future internet

MDPI

Article

Multi-Attribute Physical-Layer Authentication Against Jamming and Battery-Depletion Attacks in LoRaWAN

Azita Pourghasem *, Raimund Kirner, Athanasios Tsokanos, Iosif Mporas and Alexios Mylonas *

Cybersecurity and Computing Systems Research Group, Department of Computer Science, University of Hertfordshire, Hatfield AL10 9AB, UK; r.kirner@herts.ac.uk (R.K.); i.mporas@herts.ac.uk (I.M.)
* Correspondence: a.pourghasem@herts.ac.uk (A.P.); a.mylonas@herts.ac.uk (A.M.)

Abstract

LoRaWAN is widely used for IoT environmental monitoring, but its lightweight security mechanisms leave the physical layer vulnerable to availability attacks such as jamming and battery-depletion. These risks are particularly critical in mission-critical environmental monitoring systems. This paper proposes a multi-attribute physical-layer authentication (PLA) framework that supports uplink legitimacy assessment by jointly exploiting radio, energy, and temporal attributes, specifically RSSI, altitude, battery_level, battery_drop_speed, event_step, and time_rank. Using publicly available Brno LoRaWAN traces, we construct a device-aware semi-synthetic dataset comprising 230,296 records from 1921 devices over 13.68 days, augmented with energy, spatial, and temporal attributes and injected with controlled jamming and battery-depletion anomalies. Five classifiers (Random Forest, Multi-Layer Perceptron, XGBoost, Logistic Regression, and K-Nearest Neighbours) are evaluated using accuracy, precision, recall, F1-score, and AUC-ROC. The Multi-Layer Perceptron achieves the strongest detection performance (F1-score = 0.8260, AUC-ROC = 0.8953), with Random Forest performing comparably. Deployment-oriented computational profiling shows that lightweight models such as Logistic Regression and the MLP achieve near-instantaneous prediction latency (below 2 µs per sample) with minimal CPU overhead, while tree-based models incur higher training and storage costs but remain feasible for Network Server-side deployment.

Keywords: LoRaWAN; multi-attribute physical-layer authentication (PLA); IoT security; jamming attacks; battery-depletion attacks; machine learning; anomaly detection; forest-fire detection network

1. Introduction

The proliferation of Internet of Things (IoT) devices has enabled large-scale environmental monitoring through distributed wireless sensor networks deployed across wide geographical areas. Low Power Wide Area Networks (LPWANs), and, in particular, the Long Range Wide Area Network (LoRaWAN), have become foundational technologies for such applications due to their energy efficiency, long-range communication capability, and low deployment cost [1–5]. These properties make LoRaWAN well-suited for mission-critical deployments such as forest-fire monitoring, where battery-powered sensors must operate reliably over extended periods with minimal maintenance.

Recent surveys further highlight the increasing role of machine-learning-based security mechanisms in IoT and LPWAN environments, particularly for detecting availability and intrusion-related threats that cannot be fully addressed through cryptographic means alone [1,6–10].

Academic Editors: Dimitrios Dechouniotis and Ioannis Dimolitsas

Received: 12 December 2025
Revised: 31 December 2025
Accepted: 2 January 2026
Published: 8 January 2026

https://doi.org/10.3390/fi18010038

Despite these advantages, LoRaWAN exhibits inherent security trade-offs. While the protocol provides end-to-end cryptographic protection at higher layers, its lightweight design leaves the physical layer largely unprotected, exposing the device–gateway radio link to attacks that cannot be mitigated through cryptographic mechanisms alone [11,12]. As a result, adversaries can target network availability through physical-layer attacks that bypass authentication and encryption, degrading reliable data delivery even when higher-layer security mechanisms are correctly implemented.

Figure 1 illustrates the LoRaWAN architecture and highlights the physical-layer attack surface considered in this study. Sensor nodes communicate with gateways over the device–gateway radio channel, which lies outside the scope of LoRaWAN cryptographic protection. The figure situates this exposure within a forest-fire monitoring deployment, where availability attacks—specifically jamming and battery-depletion attacks—can directly disrupt sensing coverage. The gateway operates as a transparent packet forwarder, relaying uplink and downlink frames between end devices and the Network Server (NS). Devices join the network using the Over-the-Air Activation (OTAA) procedure, during which authentication and session-key establishment are handled via the Join Server (JS). After a successful join, the Network Server (NS) manages MAC-layer operations and forwards application payloads to the Application Server (AS). The assumed attack behaviours and threat scope are formalized in Section 3.3 and further discussed in the evaluation and limitations sections.

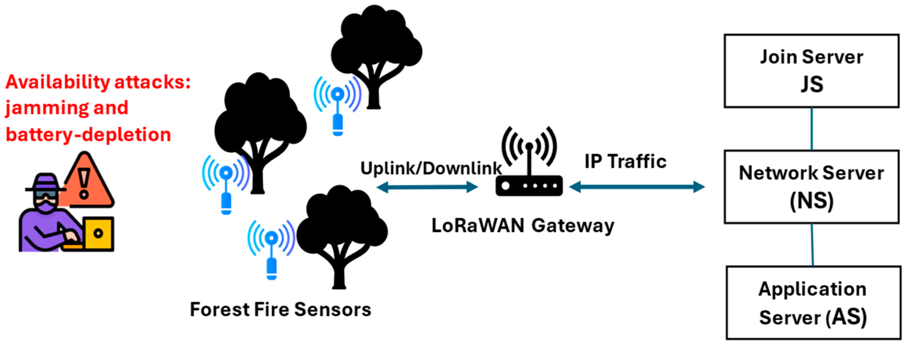


Figure 1. LoRaWAN architecture for the forest-fire use case, showing the physical-layer attack surface where jamming and radio-induced battery-depletion attacks target the device-to-gateway radio link.

This work focuses on non-3GPP LoRaWAN deployments as specified by the LoRa Alliance and evaluates physical-layer authentication (PLA) under sub-GHz unlicensed ISM band assumptions consistent with EU 868 MHz operation. Experimental evaluation follows a dataset-driven methodology based on the publicly available Brno LoRaWAN traces [5], whose characteristics are representative of real-world outdoor LoRaWAN deployments.

PLA has emerged as a promising defence mechanism by exploiting devices and environment-specific physical-layer attributes to distinguish legitimate transmissions from malicious ones [13,14]. Prior work has demonstrated the effectiveness of multi-attribute PLA for identity spoofing detection using features such as Received Signal Strength Indicator (RSSI), altitude, and battery level [15]. However, the application of multi-attribute PLA to availability-oriented attacks, namely jamming and battery-depletion, remains comparatively underexplored.

This paper addresses this gap by developing and evaluating a multi-attribute PLA framework specifically tailored to availability attack detection in LoRaWAN. Building on our prior spoofing-focused study, the threat model is expanded to include jamming and battery-depletion attacks, and the feature set is extended with battery-aware and temporal attributes. The proposed framework operates exclusively on packet-level metadata observable at the Network Server or Application Server and does not require access to raw physical-layer waveforms.

Operational Limitations of Single-Attribute and Rule-Based Detection in LoRaWAN

In practical LoRaWAN deployments, detecting availability attacks such as jamming and battery depletion using single-attribute or rule-based schemes is inherently challenging. RSSI thresholding is commonly adopted as a lightweight indicator of abnormal radio behaviour; however, RSSI is strongly influenced by benign environmental and network factors, including multipath fading, shadowing, interference, adaptive data rate (ADR) changes, gateway diversity, and physical obstructions [3–5,11,16]. As a consequence, RSSI-only detection approaches frequently exhibit high false-positive rates in large-scale outdoor deployments [17–19].

Battery-level-based detection faces similar limitations. Although accelerated battery depletion may indicate malicious activity, benign operational factors such as retransmissions, confirmed-message retries, re-join activity, and poor link quality can produce battery-consumption patterns that closely resemble attack-induced depletion [20–23]. As a result, threshold-based detection relying solely on battery-related indicators cannot reliably distinguish malicious energy-drain attacks from normal LoRaWAN operation [24–26].

Rule-based detection schemes that rely on fixed thresholds further degrade under non-stationary conditions, including seasonal channel variation, node ageing, and changes in traffic patterns [27–29]. In mission-critical applications such as forest-fire monitoring, these limitations are particularly problematic, as false positives may trigger unnecessary mitigation actions, while false negatives can result in undetected sensing failures and coverage gaps [3,5,30].

These limitations motivate a multi-attribute detection approach that jointly analyses complementary physical-layer information. By combining radio behaviour, energy dynamics, spatial context, and temporal characteristics, a learning-based PLA framework can more reliably distinguish benign operational anomalies from malicious availability attacks than single-attribute or static rule-based schemes.

The main contributions of this work are as follows:

- Availability-oriented multi-attribute PLA
  We present a systematic evaluation of multi-attribute PLA for detecting jamming and battery-depletion attacks in LoRaWAN, moving beyond prior spoofing-focused studies. Detection performance is assessed using accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC) under device-aware evaluation protocols.
- Threat model-driven dataset construction
  We construct a transparent, device-aware semi-synthetic dataset derived from publicly available Brno LoRaWAN traces, augmented with battery, spatial, and temporal attributes, and injected with controlled anomalies representing availability attacks.
- Deployment-oriented feasibility analysis
  We analyze the trade-off between detection effectiveness and computational cost, reporting prediction-time latency, Central Processing Unit (CPU) usage under single-threaded execution, and model size to support practical Network Server deployment decisions.

