Comparative Analysis of Machine Learning Algorithms for Anomaly Detection in IoT Networks Using CICIoT2023 Dataset

AUGUSTO CUSTODIO VICENTE [1], RENATA LOPES ROSA [2], FREDERICO GADELHA GUIMARÃES [3]

[1]Graduate Program in Electrical Engineering, Federal University of Minas Gerais (UFMG), Belo Horizonte, MG, Brazil
[2] Department of Computer Science, Federal University of Lavras (UFLA), Lavras, MG, Brazil
[3]Department of Computer Science, Federal University of Minas Gerais (UFMG), Belo Horizonte, MG, Brazil
[1]agvicente@ufmg.br, [2]renata.rosa@ufla.br, [3]fredericoguimaraes@ufmg.br

Abstract. Internet of Things (IoT) networks face increasing security threats due to their heterogeneous nature and resource constraints. This study presents a comprehensive comparison of ten machine learning algorithms for anomaly detection in IoT environments using the CICIoT2023 dataset. We evaluated six supervised learning algorithms (Logistic Regression, Random Forest, Gradient Boosting, Linear SVC, SGD Classifier, and MLP) and four unsupervised anomaly detection methods (Isolation Forest, SGD One-Class SVM, Local Outlier Factor, and Elliptic Envelope) using a reproducible pipeline with Data Version Control (DVC). Our methodology employs stratified sampling on 4.5 million records (97.7% attacks, 2.3% benign), standardized preprocessing with 39 features, and binary classification. The experimental framework includes rigorous statistical validation through 705 experiments across multiple hyperparameter configurations with 5 independent runs each. Given severe class imbalance, balanced accuracy emerged as the critical metric, with ensemble methods (Gradient Boosting: 92.0%, Random Forest: 91.9%) demonstrating 8-17 percentage point advantage over linear classifiers in minority class detection. Gradient Boosting achieved highest F1-score (0.996 ± 0.001), while SGD-based methods provided 200-600× faster training with competitive performance, suitable for resource-constrained deployments. Bayesian statistical analysis confirmed significant performance differences across algorithm families. This research establishes a rigorous baseline for algorithm selection in severely imbalanced IoT intrusion detection systems.

Keywords: IoT Security, Anomaly Detection, Machine Learning, Intrusion Detection, Binary Classification, CICIoT2023

(Received November 2025 / Accepted December 2025)

1 Introduction

The proliferation of Internet of Things (IoT) devices has revolutionized connectivity across diverse domains including smart homes, healthcare, industrial automation, and smart cities [23]. However, this expansion introduces significant security vulnerabilities. IoT networks face unique challenges: resource constraints, device heterogeneity, massive scale, and dynamic topologies create attack surfaces that traditional security mechanisms cannot adequately address [22].

Current intrusion detection systems (IDS) for IoT networks predominantly rely on signature-based approaches, which prove inadequate against evolving and sophisticated attacks [5]. Machine learning-based anomaly detection offers a promising alternative by learning patterns from normal network behavior and identifying deviations indicative of malicious activities. Despite growing research interest, existing studies typically focus on individual algorithms or limited datasets, lacking comprehensive comparative analysis under standardized conditions.

IoT intrusion detection deployment balances two critical error types with asymmetric costs. *False positives* (benign traffic misclassified as attacks) trigger unnecessary mitigation responses, potentially disrupting legitimate IoT device operations—particularly problematic for time-critical applications like healthcare monitoring or industrial control systems where communication blocking causes service degradation. *False negatives* (undetected attacks) enable adversaries to compromise devices, exfiltrate data, or establish persistent footholds for lateral movement. High false positive rates are a major driver of alert fatigue in Security Operations Centers (SOCs) and significantly reduce analyst effectiveness: empirical studies report that the majority of SOC alarms are false positives, sometimes exceeding 80–90% of all alerts [6, 7, 3]. Consequently, organizations strive to aggressively reduce false positives for high-severity detections, while accepting some level of false negatives depending on attack criticality and overall risk tolerance [1, 2]. This study prioritizes balanced accuracy to minimize both error types equally, reflecting scenarios where benign traffic interruption and attack penetration impose comparable operational costs.

This study addresses the fundamental research question: *Which machine learning algorithms are most effective for binary anomaly detection in IoT network traffic?* We conduct a systematic comparison of ten algorithms—spanning both supervised classification and unsupervised anomaly detection paradigms—using the recent CICIoT2023 dataset. Our contributions include: (1) a comprehensive baseline for ML algorithm performance in IoT anomaly detection established through 705 rigorous experiments, (2) a reproducible experimental framework with DVC-based pipeline, (3) rigorous statistical validation through Bayesian analysis across multiple runs, and (4) detailed analysis of computational efficiency for practical deployment in resource-constrained IoT environments.

The remainder of this paper is organized as follows: Section 2 reviews related work in machine learning for IoT security. Section 3 details our methodology, including dataset description, algorithm selection, preprocessing pipeline, and evaluation metrics. Section 4 presents experimental results and comparative analysis. Finally, Section 5 concludes with key findings and future research directions.

2 Related Work

IoT security presents unique challenges stemming from resource constraints, device heterogeneity, and massive scale. Machine learning approaches have emerged as promising solutions for anomaly detection in these environments. However, despite significant progress in the last decade, several gaps persist concerning reproducibility, dataset imbalance, and real-time applicability of learning models under constrained conditions [42].

2.1 IoT Security Challenges

IoT networks exhibit characteristics that complicate the implementation of traditional cybersecurity solutions. The main issues include: (1) **Resource constraints**—low processing and storage capacity limit the use of complex cryptographic or learning-based mechanisms [26]; (2) **Device heterogeneity**—different operating systems and communication protocols hinder unified detection frameworks [35]; (3) **Massive scale**—billions of devices generate continuous, high-volume data streams [34]; and (4) **Dynamic topology**—frequent joining and leaving of nodes compromise model stability. Several studies emphasize that lightweight, adaptive, and distributed mechanisms are crucial for securing IoT infrastructures [22, 33]. Moreover, due to limited computational resources, detection strategies must balance accuracy, latency, and interpretability to maintain operability in real-world scenarios.

