here.

1.8 Research Contributions

The proposed study makes four focused contributions. First, it defines a unified evaluation protocol in which a rule-based detector and several lightweight machine-learning classifiers solve the same node-level VNA identification task under common data and metrics. Second, it introduces a version-feature ablation analysis by reporting machine-learning performance with and without DODAG version-related features, reducing the risk of treating an attack signature as evidence of broader learning. Third, it separates detection performance from deployment feasibility by reporting centralized inference cost, constrained-node overhead, and any communication required for feature construction. Fourth, it evaluates network-size generalization by training on 20- and 40-node scenarios and testing on an unseen 60-node scenario, while treating attack intensity as an independent experimental factor.

Chapter 2: Literature Review and Gap Analysis

2.1 Literature Review

Mayzaud, Badonnel, & Chrisment (2017) [1]

Mayzaud et al. proposed a distributed monitoring strategy specifically for Version Number Attacks in RPL networks. Their work is important because it treats the attack as a protocol-behavior problem and distributes monitoring rather than relying only on a centralized detector. The approach shows that VNA activity can be identified by observing version-number propagation and routing behavior. Its strength is direct alignment with the semantics of RPL and relatively lightweight monitoring. However, the approach predates recent data-driven IDS methods and does not provide the same adaptive classification capability expected from modern ML models. It also leaves room for broader validation under heterogeneous network sizes and conditions.

Almusaylim, Jhanjhi, & Alhumam (2020) - SRPL-RP [2]

SRPL-RP detects and mitigates both rank and version number attacks. The reported evaluation showed strong packet delivery and attack-detection performance while considering control messages and energy consumption. This is valuable because it evaluates security together with network operation rather than accuracy alone. The approach is nevertheless a protocol/security framework rather than a learned detector, and its conclusions are tied to the evaluated simulation configurations. For the proposed research, SRPL-RP provides a useful benchmark for the idea that a security solution must be judged by both detection and network overhead.

Osman et al. (2021) - ML-LGBM [3]

Osman et al. proposed ML-LGBM, a machine-learning detector based on Light Gradient Boosting Machine for VNA detection. The study generated attack data in a Cooja-based environment, performed preprocessing and feature selection, and reported very high classification performance, including approximately 99%+ values across major detection metrics. The paper is especially relevant because it demonstrates that a tree-based boosting model can provide strong VNA classification with comparatively modest memory requirements. A

limitation for broader interpretation is that performance depends on the generated dataset and simulation design; therefore, cross-scenario robustness and deployment cost should be independently examined.

Alsukayti & Singh (2022) - CDRPL [4]

Alsukayti and Singh introduced CDRPL, a collaborative and distributed lightweight scheme that modifies RPL behavior to verify version changes and mitigate simple and composite VN attacks. The work emphasizes scalability, fast detection, topology convergence, QoS, energy consumption, and low communication overhead. This makes it a strong non-ML reference point for lightweight defense. Its main limitation relative to the present proposal is that it relies on protocol logic rather than learning traffic patterns, so it does not directly answer whether a compact ML model can offer comparable security with adaptive classification.

Sharma, Grover, & Verma (2023) - Mobile VNA Evaluation [5]

Sharma et al. examined the impact of Version Number Attacks in mobile RPL-based IoT networks. Their experiments focused on packet delivery ratio, average end-to-end delay, and power consumption while varying attacker conditions and considering node mobility. The study highlights a key issue often missing from static experiments: IoT nodes may move, and mobility can alter both baseline routing behavior and the observed impact of an attack. The work is primarily an impact evaluation rather than a complete ML detector, but it demonstrates why robustness should be tested under changing network conditions rather than a single fixed topology.

Sharma, Grover, & Verma (2023) - QSec-RPL [6]

QSec-RPL uses Q-Learning to detect Version Number Attackers in mobile RPL-based IoT. The approach is motivated by the need to avoid heavier cryptographic overhead and to adapt to dynamic conditions. It evaluates network measures including packet delivery, end-to-end delay, and power consumption and compares the proposed strategy with existing mechanisms. Its strength is the explicit treatment of mobility and adaptive decision-making. However, reinforcement learning introduces its own training and state-design considerations, and the study does not eliminate the need to compare simpler supervised models that may be easier to train, interpret, and deploy.