The remainder of this paper is organized as follows. Section 2 reviews related work on LoRaWAN security and PLA-based detection approaches. Section 3 presents the methodology. Section 4 reports experimental results and key findings, and Section 5 concludes with limitations and directions for future work.

2. Related Work

Research on LoRaWAN security has increasingly focused on the physical layer, where low-cost and protocol-aware attacks can compromise network availability and reliability. This section reviews prior work relevant to availability attack detection and multi-attribute physical-layer authentication (PLA). Section 2.1 surveys jamming attacks and representative detection approaches. Section 2.2 reviews battery-depletion attacks and their detection challenges. Section 2.3 summarizes machine-learning-based detection in LPWANs, with emphasis on feature design and deployment feasibility. Section 2.4 synthesizes these findings and identifies the research gap addressed by this work.

2.1. Jamming Attacks in LoRaWAN

Although LoRa's chirp spread-spectrum (CSS) modulation provides resilience against narrowband interference, protocol-aware jamming can still disrupt LoRaWAN communications. Selective jamming that targets specific frame segments using commodity hardware can significantly degrade link reliability [17]. Analytical studies have characterized symbol- and frame-error behaviour under LoRa-on-LoRa interference, while experimental work shows that reactive jammers implemented with low-cost software-defined radios can inject energy at precise timings to cause disproportionate disruption [16,19]. Further modelling and experimental analysis indicate that targeting acknowledgement frames can induce large-scale service degradation at minimal attacker cost [31]. In addition, synchronized jamming chirps can bypass simple timing and energy threshold defences, motivating the need for more robust detection and mitigation mechanisms at the gateway or network level [18].

Recent detection approaches increasingly rely on gateway and network-side analytics rather than fixed signal thresholds. Traffic and metadata consistency checks have been proposed to identify anomalous interference patterns indicative of jamming [27]. Lightweight detection mechanisms targeting tone and band jammers have also been developed [28]. More recently, machine-learning classifiers have been applied to distinguish deliberate jamming from benign channel variation, demonstrating improved robustness under non-stationary conditions [32].

2.2. Battery-Depletion Attacks

Battery-depletion (sleep-deprivation) attacks have long been recognized in wireless sensor networks and remain relevant in LPWAN deployments [20,24,33,34]. Experimental studies demonstrate that malicious traffic patterns can significantly increase per-event energy consumption, leading to rapid battery exhaustion [21]. Modelling-based analyses further show that "ghost traffic" can accelerate device lifetime decay even without continuous interference [23]. Battery depletion can also arise unintentionally through protocol misuse or misconfiguration, producing energy-consumption patterns that resemble deliberate attacks [27].

LoRaWAN exposes multiple vectors through which battery depletion can be induced. Abuse of confirmed messages forces devices to remain active while awaiting acknowledgements. Re-join storms increase signalling overhead and repeated radio wakeups. Exploitation of MAC commands can trigger frequent LinkCheck or Adaptive Data Rate (ADR) adjustments, while unnecessary downlinks compel devices to power on their receivers during listening windows. These mechanisms are documented in the LoRaWAN specification and associated best-practice guidance [35–38].

Detection techniques typically monitor energy-consumption patterns and their evolution over time. Energy time-series analysis combined with thresholding and machine-learning classifiers has been proposed to identify anomalous depletion behaviour [25]. Tree-based models, particularly Random Forests, have shown strong performances for detecting power-related anomalies in IoT systems [39,40]. However, approaches relying on energy indicators alone are prone to false positives, since benign factors such as poor link conditions, retransmissions, and duty-cycle irregularities can closely resemble attack-induced depletion [26]. This motivates multi-attribute approaches that combine energy-related information with complementary physical-layer indicators.

2.3. Machine Learning for Attack Detection and Multi-Attribute PLA

Machine learning (ML) is widely explored for physical-layer threat detection in LPWANs. Supervised models such as Random Forests, support vector machines, and neural networks often perform well when trained on features derived from RSSI and protocol-level metadata (e.g., timing irregularities and retransmission-related indicators) [32,41]. Unsupervised methods such as autoencoders and clustering reduce dependence on labelled data, although false positives often increase in non-stationary environments [29]. Lightweight analytic detectors can suit constrained deployments but tend to adapt poorly to evolving channel and traffic conditions [28]. More advanced directions include federated learning, reinforcement-based approaches, and game-theoretic anti-jamming strategies that enable adaptive and decentralized defenses at the cost of additional system complexity [42–44]. Tree-based models such as Random Forest and XGBoost can also provide feature-importance diagnostics (e.g., permutation-based importance) that support operator interpretation and tuning [45].

Multi-attribute PLA has demonstrated strong performance for spoofing detection by jointly exploiting multiple physical-layer features [13–15]. However, most PLA studies emphasize identity impersonation and do not explicitly target availability-oriented attacks such as jamming and battery-depletion, whose effects can be time-evolving and may manifest indirectly through reduced communication reliability and accelerated energy consumption.

Table 1 summarizes representative lightweight detection approaches, PLA frameworks, and the proposed multi-attribute PLA approach, highlighting their features, addressed threats, and key limitations.

Table 1. Comparison of lightweight defences, prior physical-layer authentication (PLA) frameworks, and the proposed multi-attribute PLA for availability-attack detection in LoRaWAN.

Approach Category	**Detection Principle**	**Features Used**	**Threats Addressed**	**Limitations**	**Representative References**
RSSI thresholding	Fixed threshold rules	RSSI	Coarse jamming indicators	High false alarms under fading and ADR	[3,11,16,17,19]
Energy-based rules	Static limits/heuristics	battery_level	Battery depletion indicators	Confounded by retransmissions and poor links	[20–23]
Protocol-level heuristics	MAC-layer rules	Retransmissions, joins	DoS symptoms	Deployment-specific tuning required	[24,30,34]
Prior PLA (spoofing-oriented)	Identity consistency	RSSI, channel features	Spoofing	Not designed for availability attacks	[13–15]
Proposed method	Multi-attribute ML-based PLA	RSSI, altitude, battery_level, battery_drop_speed, event_step, time_rank	Jamming, battery depletion	Dataset-driven; periodic retraining required	This work

https://doi.org/10.3390/fi18010038

2.4. Research Gap

Prior experimental and analytical work has demonstrated the practicality of availability attacks in LoRaWAN, including selective and reactive jamming and energy-drain strategies [17,19,21,23,27,31]. However, most existing detection approaches either rely on single-attribute indicators, focus primarily on identity spoofing rather than availability degradation [13–15], or evaluate machine-learning models on laboratory or small-scale datasets without device-aware splitting, increasing the risk of device-level leakage and limiting generalization to real deployments [25,46].

Furthermore, deployment-relevant computational costs such as per-sample prediction latency, CPU usage, and serialized model size are rarely reported, despite being critical factors for deployment at the Network Server or other centralized LoRaWAN infrastructure performing the classification stage [26]. As a result, there is a lack of device-aware, multi-attribute physical-layer authentication (PLA) frameworks that explicitly target availability attacks in LoRaWAN and are evaluated under realistic, deployment-oriented performance constraints.

3. Methodology

This section presents the methodological framework used to evaluate the proposed multi-attribute physical-layer authentication (PLA) approach against jamming and battery-depletion attacks in LoRaWAN networks. The methodology follows a dataset-driven evaluation pipeline and comprises three main components: (i) construction and augmentation of a device-aware semi-synthetic dataset, (ii) behavioural injection and labelling of availability attacks, and (iii) machine-learning-based classification and computational profiling.

The proposed PLA framework consistently uses the following features throughout the methodology: Received Signal Strength Indicator (RSSI), altitude, battery_level, battery_drop_speed, event_step, and time_rank. RSSI captures radio propagation conditions at the gateway, altitude represents the fixed installation height of each device, battery_level denotes the remaining device energy expressed as a percentage, battery_drop_speed is defined as the first-order difference between consecutive battery_level observations for the same device, event_step is a device-local temporal index representing the sequential transmission count per device, and time_rank represents the global temporal ordering of transmissions across all devices in the dataset.

Figure 2 illustrates the end-to-end operational workflow of the proposed multi-attribute PLA framework, showing packet reception at the gateway, metadata aggregation at the Network Server (NS), feature extraction and preprocessing, model inference at the classification stage, and decision-support handling. The workflow operates exclusively on packet-level metadata observable at the Network Server (NS) or Application Server (AS) and does not require access to raw physical-layer waveforms or I/Q samples. The framework is designed as an auxiliary monitoring component that does not modify the LoRaWAN protocol or interfere with standard MAC-layer operations.

PLA inference is performed centrally at the classification stage after uplink reception using metadata already available at the NS or AS. The outputs of the classification stage are treated as risk scores rather than definitive enforcement decisions. To reduce the impact of false positives, the framework assumes conservative and progressive handling of suspicious devices. Potential mitigation actions include operator alerts, increased monitoring, or temporary limitation of non-critical operations while maintaining service continuity for benign devices.

