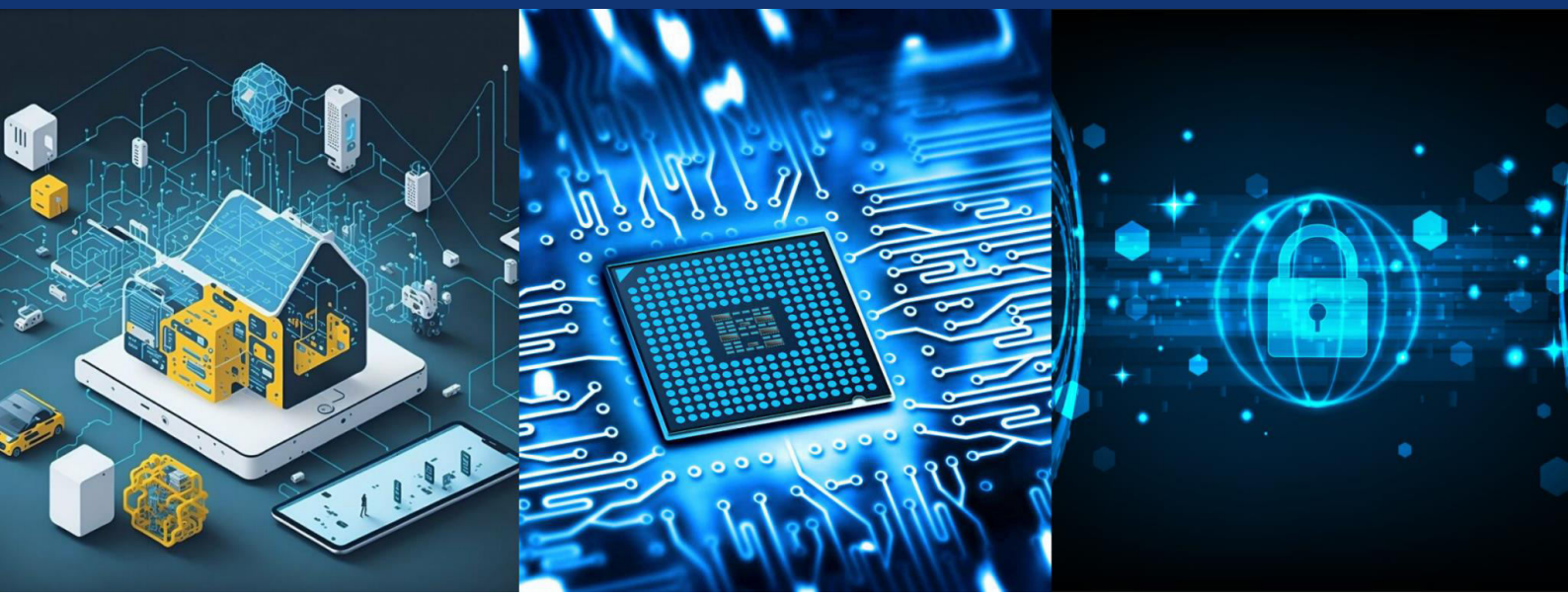
IJIRCCE

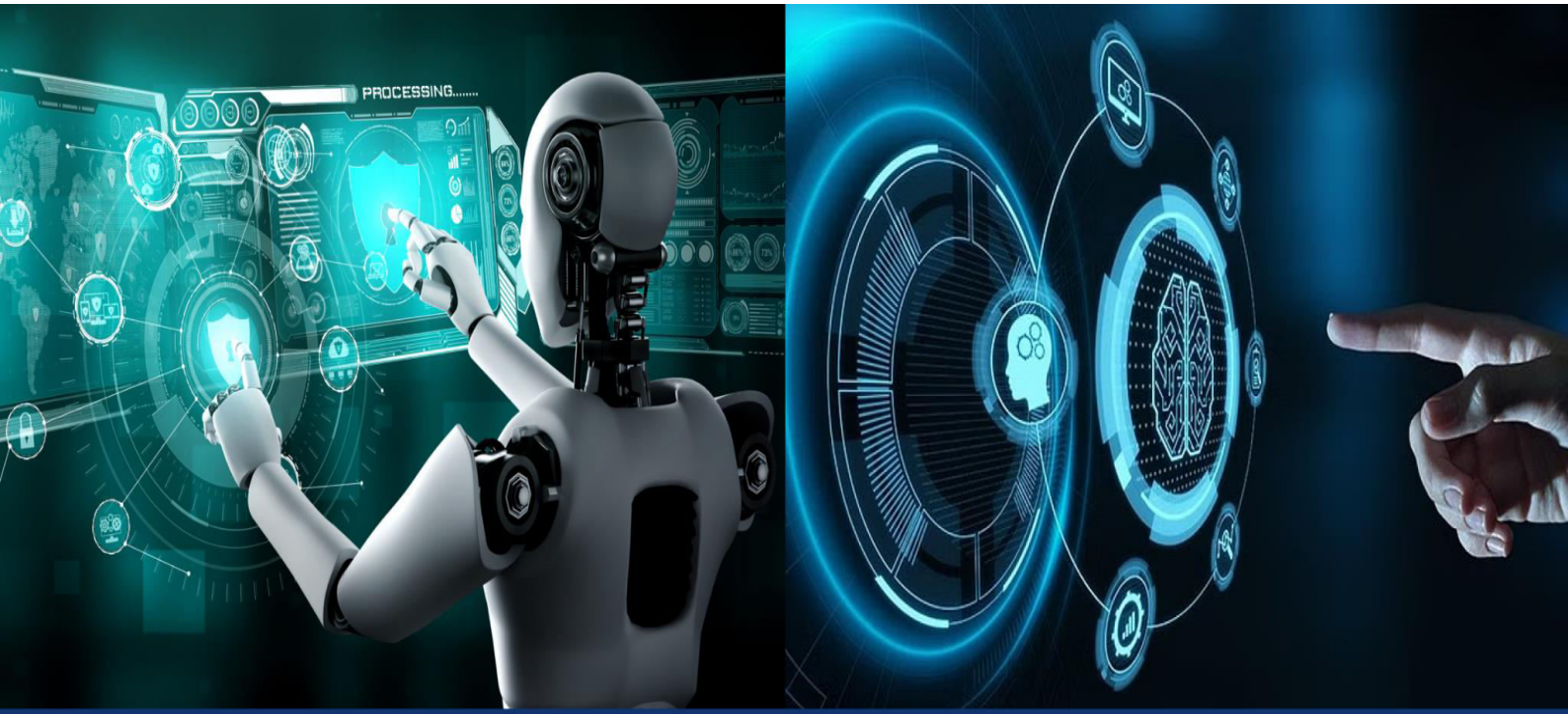
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 8, August 2026