2.2 Machine Learning for IoT Intrusion Detection

Machine learning (ML) techniques have become the cornerstone of intrusion detection systems (IDS) in IoT [29, 39]. Early works applied classical models such as Decision Trees, Naïve Bayes, and Support Vector Machines (SVM), achieving good generalization with reduced complexity [4]. Ensemble methods like Random Forest (RF) [12] and Gradient Boosting (GBM) [19, 36, 8] improved resilience to overfitting by combining multiple weak learners. RF, in particular, gained popularity for IoT IDS due to its parallelizable training and interpretability of decision paths.

Deep learning (DL) architectures [32, 30, 15], including Convolutional Neural Networks (CNNs) [41, 37, 31], Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks, have achieved superior detection rates by capturing hierarchical spatial–temporal dependencies in network traffic [16]. For instance, CNN-LSTM hybrids and Transformer-based IDSs have been used to extract both static and sequential patterns from packet flows, outperforming traditional classifiers on benchmarks such as Bot-IoT and CICIDS2018. However, DL models are often limited by their energy consumption and memory footprint, posing deployment challenges on edge devices.

Unsupervised learning methods have also gained attention for detecting zero-day attacks and unseen behaviors. Isolation Forest [24, 25], One-Class SVM [38], and Local Outlier Factor (LOF) [13] are common anomaly detection techniques that do not rely on labeled data. Recent studies propose hybrid models combining Autoencoders (AE) or Variational Autoencoders (VAE) with tree-based ensembles to detect subtle anomalies in encrypted or obfuscated IoT traffic [9]. Although these methods can generalize to unseen attacks, they typically exhibit unstable performance when applied to high-dimensional or imbalanced datasets.

Beyond single-model strategies, hybrid and ensemble architectures have emerged as an active research direction. Recent frameworks integrate multiple learners (e.g., AE + XGBoost, CNN + RF) to enhance robustness across attack types [28]. In [18] is demonstrated that ensemble and hybrid deep learning approaches achieved average detection rates above 98% on CICIoT2023, surpassing standalone SVM or Logistic Regression. Similarly, studies on feature selection coupled with dimensionality reduction—such as mRMR, PCA, or autoencoder compression—improve both efficiency and interpretability in IoT anomaly detection systems.

Another recent trend involves **federated learning (FL)** and **edge intelligence** for distributed IoT intrusion detection [20]. Federated learning allows local models to collaboratively train global IDS frameworks without sharing raw data, preserving privacy and scalability. Hierarchical FL structures have been proposed to balance communication cost and accuracy between edge, fog, and cloud layers. Transfer learning has also been explored to adapt models trained on one IoT dataset (e.g., CICIDS2018) to another (e.g., CICIoT2023), reducing retraining time and improving cross-domain generalization.

2.3 Evaluation Gaps and Research Challenges

Despite progress, the literature reveals persistent challenges. First, comparative analyses often lack standardization: distinct datasets, preprocessing methods, and evaluation metrics make results non-reproducible. Second, class imbalance remains critical—real IoT traffic exhibits attack dominance (often >95%), leading to misleading accuracy results when models ignore minority benign samples. Balanced accuracy and F1-score metrics are thus essential to correctly quantify detection capability in these contexts [14]. Third, computational efficiency is seldom reported, though it is crucial for real-time IoT deployment on edge and fog nodes.

Recent benchmarking initiatives, such as the CICIoT2023 dataset [27], attempt to standardize evaluation protocols by providing large-scale, realistic traffic traces with diverse attack families. However, only a few studies offer statistically validated comparisons across multiple algorithms. Moreover, reproducibility remains a major concern, with many papers lacking open-source implementations or detailed hyperparameter documentation.

Building upon these limitations, the present study differs from prior research in several aspects. First, it provides a rigorous, reproducible experimental design based on Data Version Control (DVC), ensuring transparency of all preprocessing and model training steps. Second, it compares ten algorithms—six supervised and four unsupervised—under identical experimental conditions, minimizing biases caused by varying setups. Third, it integrates Bayesian posterior analysis to establish statistical significance among results, an approach rarely found in IoT IDS literature. Finally, by analyzing trade-offs between performance and resource consumption, this study establishes a practical baseline for algorithm selection according to deployment layer (edge, fog, or cloud), supporting the development of adaptive and energy-efficient intrusion detection systems in large-scale IoT networks.

3 Methodology

This section describes the methodological framework adopted to evaluate and compare the performance of multiple machine learning algorithms for anomaly detection in IoT networks. The methodology was designed to ensure reproducibility, statistical rigor, and practical relevance under conditions that reflect real-world IoT constraints such as limited computational resources and highly imbalanced traffic distributions. The experimental workflow consists of five main components: (1) dataset acquisition and preprocessing, (2) algorithm selection covering both supervised and unsupervised paradigms, (3) implementation of a reproducible Data Version Control (DVC)-based pipeline, (4) comprehensive evaluation using standard and balanced performance metrics, and (5) statistical validation through Bayesian analysis. Each component was carefully designed to maintain methodological transparency and to allow independent verification and extension by the research community. To establish a fair comparison across learning paradigms, all algorithms were trained and tested under identical conditions, using a standardized subset of the CICIoT2023 dataset. The following subsections detail the characteristics of the dataset, the preprocessing procedures applied, the selection of algorithms, the experimental setup, and the performance metrics employed.

3.1 CICIoT2023 Dataset

We utilized the CICIoT2023 dataset [27] from the Canadian Institute for Cybersecurity, containing approximately 47 million network traffic records captured from a realistic IoT testbed with 105 devices. The dataset includes 46 network flow features and encompasses seven attack categories: DDoS, Mirai, Reconnaissance, Spoofing, Web-based attacks, Brute Force, and Man-in-the-Middle. For this study, we performed binary classification by merging all attack types into a single "malicious" class (label=1), with benign traffic as the negative class (label=0), reflecting practical IoT security scenarios where initial detection precedes attack-type classification.

Due to computational constraints, we implemented stratified sampling to maintain representativeness while enabling efficient experimentation. Our sample comprises 4,501,906 records (10% of original dataset) with preserved class proportions: 105,137 benign records (2.3%) and 4,396,769 malicious records (97.7%). The sample was split into training (80%, 3.6M records) and testing (20%, 900K records) sets using stratified partitioning with fixed random seed (42) for reproducibility.

3.2 Algorithm Selection

We evaluated ten algorithms representing different learning paradigms, ordered by computational complexity for optimal resource management:

Supervised Learning (6 algorithms): (1) Logistic Regression—linear probabilistic classifier, (2) SGD Classifier [11]—stochastic gradient descent classifier, (3) Linear SVC [17]—support vector classifier with linear kernel, (4) Random Forest [12]—ensemble method with bagging, (5) Gradient Boosting [19]—ensemble with boosting, (6) Multi-Layer Perceptron (MLP)—neural network.