Osman, He, Zhu, & Mokbal (2024) - ELG-IDS [7]

The ELG-IDS study proposed an ensemble-learning intrusion detection framework with genetic feature selection for multiple RPL attacks, including Version Number, decreased-rank, and DIS flooding attacks. A major contribution is the construction of a large RPL-ELIDS dataset with hundreds of thousands of records and the use of feature selection to reduce unnecessary inputs. Ensemble learning can improve predictive performance and stability, but stacking and optimization can increase model complexity. For a resource-constrained VNA-specific detector, this creates a useful comparison point between maximum predictive performance and lightweight deployment.

Al Sawafi, Touzene, & Hedjam (2023) - Hybrid Deep-Learning IDS [8]

The hybrid deep-learning IDS combines supervised and semi-supervised learning to identify known and unknown abnormal behavior in RPL IoT traffic. Its main advantage is the ability to address patterns beyond a single fixed signature and to exploit deep representation learning. However, deep-learning architectures generally require more computation, memory, and training effort than compact tree-based or linear models. The study therefore reinforces the need to report resource cost together with classification metrics when the target environment contains constrained devices.

Shahid et al. (2024) - Hybrid ML/DL IDS [9]

Shahid et al. developed a hybrid machine-learning and deep-learning IDS for RPL IoT networks. The work reflects a recent trend toward combining multiple intelligent techniques to improve attack detection across RPL threats. Such hybrid models can achieve strong predictive performance, but increased architectural complexity can make deployment, explainability, and resource budgeting more difficult. This proposal uses that trade-off as motivation to investigate whether a smaller model can retain competitive VNA detection while providing clearer resource measurements.

Azzaoui et al. (2024) - Lightweight Cooperative IDS [10]

Azzaoui et al. focused explicitly on a lightweight cooperative IDS for RPL-based IoT, emphasizing that detection quality and deployment strategy must both respect limited device memory and energy. The cooperative design distributes detection responsibility and is closely aligned with the operational realities of constrained networks. While the study addresses broader RPL intrusion detection rather than only VNA classification, it strengthens the argument that a proposed security model should include deployment-oriented measurements instead of reporting accuracy alone.

Boudouaia, Tournois, Ouchani, & Abuarqoub (2025) - VNA State of the Art [12]

Boudouaia et al. reviewed Version Number Attacks in RPL-based IoT networks and surveyed recent detection and mitigation directions, including machine learning, blockchain, and intrusion-detection mechanisms. The study is useful for positioning VNA as a continuing RPL security problem rather than a closed issue and for identifying the diversity of proposed defenses. As a state-of-the-art review, however, it does not provide the controlled node-level comparison of simple protocol rules, lightweight classifiers, feature ablation, and deployment cost targeted in this proposal.

Rouissat et al. (2026) - Tailored Lightweight ML IDS [13]

Rouissat et al. proposed a lightweight machine-learning IDS for five RPL-specific attacks, including Version Number Attack, and generated a new dataset from simulated RPL IoT scenarios. The study combines attack profiling with feature optimization and is directly relevant to the present work because it confirms continuing interest in lightweight, data-driven RPL detection using newly generated traffic rather than only legacy datasets. Its broader multi-attack objective differs from this proposal's narrower VNA-specific question, where the value of machine learning will be tested against a direct protocol-rule baseline and through removal of version-related features.

Bas, Kaya, Ak, & Gunduz Oguducu (2026) - Incremental RPL IDS [14]

Bas et al. investigated incremental learning for evolving RPL IoT attacks, including version-number manipulation, and compared several model families with attention to adaptation, catastrophic forgetting, and training-time efficiency. Their work highlights an important limitation of static classifiers: strong performance on known attacks does not guarantee resilience when the threat set evolves. The present proposal does not attempt incremental or zero-day learning; instead, it addresses a complementary methodological question by testing whether a simpler VNA-specific detector remains justified after comparison with a rule baseline, feature ablation, resource accounting, and unseen network-size evaluation.