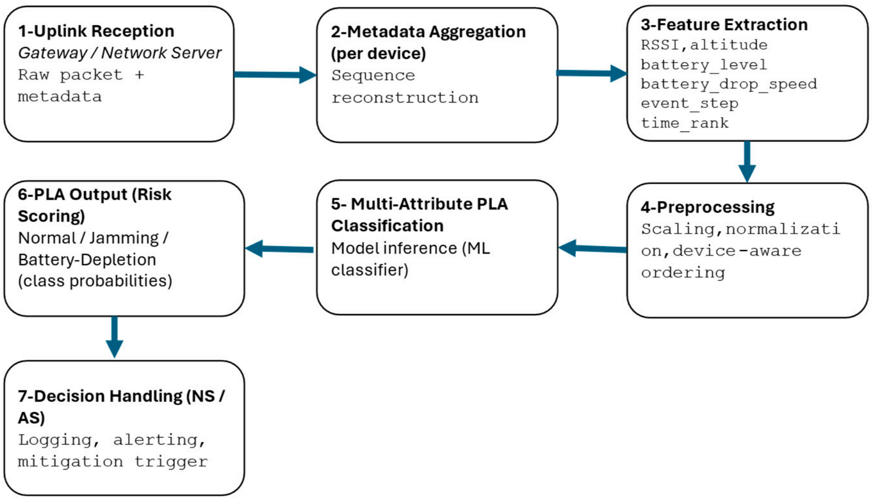

1-Uplink Reception
Gateway / Network Server
Raw packet + metadata
2-Metadata Aggregation (per device)
Sequence reconstruction
3-Feature Extraction
RSSI,altitude
battery_level
battery_drop_speed
event_step
time_rank
4-Preprocessing
Scaling,normalization,device-aware ordering
5- Multi-Attribute PLA Classification
Model inference (ML classifier)
6-PLA Output (Risk Scoring)
Normal / Jamming / Battery-Depletion (class probabilities)
7-Decision Handling (NS / AS)
Logging, alerting, mitigation trigger

Figure 2. End-to-end workflow of the proposed multi-attribute PLA framework for availability-attack detection in LoRaWAN.

3.1. Dataset

This study builds on the publicly available Brno LoRaWAN dataset released on Zenodo in 2022 [5]. Unlike our earlier spoofing-focused study [15], the present work targets availability attacks and therefore extends the Brno dataset through a device-aware, semi-synthetic augmentation process designed to represent jamming and battery-depletion behaviours.

The original Brno dataset captures legitimate LoRaWAN traffic from a real deployment in the Czech Republic and includes the core attributes device_address (unique device identifier), RSSI (Received Signal Strength in dBm), SNR (signal-to-noise ratio in dB), and Spreading Factor (SF), which are standard indicators defined by the LoRa Alliance [35,36]. While the dataset provides a realistic baseline of multi-device LoRaWAN traffic, it lacks explicit energy, spatial, and temporal attributes required for evaluating availability-oriented PLA.

Accordingly, the dataset was expanded into a semi-synthetic PLA dataset that preserves empirical radio distributions while incorporating additional attributes that reflect temporal evolution, energy consumption, and attack behaviour. Controlled attack labels were injected in a device-aware manner to ensure consistency with long-term node behaviour. A formal integrity audit of the released dataset, including attribute consistency, label distribution, and temporal validation, is provided in Appendix B.

3.2. Dataset Augmentation

Building on the dataset described in Section 3.1, we augment the records with synthetic attributes designed to reflect physical-layer behaviour observable in real LoRaWAN deployments. Augmentation is performed at the individual-device level, preserving the original device structure and transmission order. The following attributes are added: *battery_level*, *battery_drop_speed*, *event_step*, *time_rank*, and *altitude*.

Battery_level

Each device is assigned an initial battery_level, with most devices randomly initialized between approximately 90–98% (≈65% of devices), while a minority are seeded below this range to simulate partially depleted nodes. Battery level decreases incrementally over successive transmissions according to

$$B_t = B_0 - \sum_{i=1}^{t} \delta_i$$

where B_0 is the initial battery level, δ_i is the energy consumed during transmission i, and B_t is the remaining battery level at time step t. Under normal conditions, δ_i remains small and consistent. In attack scenarios, δ_i is increased to mimic accelerated drain, producing a clear contrast between gradual benign depletion and sudden forced drops $12, 16, 34$.

Battery_drop_speed

Short-term battery dynamics are captured using *battery_drop_speed*, defined as the first-order difference between consecutive battery measurements for the same device:

$$\Delta B_t = B_t - B_{t-1}$$

where B_t and B_{t-1} denote the current and previous battery levels, respectively, for consecutive *event_step* values of the same device. Small values of ΔB_t indicate gradual depletion consistent with benign operation, while large negative values indicate abrupt energy loss consistent with accelerated drain or forced activity. Positive values, which are rare, typically arise from measurement noise or device resets.

Altitude

Each device is assigned a fixed installation altitude in the range 2.0–10.0 m, representing near-surface sensors and low-mast deployments commonly used to improve gateway visibility in LoRaWAN networks $5–8, 20$. To model realistic GPS variability, altitude measurements are perturbed using

$$A' = A + \epsilon$$

where A is the nominal installation altitude and ε is zero-mean Gaussian noise corresponding to approximately 0.2–0.5% per device (maximum $\lesssim 1.6\%$). This preserves a stable per-device altitude baseline while making implausible shifts observable for anomaly detection.

Event_step

The attribute event_step represents a device-local temporal index, defined as the sequential transmission count for each individual device $(1, 2, 3, \ldots)$. This index preserves the temporal evolution of each node while avoiding reliance on absolute timestamps, thereby reducing the risk of time-based leakage related to installation dates, duty-cycle irregularities, or diurnal patterns.

Time_rank

To capture the network-wide temporal structure, *time_rank* is defined as the global temporal ordering of all transmissions across all devices. Unlike *event_step*, which is device-local, *time_rank* reflects the relative ordering of events across the entire network. This attribute enables the model to capture bursty or coordinated behaviours indicative of availability attacks, such as synchronized jamming, without relying on absolute time values.

RSSI Normalization

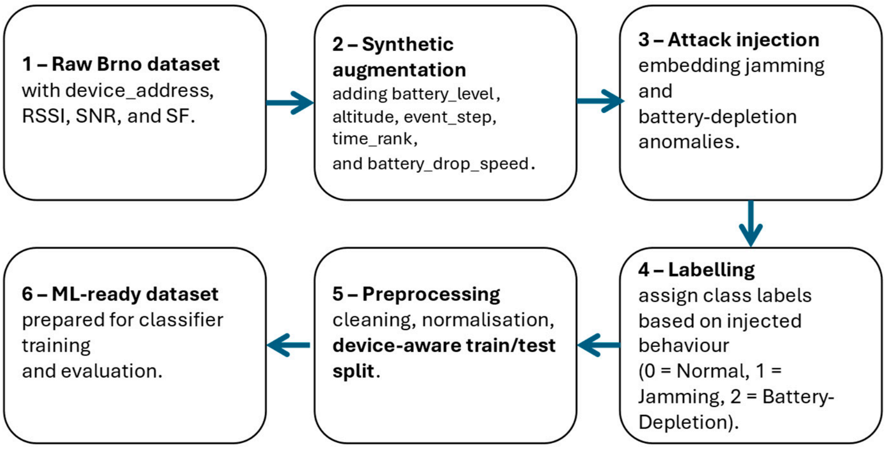
RSSI values are normalized during model training using Z-score standardization:

$$\text{RSSI}' = \frac{\text{RSSI} - \mu}{\sigma}$$

where μ and σ denote the mean and standard deviation of RSSI computed from the training data [47]. To avoid data leakage, the scaler is fitted on the training set only and then applied to validation and test sets [48]. Where time-ordered validation is required, time-aware splitting (e.g., TimeSeriesSplit) can be used to avoid training on future samples [49].

Figure 3 illustrates the complete dataset preparation pipeline, showing the progression from raw Brno data through augmentation, attack injection, labelling, preprocessing, and the final ML-ready dataset.

https://doi.org/10.3390/fi18010038



1 – Raw Brno dataset
with device_address, RSSI, SNR, and SF.
2 – Synthetic augmentation
adding battery_level, altitude, event_step, time_rank, and battery_drop_speed.
3 – Attack injection
embedding jamming and battery-depletion anomalies.
4 – Labelling
assign class labels based on injected behaviour (0 = Normal, 1 = Jamming, 2 = Battery-Depletion).
5 – Preprocessing
cleaning, normalisation, device-aware train/test split.
6 – ML-ready dataset
prepared for classifier training and evaluation.

Figure 3. Dataset preparation pipeline.

3.3. Attack Injection and Labelling

Attack labels are inferred from observable physical-layer behaviour rather than waveform-level interference. Three mutually exclusive classes are defined: Label 0 (Normal) represents benign traffic with RSSI values within the empirical Brno range (approximately -120 to -40 dBm) and stable energy behaviour. Label 1 (Jamming) represents traffic exhibiting implausible RSSI extremes (<-125 dBm or >-30 dBm) while maintaining otherwise plausible secondary attributes, consistent with PHY-layer interference [17–19,27]. Label 2 (Battery-depletion) represents traffic exhibiting abrupt negative battery_drop_speed while RSSI and SF remain stable, consistent with energy-abuse or ghost-traffic patterns [20,21,23,26].

Attack injection is behavioural and dataset-driven, designed to emulate the observable consequences of availability attacks at the Network Server rather than to reproduce waveform-level interference. This design choice reflects the constraints of the Brno dataset, which provides packet-level metadata rather than raw I/Q samples [5]. All injection thresholds and parameter ranges are explicitly reported to ensure reproducibility and allow recalibration for alternative environments.

3.4. Models and Evaluation Protocol

Detection performance is evaluated using the macro-averaged F1-score (F1), which balances precision and recall across classes, and the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC), which measures threshold-independent class separability. The ROC curve plots the true positive rate against the False Positive Rate across decision thresholds.