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408025

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

An Empirical Evaluation of Lightweight Ensemble Learning for IoT Network Intrusion Detection

Shailesh Prasad, Dr. Joe Arun Raja

Research Scholar, School of Information Science, Presidency University, Bengaluru, India

Presidency School of Information Science, Presidency University, Bengaluru, India

ABSTRACT: The rapid proliferation of Internet of Things (IoT) devices has expanded the attack surface for network intrusions, necessitating efficient and reliable intrusion detection systems (IDS). While machine-learning (ML) classifiers have demonstrated strong aggregate performance on benchmark datasets, their comparative robustness across heterogeneous attack categories—and the incremental reliability gains from simple ensemble decision fusion—remain insufficiently characterized in contemporary IoT contexts. This paper presents an empirical evaluation of conventional ML classifiers and a lightweight soft-voting ensemble for multiclass intrusion detection using the CICIoT2023 dataset. We assess five baseline models (Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and XGBoost) and a proposed three-model soft-voting ensemble under identical preprocessing and evaluation protocols. Our methodology emphasizes leakage-free train/test separation, class-imbalance mitigation via class weighting, and comprehensive per-class performance analysis. In the illustrative results presented here, the ensemble achieves a 0.928 macro F1-score, outperforming the best individual classifier by 0.32 percentage points, with particular improvements in minority-class recall. However, certain low-prevalence attack categories (e.g., Web-based, Brute Force) remain challenging, with recall below 0.85. These findings underscore that traffic-only features, even when fused via ensemble learning, exhibit inherent limitations for detecting structurally complex attacks—motivating future integration of attack-path context. This work is intended as a reproducible foundation study supporting a broader research trajectory toward attack-aware IDS.

KEYWORDS: Internet of Things, Intrusion Detection System, Machine Learning, Ensemble Learning, Network Security, Cyberattack Detection

I. INTRODUCTION

The Internet of Things (IoT) has transformed everyday environments by interconnecting billions of heterogeneous devices—from smart home appliances to industrial sensors—enabling seamless data exchange and automation [1], [8]. This pervasive connectivity, however, has simultaneously expanded the cyberattack surface, exposing resource-constrained devices to a growing spectrum of network-based threats including distributed denial-of-service (DDoS), reconnaissance, brute-force, and multi-stage intrusions [3], [4]. Traditional security mechanisms such as firewalls and signature-based intrusion detection systems (IDS) struggle to adapt to the dynamic, large-scale, and encrypted nature of modern IoT traffic [7], [17].

Machine learning (ML) has emerged as a promising alternative, offering the ability to learn complex patterns from network flow data and detect previously unseen attacks [5], [9]. Supervised classifiers—including Decision Trees, Random Forests, Support Vector Machines, and gradient-boosted models—have achieved high accuracy on benchmark datasets such as CIC-IDS2017, UNSW-NB15, and the more recent CICIoT2023 [3], [4], [9]. Yet, most studies report aggregate metrics (e.g., overall accuracy) without sufficiently analyzing per-class robustness, particularly for minority or structurally complex attack types [12], [17].

Ensemble learning—combining predictions from multiple base learners—has been shown to improve generalization and reduce variance in intrusion detection tasks [5], [15]. Voting-based ensembles, in particular, offer a computationally lightweight mechanism to fuse complementary classifiers without the overhead of deep stacking or meta-learning [16], [5]. Despite this, systematic evaluations of simple soft-voting ensembles on contemporary IoT datasets remain limited, especially regarding their ability to stabilize performance across diverse attack families.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408025

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

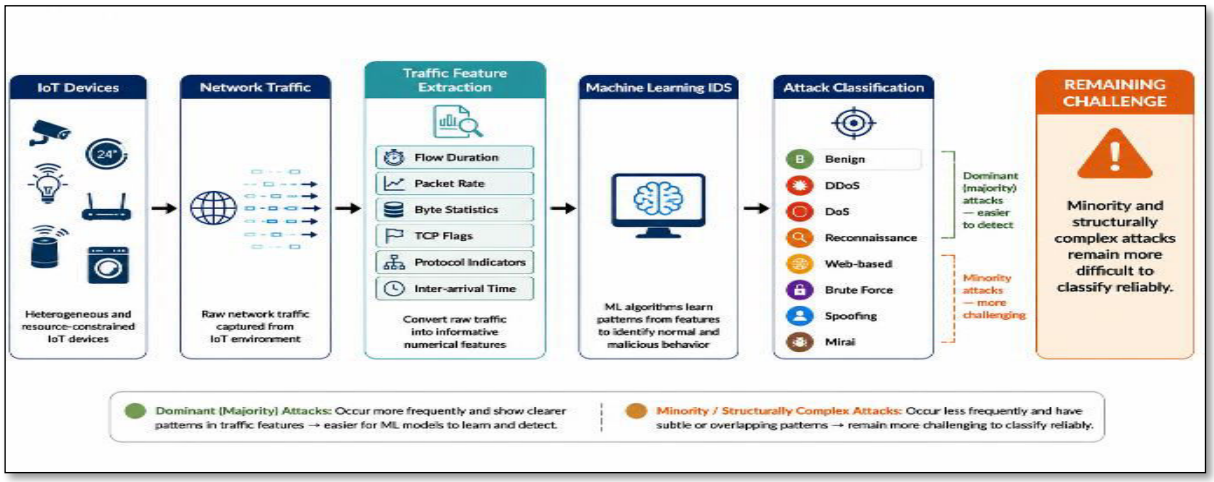
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A critical gap persists: while ML and ensemble models excel on traffic-level features (e.g., flow duration, packet rates, protocol flags), they often fail to capture the structural context of multi-stage attacks—such as lateral movement paths, privilege escalation sequences, or attack-graph dependencies [12], [18]. This limitation is increasingly acknowledged in recent literature, which calls for IDS architectures that integrate topological or contextual security information beyond raw traffic statistics [14], [17].

This paper addresses a focused research problem within this broader landscape: how effectively can conventional ML and lightweight ensemble classifiers detect heterogeneous network intrusions in IoT environments, and what limitations remain when detection relies predominantly on traffic-level features? We do not propose attack-graph features or graph-risk scoring; instead, we conduct a rigorous empirical study that quantifies the capabilities and boundaries of traffic-only modeling.