2.2 Comparative Analysis of Reviewed Studies

The literature shows a progression from protocol-aware monitoring and rule-based mitigation toward supervised machine learning, reinforcement learning, ensemble learning, deep learning, and more recent

adaptive/lightweight approaches. Studies published in 2025-2026 continue to treat VNA and other RPL routing attacks as active research problems, including state-of-the-art VNA review work, newly generated RPL datasets, lightweight feature-optimized IDS designs, and incremental learning for evolving attacks [12]-[14]. Traditional approaches tend to be easier to interpret, while intelligent approaches can learn combinations of routing behavior. Direct comparison remains difficult because studies use different datasets, topologies, attack mixes, units of analysis, and resource measures. This inconsistency motivates the common evaluation protocol used in the proposed study.

A second pattern is that strong classification performance does not automatically demonstrate practical deployability. For example, boosting, ensemble, and deep-learning approaches can achieve high predictive performance, yet constrained IoT environments require attention to model size, inference time, memory, and energy. Conversely, lightweight protocol schemes may have low overhead but less adaptability to traffic variation. A useful research contribution is therefore a controlled evaluation framework in which multiple lightweight ML classifiers are trained and compared on the same data and then assessed with both security and resource metrics.

2.3 Gap Analysis

Study / Approach	Evaluation Type	Computational / Resource Cost	Network Conditions / Scope	Dataset / Evidence	Gap Relevant to Proposed Study
Mayzaud et al. (2017)	Distributed monitoring	Protocol-oriented lightweight monitoring	Single-attacker/static assumptions in original monitoring context	RPL experimental/simulation evidence	Does not provide a common ML model comparison with deployment-cost metrics
SRPL-RP (2020)	Protocol detection and mitigation	Energy and control-message performance reported	Multiple network topologies evaluated	Simulation-based evidence	Effective protocol solution, but not a supervised ML comparison
ML-LGBM (2021)	Supervised ML	Execution time and memory requirements reported	Generated RPL/VNA scenarios	Cooja-generated dataset	Strong ML baseline; broader common-scenario comparison remains useful
CDRPL (2022)	Collaborative protocol mitigation	Designed for lightweight node-local processing	Simple/composite VN attack setups	Simulation-based evidence	Protocol modification rather than data-driven classifier comparison
Mobile VNA study (2023)	Attack impact evaluation	Power consumption included	Mobile-network impact study (outside proposed static scope)	Simulation scenarios	Evaluates VNA impact rather than proposing the target lightweight supervised detector
QSec-RPL (2023)	Q-Learning	Network metrics include PDR, delay, power	Mobile IoT focus (outside proposed static scope)	Simulation-based evidence	Adaptive RL approach; does not answer which simpler supervised model offers best cost/performance trade-off
Osman et al. (2024) - ELG-IDS	Ensemble learning + feature selection	Feature reduction considered	Multi-attack setting	Large RPL IDS dataset	Broader multi-attack ensemble may be more complex than a VNA-specific lightweight model
Al Sawafi et al. (2023)	Deep/semi-supervised learning	Higher computational complexity is a deployment consideration	Broad abnormal-behavior focus	IoT/RPL traffic data	Motivates explicit resource-cost reporting for constrained deployment
Hybrid ML/DL IDS (2024)	Hybrid intelligent IDS	Architectural complexity is higher than simple baselines	Broad RPL attack coverage	Experimental dataset	Need for transparent comparison against simpler lightweight models
Lightweight Cooperative IDS (2024)	Cooperative IDS	Explicit constrained-device focus	Distributed deployment	RPL evaluation	Broader IDS scope; VNA-specific supervised trade-off remains a useful research question
Alfriehat et al. (2024) [11]	Review and taxonomy	Compares RPL attack-detection approaches	Broad RPL attack scope	Published literature	Review evidence; no common node-level VNA experiment
Boudouaia et al. (2025) [12]	VNA state-of-the-art review	Discusses recent defense directions	VNA-focused literature scope	Published literature	No controlled baseline/ML comparison or feature ablation
Rouissat et al. (2026) [13]	Lightweight ML IDS	Feature optimization and lightweight detection	Multiple RPL attacks	New simulated RPL dataset	Broader multi-attack setting; no VNA-specific rule-baseline/ablation design
Bas et al. (2026) [14]	Incremental-learning IDS	Training/adaptation efficiency considered	Evolving RPL attacks	Experimental RPL attack data	Focuses adaptation; not the proposed static VNA node-level baseline/ablation comparison