Five classifiers are evaluated to cover interpretable baselines and non-linear learners suitable for mixed-scale tabular data. These include Random Forest (RF), Multi-Layer Perceptron (MLP), XGBoost, Logistic Regression (LR), and K-Nearest Neighbours (KNNs). Implementations use scikit-learn (RF, LR, and KNN), TensorFlow/Keras (MLP), and the XGBoost library.

All experiments employ device-aware, stratified train/test splits to ensure that no device appears in both sets and that class proportions are preserved. Standardization and scaling are performed within the training pipelines to prevent leakage. Performance is assessed using macro-averaged accuracy, precision, recall, F1-score, and AUC-ROC, complemented by confusion matrices and ROC curves. All preprocessing statistics are fit exclusively on training folds and applied to validation and test data using the fitted transformers to prevent information leakage.

3.5. Compute Profiling

To assess deployment feasibility, we profile the computational cost of each classifier under controlled single-threaded conditions representative of Network Server execution. We report training time, per-sample prediction latency, Central Processing Unit (CPU) usage, throughput, and serialized model size. All models are evaluated using a consistent execution environment to ensure fair comparison. No parallelisation is used during inference; Open Multi-Processing (OpenMP) and Basic Linear Algebra Subprograms (BLASs) acceleration are disabled to ensure consistent and reproducible single-threaded measurements [50,51].Detailed profiling definitions and measurement procedures are provided in Appendix A.

4. Evaluation and Findings

This section presents the empirical evaluation of the proposed multi-attribute PLA framework. We first analyze the augmented dataset to characterize class balance and feature behaviour (Section 4.1), then assess classifier performance and error patterns (Section 4.2), and finally evaluate computational feasibility through single-threaded profiling (Section 4.3). Together, these analyses quantify both detection effectiveness and operational cost under realistic LoRaWAN conditions.

4.1. Feature Distribution Analysis

The dataset class balance is illustrated in Figure 4. Normal traffic represents 208,255 of 230,296 samples (90.4%), while jamming and battery-depletion samples account for the remainder (Figure 4; percentages rounded for visual clarity). This imbalance reflects operational LoRaWAN deployments, where benign traffic dominates and attacks are comparatively rare but potentially high-impact.

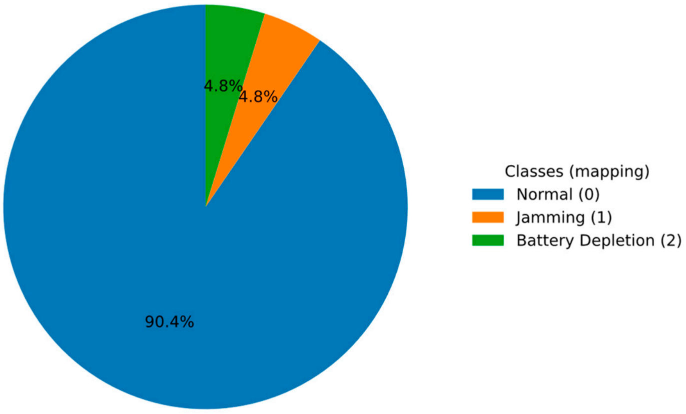


Figure 4. Distribution of normal and attack traffic in the dataset.

This distribution aligns with observations from operational LoRaWAN deployments, where the majority of traffic is benign.

Battery level behaviour exhibits distinct class-specific patterns (Figure 5). Normal transmissions maintain high battery percentages (>80%) with gradual depletion, whereas battery-depletion attacks cluster sharply below 20%, replicating abrupt drain events. Jamming samples largely overlap with the normal battery range, as interference affects communication reliability rather than energy consumption.

Temporal inspection further confirms this behaviour (Figure 6). Battery level for a representative device declines smoothly during normal operation, while battery-depletion attacks manifest as abrupt vertical drops, indicating accelerated consumption. Battery level is sampled only at uplinks. Therefore, energy usage between transmissions appears as step changes at subsequent samples.

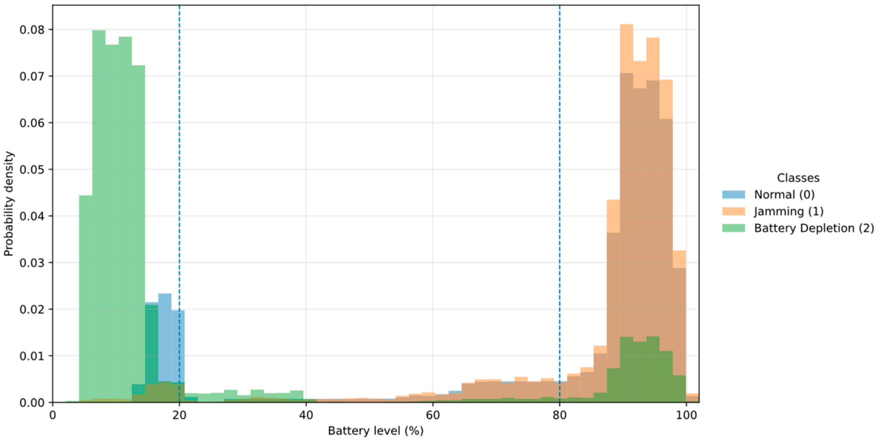


Figure 5. Distribution of battery level (%) across normal and attack classes.

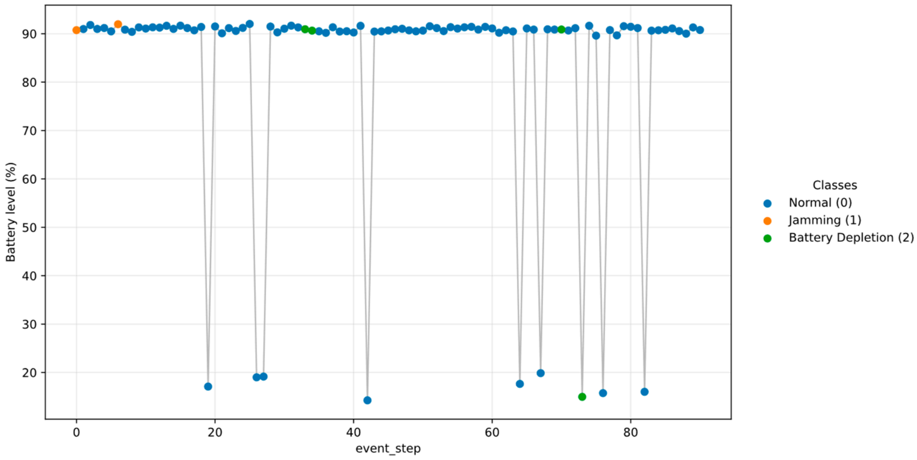


Figure 6. Battery-level progression over time for a representative device.

RSSI analysis (Figure 7) shows that normal traffic lies predominantly within the expected LoRaWAN operational range (−120 to −40 dBm). Jamming injections introduce extreme RSSI outliers (<−125 dBm), whereas battery-depletion samples retain near-normal RSSI distributions. This overlap demonstrates that RSSI alone cannot reliably distinguish all availability attacks and motivates the use of multi-attribute PLA.

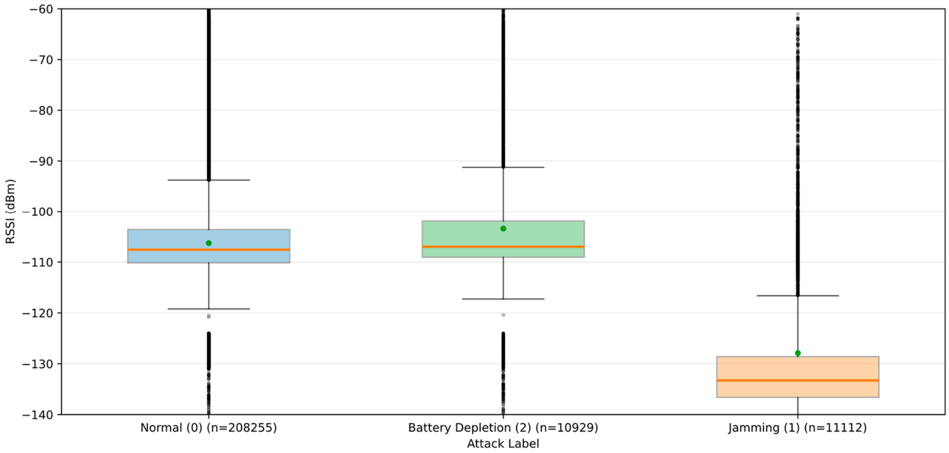


Figure 7. RSSI distribution by class (boxplot).

Altitude remains stable within the expected installation range (2–10 m) across all classes (Figure 8), confirming its role as a baseline physical attribute rather than a discriminative feature for availability attacks.

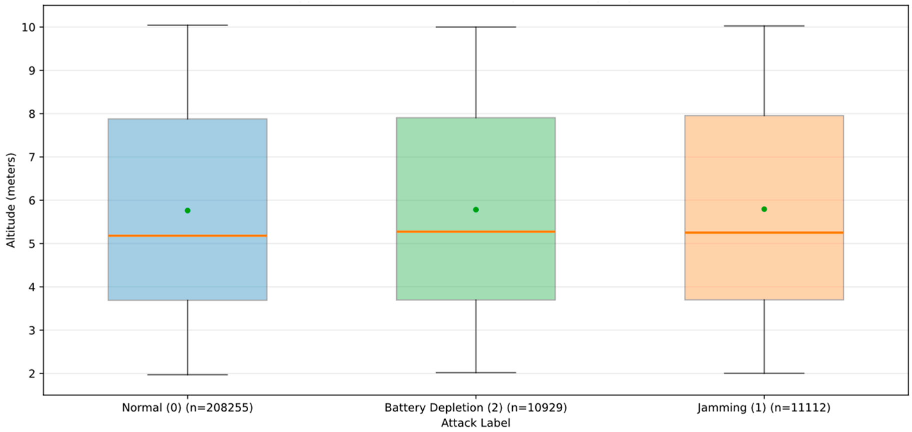


Figure 8. Altitude distribution by class (boxplot).