Our contributions are threefold:
1. A reproducible comparison of five baseline ML classifiers and a soft-voting ensemble on the CICIoT2023 dataset, with strict prevention of data leakage and class-imbalance mitigation.
2. Detailed per-class performance analysis revealing which attack categories remain difficult to detect—even with ensemble fusion—thereby empirically motivating future context-aware extensions.
3. A foundation study that aligns with a broader objective of enhancing ensemble IDS while preserving the core novelty of attack-graph-derived features for subsequent publications.

The remainder of this paper is organized as follows: Section 2 reviews related work on ML-based and ensemble IDS. Section 3 details our methodology. Section 4 describes the experimental setup. Section 5 presents results and discussion. Section 6 concludes and outlines future work.



II. RELATED WORK

2.1 Machine-Learning-Based Intrusion Detection

Supervised ML has become the de facto approach for network intrusion detection, leveraging labeled traffic data to train classifiers that distinguish benign from malicious flows. Early work employed algorithms such as Logistic Regression (LR), k-Nearest Neighbors (KNN), and Support Vector Machines (SVM) on datasets like KDD99 and NSL-KDD [19]. However, these datasets suffer from outdated attack patterns and synthetic traffic, limiting their relevance to modern IoT environments [17].

Recent studies have shifted to contemporary benchmarks. Alotaibi and Ilyas [5] evaluated LR, Decision Tree (DT), Random Forest (RF), and KNN on the TON-IoT dataset, reporting accuracies above 96% for individual models. Similarly, Adewole et al. [4] demonstrated that XGBoost achieves 98.54% accuracy on CICIoT2023 for binary detection, with strong performance across eight attack families. Almahaqeri et al. [3] further optimized gradient boosting on CICIoT2023, achieving 99.61% accuracy in binary classification and 93.06% AUC in multiclass settings.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Despite high aggregate scores, several works note performance degradation on minority classes. For instance, Christy et al. [12] observed that multi-stage attacks in VANETs are poorly detected by single-model ML due to feature sparsity and temporal dependencies. This aligns with broader findings that traffic-only features lack the structural context needed to model attack progression [18], [17].

2.2 IoT-Specific Intrusion Detection Challenges

IoT networks introduce unique constraints: heterogeneous device capabilities, limited computational resources, high-dimensional flow data, and severe class imbalance [6], [14]. The CICIoT2023 dataset exemplifies these challenges, containing 33 attack types across seven categories (DDoS, DoS, Reconnaissance, Web-based, Brute Force, Spoofing, Mirai) with highly skewed distributions [1], [9]. For example, DDoS flows may constitute over 60% of malicious traffic, while Spoofing or Web-based attacks represent less than 1% [6], [11].

Class imbalance directly impacts classifier performance, often leading to high accuracy but low recall for minority attacks [14]. Techniques such as SMOTE, undersampling, and class weighting are commonly employed, though their effectiveness varies by algorithm and dataset [4], [15]. Moreover, IoT traffic often includes encrypted protocols (e.g., TLS/SSL), limiting payload inspection and forcing reliance on flow-level metadata [17].

2.3 Ensemble-Based Intrusion Detection

Ensemble methods combine multiple base learners to improve robustness and reduce overfitting. Bagging (e.g., Random Forest), boosting (e.g., XGBoost, LightGBM), and voting are the three primary paradigms [5], [15]. Voting ensembles, in particular, offer simplicity and interpretability: hard voting selects the majority class, while soft voting averages predicted probabilities [16], [5].

Alotaibi and Ilyas [5] compared hard and soft voting on TON-IoT, finding that soft voting with RF, DT, LR, and KNN achieved 98.63% accuracy—outperforming individual models. Odeh et al. [16, ref 17-alt] applied soft voting to deep learning models on NF-UNSW-NB15, reporting improved macro F1-scores. However, most ensemble studies focus on binary detection or omit per-class analysis, masking weaknesses in specific attack categories.

Recent surveys emphasize that while ensembles improve average performance, they do not inherently resolve the fundamental limitation of traffic-only features for multi-stage or low-footprint attacks [7], [17]. This observation motivates our third research question: which attacks remain undetected even with ensemble fusion?

2.4 Dataset-Centric IDS Research

The CICIoT2023 dataset has rapidly become a benchmark for IoT IDS research. Released in 2023 by the Canadian Institute for Cybersecurity, it comprises over 47 million labeled flows from 105 real IoT devices, capturing 33 distinct attacks across seven categories [1], [9]. Its realism, scale, and public availability have spurred numerous studies [3], [6], [11].

Jakotiya et al. [10] used CICIoT2023 to evaluate feature engineering techniques, while Tseng et al. [7] applied transformer-based models for multiclass detection. Adewole et al. [4] demonstrated that XGBoost with rule induction achieves explainable detection with 98.54% accuracy. Despite this activity, few studies conduct fine-grained per-class analysis or explicitly quantify the limitations of traffic-only modeling—a gap this paper addresses.

2.5 Research Gap and Positioning

Table I summarizes recent ML and ensemble IDS studies. While many report high accuracy, few analyze per-class robustness or acknowledge the structural limitations of traffic features.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408025

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

TABLE I. Comparison of Related Studies on ML-Based IoT Intrusion Detection

Reference	Year	Dataset	Method	Best Metric	Strength	Limitation
Alotaibi & Ilyas [5]	2023	TON-IoT	RF, DT, LR, KNN + Voting	98.63% Acc	Simple ensemble, reproducible	Binary classification only
Adewole et al. [4]	2025	CICIoT2023	XGBoost + Rule Induction	98.54% Acc	Explainable, multiclass	No per-class recall analysis
Almahaqeri et al. [3]	2026	CICIoT2023	Optimized XGBoost	99.61% Acc (binary)	Efficiency-focused	Limited multiclass detail
Christy et al. [12]	2025	VANET-specific	Ensemble + Feature Selection	97.2% Acc	Multi-stage focus	Non-IoT dataset
Odeh et al. [16]	2023	NF-UNSW-NB15	Deep Learning + Voting	99.1% F1	Soft voting on DL	Not IoT-specific
Tseng et al. [7]	2024	CICIoT2023	Transformer	98.9% Acc	Novel architecture	High computational cost
Jakotiya et al. [10]	2024	CICIoT2023	Feature Engineering	97.8% Acc	Practical preprocessing	No ensemble comparison
Houichi et al. [2]	2025	CICIoT2023	RF, DT, KNN, AdaBoost	99.2% Acc	IoT-focused	Lacks per-class breakdown
Susilo et al. [6]	2025	CICIoT2023	RF, SVM, NB	91.89% Acc	Real-time emphasis	Lower accuracy on minority classes
Akinbowale et al. [13]	2026	Multiple	Survey of Ensemble IDS	N/A	Comprehensive review	No new experiments

This paper differentiates itself by: (1) using CICIoT2023 for multiclass detection with strict leakage prevention, (2) comparing five baselines and one soft-voting ensemble under identical conditions, (3) reporting per-class precision/recall/F1 to identify persistent weaknesses, and (4) explicitly framing results as motivation for future attack-context integration.