2.4 Synthesis and Identified Research Gap

Across the reviewed studies, three dimensions repeatedly determine whether a VNA defense is practically useful: detection effectiveness, computational/resource cost, and robustness across network conditions. The literature contains strong individual solutions, but the studies are difficult to compare because they use different datasets, topologies, attack settings, and metrics [3]-[11]. Protocol-based approaches show that low-overhead

protection is possible [1], [2], [4], while ML, ensemble, deep-learning, and reinforcement-learning studies show that high detection performance is possible [3], [6]-[10]. The unresolved question for this proposal is narrower: whether a comparatively simple supervised classifier can retain strong VNA detection while remaining small and fast across several static RPL network sizes and attack intensities.

The proposed study addresses this gap with a common data-generation, feature, validation, and evaluation procedure for all candidate classifiers. Detection metrics will be reported together with model size and inference time, and complete simulation runs will be separated across training and testing to reduce leakage. Network size and attack intensity will be varied explicitly. Mobility is not treated as a missing dimension in this study because it is outside the defined scope and has been investigated separately in prior mobile-RPL work [5], [6].

Chapter 3: Research Methodology

3.1 Research Design

This study uses a quantitative experimental design. The manipulated experimental factors are network size and VNA condition/intensity; the dependent measures are RPL network-performance indicators and machine-learning detection/resource metrics. This design follows the course methodology sequence of defining the research problem and objectives, selecting a research design, collecting measurable data, analyzing the data, and interpreting the findings. It is suitable because the objectives require controlled comparison rather than qualitative interpretation.

3.2 Experimental Environment and Data Collection

Primary quantitative data will be generated using Contiki-NG release 4.8 with Cooja under controlled RPL scenarios. A normal baseline will first be established without malicious behavior. VNA scenarios will then configure compromised nodes to advertise illegitimate higher DODAG version numbers. Cooja/Contiki-NG radio logs and packet traces will provide node-level routing and traffic observations, while Energest will provide software-based activity measurements for CPU, low-power mode, and radio operation. Secondary data from the reviewed literature will be used for contextual comparison only.

The data-collection plan follows the course sequence of defining objectives and required data, selecting the collection method, designing scenarios, pilot testing, standardizing the procedure, and executing the analysis plan. Network sizes of 20, 40, and 60 nodes will be used. When network size is varied, the malicious-node proportion will be held at approximately 5%; attack intensity will be varied independently through the forged-version advertisement rate. Every defined scenario will be executed with 10 independent random seeds. Final radio, topology, timing, compiler, and simulator parameters will be documented before the main experiment so that the dataset can be reproduced.

