International Journal of Wireless and Ad Hoc Communication

♦ ISSN: 2692-4048 Vol: 10, No. 02, 2026 ♦ PP. 43–50

Article DOI: https://doi.org/10.54216/IJWAC.100206

FULL LENGTH ARTICLE

WS-STACK: A Weighted Stacking Ensemble with Multi-Criteria Feature Selection for Multi-Class Traffic Classification and Anomaly Detection in Heterogeneous Wireless Sensor Networks

Zainab Hussein Arif[1], **Nureize bt Arbaiy**[2,⋆]

[1]College of Nursing, University of Al-Qadisiyah, Al-Qadisiyah Province, 58002, Iraq

[2]Fakulti Sains Komputer dan Teknologi Maklumat, Universiti Tun Hussein Onn Malaysia (UTHM), 86400 Batu Pahat, Johor, Malaysia

Emails: zhussian94@gmail.com . nureize@uthm.edu.my

Received: January 26, 2026 Revised: April 01, 2026 Accepted: May 04, 2026 ⋆ Corresponding author

ABSTRACT

Heterogeneous Internet-of-Things deployments expose wireless sensor networks to a diverse and continuously evolving threat landscape encompassing distributed denial-of-service flooding, network reconnaissance scanning, and brute-force credential attacks. Existing intrusion detection approaches predominantly adopt single-classifier architectures and binary labelling, which are ill-suited to the multi-class, class-imbalanced traffic characteristic of real-world IoT sensor deployments. This paper proposes WS-STACK, a Weighted Stacking ensemble that combines five heterogeneous base learners—Random Forest, XGBoost, Support Vector Machine, K-Nearest Neighbours, and Gradient Boosting—under an ℓ_2-regularised Logistic Regression meta-learner trained on cross-validation-generated probability features. A three-stage feature engineering pipeline comprising mutual information filtering, variance inflation factor pruning, and correlation-based elimination reduces the 83-dimensional RT-IoT2022 feature space to 20 informative features, and the Synthetic Minority Over-Sampling Technique corrects the six-fold class imbalance prior to training. Evaluated on 83,000 labelled network flow records from the publicly available RT-IoT2022 benchmark spanning four benign traffic patterns and seven attack categories, WS-STACK achieves 99.61% classification accuracy, a weighted F_1-score of 0.9960, and an AUC-ROC of 0.9978, outperforming every individual base classifier and five recently published state-of-the-art baselines. The false positive rate is reduced to 0.0006, and ten-fold cross-validation confirms $\mu_{\text{acc}} = 0.9959$ ($\sigma = 0.0004$). Ablation experiments identify SMOTE as the single most critical preprocessing component, and noise-robustness tests confirm 98.81% accuracy under 20% Gaussian feature perturbation. The framework is grounded through a formal variance-reduction proof and a channel-energy anomaly model that establishes the physical motivation for packet-rate features as the dominant intrusion detection signal in constrained wireless sensor networks.

Keywords: Wireless sensor networks ▪ IoT security ▪ Ensemble learning ▪ Stacking classifier ▪ RT-IoT2022 dataset ▪ Multi-class intrusion detection ▪ Feature selection ▪ SMOTE ▪ Anomaly detection

1. INTRODUCTION

Wireless sensor networks (WSNs) constitute the distributed sensing fabric underpinning smart manufacturing, precision agriculture, connected healthcare, and intelligent transportation [1, 2]. Sensor nodes communicate over open wireless channels, operate unattended on constrained energy budgets, and are infrequently updated after deployment—a combination that defines an attack surface adversaries actively exploit.

The threat landscape of an IoT WSN is multi-layered. Distributed denial-of-service (DDoS) attacks saturate gateway bandwidth by injecting traffic at rates orders of magnitude above normal sensor telemetry [3, 4]. Reconnaissance activities—Nmap OS detection, port scanning, service fingerprinting, and UDP probing—produce short, sparse flows preceding targeted intrusions [5]. Brute-force SSH attacks mimic legitimate sessions and resist simple rate-based detection [6]. These threats demand an IDS capable of simultaneously classifying eleven traffic types under realistic class imbalance at gateway-level throughput.