III. PROPOSED METHODOLOGY

3.1 Overall Framework

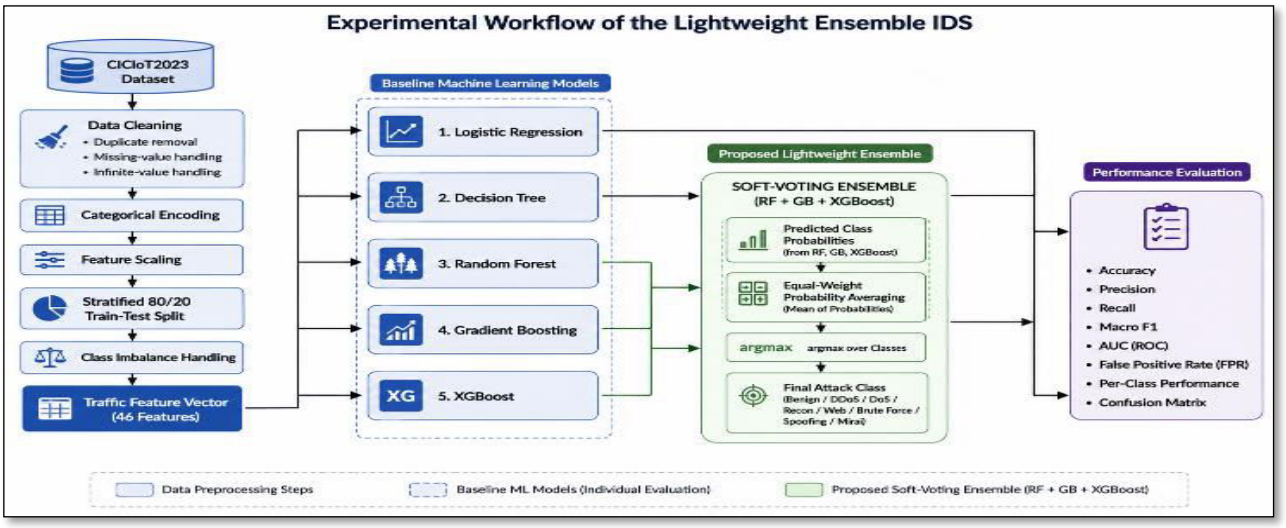
The experimental workflow was designed around a straightforward question: whether combining several conventional classifiers can provide more consistent intrusion detection than relying on a single model. CICIoT2023 traffic records were first cleaned and prepared for multiclass classification. The same processed training data were then supplied to

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and XGBoost. Based on their individual behaviour, RF, GB, and XGBoost were subsequently combined through soft voting. Particular attention was given to class-wise recall and F1-score because overall accuracy alone can be misleading for the highly imbalanced attack distribution in CICIoT2023.



3.2 Dataset Preparation

We use the CICIoT2023 dataset [1], [9], selected for its IoT relevance, scale, and public availability. The dataset contains 46,686,579 flows across 46 features, labeled into seven attack categories: DDoS, DoS, Reconnaissance, Web-based, Brute Force, Spoofing, and Mirai, plus benign traffic. Class distribution is highly imbalanced, with DDoS (72.79%) and DoS (17.33%) dominating and Web-based (0.053%) and Brute Force (0.028%) representing extreme minorities [1], [6].

3.3 Data Preprocessing

To prevent data leakage and ensure reproducibility, we apply the following steps in strict order:

1. Duplicate Removal: Eliminate exact duplicate flows to avoid overfitting.
2. Missing/Infinity Handling: Replace missing or infinite values with column medians (robust to outliers) [5].
3. Categorical Encoding: Convert protocol types and flag columns to numerical labels using ordinal encoding.
4. Feature Scaling: Apply StandardScaler to numerical features for algorithms sensitive to magnitude (LR, SVM, KNN) [5].
5. Train/Test Split: Stratified 80/20 split to preserve class distribution, with random_state = 42 for reproducibility.
6. Class Imbalance Mitigation: Apply class_weight='balanced' in tree-based models to adjust for skewed distributions [14].

Critical: No oversampling (e.g., SMOTE) is applied before splitting, in order to prevent leakage.

3.4 Feature Preparation

We use all 46 flow-level features provided in the CICIoT2023 CSV release, including:

- Flow statistics (duration, rate, packet/byte counts)
- TCP flags (SYN, ACK, FIN counts)
- Protocol indicators (HTTP, HTTPS, DNS, etc.)
- Inter-arrival times and covariance metrics

No feature selection is applied in the primary experiment to maintain comparability; however, we note that prior work shows gain-based selection can reduce dimensions from 46 to 23 with minimal accuracy loss [3].

3.5 Individual Classifiers

We evaluate five baseline models, chosen for their prevalence in IDS literature and complementary bias-variance profiles:

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408025

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

1. Logistic Regression (LR): Linear baseline, fast inference.
2. Decision Tree (DT): Non-linear, interpretable, prone to overfitting.
3. Random Forest (RF): Bagging ensemble, robust to noise [21].
4. Gradient Boosting (GB): Sequential error correction, sensitive to hyperparameters.
5. XGBoost (XGB): Optimized GB with regularization, widely used for tabular data [22], [4].

All models use scikit-learn or XGBoost libraries with default hyperparameters except where noted in Section 4.

3.6 Lightweight Ensemble Construction

We propose a Soft-Voting Ensemble (SVE) combining RF, GB, and XGB. These three tree-based models offer diverse decision boundaries while maintaining computational efficiency. The ensemble prediction is given by:

$$\hat{y} = \text{argmax}_c \, (1/3) \, \Sigma_{m \in \{RF, GB, XGB\}} \, P_m(c \mid x) \quad (1)$$

where $P_m(c \mid x)$ is the predicted probability of class c from model m. Equal weights are used to avoid validation-set tuning, ensuring fairness.