3.3 Candidate Features

The classification task is explicitly node-level: each observation represents one node during a fixed time window, and the label indicates whether that node is benign or is the malicious node performing the Version Number Attack during that window. This definition keeps the machine-learning task consistent with the rule-based baseline, which also attempts to identify the offending node rather than merely detect that the network is disturbed. Candidate features include DIO transmission rate, observed DODAG version changes, time between version changes, node-rank variation, parent changes, control-to-data packet ratio, retransmission-related measures, and local traffic rates that can be associated with the observed node. Because a DODAG version increase is close to the VNA mechanism itself, version-related features may behave as an oracle-like signal. Therefore, every machine-learning model will be evaluated twice: first with the version-related feature(s), and then after removing them. The second setting tests whether the model can identify malicious nodes from broader routing behavior rather than simply reproducing the attack signature. Feature construction will use compact periodic node monitoring reports sent to the root/border router once per analysis window; the report size, transmission count, and resulting communication/energy overhead will be measured rather than treated as free. Only features available from these reports and traffic already observable by the root/border-router monitoring process will be used. A 60-second non-overlapping analysis window will be used, and each simulation run will last 30 minutes.

3.4 Deployment Architecture and Reference Platform

The proposed detector will use a centralized architecture at the RPL root/border router rather than executing the full machine-learning model on every constrained sensor node. Node-level observations will be derived from routing/control information available to the monitoring process. The simulation will be pinned to Contiki-NG release 4.8 for the Zolertia Z1 reference mote, because the release-v4.8 source tree explicitly contains the Z1 platform implementation and includes the Energest subsystem. Zolertia Z1 provides a concrete legacy constrained-device context with approximately 8 KB RAM and 92 KB flash and an MSP430F2617-class MCU. Energest will be used to obtain software-based estimates of CPU, low-power-mode, and radio activity for the simulated node platform. Because MSP430/Z1 support is legacy and may not be maintained in later Contiki-NG releases, the exact release, compiler/toolchain, and simulation configuration will be recorded and kept fixed for reproducibility. Inference itself is centralized; therefore, border-router model size and inference time will be reported separately from node-side memory additions, Energest activity, and any extra communication needed to construct node-window features.

3.5 Data Preprocessing

- With 30 non-overlapping 60-second windows per node in each 30-minute run, the eight planned scenarios and 10 seeds yield approximately 96,000 node-window observations; about 3,000 are expected to be malicious-node observations (approximately 3.1%), confirming substantial class imbalance before model training. Remove duplicate, malformed, or incomplete node-window observations and document all exclusion rules.
- Assign labels using the controlled simulation state and the node/window definition established in Section 3.3.
- Encode categorical variables when required and scale numeric features only for algorithms that benefit from scaling.
- Assess class balance after data generation. If imbalance is material, use stratified training procedures and class weights first; any resampling will be applied only to training data and documented.
- Keep complete simulation runs grouped during train/validation/test partitioning so observations from the same run cannot appear in both training and testing, reducing data leakage.
- Apply correlation/importance analysis and a lightweight feature-selection step using training data only to reduce redundant inputs.

3.6 Machine-Learning Models

The primary comparison will use lightweight supervised classifiers such as Logistic Regression, Decision Tree, Random Forest, and LightGBM. A protocol-aware rule-based baseline will identify a node as suspicious when that non-root node originates or propagates an illegitimate DODAG version increase according to information available to the monitoring process. Both the baseline and machine-learning models therefore solve the same node-level classification task. Hyperparameters will be selected using training data only, and final test scenarios will remain unseen during model selection. Model development and analysis will use Python, pandas, scikit-learn, and LightGBM; exact software versions will be recorded when the experiment is executed. Cooja/Contiki-NG radio logs and packet captures will be used for extraction, with Wireshark used where packet-level inspection is required. For the rule baseline, the originator is defined as the non-root node whose transmitted DIO first introduces the illegitimate higher DODAG version number, not a downstream node that subsequently propagates that version.