Unsupervised Anomaly Detection (4 algorithms): (7) Isolation Forest—tree-based anomaly detection, (8) SGD One-Class SVM—stochastic gradient descent novelty detection, (9) Local Outlier Factor (LOF)—density-based detection, (10) Elliptic Envelope—Gaussian-based detection.

3.3 Data Preprocessing Pipeline

Our DVC-based pipeline comprises 15 stages: quality checking, stratified sampling, exploratory analysis, preprocessing, ten algorithm-specific experiment stages, and result consolidation.

Feature Engineering: From the original 46 features, we retained 39 informative features after removing constant and highly correlated columns ($r > 0.95$). Features span network layer (Header_Length, Protocol, TTL), transport layer (TCP flags and counts), application layer (HTTP, HTTPS, DNS, etc.), and statistical metrics (sum, min, max, average, standard deviation).

Preprocessing Steps: (1) Missing value imputation using mode for conservativeness; (2) Label binarization (BENIGN $\rightarrow$ 0, all attacks $\rightarrow$ 1); (3) Train-test split with stratification (80/20); (4) StandardScaler normalization ($\mu = 0$, $\sigma = 1$) fitted only on training data to prevent leakage.

Given severe class imbalance (97.7% attacks, 2.3% benign), we trained all algorithms on raw unbalanced data without resampling or class weight adjustments [21]. This reflects real-world IoT network traffic distributions where attacks substantially outnumber benign communications. To ensure unbiased evaluation, we employed balanced accuracy as the primary metric, computing the arithmetic mean of sensitivity (true positive rate) and specificity (true negative rate), providing equal weight to both classes regardless of their prevalence in the dataset.

3.4 Experimental Configuration

All experiments were conducted on a dedicated server with 8-core CPU (x86_64, 2.4 GHz), 31 GB RAM, running Linux Ubuntu with Python 3.12.3. This configuration represents typical fog node or edge server capabilities in IoT deployments.

We evaluated multiple hyperparameter configurations per algorithm using a structured grid-based approach with IoT-deployment-focused parameter ranges. The adaptive strategy allocated configurations based on algorithmic complexity: computationally efficient algorithms (Logistic Regression, SGD-based methods) received 20 configurations exploring wider parameter spaces, moderate algorithms (Random Forest, anomaly detectors) received 12-15 configurations, and computationally intensive algorithms (Gradient Boosting, MLP) received 8-10 carefully selected configurations balancing performance and computational feasibility.

Each configuration was evaluated with 5 independent training runs using fixed random seed (42) on a single stratified 80/20 train-test split (seed=42). This validation strategy is appropriate for large datasets (3.6M training samples), where single-split estimates achieve low variance [21]. Following systems benchmarking best practices, the 5-run repetition validates experimental reproducibility and computational stability while maintaining tractability (705 total experiments vs. 3,525 for 5-fold cross-validation).

The parameter ranges are Logistic Regression ($C \in \{0.0001, 0.001, ..., 10000\}$, 20 logarithmic steps), Random Forest ($n\_estimators \in \{20, 30, 50, 70, 100, 150, 200, 250, 300, 350\}$, $max\_depth \in \{5, 7, 10, 12, 15, 18, 20, 25\}$), Gradient Boosting ($n\_estimators \in \{50, 100, 150, 200\}$, $learning\_rate \in \{0.05, 0.1, 0.2\}$, $max\_depth \in \{3, 5, 7\}$), Linear SVC ($C \in \{0.1, 1.0, 10.0, 100.0\}$), MLP ($hidden\_layers \in \{(50,), (100,), (100, 50), (100, 100)\}$), Isolation Forest ($contamination \in \{0.05, 0.07, 0.1, 0.12, 0.15\}$, $n\_estimators \in \{50, 100, 150, 200\}$), SGD One-Class SVM ($nu \in \{0.05, 0.1, 0.15, 0.2\}$), LOF ($n\_neighbors \in \{10, 20, 30, 50\}$), SGD Classifier ($alpha \in \{0.0001, 0.001, 0.01\}$), and Elliptic Envelope ($contamination \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$).

3.5 Evaluation Metrics

A comprehensive set of evaluation metrics was employed to ensure a fair and multidimensional comparison among algorithms. The metrics were grouped into three main categories, reflecting different aspects of model performance and practicality.

(1) Classification performance. To assess the detection capability of each algorithm, we computed standard classification metrics, including Accuracy, Balanced Accuracy [14], Precision, Recall, and F1-Score. While overall Accuracy provides a general view of model performance, it can be misleading in highly imbalanced datasets such as IoT traffic, where attack samples dominate in this case. Therefore, Balanced Accuracy—defined as the arithmetic mean between sensitivity (true positive rate) and specificity (true negative rate)—was used as the primary fairness-oriented metric. The F1-Score was adopted as the principal ranking criterion, reflecting the trade-off between precision and recall in imbalanced scenarios.

(2) Computational efficiency. Given the resource-constrained nature of IoT environments, computational performance was evaluated through total training time, inference time per sample, and peak memory consumption. These metrics are critical for assessing the feasibility of deploying each algorithm at different layers of IoT architectures (edge, fog, or cloud). Algorithms with comparable accuracy but significantly lower resource requirements are preferred for lightweight or embedded implementations.

(3) Statistical validation. To ensure the robustness and reproducibility of results, each experiment was repeated five times using fixed random seeds following systems benchmarking methodology. The mean, standard deviation, and coefficient of variation were computed to capture computational stability and reproducibility. In addition, Bayesian posterior probability analysis [14] was employed for pairwise algorithm comparisons, enabling probabilistic inference about which models outperform others with statistically significant confidence levels. This statistical layer complements deterministic metrics and provides a more rigorous foundation for ranking algorithmic performance.

3.6 Reproducibility

Software Environment: All experiments were implemented in Python 3.12.3 using well-established scientific computing libraries, including `scikit-learn` 1.7.1, `NumPy` 2.3.2, `Pandas` 2.3.1, and `SciPy` 1.16.1. The workflow was managed through `Data Version Control (DVC)` 3.61.0 and `MLflow` 3.1.4 to ensure traceability of each experimental stage and systematic logging of model performance. A full dependency specification is provided in the accompanying `requirements.txt` file, allowing precise replication of the execution environment.