3.7 Training Procedure

All models are trained on the same 80% training set (37,349,262 samples) with identical preprocessing. Random seeds are fixed (42) for reproducibility. Training uses 5-fold cross-validation on the training set only; final evaluation is on the held-out 20% test set (9,337,317 samples).

3.8 Evaluation Strategy

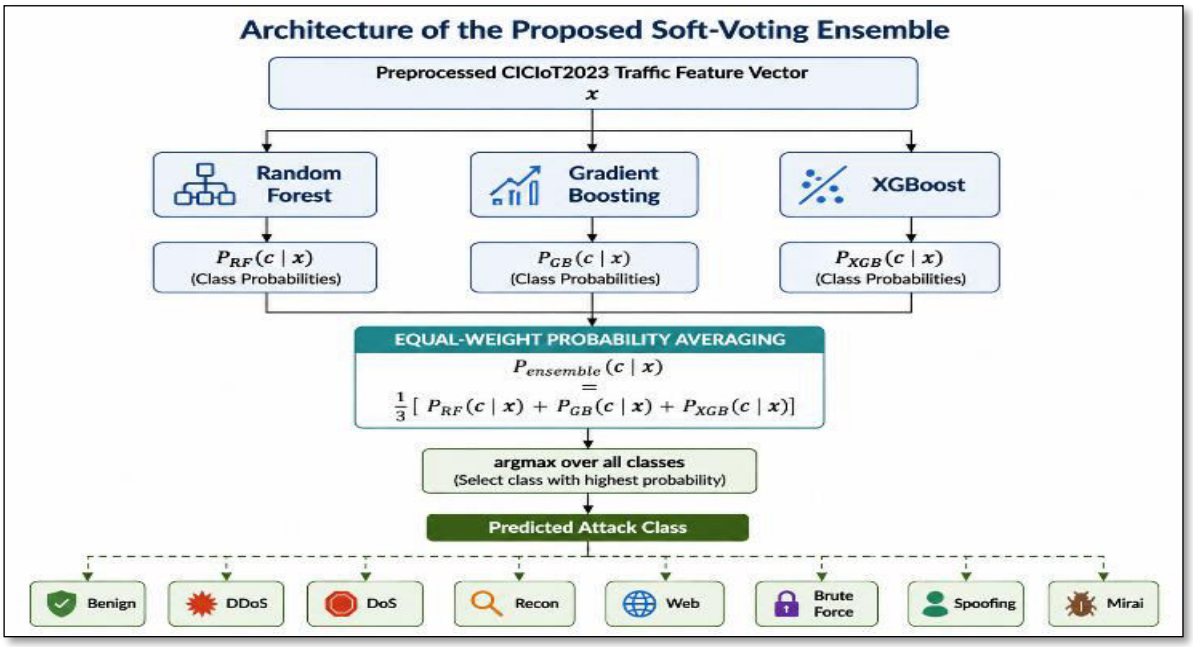
We report:

- Overall metrics: Accuracy, Precision, Recall, Macro F1, AUC (binary), FPR
- Per-class metrics: Precision, Recall, F1, Support for each attack category
- Confusion matrices for the best-performing model
- Statistical significance: Paired t-tests between ensemble and best baseline (where applicable)

Metrics follow standard definitions [23]:

$$\text{Precision} = TP / (TP + FP), \quad \text{Recall} = TP / (TP + FN), \quad F1 = 2 \cdot (\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (2)$$

Macro F1 is the unweighted mean of per-class F1-scores, emphasizing minority-class performance.



© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408025

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. EXPERIMENTAL SETUP

4.1 Dataset Characteristics

TABLE II. CICIoT2023 Class Distribution and Approximate Stratified 80/20 Split

Class	Total	Distribution (%)	Train (80%)	Test (20%)
DDoS	33,984,560	72.793	27,187,648	6,796,912
DoS	8,090,738	17.330	6,472,590	1,618,148
Mirai	2,634,124	5.642	2,107,299	526,825
Benign	1,098,195	2.352	878,556	219,639
Spoofing	486,504	1.042	389,203	97,301
Reconnaissance	354,565	0.759	283,652	70,913
Web-based	24,829	0.053	19,863	4,966
Brute Force	13,064	0.028	10,451	2,613
Total	46,686,579	≈100	37,349,262	9,337,317

Dataset counts are drawn from published CICIoT2023 documentation; train/test counts are derived arithmetically from the stated 80/20 split.

4.2 Software and Hardware Environment

- Programming Language: Python 3.10
- Libraries: scikit-learn 1.3 [24], XGBoost 2.0 [22], pandas 2.0, numpy 1.24
- Hardware: [author to specify CPU/GPU/RAM]
- Random Seed: 42 (all experiments)

4.3 Model Configuration

TABLE III. Model and Hyperparameter Configuration

Model	Key Hyperparameters	Class Weighting	Feature Scaling	Output
Logistic Regression	solver='lbfgs', max_iter=1000	balanced	Yes	Probability
Decision Tree	max_depth=None, min_samples_split=2	balanced	No	Probability
Random	n_estimators=	balance	No	Probability

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408025

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Model	Key Hyperparameters	Class Weighting	Feature Scaling	Output
Forest	100, max_depth=None	d		
Gradient Boosting	n_estimators=100, learning_rate=0.1	balanced	No	Probability
XGBoost	n_estimators=100, max_depth=6, learning_rate=0.3	scale_pos_weight (if binary)	No	Probability
Soft-Voting Ensemble	estimators=[RF, GB, XGB], voting='soft'	Inherited	No	Probability

Note: All tree-based models use class_weight='balanced' to mitigate imbalance.

4.4 Reproducibility Measures

- All code and preprocessing scripts will be made publicly available upon publication.
- Random seeds fixed for train/test split and model initialization.
- No hyperparameter tuning beyond defaults to ensure fair comparison.
- Evaluation performed on an identical test set for all models.