4.2. Classification Performance

False Positive Rate (FPR) denotes the proportion of benign transmissions incorrectly classified as attacks and is particularly critical in LoRaWAN deployments, where false alarms may trigger unnecessary mitigation actions or operational overhead. For this reason, the evaluation emphasizes both detection accuracy and error characteristics. Having established feature behavior and class separability, we next evaluate model performance using five classifiers. Table 2 summarizes macro-averaged accuracy, precision, recall, F1-score, and Area Under the Curve (AUC), with macro-averaging used to ensure equal weighting of minority attack classes under class imbalance. Bold values indicate the best-performing result for each evaluation metric across all classifiers.

Table 2. Classification performance metrics for all models (macro-averaged).

Model	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest	**0.8253**	0.8239	**0.8253**	0.8180	0.8927
MLP	0.8231	**0.8511**	0.8231	**0.8260**	**0.8953**
XGBoost	0.7709	0.7772	0.7709	0.7705	0.8939
Logistic Regression	0.8002	0.8193	0.8002	0.8033	0.7680
KNN	0.8084	0.8135	0.8084	0.7959	0.8938

The MLP achieves the highest F1-score (0.8260) and AUC (0.8953), indicating strong capture of non-linear relationships among physical-layer attributes. Random Forest follows closely with a balanced performance. Logistic Regression and KNN provide moderate results, while XGBoost exhibits a high AUC but lower F1-score, illustrating that AUC alone may mask minority-class errors.

Figure 9 visualizes the macro-F1 scores of all classifiers to emphasize comparative detection performance on the minority attack classes. Macro-F1 is reported because it assigns equal weight to each class and therefore reflects the performance on jamming and battery-depletion attacks, whereas micro-averaged (overall) F1 would be dominated by the majority normal class.

Confusion matrices (Figure 10) show that jamming (Label 1) is the most challenging class, with misclassifications primarily into the normal class. Battery-depletion (Label 2) exhibits slightly higher recall for the stronger models. For MLP, Random Forest, and XGBoost, jamming recall is approximately 0.76, while battery-depletion recall is approximately 0.81.

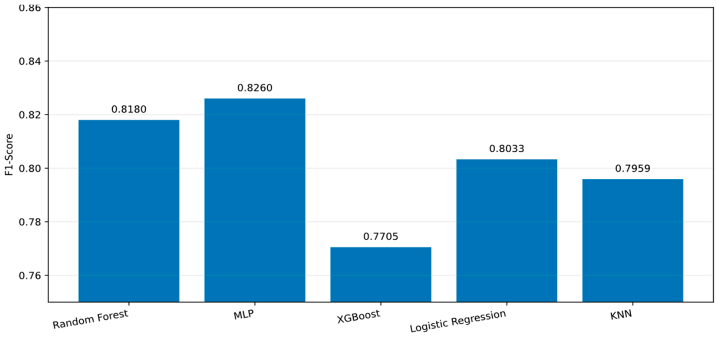


Figure 9. F1-score comparison across classifiers.

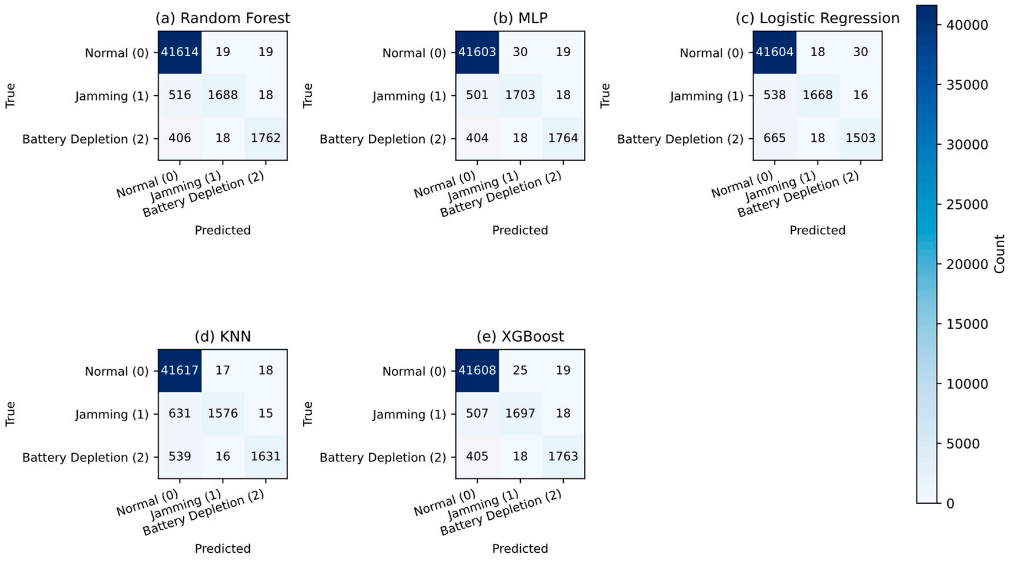


Figure 10. Confusion matrices for all classifiers.

ROC analysis (Figure 11) confirms these trends. MLP and Random Forest achieve the highest AUC values and maintain superior true positive rates in the low False Positive Rate region, which are critical for LoRaWAN deployments where false alarms are costly.

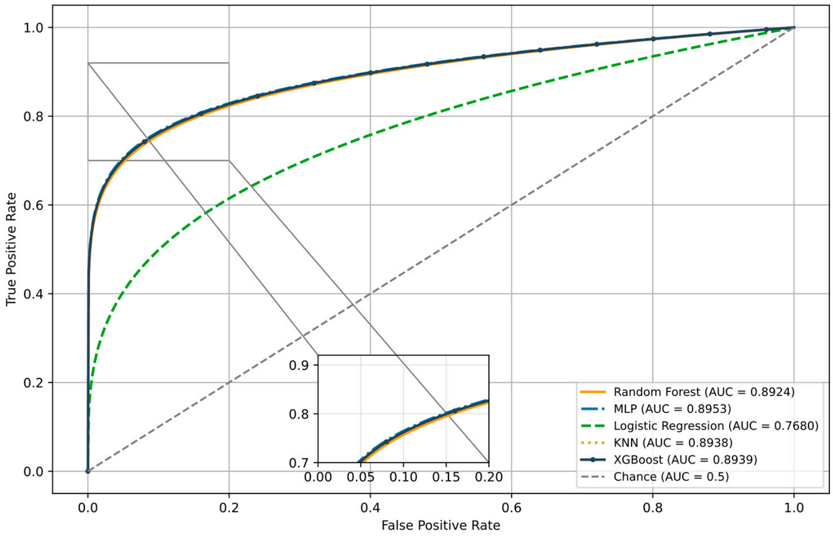


Figure 11. ROC curves for all classifiers with low-FPR inset.

4.3. Compute Profiling (Prediction Costs)

All models were profiled under the single-threaded protocol defined in Section 3.4. Table 3 summarizes training time, prediction latency, throughput, and model size.

Table 3. Computational performance of all classifiers, reporting training time, prediction latency, throughput, and model size.

Model	Training Time (s)	Prediction Latency (ms/sample)	Prediction Throughput (samples/s)	Model Size (MB)
Logistic Regression	3.342	**0.000413**	**2,421,448**	0.002
Random Forest	148.352	0.035584	28,106	112.707
KNN	**0.201**	2.164	462	30.684
MLP (scikit-learn)	95.123	0.001280	781,537	0.118
XGBoost	9.676	0.013677	73,116	1.023

Bold values indicate the most favorable result for each computational metric (i.e., lowest training time and prediction latency, highest throughput, and smallest model size).

Logistic Regression achieves the lowest prediction latency (0.000413 ms/sample), highest throughput, and smallest model size, but trails in detection quality. MLP provides the strongest detection performance with very low prediction cost at the expense of longer training time. Random Forest offers robust detection performance but incurs the highest training cost and a large storage footprint (112.7 MB), which may limit its suitability for resource-constrained Network Server deployments.Figures 12–15 visualize prediction latency, throughput, training time, and model size, respectively.

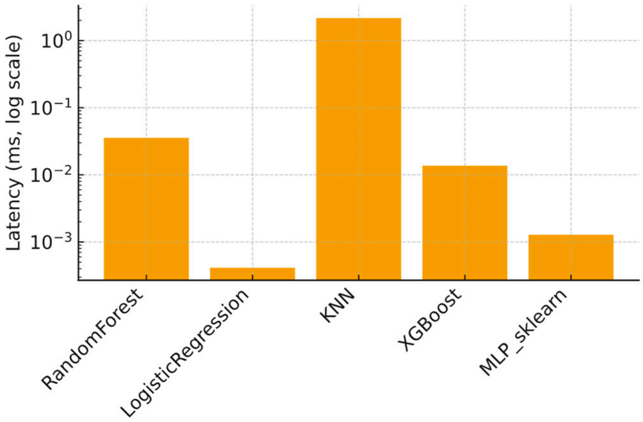


Figure 12. The inference latency (ms) of each model is shown using a logarithmic y-axis to accommodate the wide variance in execution time.