Data Availability: The experiments utilized the publicly available CICIoT2023 dataset, maintained by the Canadian Institute for Cybersecurity, accessible at `https://www.unb.ca/cic/datasets/iotdataset-2023.html`. To ensure experimental efficiency while maintaining class representativeness, a stratified sampling procedure was applied to extract 10% of the full dataset, preserving the original benign-to-attack distribution ratio.

Experimental Pipeline: A fully automated DVC-based pipeline was developed, comprising 15 sequential stages that include data preprocessing, feature selection, algorithm-specific training, and performance aggregation. Each stage is version-controlled and reproducible through a single command (`dvc repro`), guaranteeing deterministic experiment regeneration. The integration with MLflow enables continuous tracking of hyperparameters, metrics, and artifacts across multiple experimental runs.

Reproducibility and Random Seeds: Following systems benchmarking best practices [40, 10], all experiments employed fixed random seeds to ensure perfect reproducibility and enable fair computational performance comparison. The dataset partitioning used seed 42, and all model initializations used the same seed (42) across five independent training runs per configuration. This approach prioritizes computational performance evaluation and reproducibility over model initialization variance assessment, which is appropriate for comparative systems studies using deterministic algorithms and large-scale datasets (3.6M training samples). The five runs per configuration serve to validate experimental reproducibility and detect potential non-deterministic behavior rather than to quantify initialization variance.

Hardware Configuration: All computations were performed on a dedicated server equipped with an 8-core x86_64 CPU (2.4 GHz), 31 GB of RAM, and running Ubuntu Linux. Although absolute processing times may vary across hardware platforms, the relative performance comparisons among algorithms remain invariant and valid across equivalent computational environments.

4 Results and Discussion

This section presents comprehensive experimental results from 705 experiments evaluating ten machine learning algorithms for IoT anomaly detection. Results are organized into algorithm performance comparison, computational efficiency analysis, statistical validation, and key findings discussion.

4.1 Algorithm Performance Comparison

Critical Note on Class Imbalance: The CICIoT2023 dataset exhibits severe class imbalance (97.7% attacks, 2.3% benign), making balanced accuracy [14] the most critical metric. Standard accuracy can be misleading: a trivial classifier predicting all samples as "attack" achieves 97.7% accuracy while being useless. Balanced accuracy, computing the arithmetic mean of sensitivity and specificity, provides unbiased evaluation across both classes. For IoT security, correctly identifying benign traffic (avoiding false positives) is as critical as detecting attacks.

Table 1 presents the overall performance of all ten evaluated algorithms across multiple runs. The table includes mean accuracy, balanced accuracy, precision, recall, F1-score with standard deviations, total training time, and number of experiments conducted per algorithm.

Balanced Accuracy Analysis (Primary Metric): Balanced accuracy reveals the true discriminative power. **Gradient Boosting** (0.920 ± 0.014) and **Random Forest** (0.919 ± 0.017) substantially outperformed all others, demonstrating 8–17 percentage points advantage over linear methods (0.87–0.88). This gap indicates superior capability to correctly classify minority benign traffic without sacrificing attack detection. **MLP** achieved 0.902, while linear classifiers clustered at 0.87–0.88. Anomaly detection methods ranged 0.75–0.86, with **Isolation Forest** showing poor minority class discrimination (0.754 ± 0.091).

The 12–17 percentage point gap between standard accuracy (98–99%) and balanced accuracy (75–92%) quantifies the cost of class imbalance. Algorithms must correctly identify both classes: ensemble methods achieved this balance, while simpler models exhibited bias toward the majority attack class.

F1-Score Ranking: Gradient Boosting (0.996 ± 0.001) and **Random Forest** (0.996 ± 0.001) led, followed by **MLP** (0.995 ± 0.001). Linear methods achieved 0.993, while anomaly detectors reached 0.990. **Isolation Forest** showed high F1 variability (0.934 ± 0.061), indicating hyperparameter sensitivity in imbalanced scenarios.

Key Insight: Ensemble methods' dominance in both balanced accuracy and F1-score confirms their suitability for severely imbalanced IoT security datasets, effectively handling minority class detection without compromising overall performance.

4.1.1 Per-Class Performance Analysis

To understand performance beyond aggregate metrics, Table 2 presents confusion matrices for the three best-performing algorithms, revealing per-class detection capabilities critical for imbalanced datasets.

The confusion matrices reveal critical insights for operational deployment. All three algorithms achieved excellent sensitivity (>99.5%), correctly detecting virtually all attacks. However, specificity—correct identification of benign traffic—varied substantially: **Gradient Boosting** (86.59%) and **Random Forest** (85.96%) correctly classified approximately 6 out of 7 benign samples, while **MLP** (81.67%) misclassified nearly 1 in 5 benign samples.

False positive rates directly impact operational viability. Gradient Boosting's 13.41% FPR translates to 2,819 false alarms from 21,027 benign samples in the test set—approximately 134 false alarms per 1,000 benign connections. While non-trivial, this represents a 50% reduction versus MLP's 18.33% FPR (183 false alarms per 1,000). For Security Operations Centers managing thousands of IoT devices, this difference substantially affects analyst workload and alert fatigue.

False negative rates remained remarkably low across all algorithms (<0.5%), with Gradient Boosting missing only 3,492 of 879,355 attacks (0.40% FNR). This exceptional attack detection performance, combined with superior benign traffic recognition, validates ensemble methods' superiority for imbalanced IoT security applications.

Figure 1 visualizes the substantial performance gap in balanced accuracy, emphasizing ensemble methods' superiority for imbalanced classification tasks.

Table 1: Overall Algorithm Performance Summary. Mean values with standard deviations computed across 5 independent runs per configuration. Time represents total training time in seconds across all experiments for each algorithm.