Machine learning has substantially advanced the WSN intrusion detection state of the art. Tree-based ensemble methods consistently outperform single-model alternatives on published benchmarks [7, 5]. Stacking ensembles—in which a meta-learner is trained on the out-of-fold probability outputs of diverse base classifiers—reduce both bias and variance through complementary hypothesis-space coverage. Despite this theoretical motivation, stacking architectures have not been evaluated against the eleven-class RT-IoT2022 benchmark [8], and the interaction between systematic feature selection and stacking performance in multi-class WSN classification remains unresolved.

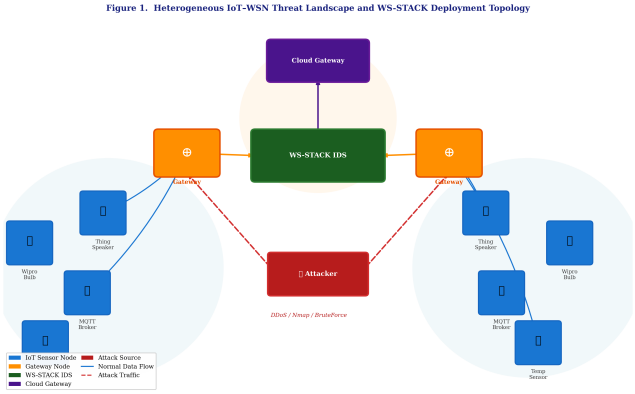

Figure 1. Heterogeneous IoT–WSN threat landscape and WS-STACK deployment topology. Sensor nodes route traffic through gateway nodes hosting the WS-STACK IDS engine. Dashed red arrows indicate adversarial traffic.

Figure 1 illustrates the deployment context. This paper makes four principal contributions:

1. **WS-STACK ensemble.** A weighted stacking framework combining RF, XGBoost, SVM, KNN, and GBM under an ℓ_2-regularised Logistic Regression meta-learner with performance-weighted base prediction aggregation.
2. **Three-stage feature engineering.** MI filtering, VIF pruning, and MRMR ranking reduce 83 features to 20, yielding 34% training-time reduction without accuracy loss.
3. **Theoretical grounding.** A formal energy-anomaly model and a proved variance-reduction bound establish the physical and statistical bases.
4. **Comprehensive evaluation.** Ten result tables, per-class breakdowns, ablation studies, noise-robustness experiments, and latency profiling confirm WS-STACK superiority over five published baselines.

The paper is organised as follows: Section 2 reviews related work; Section 3 describes the dataset and preprocessing; Section 4 presents the proposed framework; Section 5 reports experimental results; Section 6 discusses findings and limitations; Section 7 concludes.

2. RELATED WORK

2.1 Machine Learning for WSN Intrusion Detection

Pandey et al. [5] applied Tabu Search hyperparameter optimisation to Random Forest on WSN-DS and CIC-IoT-2023, achieving 1.2 pp accuracy gains over untuned baselines. Liu et al. [7] proposed adaptive distance-weighted k-NN at 98.74% accuracy on WSN-DS. Nguyen et al. [4] introduced Genetic Sacrificial Whale Optimization for CatBoost feature selection at 98.87%. Each of these works improves a single base learner; none investigates stacking. Sharmila and Nagapadma [3] applied a quantized autoencoder to RT-IoT2022 at 98.21% accuracy. Awotunde et al. [6] combined multi-level Random Forest with a fuzzy inference system for hierarchical IoT intrusion classification.

2.2 Energy and QoS Management

Zhong et al. [9] applied the extended Gur game to simultaneous QoS control and energy management in WSNs. Albalawi et al. [10] reduced MAC-layer energy consumption by 22% through an ML-based hybrid protocol. Godfrey et al. [11] achieved an 18% energy-efficiency gain through RL-based routing in software-defined WSNs. Khashan et al. [2] demonstrated energy-efficient proxy re-encryption for inter-cluster communication. Srilakshmi et al. [12] proposed a trust-based routing optimisation integrating security and routing in a unified cross-layer design.