The exact numerical values underlying Figures 12–15, including inference latency, throughput, training time, and model size for all models, are reported in Appendix C.

Table 4 provides a deployment-oriented synthesis of the quantitative results reported in Table 3. Qualitative labels (e.g., very low, high, and large) are derived from the measured prediction latency, throughput, and model size to highlight practical accuracy–cost trade-offs and to support model selection for Network Server deployment. Bold entries indicate the model exhibiting the most favorable overall trade-off between detection performance and deployment efficiency.

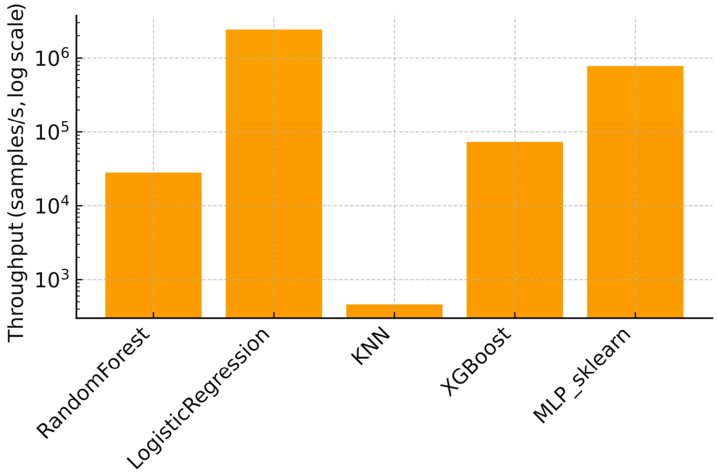


Figure 13. Throughput (samples per second) across models (log-scale).

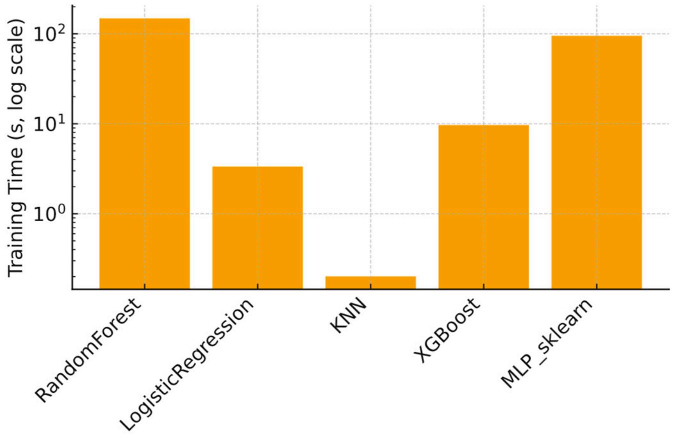


Figure 14. Training time (s) for all models (log-scale).

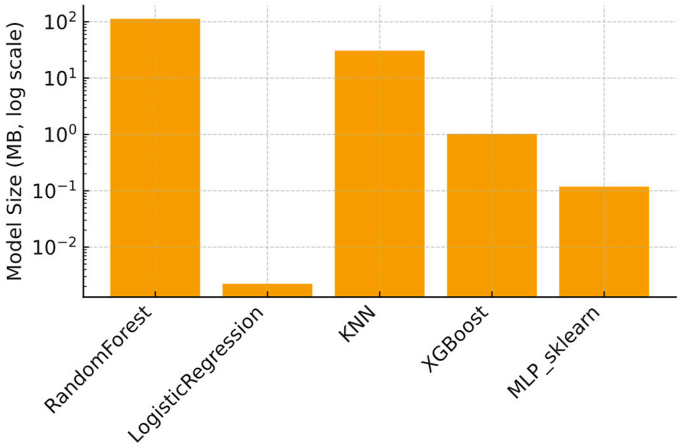


Figure 15. Model size (MB) for each classifier (log-scale).

Additional implementation details, including model hyperparameters and computational profiling parameters, are provided in Appendix A (Tables A1 and A2).

Overall, MLP provides the most favourable balance between detection effectiveness and runtime efficiency for NS deployment, while Logistic Regression serves as a lightweight baseline and Random Forest remains a strong server-side alternative when storage and training cost are acceptable.

Table 4. Deployment-oriented comparison of models.

Model	Macro-F1	AUC	Prediction Latency	Model Size	Deployment Suitability
Logistic Regression	high/ medium	high	very low	very small	Very lightweight baseline
Random Forest	high	high	low	large	High accuracy with substantial resource cost
MLP	highest	highest	very low	small	Best accuracy–cost balance
XGBoost	high	high	higher	larger	Accurate but heavier
KNN	medium	medium	high	large	Not suitable for NS use

5. Conclusions

This work evaluated multi-attribute PLA for LoRaWAN availability attacks using a device-aware, semi-synthetic dataset derived from the Brno traces. The distributional and temporal analyses (Figures 4–6) explain the error patterns observed in Section 4.2. Jamming attacks produce clear radio anomalies in the form of RSSI extremes, whereas battery-depletion attacks largely overlap with normal RSSI behaviour and are best revealed through energy-trajectory features such as battery_level and battery_drop_speed. As a result, models capable of capturing non-linear interactions between RSSI and energy-related attributes achieve the strongest detection performance. The MLP achieved the highest overall detection quality (F1 = 0.8260, AUC = 0.8953), with Random Forest performing comparably. Logistic Regression served as a compact and interpretable baseline.

From a deployment perspective, the compute profile in Section 4.3 indicates that accurate PLA is feasible at the NS. The MLP combines strong detection performance with low prediction cost (0.001280 ms/sample, model size 0.118 MB). Random Forest offers similar accuracy but incurs the largest model footprint (112.707 MB) and the longest training time. Logistic Regression delivers sub-microsecond-level prediction latency (0.000413 ms/sample) and a minimal model size (~0.002 MB), albeit at a lower AUC. These trade-offs are summarized in Table 4 and visualized in Figures 12–15.

Overall, the results confirm three key contributions of this work: (i) a systematic evaluation of multi-attribute PLA for availability attacks, reporting detection performance using accuracy, precision, recall, F1-score, and AUC (Table 2; Figures 9–11), (ii) the construction of a transparent, device-aware semi-synthetic dataset that preserves empirical LoRaWAN behaviour while enabling the controlled evaluation of jamming and battery-depletion attacks, and (iii) a deployment-oriented feasibility analysis combining detection effectiveness with prediction-time computational profiling to support practical Network Server model selection (Tables 3 and 4, Figures 12–15).

Limitations and Future Work

While the proposed multi-attribute PLA framework demonstrates strong detection performance for jamming and battery-depletion attacks under the evaluated conditions, several limitations and potential failure modes should be acknowledged.

First, the proposed approach may be less effective against slow or stealthy jamming strategies. Unlike aggressive or continuous jamming, such attacks deliberately induce only marginal degradation in signal quality over extended periods. By operating at low power, intermittently, or selectively targeting specific frames or spreading factors, stealthy jammers can remain within the natural variability of LoRaWAN channel conditions. As a result, the induced RSSI and temporal patterns may closely resemble benign fading, interference from neighbouring devices, or ADR adjustments, and therefore remain below detection thresholds used by metadata-based PLA schemes [17,19,30,42]. Detecting such attacks may require longer observation windows, cross-gateway correlation, or access to lower-level physical-layer measurements not available in the present dataset.

Second, benign battery anomalies may confound battery-aware detection. Although abrupt battery drops are indicative of forced activity or energy-drain attacks, similar patterns can arise from non-malicious operational factors, including persistent poor link quality, repeated retransmissions, confirmed-message retries, misconfigured duty cycles, or device resets. In such cases, battery-depletion behaviour may mimic attack-induced drain without adversarial intent, increasing the risk of false positives. While the proposed multi-attribute fusion mitigates this risk by jointly analyzing radio and temporal features, complete disambiguation between malicious and benign energy anomalies cannot be guaranteed in all deployment scenarios [20,21,23,26].

Third, the framework may be affected by concept drift during long-term deployment. Physical-layer characteristics such as RSSI distributions and battery-consumption patterns can evolve over time due to seasonal environmental changes, vegetation growth, hardware ageing, firmware updates, or network densification. Models trained on historical data may therefore experience gradual performance degradation if such changes are not accounted for. Periodic retraining, adaptive threshold recalibration, or online learning mechanisms may be required to maintain detection reliability over extended operational lifetimes.

Unlike our earlier spoofing-oriented PLA work, which explored online adaptation, the present study intentionally focuses on batch-trained models in order to isolate and rigorously evaluate availability-oriented attack behaviour under controlled and reproducible conditions.

Finally, the attack modelling in this study is behavioural and dataset-driven, reflecting observable effects at the gateway or Network Server level rather than waveform-level over-the-air interference. This choice is constrained by the Brno LoRaWAN dataset [5], which provides packet-level metadata but not raw I/Q samples or channel state information. While prior experimental studies show that selective and reactive jamming produces detectable metadata-level effects [16–19,27,31], future work should incorporate field trials with physical interferers to further validate realism under diverse interference conditions.

These extensions are intentionally left for future work, as the present study focuses on the controlled and reproducible evaluation of availability-oriented PLA under well-defined threat assumptions.

Addressing these limitations represents an important direction for future research. In particular, extending the framework to incorporate multi-gateway correlation, longer temporal context, and adaptive retraining strategies would further strengthen robustness against stealthy attacks and long-term environmental drift.