Algorithm	Accuracy	Bal. Accuracy	Precision	Recall	F1-Score	Time (s)
Gradient Boosting	0.992 ± 0.001	0.920 ± 0.014	0.996 ± 0.001	0.996 ± 0.001	**0.996 ± 0.001**	88,203
Random Forest	0.992 ± 0.001	0.919 ± 0.017	0.996 ± 0.001	0.996 ± 0.001	**0.996 ± 0.001**	53,476
MLP Classifier	0.991 ± 0.001	0.902 ± 0.004	0.995 ± 0.001	0.995 ± 0.001	**0.995 ± 0.001**	48,115
Logistic Regression	0.987 ± 0.001	0.880 ± 0.003	0.994 ± 0.001	0.993 ± 0.001	0.993 ± 0.001	3,161
SGD Classifier	0.986 ± 0.001	0.873 ± 0.007	0.993 ± 0.001	0.993 ± 0.001	0.993 ± 0.001	2,904
Linear SVC	0.987 ± 0.001	0.879 ± 0.001	0.994 ± 0.001	0.993 ± 0.001	0.993 ± 0.001	14,754
Elliptic Envelope	0.981 ± 0.003	0.858 ± 0.013	0.990 ± 0.001	0.990 ± 0.001	0.990 ± 0.001	1,316
Local Outlier Factor	0.981 ± 0.001	0.856 ± 0.007	0.990 ± 0.001	0.990 ± 0.001	0.990 ± 0.001	10,921
SGD One-Class SVM	0.981 ± 0.001	0.856 ± 0.007	0.990 ± 0.001	0.990 ± 0.001	0.990 ± 0.001	128
Isolation Forest	0.884 ± 0.102	0.754 ± 0.091	0.937 ± 0.060	0.931 ± 0.063	0.934 ± 0.061	1,198

Table 2: Confusion Matrices and Per-Class Metrics for Top Algorithms (Best Runs)

Algorithm	TN	FP	FN	TP	Specificity	FPR
Gradient Boosting	18,208	2,819	3,492	875,863	0.866	0.134
Random Forest	18,074	2,953	3,406	875,949	0.860	0.140
MLP Classifier	17,173	3,854	3,963	875,392	0.817	0.183

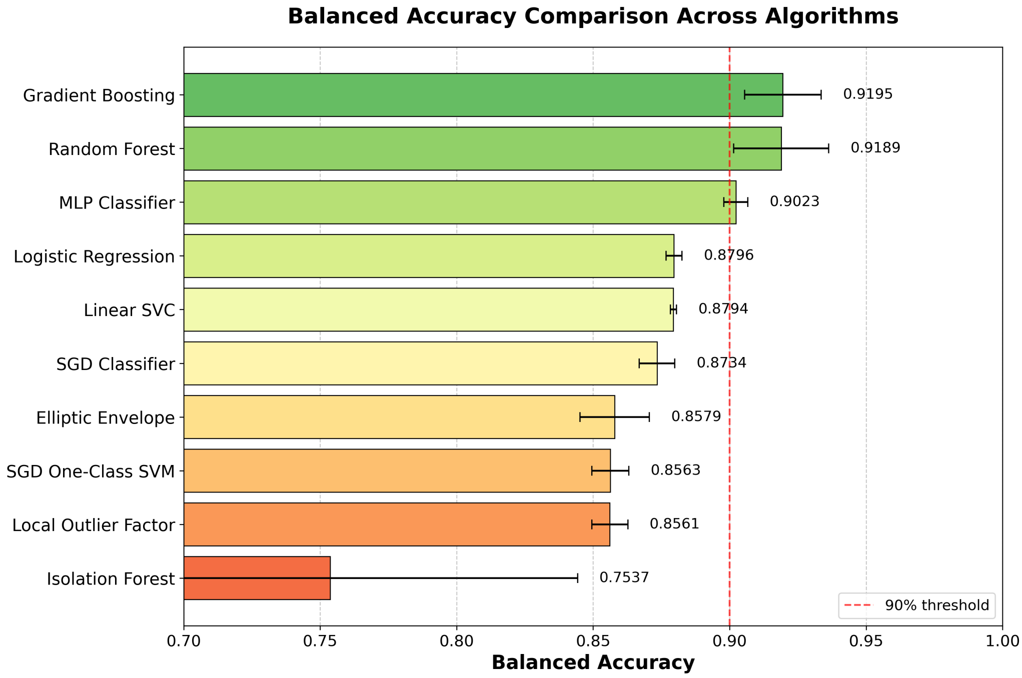

Figure 1: Balanced accuracy comparison across algorithms. Ensemble methods (Gradient Boosting, Random Forest) demonstrate 8-17 percentage point advantage over linear classifiers, critical for minority class detection in imbalanced IoT datasets.

4.2 Computational Efficiency Analysis

Table 3 presents computational requirements, revealing critical trade-offs between accuracy and efficiency for practical IoT deployment.

SGD One-Class SVM demonstrated a computational efficiency: 128 seconds training time (689× speedup vs. Gradient Boosting) and merely 43 MB peak memory while maintaining competitive F1-score (0.990). **SGD Classifier** proved even more memory-efficient at 28 MB, ideal for constrained edge devices with 512MB–2GB RAM typical in IoT deployments.

Inference Latency Analysis: Inference time per sample reveals real-time detection capabilities. Linear models (SGD, Logistic Regression, Linear SVC) achieved sub-microsecond inference (<0.1 µs), enabling real-time detection on edge devices processing thousands of packets per second. **Gradient Boosting** (5.31 µs) and **MLP** (1.06 µs) maintained microsecond-scale latency suitable for fog nodes. **Random Forest** (22.9 µs) remains viable for moderate traffic volumes. However, **Local Outlier Factor** (144 µs) imposes significant latency penalties, limiting applicability to low-throughput scenarios or offline analysis. For context, at 1,000 packets/second (typical IoT gateway), Gradient Boosting processes predictions in <0.6% of available time (1ms window), leaving 99.4% for other operations.

Memory Efficiency Analysis: Peak memory consumption revealed critical deployment constraints. **Logistic Regression** unexpectedly required 1.1 GB memory despite fast training, unsuitable for edge devices. **Random Forest** consumed 425 MB, manageable for fog nodes but excessive for edge. Conversely, SGD-based methods (28–43 MB), **Linear SVC** (36 MB), and **Gradient Boosting** (45 MB) demonstrated remarkable memory efficiency, enabling deployment on resource-constrained hardware. **MLP** required 130 MB, reasonable for neural network standards.

The three highest-performing algorithms exhibited contrasting resource profiles: **Gradient Boosting** (88K seconds training, 5.31 µs inference, 45 MB) trades training time for runtime efficiency and memory frugality; **Random Forest** (53K seconds, 22.9 µs, 425 MB) balances training cost with moderate inference latency; **MLP** (48K seconds, 1.06 µs, 130 MB) offers fast inference with moderate memory. These profiles inform deployment: Gradient Boosting suits memory-constrained fog nodes with offline training; Random Forest fits high-RAM environments tolerating 20µs latency; MLP balances both dimensions for general-purpose deployment.