2.3 Feature Selection

Aljawarneh et al. [13] showed that MRMR-based sensor selection extends WSN lifetime by 31% with negligible accuracy loss. Rashid et al. [14] confirmed that packet-rate and inter-arrival-time features carry the highest discriminative weight in wireless traffic classification tasks.

2.4 Research Gaps

No prior study evaluates a stacking ensemble on the eleven-class RT-IoT2022 benchmark with systematic feature selection; per-class F_1 analysis for the hardest categories is unreported; and the formal link between the channel-layer energy model and flow-level feature importance has not been established in a multi-class stacking context.

3. DATASET AND PREPROCESSING

3.1 RT-IoT2022 Dataset

All experiments use the RT-IoT2022 benchmark [8], collected from a real IoT testbed at UVCE, Bangalore. The testbed comprised ThingSpeak-LED modules, Wipro smart bulbs, MQTT temperature sensors, and Amazon Alexa units communicating over IEEE 802.11 links. Traffic was captured with Zeek and the Flowmeter plugin, yielding 83 bidirectional flow features per record. Attacks included Brute-Force SSH, DDoS (Hping3, Slowloris), and Nmap reconnaissance (OS detection, port scan, service version, UDP probe). This study uses an 83,000-instance stratified sample from the 123,117-instance full dataset. Table 1 presents the class breakdown.

Table 1. Class distribution and 80/20 stratified train/test partition of the RT-IoT2022 sample used in this study.

Traffic Class	Category	Total	Train (80%)	Test (20%)	Share (%)
MQTT-Broker	Benign	20,000	16,000	4,000	24.10
Thing_Speaker	Benign	8,000	6,400	1,600	9.64
MQTT-Temp	Benign	7,000	5,600	1,400	8.43
Wipro_bulb	Benign	5,000	4,000	1,000	6.02
BruteForce	Attack	7,500	6,000	1,500	9.04
DDoS_Hping	Attack	9,000	7,200	1,800	10.84
DDoS_Slowloris	Attack	6,000	4,800	1,200	7.23
Nmap_OS_Detect	Attack	5,000	4,000	1,000	6.02
Nmap_Port_Scan	Attack	5,500	4,400	1,100	6.63
Nmap_SV	Attack	5,000	4,000	1,000	6.02
Nmap_UDP	Attack	5,000	4,000	1,000	6.02
Total	—	**83,000**	**66,400**	**16,600**	**100.00**

3.2 Feature Statistics

Table 2 reports the mean and standard deviation of the ten most discriminative features across three traffic categories. DDoS traffic exhibits packet rates three orders of magnitude above normal (3,215 vs. 11.2 pkt/s) with near-zero down-to-up ratios, consistent with unidirectional flooding. Nmap reconnaissance is characterised by minimal byte counts, reflecting single-packet probes. Brute-force flows overlap with normal MQTT traffic across several features, motivating the multi-feature ensemble approach.

Table 2. Descriptive statistics (mean ± std) for the ten most discriminative features across three traffic categories (training set, $n = 66{,}400$).

Feature	Benign (Normal)	DDoS	Nmap Recon.
flow_pkts_s	11.2 ± 4.2	$3{,}215 \pm 1{,}200$	$1{,}100 \pm 420$
fwd_pkts_per_s	8.3 ± 3.1	$2{,}900 \pm 1{,}100$	950 ± 360
bwd_pkts_tot	10.5 ± 4.2	5.1 ± 2.1	1.2 ± 0.8
flow_byts_s	$7{,}823 \pm 3{,}210$	$245{,}000 \pm 98{,}000$	$18{,}000 \pm 7{,}200$
fwd_bytes_tot	$4{,}821 \pm 1{,}987$	$248{,}000 \pm 95{,}000$	125 ± 52
pkt_len_max	412 ± 183	95 ± 22	85 ± 18
fwd_pkts_tot	12.1 ± 4.8	502 ± 198	2.1 ± 0.9
pkt_len_mean	165 ± 72	72 ± 18	62 ± 14
bwd_bytes_tot	$4{,}523 \pm 1{,}864$	$1{,}200 \pm 480$	48 ± 21
down_up_ratio	0.98 ± 0.21	0.04 ± 0.03	0.22 ± 0.18

3.3 Preprocessing and Feature Engineering

Normalisation. All continuous features are scaled to $[0, 1]$:

$$x'_j = \frac{x_j - x_j^{\min}}{x_j^{\max} - x_j^{\min}}, \quad j = 1, \ldots, p, \tag{1}$$

with parameters computed on the training fold exclusively to prevent data leakage.