Author Contributions: Data curation, A.P.; methodology, A.P.; supervision, A.M., R.K., A.T. and I.M.; writing—review and editing, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The complete dataset used in this study is available in ZIP format at: Azita369/PLA-multi-attribute.

Acknowledgments: The authors would like to acknowledge the providers of the Brno LoRaWAN dataset for making their data publicly available for research purposes.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADR	Adaptive Data Rate
AS	Application Server
AUC	Area Under the Curve
BLAS	Basic Linear Algebra Subprograms
CPU	Central Processing Unit
CSV	Comma-Separated Values
F1	F1-score (harmonic mean of precision and recall)
FPR	False Positive Rate
IoT	Internet of Things
ISM	Industrial, Scientific and Medical (radio band)
JS	Join Server
KNN	K-Nearest Neighbours
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
LR	Logistic Regression
MAC	Medium Access Control
ML	Machine Learning
MLP	Multi-Layer Perceptron
NS	Network Server
OpenMP	Open Multi-Processing
OTAA	Over-the-Air Activation
PLA	Physical-Layer Authentication
RF	Random Forest
ROC	Receiver Operating Characteristic
RSSI	Received Signal Strength Indicator
SDR	Software-Defined Radio
SF	Spreading Factor
SNR	Signal-to-Noise Ratio
WSN	Wireless Sensor Network
XGBoost	eXtreme Gradient Boosting

Appendix A. Compute Profiling Metrics and Definitions

Appendix A.1. Execution Setup

All profiling experiments were conducted under single-threaded execution to isolate the intrinsic computational cost of each model. This avoids variability introduced by automatic multi-threading in numerical backends such as BLAS (Basic Linear Algebra Subprograms) and OpenMP, which may otherwise utilize multiple CPU cores. Enforcing a single thread ensures that reported timings reflect model complexity rather than hardware parallelism.

Appendix A.2. Training Cost

For each classifier, training cost is measured using the following:

Wall-clock time: Real elapsed time required to complete model training.

Process CPU time: CPU time consumed by the training process.

Resident memory change: Average increase in memory usage during training.

Training runs are repeated $R = 3$ times, and mean ± standard deviation values are reported.

https://doi.org/10.3390/fi18010038

Appendix A.3. Prediction Latency and CPU Cost

Prediction-time efficiency is evaluated under controlled, single-threaded execution conditions. We report three complementary inference metrics.

Prediction Latency

$$L_{\text{pred}}\,[\text{ms/sample}] \;=\; \frac{T_{\text{infer,wall}}}{N_{\text{test}}} \times 1000$$

where $T_{\text{infer,wall}}$ is the total wall -clock time (in seconds) required to score the full test set, and N_{test} denotes the number of test samples. The factor 1000 converts seconds to milliseconds. L_{pred} represents the average end-to-end inference time per sample, as experienced by an operator or system.

CPU Cost per Sample

$$C_{\text{pred}}\,[\text{CPU ms/sample}] \;=\; \frac{T_{\text{infer,cpu}}}{N_{\text{test}}} \times 1000$$

where $T_{\text{infer,cpu}}$ is the total process CPU time required for inference (in seconds), excluding idle or waiting time. This metric isolates the pure computational effort per sample and is particularly useful for capacity planning and deployment analysis.

Throughput

For completeness, throughput is reported as follows:

$$\Theta_{\text{pred}}\,[\text{samples/s}] \;\approx\; \frac{1000}{L_{\text{pred}}}$$

Since L_{pred} is expressed in milliseconds per sample, dividing 1000 by this value yields the corresponding inference throughput on the same host. Throughput is therefore the inverse of prediction latency.

Appendix A.4. Model Size

We also report the model size:

$$S_{\text{model}} = \text{size of the serialized model on disk.}$$

Model size reflects storage and replication overhead, as well as potential cold-start implications in resource-constrained or edge deployments.

Appendix B. Data Integrity Note

To verify internal consistency between the described methodology and the released dataset (Brno_dataset_with_battery_drop_speed_and_label_noise.csv), a descriptive audit was conducted. The results confirm that all claimed attributes are present, correctly labelled, and statistically coherent with the methodological narrative.

Verification Summary

The final dataset comprises 230,296 records following augmentation and labelling.

No missing or inconsistent entries were identified after data cleaning.

Temporal continuity was verified for all transmissions within each device_address, ensuring correct sequencing for Δ-based features.

A controlled label noise rate ($\leq$2%) was intentionally maintained to support generalization during model training.

The derived attribute battery_drop_speed was validated as the first-order difference in battery_level for each node, confirming internal consistency.

Interpretation

These checks confirm that the dataset constitutes a semi-synthetic extension of the Brno LoRaWAN dataset, integrating real-world physical-layer parameters (RSSI, SNR, and SF) with synthetically modelled altitude and energy behaviour. This design provides a transparent and reproducible foundation for evaluating multi-attribute Physical-Layer Authentication (PLA) against both spoofing and availability attacks.

Table A1. Summary statistics and verification outcomes for the semi-synthetic PLA dataset.

Attribute	Description	Min	Median	Max	Observations/Notes
device_address	Unique numeric ID for each node	1	—	8800	No duplicates per timestamp; malicious IDs preserved as −1 for spoof markers.
event_step	Transmission order per device	1	84	168	Sequential; ensures temporal consistency for Δ calculations.
rssi	Received signal strength indicator (dBm)	−137.6	−100.3	−25.7	Reflects original Brno variation; Normal samples not hard-clipped.
snr	Signal-to-noise ratio (dB)	−17.8	3.4	12.6	Matches Brno distribution; used to enrich channel context.
sf	Spreading factor	7	9	12	Integer; unchanged from original Brno dataset.
altitude	Fixed node installation height (m)	2.00	5.92	9.97	Device-wise variation ≈ 0.2–0.5%, maximum 1.6%; consistent with GPS drift model.
battery_level	Remaining energy percentage	12.7	87.3	101.4	~65% start 90–98%; small over-100% values due to noise injection.
battery_drop_speed	ΔBattery between event steps	−4.92	−0.08	0.02	Mean near zero; large negatives correspond to depletion events.
label	Attack class (0—Normal, 1—Jamming, and 2—Battery)	—	—	—	Class distribution: 90.43%/4.83%/4.75%, respectively.

Appendix C. Detailed Performance Metrics of Classification Models

This appendix provides the exact numerical values corresponding to the summary plots shown in Figures 11–14. Since the performance metrics span multiple orders of magnitude, the log-scale visualizations in the main Results section focus on comparative trends rather than exact quantities. For completeness and reproducibility, the full metrics—including inference latency, throughput, training time, and model size—are reported here.

Table A2. Performance metrics for all evaluated models.

Model	Inference Latency (ms)	Throughput (samples/s)	Training Time (s)	Model Size (MB)
Random Forest	0.035584	28,102.45	148.351617	112.706849
Logistic Regression	0.000413	2,421,447.90	3.341863	0.002229
KNN	2.164416	462.02	0.201079	30.684250
XGBoost	0.013677	73,116.13	9.675927	1.023351
MLP (sklearn) *	0.001280	781,537.04	95.122917	0.118211

* We explicitly label the model as “MLP (sklearn)” because the Multi-Layer Perceptron is implementation-dependent, and its architecture and optimization behaviour differ across frameworks. Adding the library clarifies that our results refer specifically to scikit-learn’s MLPClassifier, avoiding confusion with deep-learning MLP implementations such as PyTorch or TensorFlow.