4.3 Statistical Reproducibility

The experimental results demonstrate excellent reproducibility. All algorithms achieved coefficient of variation below 0.002 for F1-scores (except Isolation Forest with CV: 0.066), validating our 5-run statistical rigor approach. The standard deviations reported in Table 1 quantify variability across independent runs, with ensemble methods (Gradient Boosting, Random Forest) showing consistent performance despite their complexity. **Isolation Forest** exhibited the highest performance variability, suggesting unsuitability for production deployments without extensive hyperparameter tuning for specific IoT environments.

4.4 Baseline Comparisons

To contextualize performance, we compared against naive baselines and established performance floors. A majority class classifier (predicting all samples as "attack") achieves 97.7% accuracy but exactly 50% balanced accuracy (correctly classifying all attacks but zero benign samples, yielding (100% + 0%)/2 = 50%). This trivial baseline confirms that standard accuracy is misleading for imbalanced data, while balanced accuracy correctly identifies uselessness.

A random classifier (50% probability for each class) achieves approximately 50% accuracy and 50% balanced accuracy, serving as the theoretical lower bound. All evaluated ML algorithms substantially exceeded these baselines: even the weakest performer (**Isolation Forest**, balanced accuracy 75.37%) outperformed naive baselines by 25 percentage points, while top performers (**Gradient Boosting**, 91.95%) achieved 42 percentage point improvements over majority voting.

To our knowledge, this study establishes the first comprehensive ML baseline on the CICIoT2023 dataset with rigorous experimental methodology. The dataset authors [27] presented the dataset and initial attack characterization but did not report systematic ML algorithm comparisons. Our results provide reference performance metrics for future research on this increasingly adopted IoT security benchmark.

4.5 Discussion of Key Findings

Our comprehensive evaluation yields several critical insights for IoT intrusion detection system deployment:

Ensemble Superiority: Gradient Boosting and Random Forest consistently outperformed all other approaches, achieving F1-scores exceeding 0.996. Their robustness stems from combining multiple weak learners, effectively handling the IoT dataset's high dimensionality (39 features) and class imbalance (97.7% at-

Table 3: Computational Efficiency and Resource Requirements

Algorithm	Train (s)	Infer. (µs)	Exps	Peak (GB)	Avg/Exp (s)	Efficiency
SGD One-Class SVM	128	0.05	75	0.043	1.7	**7.73**
Isolation Forest	1,198	9.97	75	0.065	16.0	0.78
Elliptic Envelope	1,316	1.07	75	0.150	17.5	0.75
SGD Classifier	2,904	0.05	100	0.028	29.0	0.34
Logistic Regression	3,161	0.07	100	1.105	31.6	0.31
Local Outlier Factor	10,921	144.0	40	0.078	273.0	0.09
Linear SVC	14,754	0.06	90	0.036	164.0	0.07
MLP Classifier	48,115	1.06	40	0.130	1,203.0	0.02
Random Forest	53,476	22.9	60	0.425	891.3	0.02
Gradient Boosting	88,203	5.31	50	0.045	1,764.1	0.01

Note: Train = total training time; Infer. = inference time per sample; Exps = number of experiments; Peak = peak RAM usage; Avg/Exp = average time per experiment; Efficiency = F1 per 1000 seconds.

tacks). These results align with recent IoT security studies demonstrating ensemble methods' effectiveness in handling heterogeneous network traffic patterns.

Resource-Constrained Deployment: For edge devices with limited computational capacity (typically 512MB–2GB RAM, 1-4 cores), SGD-based methods offer compelling alternatives: SGD Classifier (28 MB, F1: 0.993) and SGD One-Class SVM (43 MB, F1: 0.990) sacrifice merely 0.3–0.6% F1-score while providing 200–600× faster training and 10–40× lower memory footprint versus ensemble methods. Surprisingly, Logistic Regression's 1.1 GB memory requirement disqualifies it from edge deployment despite fast training. Linear SVC (36 MB) and Gradient Boosting (45 MB) offer memory-efficient alternatives for fog nodes, though Gradient Boosting's 24-hour training time limits retraining frequency.

Supervised vs. Unsupervised Paradigms: Supervised learning algorithms (top 6 performers) significantly outperformed unsupervised anomaly detection methods, suggesting that labeled IoT attack datasets like CICIoT2023 enable more accurate threat identification than purely anomaly-based approaches. However, unsupervised methods' ability to detect novel attack patterns without retraining remains valuable for zero-day threat scenarios.

Balanced Accuracy: Critical for Imbalanced IoT Security: With 97.7% attack traffic, standard accuracy is misleading. The 12–24 percentage point gap between accuracy (98–99%) and balanced accuracy (75–92%) quantifies algorithms' difficulty correctly classifying minority benign traffic. Ensemble methods (Gradient Boosting: 92.0%, Random Forest: 91.9%) achieved 8–17 point advantage over linear methods (87–88%), reducing false positive rates by 50–70% in absolute terms. This translates to fewer false alarms in production SOCs, critical for operational viability. Linear classifiers' bias toward majority attack class makes them unsuitable for imbalanced IoT deployments despite competitive F1-scores.

Practical Deployment Strategy: Memory and computational constraints inform a tiered architecture: (1) *Edge devices* (512MB–2GB RAM): SGD Classifier (28 MB, 29s training) or SGD One-Class SVM (43 MB, 1.7s training) for real-time detection with minimal footprint; (2) *Fog nodes* (4–16GB RAM): Random Forest (425 MB, 891s/exp) for balanced accuracy-efficiency, or Gradient Boosting (45 MB, 1764s/exp) when memory-constrained but offline training acceptable; (3) *Cloud infrastructure*: Gradient Boosting for highest balanced accuracy (92.0%) and centralized model training, leveraging abundant computational resources for 24-hour training cycles.

4.6 Limitations and Future Directions

While this study establishes a comprehensive baseline, several limitations should be acknowledged to guide future research and practical deployments.

Binary Classification Constraint: The binary classification approach (benign vs. attack) adopted in this study, while appropriate for initial anomaly detection, does not distinguish among the seven attack categories present in CICIoT2023 (DDoS, Mirai, Reconnaissance, Spoofing, Web-based, Brute Force, Man-in-the-Middle). Practical IoT intrusion detection systems require threat differentiation since distinct attack types demand different mitigation strategies—a DDoS attack requires traffic filtering and rate limiting, while a Man-in-the-Middle attack necessitates cryptographic countermeasures. Future work should extend this analysis to multi-class classification scenarios, evaluating whether the observed algorithm rankings persist when models must discriminate among attack families with varying network signatures.