3.7 Evaluation Metrics

Node-level classification performance will be measured using accuracy, precision, recall, F1-score, confusion matrix, and false-positive rate. Precision, recall, F1-score, and false-positive rate will be emphasized because class imbalance can make accuracy misleading. The same node-level metrics will be reported for the rule-based baseline and each machine-learning model. Machine-learning results will also be reported with and without version-related features. Deployment evaluation will separately report border-router model size and inference time, any additional node-side memory requirement, and communication/control overhead required to construct the features. Network-level effects will include packet delivery ratio, average end-to-end delay, control overhead, and energy/power indicators available from the simulation. Deployment feasibility is an acceptance criterion, not part of H1: the selected design must not require the constrained Z1-class nodes to exceed their 8 KB RAM/92 KB flash context, and any additional communication cost must be reported explicitly. Inference time will be measured on a fixed host computer rather than on the simulated Cooja border router; the planned timing host is an Intel Core i7-12700H CPU with 16 GB RAM running 64-bit Ubuntu 22.04, and the exact Python/library versions and CPU power mode will be recorded with the results. Detection latency, measured from the first malicious version advertisement to the first correct node-level alert, will also be reported.

3.8 Objective-to-Method Mapping

Research Objective	Method / Activity	Main Output / Measure
O1: Quantify VNA impact	Compare normal/VNA runs at controlled size/intensity	PDR, delay, control overhead, energy
O2: Generate dataset	Create node-window observations; 10 seeds/scenario; control leakage/imbalance	Labeled node-level dataset
O3: Train, compare, evaluate	Rule baseline + LR/DT/RF/LightGBM; with/without version features	Node-level F1, precision, recall, FPR; resource/overhead measures
O4: Select and test stability	Train on 20/40; test unseen 60; vary attack intensity independently	Generalization and robustness evidence

3.9 Data Analysis Plan

Each defined scenario will be executed with 10 independent random seeds. Complete simulation runs will remain separated between training and testing to reduce leakage, so the seed/run—not individual rows randomly mixed across sets—provides the independent experimental replication. Class imbalance will be reported explicitly; class weighting and stratified procedures will be applied only within training where appropriate. Pairwise F1-score comparisons across matched runs will use a paired t-test if the paired differences reasonably satisfy normality assumptions; otherwise the Wilcoxon signed-rank test will be used, with $\alpha = 0.05$. Effect sizes and confidence intervals will be reported where practical. For network-size generalization, models will be trained on observations from the 20- and 40-node scenarios and tested on the unseen 60-node scenarios. The rule baseline and the version-feature ablation will be evaluated on the same test conditions. Statistical evidence will address H0/H1, while deployment feasibility will be judged separately using the architecture and resource/overhead criteria defined above. The rule-based baseline will be included in the same Holm-corrected comparison family as the four machine-learning models; the primary planned comparisons are the four ML-

versus-baseline tests, each based on 10 matched seed-level F1-score pairs. Holm's step-down correction will control the family-wise error rate across these four primary comparisons.

3.10 Validity, Reliability, and Limitations

Internal validity will be supported through controlled attack injection, repeated runs, standardized preprocessing, and fixed evaluation procedures. Grouping complete simulation runs during partitioning will reduce leakage between highly similar node-window observations. Class distribution will be reported and imbalance will be handled only within training data. Reliability and reproducibility will be supported by recording scenario parameters, seeds, software versions, feature definitions, labeling rules, and model settings. The principal external-validity limitation is the use of static simulated RPL networks rather than physical or mobile deployments; therefore, conclusions will not be generalized beyond the evaluated scenarios. The inferential comparisons are based on 10 independent matched seeds per scenario; this modest sample size limits statistical power, so effect sizes and confidence intervals will be interpreted alongside p-values.

3.11 Ethical Considerations and Reproducibility

The study does not involve human participants or personal data; however, responsible research practices remain relevant. Simulation data, preprocessing decisions, model settings, and evaluation procedures will be documented transparently to support reproducibility. Results will be reported without selectively omitting unfavorable outcomes, and limitations of simulation-based evidence will be stated explicitly. If code or data cannot be publicly shared, sufficient configuration details will still be recorded to allow the experiment to be reconstructed. Any use of software or automated tools will be disclosed in accordance with university requirements.