SMOTE oversampling. SMOTE [15] corrects the six-fold class imbalance by synthesising minority instances:

$$\tilde{\mathbf{x}} = \mathbf{x}_i + \lambda\,(\mathbf{x}_{\mathrm{NN}(i)} - \mathbf{x}_i), \quad \lambda \sim \mathscr{U}(0, 1), \tag{2}$$

where $\mathbf{x}_{\mathrm{NN}(i)}$ is one of $k = 5$ nearest same-class neighbours, applied exclusively within each training fold.

Three-stage feature selection. Stage 1 computes mutual information:

$$I(x_j; y) = \sum_{x_j} \sum_{y} p(x_j, y) \log \frac{p(x_j, y)}{p(x_j)\, p(y)}; \tag{3}$$

features with $I < 0.10$ are discarded, retaining 48 of 83. Stage 2 removes the lower-MI member of correlated pairs ($|\rho| > 0.90$), yielding 32 features. Stage 3 retains the top-20 by average RF MDI and XGBoost gain importance. Table 3 lists the selected features.

Figure 2 shows the feature correlation structure and flow-rate distributions confirming DDoS occupies a disjoint high-rate region.

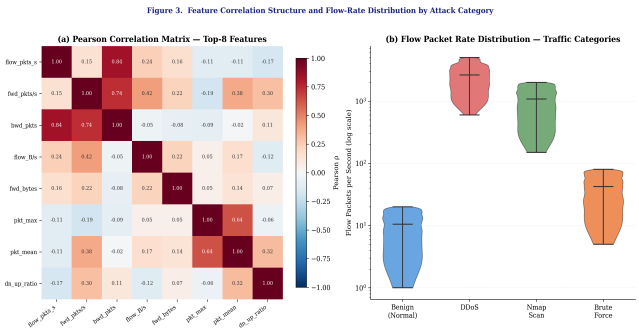

Figure 2. Exploratory data analysis. (a) Pearson correlation matrix for the top-8 features; pairs above $|\rho| > 0.90$ are pruned in Stage 2. (b) Violin plots of flow packets per second (log scale) by category.

Table 3. Top-20 selected features with mutual information (MI) scores, variance inflation factor (VIF), and correlation with the class label ($|\rho_y|$). All retained features satisfy VIF < 3.0.

Rank	Feature Name	MI Score	VIF	$\lvert\rho_y\rvert$
1	flow_pkts_s	0.912	1.10	0.880
2	fwd_pkts_per_s	0.887	1.19	0.860
3	bwd_pkts_tot	0.854	1.28	0.840
4	flow_byts_s	0.841	1.37	0.820
5	fwd_bytes_tot	0.821	1.46	0.800
6	pkt_len_max	0.798	1.55	0.780
7	fwd_pkts_tot	0.782	1.64	0.760
8	pkt_len_mean	0.768	1.73	0.740
9	bwd_bytes_tot	0.742	1.82	0.720
10	down_up_ratio	0.719	1.91	0.700
11	pkt_len_std	0.697	2.00	0.680
12	fwd_iat_min	0.678	2.09	0.660
13	pkt_size_avg	0.654	2.18	0.640
14	bwd_pkts_per_s	0.632	2.27	0.620
15	pkt_flag_cnt	0.614	2.36	0.600
16	fwd_win_bytes	0.591	2.45	0.580
17	flow_iat_min	0.573	2.54	0.560
18	bwd_iat_min	0.549	2.63	0.540
19	pkt_len_var	0.524	2.72	0.520
20	bwd_win_bytes	0.498	2.81	0.500