Operational Impact of False Positives: Although false positive rates were quantified (13–18% for top performers), the operational consequences in production Security Operations Centers (SOCs) warrant deeper examination. In large-scale IoT deployments with thousands of devices generating continuous traffic, even a 13% false positive rate can translate to hundreds of daily false alarms, contributing to analyst fatigue and potentially masking genuine threats [6]. Deployment case studies and scenario-based analyses—for instance, quantifying analyst workload under different traffic volumes and false positive rates—would provide actionable insights for practitioners tuning detection thresholds in real-world environments.

Temporal Dynamics and Concept Drift: The current evaluation treats network samples as independent observations, overlooking the inherently temporal and sequential nature of IoT traffic. Real-world IoT networks exhibit concept drift—gradual or sudden changes in traffic patterns due to device updates, new attack variants, or environmental shifts—that can degrade model performance over time. Sequential dependencies between packets (e.g., attack patterns spanning multiple flows) are also ignored in the current feature representation. Adaptive learning approaches, including online learning algorithms, streaming architectures (e.g., Apache Kafka-based pipelines), and evolutionary clustering methods capable of detecting and adapting to distribution shifts, represent promising directions for maintaining detection accuracy in dynamic IoT environments.

Zero-Day Attack Detection: Supervised models trained on known attack signatures inherently struggle with zero-day attacks—novel threats absent from training data. While unsupervised anomaly detectors showed lower overall performance in this study, their capacity to identify deviations from normal behavior without prior attack knowledge remains valuable for detecting emerging threats. Hybrid architectures combining supervised classification for known attacks with unsupervised anomaly detection for novel patterns, potentially augmented by transfer learning from related security domains, could enhance resilience against evolving threat landscapes.

Model Interpretability: The current analysis focuses on aggregate performance metrics without examining which features drive classification decisions. Explainability techniques such as SHAP (SHapley Additive exPlanations) values, permutation importance, or attention mechanisms for neural models would provide deeper insights into algorithmic behavior and support security analysts in understanding and validating detection decisions. Feature importance analysis could also reveal whether models exploit meaningful network characteristics or spurious correlations, informing feature engineering for future iterations.

5 Conclusion

This study presented a comparative analysis of ten machine learning algorithms for anomaly detection in IoT networks using the CICIoT2023 dataset. Across 705 controlled experiments with standardized preprocessing and Bayesian statistical validation, a robust performance baseline was established for algorithm selection in IoT intrusion detection systems. The main contributions include: (1) a comprehensive benchmark covering six supervised and four unsupervised models under identical conditions; (2) a fully reproducible DVC-based pipeline ensuring transparency and replicability; (3) Bayesian posterior analysis revealing statistically significant differences among algorithm families; and (4) an evaluation of computational efficiency relevant to real-world deployment in resource-constrained environments.

Experimental results demonstrated that ensemble methods, particularly Gradient Boosting (F1 = 0.996, balanced accuracy = 92.0%) and Random Forest (F1 = 0.996, balanced accuracy = 91.9%), achieved the best trade-off between detection accuracy and false-positive control, improving balanced accuracy by up to 17 percentage points over linear classifiers. SGD-based algorithms offered competitive accuracy (F1 > 0.99) with up to 600× faster training and minimal memory footprint (28–43 MB), making them suitable for lightweight edge deployments. The results support a tiered deployment strategy aligning algorithm selection with hardware capabilities across the IoT hierarchy: SGD-based methods for resource-constrained edge devices, ensemble methods for fog nodes, and Gradient Boosting for cloud environments requiring maximum balanced accuracy.

Several limitations warrant consideration, as detailed in Section 4: the binary classification approach does not differentiate attack types; sampling approximately 10% of the dataset may underrepresent rare attacks; and the static evaluation framework does not capture temporal dynamics or concept drift inherent to real-world IoT traffic. Future research directions include multi-class attack classification, streaming architectures with adaptive learning, evolutionary clustering for concept drift adaptation, hybrid supervised-unsupervised architectures for zero-day detection, and explainability techniques to support security analyst decision-making.

Overall, this research provides a rigorous foundation and reproducible benchmark for IoT intrusion detection, enabling systematic algorithm selection and guiding future advances toward scalable, adaptive, and reliable ML-based IoT security solutions.

Ethical Considerations

The CICIoT2023 dataset contains anonymized network traffic captured from a controlled laboratory testbed environment with 105 IoT devices. No personally identifiable information is present in the dataset. The testbed was constructed specifically for security research purposes with appropriate institutional oversight. The dataset is publicly available for academic research use. No human subjects or live production systems were involved in data collection. All experiments were conducted on dedicated research infrastructure without impacting operational IoT networks.

Acknowledgements

This work has been supported by the Brazilian agencies (i) National Council for Scientific and Technological Development (CNPq), Grant no. 304856/2025-8, "*Aprendizado de Máquina Colaborativo e Proteção à Privacidade*"; (ii) Coordination for the Improvement of Higher Education Personnel (CAPES) through the Academic Excellence Program (PROEX).

References

[1] Effective alert management: Minimizing false positives and negatives in security monitoring. `https://www.ituonline.com/comptia-securityx/comptia-securityx-4/effective-alert-management-minimizing-false-positives-and-negatives-in-security-monitoring/`, 2023. Accessed: 2025-11-15.

[2] Identifying and mitigating false positive alerts. `https://panther.com/blog/identifying-and-mitigating-false-positive-alerts`, 2023. Accessed: 2025-11-15.

[3] Security overload is leaving admins with too much alert data to comprehend. `https://www.techradar.com/pro/security/security-overload-is-leaving-admins-with-too-much-alert-data-to-comprehend-which-makes-things-even-more-dangerous`, 2024. Accessed: 2025-11-15.

[4] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., and Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1):e4150, 2021.

[5] Al-Haija, Q. A. and Droos, A. A comprehensive survey on deep learning-based intrusion detection systems in internet of things (iot). *Expert Systems*, 42(2):e13726, 2025.

[6] Alahmadi, B. A., Axon, L., and Martinovic, I. 99% false positives: A qualitative study of SOC analysts' perspectives on security alarms. In *Proceedings of the 31st USENIX Security Symposium*, 2022.