3.12 Expected Outcomes

- A documented node-window dataset containing labeled normal and VNA observations from the defined RPL scenarios.
- Measured evidence of how VNA changes packet delivery, delay, control overhead, and energy-related behavior in the simulated networks.
- A like-for-like comparison of Logistic Regression, Decision Tree, Random Forest, and LightGBM using the same partitions and metrics.
- A selected classifier supported by F1-score, statistical testing, false-positive rate, inference time, and model-size evidence.
- Scenario-wise results showing whether the selected classifier remains stable as network size and attack intensity change.
- A clear statement of the limits of the simulation evidence and the additional work required before hardware or mobile deployment.

3.13 Gantt Chart

Activity	W1 18 Aug-24 Aug	W2 25 Aug-31 Aug	W3 01 Sep-07 Sep	W4 08 Sep-14 Sep	W5 15 Sep-21 Sep	W6 22 Sep-28 Sep	W7 29 Sep-05 Oct	W8 06 Oct-12 Oct	W9 13 Oct-19 Oct	W10 20 Oct-26 Oct	W11 27 Oct-02 Nov	W12 03 Nov-09 Nov
Literature refinement & protocol design	•	•										
Cooja environment setup & pilot	•	•	•									

Activity	W1 18 Aug-24 Aug	W2 25 Aug-31 Aug	W3 01 Sep-07 Sep	W4 08 Sep-14 Sep	W5 15 Sep-21 Sep	W6 22 Sep-28 Sep	W7 29 Sep-05 Oct	W8 06 Oct-12 Oct	W9 13 Oct-19 Oct	W10 20 Oct-26 Oct	W11 27 Oct-02 Nov	W12 03 Nov-09 Nov
Normal/ VNA scenario implementat ion			•	•	•							
Data collection & labeling				•	•	•						
Preprocessin g & feature selection						•	•					
Model training & tuning							•	•	•			
Evaluation & scalability tests									•	•		
Gap comparison & results analysis										•	•	
Writing, review & final formatting											•	•

3.14 Research Milestones

- Milestone 1 (end W2 - 31 Aug 2026): literature refinement and experimental design finalized.
- Milestone 2 (end W5 - 21 Sep 2026): Cooja environment, pilot configuration, observation window, and thresholds finalized.
- Milestone 3 (end W7 - 05 Oct 2026): normal/VNA simulation runs, labeling, and dataset collection completed.
- Milestone 4 (end W9 - 19 Oct 2026): preprocessing, feature selection, model training, and tuning completed.
- Milestone 5 (end W10 - 26 Oct 2026): statistical, resource-cost, and scenario-stability evaluation completed.
- Milestone 6 (end W12 - 09 Nov 2026): final analysis, writing, reference verification, and formatting completed.