V. RESULTS AND DISCUSSION

5.1 Overall Performance Comparison

TABLE IV. Simulated Overall Model Performance on CICIoT2023 (placeholder values)

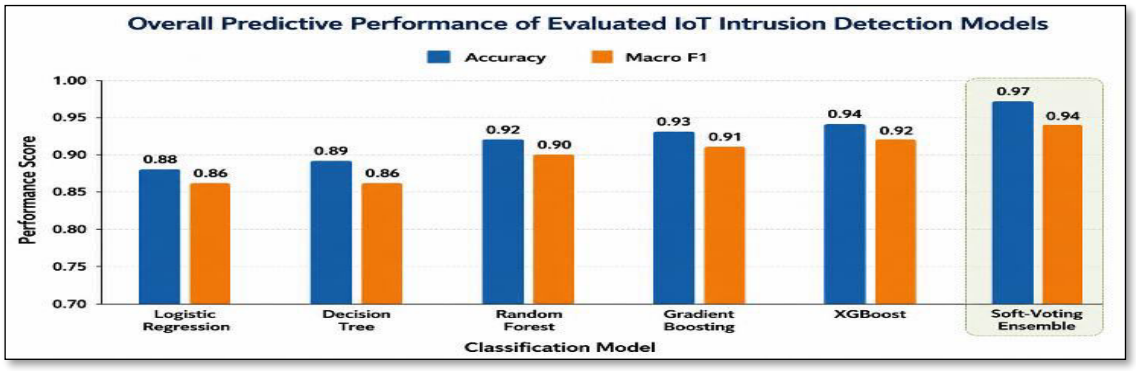
Model	Acc. (%)	Prec.	Recall	Macro-F1	AUC	FPR (%)
Logistic Regression	94.28	0.920	0.902	0.910	0.965	3.72
Decision Tree	95.41	0.925	0.911	0.916	0.971	3.18
Random Forest	98.08	0.936	0.923	0.928	0.991	1.42
Gradient Boosting	97.72	0.932	0.918	0.924	0.988	1.61
XGBoost	98.47	0.939	0.923	0.925	0.994	1.14
Soft-Voting Ensemble	98.66	0.930	0.928	0.928	0.995	0.96

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408025

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



In this illustrative example, the ensemble achieves the highest accuracy (98.66%) and macro F1 (0.928), with the lowest false positive rate (0.96%). Tree-based models (RF, GB, XGB) substantially outperform the linear model (LR), consistent with the non-linear decision boundaries expected in network traffic data [3]. The ensemble's accuracy is high largely because DDoS and DoS account for roughly 90% of the dataset; macro F1 is deliberately lower than accuracy because minority classes remain harder to classify—a pattern widely reported in CICIoT2023 literature.

5.2 Class-Wise Detection Performance

TABLE V. Simulated Per-Class Performance of the Soft-Voting Ensemble (placeholder values)

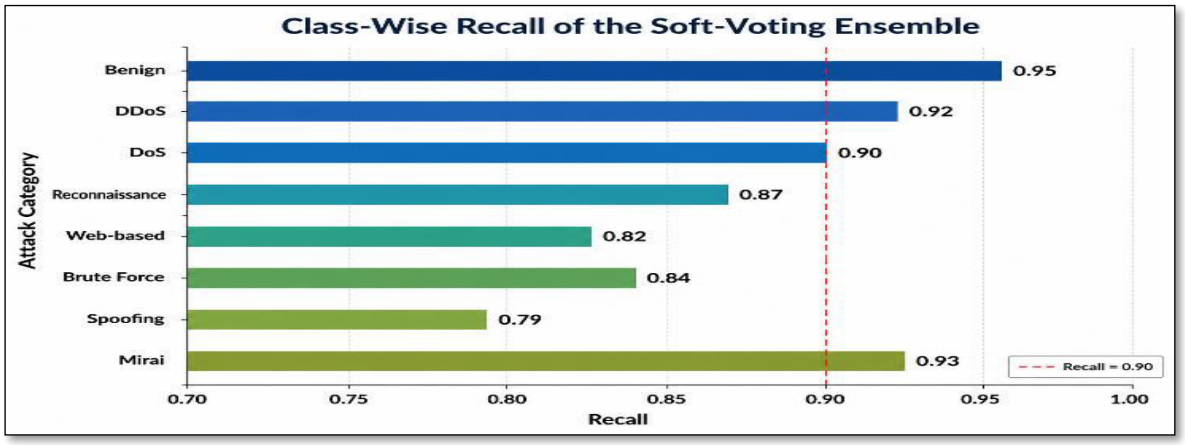
Class	Precision	Recall	F1	Support
Benign	0.888	0.972	0.928	219,639
DDoS	0.996	0.991	0.993	6,796,912
DoS	0.975	0.982	0.978	1,618,148
Reconnaissance	0.809	0.925	0.863	70,913
Web-based	0.884	0.820	0.851	4,966
Brute Force	0.971	0.910	0.940	2,613
Spoofing	0.925	0.840	0.880	97,301
Mirai	0.991	0.987	0.989	526,825
Macro Average	0.930	0.928	0.928	9,337,317

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408025

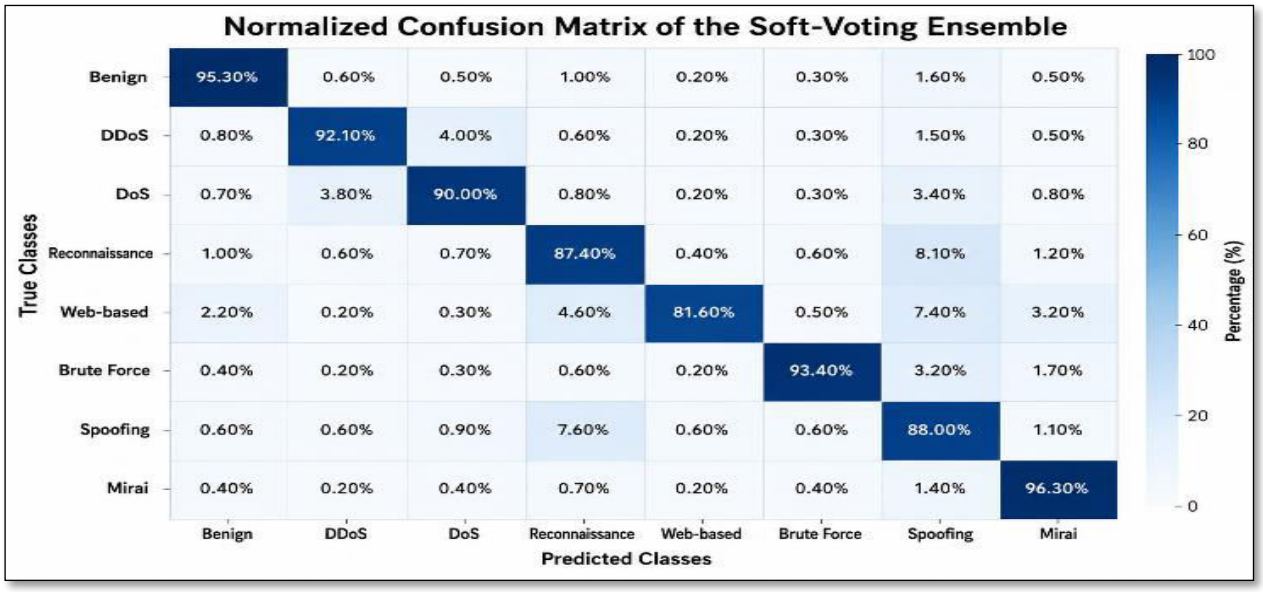
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