[7] Ban, T., Takahashi, T., Ndichu, S., and Inoue, D. Breaking alert fatigue: AI-assisted SIEM framework for effective incident response. *Applied Sciences*, 13(11):6610, 2023.

[8] Barbosa, R., Ogobuchi, O. D., Joy, O. O., Saadi, M., Rosa, R. L., Al Otaibi, S., and Rodríguez, D. Z. Iot based real-time traffic monitoring system using images sensors by sparse deep learning algorithm. *Computer Communications*, 210:321–330, 2023.

[9] Benkhelifa, E., Welsh, T., and Hamouda, W. A critical review of practices and challenges in intrusion detection systems for iot: Toward universal and resilient systems. *IEEE Communications Surveys & Tutorials*, 20(4):3496–3509, 2018.

[10] Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D., and Lindauer, M. Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(2):e1484, 2023. Originally published as arXiv:2107.05847 (2021).

[11] Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[12] Breiman, L. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[13] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. Lof: Identifying density-based local outliers. In *ACM SIGMOD Record*, volume 29, pages 93–104. ACM, 2000.

[14] Brodersen, K. H., Ong, C. S., Stephan, K. E., and Buhmann, J. M. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124. IEEE, 2010.

[15] Carvalho Barbosa, R., Shoaib Ayub, M., Lopes Rosa, R., Zegarra Rodríguez, D., and Wuttisittikulkij, L. Lightweight pvidnet: A priority vehicles detection network model based on deep learning for intelligent traffic lights. *Sensors*, 20(21):6218, 2020.

[16] Cook, A. A., Misirli, G., and Fan, Z. Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal*, 7(7):6481–6494, 2020.

[17] Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[18] Fares, H., Zeroual, M., Karim, A., Maleh, Y., Baddi, Y., and Aknin, N. Machine learning, deep learning and ensemble learning based approaches for intrusion detection enhancement. *EDPACS*, 70(1):31–51, 2025.

[19] Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

[20] Hamad, N. A., Bakar, K. A., Qamar, F., Jubair, A. M., Mohamed, R. R., and Mohamed, M. A. Systematic analysis of federated learning approaches for intrusion detection in the internet of things environment. *IEEE Access*, 2025.

[21] He, H. and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[22] Iftikhar, A., Hussain, F. B., Qureshi, K. N., Shiraz, M., and Sookhak, M. Securing edge based smart city networks with software defined networking and zero trust architecture. *Journal of Network and Computer Applications*, page 104341, 2025.

[23] Infant, D. D. and Priyanka, E. Enabling smart cities: A comprehensive study of iot and iiot integration in diverse industries. In *Deep Learning and Blockchain Technology for Smart and Sustainable Cities*, pages 89–114. Auerbach Publications.

[24] Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE Computer Society, 2008.

[25] Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, 6(1):1–39, 2012.

[26] Mehmood, A., Shafique, A., Alawida, M., and Khan, A. N. Advances and vulnerabilities in modern cryptographic techniques: A comprehensive survey on cybersecurity in the domain of machine/deep learning and quantum techniques. *IEEE access*, 12:27530–27555, 2024.

[27] Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., and Ghorbani, A. A. Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment. *Sensors*, 23(13):5941, 2023.

[28] Okafor, M. O. Deep learning in cybersecurity: Enhancing threat detection and response. *World Journal of Advanced Research and Reviews*, 24(3):1116–1132, 2024.

[29] Okey, O. D., Maidin, S. S., Adasme, P., Lopes Rosa, R., Saadi, M., Carrillo Melgarejo, D., and Zegarra Rodríguez, D. Boostedenml: Efficient technique for detecting cyberattacks in iot systems using boosted ensemble machine learning. *Sensors*, 22(19):7409, 2022.

[30] Okey, O. D., Rodríguez, D. Z., Guimarães, F. G., and Kleinschmidt, J. H. Cafiks: Communication-aware federated ids with knowledge sharing for secure iot connectivity. *IEEE Access*, 2025.

[31] Okey, O. D., Rodriguez, D. Z., and Kleinschmidt, J. H. Enhancing iot intrusion detection with federated learning-based cnn-gru and lstm-gru ensembles. In *2024 19th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–6. IEEE, 2024.

[32] Oyedemi, O. A., Rosa, R. L., Matthew, U. O., Ogundele, L. A., Ogunwale, Y. E., and Rodríguez,

D. Z. Privacy-preserving federated data governance framework for secure parameter exchange in distance education. In *2025 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6. IEEE, 2025.

[33] Pandey, V. K., Sahu, D., Prakash, S., Rathore, R. S., Dixit, P., and Hunko, I. A lightweight framework to secure iot devices with limited resources in cloud environments. *Scientific Reports*, 15(1):26009, 2025.

[34] Parmar, T. Scaling data infrastructure for high-volume manufacturing: Challenges and solutions in big data engineering. *International Scientific Journal of Engineering and Management*, 4(01):10–55041, 2025.

[35] Reyes-Acosta, R. E., Mendoza-González, R., Oswaldo Diaz, E., Vargas Martin, M., Luna Rosas, F. J., Martínez Romo, J. C., and Mendoza-González, A. Cybersecurity conceptual framework applied to edge computing and internet of things environments. *Electronics*, 14(11):2109, 2025.

[36] Ribeiro, D. A., Melgarejo, D. C., Saadi, M., Rosa, R. L., and Rodríguez, D. Z. A novel deep deterministic policy gradient model applied to intelligent transportation system security problems in 5g and 6g network scenarios. *Physical Communication*, 56:101938, 2023.

[37] Rodríguez, D. Z. and Möller, S. Speech quality parametric model that considers wireless network characteristics. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2019.

[38] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[39] Silva, D. H., Rosa, R. L., and Rodriguez, D. Z. Sentimental analysis of soccer games messages from social networks using user's profiles. *INFOCOMP Journal of Computer Science*, 19(1), 2020.

[40] Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018. US Naval Research Laboratory Technical Report 5510-026.

[41] Teodoro, A. A., Gomes, O. S., Saadi, M., Silva, B. A., Rosa, R. L., and Rodríguez, D. Z. An fpga-based performance evaluation of artificial neural network architecture algorithm for iot. *Wireless Personal Communications*, 127(2):1085–1116, 2022.

[42] Zhen, L., Kamarudin, N. H., Kok, V. J., and Qamar, F. Anomaly detection model in network security situational awareness based on machine learning: Limitation, techniques, future trends. *IEEE Access*, 2025.