References

- [1] A. Mayzaud, R. Badonnel, and I. Chrisment, "A Distributed Monitoring Strategy for Detecting Version Number Attacks in RPL-Based Networks," *IEEE Transactions on Network and Service Management*, 2017. doi: 10.1109/TNSM.2017.2705290.
- [2] Z. A. Almusaylim, N. Z. Jhanjhi, and A. Alhumam, "Detection and Mitigation of RPL Rank and Version Number Attacks in the Internet of Things: SRPL-RP," *Sensors*, vol. 20, no. 21, 5997, 2020. doi: 10.3390/s20215997.
- [3] M. Osman, J. He, F. M. M. Mokbal, N. Zhu, and S. Qureshi, "ML-LGBM: A Machine Learning Model Based on Light Gradient Boosting Machine for the Detection of Version Number Attacks in RPL-Based Networks," *IEEE Access*, 2021. doi: 10.1109/ACCESS.2021.3087175.
- [4] I. S. Alsukayti and A. Singh, "A Lightweight Scheme for Mitigating RPL Version Number Attacks in IoT Networks," *IEEE Access*, vol. 10, pp. 111115-111133, 2022. doi: 10.1109/ACCESS.2022.3215460.
- [5] G. Sharma, J. Grover, and A. Verma, "Performance Evaluation of Mobile RPL-Based IoT Networks Under Version Number Attack," *Computer Communications*, vol. 197, pp. 12-22, 2023. doi: 10.1016/j.comcom.2022.10.014.
- [6] G. Sharma, J. Grover, and A. Verma, "QSec-RPL: Detection of Version Number Attacks in RPL Based Mobile IoT Using Q-Learning," *Ad Hoc Networks*, vol. 142, 103118, 2023. doi: 10.1016/j.adhoc.2023.103118.
- [7] M. Osman, J. He, N. Zhu, and F. M. M. Mokbal, "An Ensemble Learning Framework for the Detection of RPL Attacks in IoT Networks Based on the Genetic Feature Selection Approach," *Ad Hoc Networks*, vol. 152, 103331, 2024. doi: 10.1016/j.adhoc.2023.103331.
- [8] Y. Al Sawafi, A. Touzene, and R. Hedjam, "Hybrid Deep Learning-Based Intrusion Detection System for RPL IoT Networks," *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, 21, 2023. doi: 10.3390/jsan12020021.
- [9] U. Shahid et al., "Hybrid Intrusion Detection System for RPL IoT Networks Using Machine Learning and Deep Learning," *IEEE Access*, vol. 12, pp. 113099-113112, 2024. doi: 10.1109/ACCESS.2024.3442529.
- [10] H. Azzaoui, A. Z. E. Boukhamla, P. Perazzo, M. Alazab, V. Ravi, et al., "A Lightweight Cooperative Intrusion Detection System for RPL-Based IoT," *Wireless Personal Communications*, vol. 134, pp. 2235-2258, 2024. doi: 10.1007/s11277-024-11009-2.
- [11] N. Alfriehat, M. Anbar, M. Aladaileh, I. H. Hasbullah, T. A. Shurbaji, S. Karuppayah, and A. Almomani, "RPL-Based Attack Detection Approaches in IoT Networks: Review and Taxonomy," *Artificial Intelligence Review*, vol. 57, Art. no. 248, 2024, doi: 10.1007/s10462-024-10907-y.
- [12] M. Boudouaia, V. Tournois, S. Ouchani, and A. Abuarqoub, "Version Number Attacks in RPL based IoT Networks: State of the Art and Future Directions," in *Proceedings of the 8th International Conference on Future Networks and Distributed Systems (ICFNDS 2024)*, pp. 276-282, Association for Computing Machinery, published 2025. doi: 10.1145/3726122.3726163.
- [13] M. Rouissat, H. S. A. Belkhira, M. Belkheir, A. Mokaddem, D. Ziani, and I. S. Alsukayti, "Tailored machine learning-based IDS for RPL-based IoT networks with feature optimization and attack profiling using a novel dataset," *Journal of Electrical Engineering*, vol. 77, no. 3, 2026.

[14] S. Bas, K. Kaya, E. Ak, and S. G. Oguducu, “Adaptive Intrusion Detection for Evolving RPL IoT Attacks Using Incremental Learning,” in 2026 IEEE 23rd Consumer Communications and Networking Conference (CCNC 2026), Institute of Electrical and Electronics Engineers, 2026. doi: 10.1109/CCNC65079.2026.11366367.

Appendix A: Proposed Experimental Scenario Matrix

Scenario	Nodes	Attackers	Condition	Purpose
S1	20	0	Normal baseline; 10 seeds	Baseline
S2	20	1 (~5%)	VNA, standard forged-version rate; 10 seeds	Small network
S3	40	0	Normal baseline; 10 seeds	Baseline
S4	40	2 (5%)	VNA, same standard rate; 10 seeds	Size effect at fixed attacker ratio
S5	60	0	Normal baseline; 10 seeds	Unseen-size baseline
S6	60	3 (5%)	VNA, same standard rate; 10 seeds	Unseen-size generalization
S7	40	2 (5%)	VNA, low forged-version rate; 10 seeds	Low attack intensity
S8	40	2 (5%)	VNA, high forged-version rate; 10 seeds	High attack intensity

Note: Final node counts, attacker ratios, radio parameters, and simulation duration will be confirmed after pilot testing and documented before the main experiment.