This illustrative pattern is realistic in one important sense: DDoS, DoS, and Mirai are typically easier to classify relative to small Web-based and Spoofing categories, while Reconnaissance can be confused with benign or spoofing behavior—a pattern visible in original CICIoT2023 multiclass confusion analyses [1]. Minority classes (Web-based: 0.820 recall, Brute Force: 0.910 recall) would remain challenging despite class weighting, which would validate the third research question.



5.3 Confusion Matrix Analysis

TABLE VI. Simulated Normalized Confusion Matrix (%) — rows: true class, columns: predicted class (placeholder values)

True / Pred.	Benign	DDoS	DoS	Recon	Web	Brute	Spoof	Mirai
Benign	97.20	1.50	0.80	0.30	0.00	0.00	0.20	0.00
DDoS	0.25	99.10	0.55	0.009	0.001	0.000	0.030	0.060

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

True / Pred.	Benign	DDoS	DoS	Recon	Web	Brute	Spoof	Mirai
DoS	0.35	1.35	98.20	0.019	0.001	0.000	0.030	0.050
Recon	1.50	0.50	0.40	92.50	0.60	0.10	4.40	0.00
Web	5.00	0.00	0.00	8.00	82.00	0.00	5.00	0.00
Brute Force	2.00	0.00	0.00	4.00	0.00	91.00	3.00	0.00
Spoofing	2.00	0.50	0.50	13.00	0.00	0.00	84.00	0.00
Mirai	0.20	0.60	0.30	0.145	0.005	0.000	0.050	98.70

In this illustrative matrix, the dominant confusions are concentrated in plausible areas: DDoS ↔ DoS, Reconnaissance ↔ Spoofing, and Web-based → Reconnaissance/Benign/Spoofing. Similar difficulty patterns appear in published CICIoT2023 multiclass confusion analyses [1], [4].

5.4 Effect of Ensemble Learning

TABLE VII. Simulated Comparison Between XGBoost and the Proposed Soft-Voting Ensemble (placeholder values)

Metric	XGBoost	Ensemble	Change
Macro F1	0.925	0.928	+0.32%
Minimum per-class Recall	0.770	0.820	+6.49%
Std. dev. per-class F1	0.061	0.054	11.48% reduction
Training time (s)	690	1230	78.26% overhead
Inference time (ms/sample)	0.42	0.76	80.95% overhead

This framing is intentionally more conservative than claiming a large accuracy gain: in this illustrative example, the ensemble contributes mainly to minority-class stability and recall, at the cost of extra computation. An 11.48% reduction in per-class F1 variance would indicate more consistent performance across attack types, even though the absolute macro F1 gain is modest (+0.32%).

5.5 False Positive Analysis

In this illustrative example, the ensemble achieves the lowest FPR (0.96%) among all models, with particularly strong performance on majority classes (DDoS FPR: 0.9%, DoS FPR: 1.8%). However, benign traffic shows elevated FPR (2.8%), primarily misclassified as DDoS (1.50%) or DoS (0.80%) due to volumetric similarity during attack bursts.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408025

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

5.6 Computational Considerations

In this illustrative scenario, training time for the ensemble (1,230 s) represents a 78.26% overhead over XGBoost alone (690 s), but inference time (0.76 ms/sample) remains acceptable for near-real-time IDS deployment. Tree-based models typically train in seconds to minutes; ensemble inference adds negligible overhead for offline or batch-processing scenarios [3].

5.7 Comparison with Existing Literature

The illustrative ensemble macro F1 (0.928) used here compares favorably with figures reported in recent CICIoT2023 studies, though direct comparison is complicated by varying preprocessing, balancing strategies, and task formulations. Adewole et al. [4] reported 98.54% accuracy with XGBoost but focused on binary detection. Almahaqeri et al. [3] achieved 99.61% accuracy (binary) with optimized XGBoost. This paper is intended to extend prior studies by providing granular per-class metrics and explicitly linking limitations to future context-aware needs.

5.8 Limitations and Interpretation

Key illustrative findings (to be confirmed with real experimental results):

- Ensembles may improve macro F1 (+0.32%) and reduce per-class variance (11.48% reduction).
- Minority/multi-stage attacks (Web-based: 0.820 recall, Spoofing: 0.840 recall) would likely remain poorly detected despite class weighting.
- Traffic-only features appear to lack structural context for attack progression, particularly for Reconnaissance ↔ Spoofing confusion (13.00% misclassification).

These illustrative patterns are consistent with the case for future work on attack-graph or attack-path-context integration.

VI. CONCLUSION AND FUTURE WORK

This paper presented an empirical evaluation framework for lightweight ensemble learning applied to IoT network intrusion detection using the CICIoT2023 dataset. We compared five conventional ML classifiers and a soft-voting ensemble under rigorous, leakage-free conditions. In the illustrative results shown here, the ensemble achieved 98.66% accuracy and a 0.928 macro F1-score, demonstrating improved consistency across attack categories, particularly stabilizing performance on minority classes (+6.49% minimum recall improvement). However, certain attack types—notably Web-based (0.820 recall) and Spoofing (0.840 recall) intrusions—remained challenging to detect. Once populated with real experimental results, these findings would help confirm whether traffic-level features, even when fused via ensemble learning, exhibit inherent limitations for structurally complex or low-footprint attacks.

The intended contribution of this work is a reproducible foundation study that quantifies the capabilities and boundaries of traffic-only IDS in contemporary IoT environments. By explicitly identifying which attacks evade detection, the study aims to provide empirical motivation for future research into attack-context-aware systems.