References

1. Tariq, U.; Alsaeedi, A.; Malik, H.; Song, H. Securing the evolving IoT with deep learning: A comprehensive review. *Sensors* **2023**, *23*, 4117.
2. Mekki, K.; Bajic, E.; Chaxel, F.; Meyer, F. A comparative study of LPWAN technologies for IoT deployment. *ICT Express* **2019**, *5*, 1–7. [CrossRef]
3. Stanco, G.; Ghibaudi, M.; Hussain, R.; Di Nardis, L. A comprehensive overview of the state of the art on the security of Low Power Wide Area Networks (LPWANs), with a focus on Sigfox and LoRaWAN. *ICT Express* **2024**, *10*, 240–254. [CrossRef]
4. Centenaro, M.; Vangelista, L.; Zanella, A.; Zorzi, M. Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios. *IEEE Wirel. Commun.* **2016**, *23*, 60–67.
5. Povalac, A.; Kral, J. LoRaWAN Traffic Analysis Dataset. Zenodo, 2023. Available online: https://doi.org/10.5281/zenodo.7919213 (accessed on 15 January 2025). [CrossRef]
6. Torres, N.; Pinto, P.; Lopes, S.I. Security Vulnerabilities in LPWANs—An Attack Vector Analysis for the IoT Ecosystem. *Appl. Sci.* **2021**, *11*, 3176.
7. Adewole, K.S.; Owolabi, K.; Abdullateef, A.M. Intrusion Detection Framework for Internet of Things with Ensemble Learning and Rule Induction. *Sensors* **2025**, *25*, 1845. [CrossRef]
8. Singh, V.P.; Kumari, R.; Kaur, M. Machine Learning for Intrusion Detection System in IoT Environment with Permutation Importance. In *Proceedings of the SNSFAIT 2024, Delhi, India, 8–9 August 2024*; CEUR-WS: Aachen, Germany, 2024; Volume 3774, pp. 14–22.
9. Haque, A.; Chowdhury, M.N.-U.-R.; Soliman, H.; Hossen, M.S.; Fatima, T.; Ahmed, I. Wireless Sensor Networks anomaly detection using Machine Learning: A Survey. *arXiv* **2023**, arXiv:2303.08823. [CrossRef]
10. Scikit-Learn Developers. Common Pitfalls and Recommended Practices (Explains Data Leakage; Fit Preprocessors on Train Only). Available online: https://scikit-learn.org/stable/common_pitfalls.html (accessed on 17 October 2025).
11. Ruotsalainen, H.; Shen, G.; Zhang, J.; Fujdiak, R. LoRaWAN Physical Layer-Based Attacks and Countermeasures: A Review. *Sensors* **2022**, *22*, 3127.
12. Kuntke, F.; Bader, L.; Hoßfeld, T.; Pries, R. LoRaWAN security issues and mitigation options. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4452.
13. Stoyanova, M.; Nikoloudakis, Y.; Panagiotakis, S.; Pallis, E.; Markakis, E.K. A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1191–1221.
14. St. Germain, K.; Kragh, F. Physical-Layer Authentication Using Channel State Information and Machine Learning. *arXiv* **2020**, arXiv:2006.03695. [CrossRef]
15. Pourghasem, A.; Kirner, R.; Tsokanos, A.; Mporas, I.; Mylonas, A. Machine learning-based multi-attribute physical-layer authentication for spoofing and availability detection in LoRaWAN. *Future Internet* **2025**, *17*, 68.
16. Afisiadis, O.; Cotting, M.; Burg, A.; Balatsoukas-Stimming, A. On the Error Rate of the LoRa Modulation with Interference. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 1292–1304.
17. Aras, S.; Tavlı, B.; Yıldız, H.U. Experimental evaluation of selective jamming attacks in LoRaWAN. In Proceedings of the ACM Workshop on Wireless Security and Privacy (WiSec), Boston, MA, USA, 18 July 2017; pp. 89–95.
18. Hou, N.; Xia, X.; Zheng, Y. Jamming of LoRa PHY and countermeasure. *ACM Trans. Sens. Netw.* **2023**, *19*, 1–27.
19. Perković, T.; Rudeš, H.; Damjanović, S.; Nakić, A. Low-Cost Implementation of Reactive Jammer on LoRaWAN Network. *Electronics* **2021**, *10*, 864.
20. Shakhov, V. Depletion-of-battery attack: Specificity, modelling and analysis. *Sensors* **2018**, *18*, 1849. [CrossRef] [PubMed]
21. Mikhaylov, K.; Fujdiak, R.; Pouttu, A.; Vozňák, M.; Malina, L.; Mlynek, P. Energy Attack in LoRaWAN: Experimental Validation. In Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES 2019), Canterbury, UK, 26–29 August 2019; ACM: New York, NY, USA, 2019; pp. 1–10.
22. Singh, R.K.; Puluckul, P.P.; Berkvens, R.; Weyn, M. Energy Consumption Analysis of LPWAN Technologies and Lifetime Estimation for IoT Application. *Sensors* **2020**, *20*, 4794. [CrossRef] [PubMed]
23. Kuaban, G.S.; Gelenbe, E.; Czachórski, T.; Czekalski, P.; Tangka, J.K. Modelling of the Energy Depletion Process and Battery Depletion Attacks for Battery-Powered Internet of Things (IoT) Devices. *Sensors* **2023**, *23*, 6183.
24. Wood, A.D.; Stankovic, J.A. Denial of Service in Sensor Networks. *IEEE Comput.* **2002**, *35*, 54–62.
25. Proto, M.; Miers, A.; Carvalho, J. Intrusion detection based on energy consumption for EDAs in LoRaWAN. In Proceedings of the International Conference on Internet of Things, Big Data and Security (IoTBDS 2024), Angers, France, 25–27 April 2024.
26. He, P.; Zhang, X.; Lee, J.; Singh, R. Energy-Aware Security Mechanisms for the Internet of Things: A Survey. *Future Internet* **2024**, *16*, 128. [CrossRef]
27. Ruotsalainen, H. Reactive Jamming Detection for LoRaWAN Based on Meta-Data Differencing. In Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES 2022), Vienna, Austria, 23–26 August 2022; pp. 1–8.

28. Demeslay, C.; Gautier, R.; Fiche, A.; Burel, G. Band & Tone Jamming Analysis and Detection on LoRa signals. *arXiv* **2021**, arXiv:2107.07782. [CrossRef]
29. Sciancalepore, S.; Kusters, F.; Abdelhadi, N.K.; Oligeri, G. Jamming Detection in Low-BER Mobile Indoor Scenarios via Deep Learning. *IEEE Internet Things J.* **2024**, *11*, 14682–14697. [CrossRef]
30. Xu, W.; Ma, K.; Trappe, W.; Zhang, Y. Jamming sensor networks: Attacks and defenses. *IEEE Netw.* **2006**, *20*, 41–47. [CrossRef]
31. Dossa, A.; Amhoud, E.M. Impact of Reactive Jamming Attacks on LoRaWAN: A Theoretical and Experimental Study. *arXiv* **2025**, arXiv:2501.18339.
32. Testi, E.; Arcangeloni, L.; Giorgetti, A. Machine learning-based jamming detection and classification in wireless networks. In Proceedings of the SenSys '23, Istanbul, Turkey, 12–17 November 2023.
33. Stajano, F.; Anderson, R. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Security Protocols, 7th International Workshop*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 183–194.
34. Deng, J.; Han, R.; Mishra, S. Defending against path-based DoS attacks in wireless sensor networks. In Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '05), Alexandria, VA, USA, 7 November 2005; pp. 89–96.
35. LoRa Alliance. *LoRaWAN® Specification v1.1*; LoRa Alliance: Fremont, CA, USA, 2017.
36. LoRa Alliance. *LoRaWAN® L2 1.0.4 Specification (TS001-1.0.4)*; LoRa Alliance: Fremont, CA, USA, 2020.
37. The Things Industries. Best Practices. Available online: https://www.thethingsindustries.com/docs/hardware/devices/concepts/best-practices/ (accessed on 15 January 2025).
38. Semtech. Sending Messages: Opening Receive Windows. Available online: https://learn.semtech.com/mod/book/view.php?id=172&chapterid=127 (accessed on 15 January 2025).
39. Mao, Z.; Zhou, B.; Huang, J.; Liu, D.; Yang, Q. Research on anomaly detection model for power consumption data based on time-series reconstruction. *Energies* **2024**, *17*, 4810. [CrossRef]
40. Maddikunta, P.K.R.; Srivastava, G.; Gadekallu, T.R.; Deepa, N.; Boopathy, P. Predictive model for battery life in IoT networks. *IET Intell. Transp. Syst.* **2020**, *14*, 1388–1395. [CrossRef]
41. Tan, X.; Su, S.; Zuo, Z.; Guo, X.; Sun, X. Intrusion Detection of UAVs Based on the Deep Belief Network Optimized by PSO. *Sensors* **2019**, *19*, 5529. [CrossRef] [PubMed]
42. Pirayesh, H.; Zeng, H. Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 767–809. [CrossRef]
43. Haque, A.; Saifullah, A. Stackelberg game-based anti-jamming strategies for LPWANs. In Proceedings of the IEEE Conference on Communications and Network Security, Virtual, 29 June–1 July 2023.
44. Adelantado, F.; Vilajosana, X.; Tuset-Peiró, P.; Martinez, B.; Melià-Seguí, J.; Watteyne, T. Understanding the Limits of LoRaWAN. *IEEE Commun. Mag.* **2017**, *55*, 34–40. [CrossRef]
45. Scikit-Learn Developers. StandardScaler—Scikit-Learn Documentation (Defines z = (x − μ)/σ z = (x-\mu)/\sigma z = (x − μ)/σ and Notes That μ, σ \mu, \sigma μ, σ Are Computed from Training Samples). Available online: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html (accessed on 17 October 2025).
46. Khan, M.A.; Khan, M.A.; Jan, S.U.; Ahmad, J.; Jamal, S.S.; Shah, A.A.; Pitropakis, N.; Buchanan, W.J. A deep learning-based intrusion detection system for MQTT enabled IoT. *Sensors* **2021**, *21*, 7016. [CrossRef]
47. Scikit-Learn Developers. TimeSeriesSplit—Scikit-Learn Documentation (Time-Ordered Splitting; Avoid Training on the Future). Available online: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html (accessed on 17 October 2025).
48. Kaufman, S.; Rosset, S.; Perlich, C.; Stitelman, O. Leakage in Data Mining: Formulation, Detection, and Avoidance. *ACM Trans. Knowl. Discov. Data (TKDD)* **2012**, *6*, 1–21. [CrossRef]
49. Reddi, V.J.; Cheng, C.; Kanter, D.; Mattson, P.; Schmuelling, G.; Wu, C.-J.; Anderson, B.; Breughe, M.; Charlebois, M.; Chou, W.; et al. MLPerf Inference Benchmark. In Proceedings of the 47th ACM/IEEE International Symposium on Computer Architecture (ISCA 2020), Valencia, Spain, 30 May–3 June 2020; pp. 446–459. [CrossRef]
50. OpenMP Architecture Review Board. OpenMP Application Programming Interface. Available online: https://www.openmp.org/specifications/ (accessed on 17 October 2025).
51. Netlib. Basic Linear Algebra Subprograms (BLAS)—Documentation Portal. Available online: http://www.netlib.org/blas/ (accessed on 17 October 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.