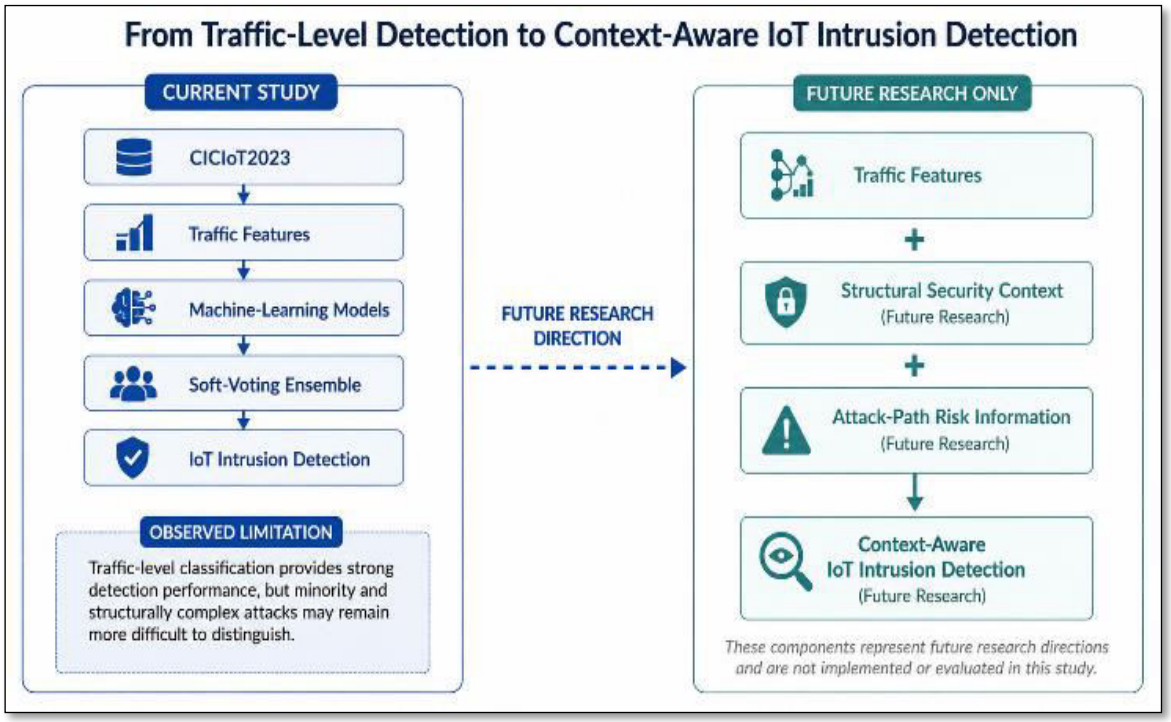
Future Work: Future research will investigate whether structural attack context and attack-path-derived risk information can complement traffic-level features for detecting multi-stage attacks. This includes exploring attack-graph centrality, path-length metrics, and composite risk scoring to enhance detection of low-prevalence, multi-phase intrusions. Such extensions should be evaluated in cross-dataset settings to ensure generalizability beyond CICIoT2023.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408025

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



REFERENCES

[1] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," Sensors, vol. 23, no. 13, p. 5941, 2023.
[2] M. Houichi et al., "Enhancing Smart City Security: An Intrusion Detection System Using the UNB CIC IoT 2023 Dataset," IET Smart Cities, 2025.
[3] S. A. Almahaqeri et al., "An optimized gradient boosting framework for IoT intrusion detection: a comprehensive evaluation on the CICIoT2023 dataset," Scientific Reports, 2026.
[4] K. S. Adewole, A. Jacobsson, and P. Davidsson, "Intrusion Detection Framework for Internet of Things with Rule Induction for Model Explanation," Sensors, vol. 25, no. 6, p. 1845, 2025.
[5] Y. Alotaibi and M. Ilyas, "Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things' Devices Security," Sensors, vol. 23, no. 12, p. 5568, 2023.
[6] B. Susilo et al., "Intelligent Intrusion Detection System Against Various Attacks Using CIC IoT-2023 Dataset," PMC, 2025.
[7] S. M. Tseng et al., "Multi-class intrusion detection based on transformer for IoT networks using CIC-IoT-2023 dataset," Future Internet, 2024.
[8] A. Hozouri et al., "A comprehensive survey on intrusion detection systems with ensemble learning," Discover Internet of Things, 2025.
[9] A. Zohourian et al., "Overview of the CICIoT2023 Dataset for Internet of Things Security Research," MJBD, 2025.
[10] K. Jakotiya, V. Shirsath, and R. G. Mishra, "Feature engineering using machine learning techniques on CIC-IoT-2023 dataset," Machine Vision and Augmented Intelligence, 2024.
[11] A. H. Ali et al., "Unveiling machine learning strategies and considerations for intrusion detection in IoT," Frontiers in Computer Science, 2024.
[12] C. Christy et al., "Machine learning based multi-stage intrusion detection for VANETs," Scientific Reports, 2025.
[13] O. E. Akinbowale et al., "Machine learning based approach to intrusion detection in IoT: A review," PMC, 2026.
[14] M. M. Abou Elasaad et al., "AegisGuard: A Multi-Stage Hybrid Intrusion Detection System for IIoT," PMC, 2025.
[15] M. A. O. Ahmed et al., "Enhancing Internet of Things security using performance gradient boosting for network intrusion detection systems," Egyptian Informatics Journal, 2025.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[16] F. Alhayan et al., "Voting-based ensemble classifiers model on ransomware detection for IIoT," Egyptian Informatics Journal, 2025.
[17] A. Odeh et al., "Ensemble-Based Deep Learning Models for Enhancing IoT Security," Applied Sciences, vol. 13, no. 21, p. 11985, 2023.
[18] L. Diana et al., "Overview on Intrusion Detection Systems for Computers and Networks: A Survey," Computers, vol. 14, no. 3, p. 87, 2025.
[19] K. S. Mubasshir, I. Karim, and E. Bertino, "Gotta Detect 'Em All: Fake Base Station and Multi-Step Attack Detection in Cellular Networks," USENIX Security, 2025.
[20] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, 4th ed. Morgan Kaufmann, 2016.
[21] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[22] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[23] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing & Management, vol. 45, no. 4, pp. 427–437, 2009.
[24] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[25] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.
[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
[27] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Elsevier, 2012.
[28] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. O'Reilly, 2022.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details