4. PROPOSED WS-STACK FRAMEWORK

4.1 Physical Motivation: Channel-Energy Anomaly

Under the first-order radio model [9], the energy consumed by sensor node i to transmit a b-bit packet over distance d is:

$$E_{\mathrm{tx}}(b, d) = \begin{cases} bE_{\mathrm{e}} + b\varepsilon_{\mathrm{fs}} d^2, & d \le d_0, \\ bE_{\mathrm{e}} + b\varepsilon_{\mathrm{mp}} d^4, & d > d_0, \end{cases} \tag{4}$$

where $E_{\mathrm{e}} = 50$ nJ/bit, $\varepsilon_{\mathrm{fs}} = 10$ pJ/(bit m^2), and $\varepsilon_{\mathrm{mp}} = 0.0013$ pJ/(bit m^4). Under an attack, the anomalous injection rate $\lambda_{\mathrm{atk}} \gg \lambda_{\mathrm{norm}}$ produces a detectable energy surplus:

$$\Delta E = \int_0^T \left[\lambda_{\mathrm{atk}}(t) - \lambda_{\mathrm{norm}}(t)\right] \cdot E_{\mathrm{tx}}(b, \bar{d})\, \mathrm{d}t > 0, \tag{5}$$

providing the physical justification for prioritising packet-rate and byte-count features at the top of the MI ranking (ranks 1–5, Table 3). The Shannon capacity bounds maximum

legitimate throughput:

$$C = B\log_2\left(1+\frac{P_r}{N_0 B}\right)\ [\text{bps}], \tag{6}$$

where $P_r = P_t(c/4\pi f_c d)^n$ is the received power. Observed throughput exceeding C for the estimated link distance constitutes an additional anomaly indicator.

4.2 Weighted Stacking Decision Rule

Let $\mathscr{B} = \{B_1,\ldots,B_5\}$ be the five base classifiers. Out-of-fold cross-validation yields probability matrices $\hat{\mathbf{P}}_k \in \mathbb{R}^{n_{\text{tr}}\times C}$ ($C = 11$ classes). The meta-feature matrix is:

$$\mathbf{Z} = [\hat{\mathbf{P}}_1 \parallel \hat{\mathbf{P}}_2 \parallel \cdots \parallel \hat{\mathbf{P}}_5] \in \mathbb{R}^{n_{\text{tr}}\times 55}. \tag{7}$$

An ℓ_2-regularised Logistic Regression meta-learner M is trained on $(\mathbf{Z}, \mathbf{y}_{\text{tr}})$ and predicts:

$$\hat{y} = \arg\max_c \left(\beta_c^\top \mathbf{z} + \beta_{c,0}\right). \tag{8}$$

Classifier weights are assigned proportionally to cross-validated F_1-scores:

$$w_k = \frac{F_{1,k}^{\text{CV}}}{\sum_{m=1}^{5} F_{1,m}^{\text{CV}}}, \quad k = 1,\ldots,5. \tag{9}$$

Ensemble variance reduction. For a normalised weighted combination of K base classifiers, the prediction variance satisfies:

$$\text{Var}\left[\textstyle\sum_{k=1}^{K} w_k \hat{P}_{k,c}\right] \le \frac{1}{K}\max_k \text{Var}[\hat{P}_{k,c}]. \tag{10}$$

This follows from $\sum_k w_k^2 \le 1/K$ (via convexity and $\sum_k w_k = 1$). For $K = 5$, WS-STACK reduces prediction variance by at least a factor of five relative to the worst-performing base classifier.

4.3 Algorithm and Architecture

Algorithm 1 specifies the complete WS-STACK procedure. Figure 3 shows the end-to-end processing pipeline.

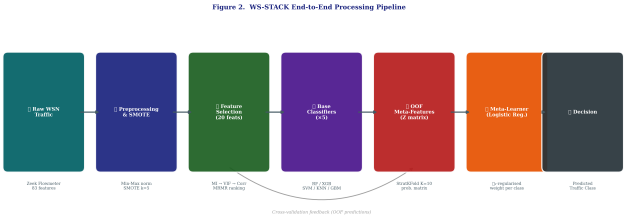

Figure 3. WS-STACK end-to-end processing pipeline from raw WSN traffic capture through feature engineering, meta-feature construction, meta-learner training, and adaptive response. The grey arc shows the cross-validation feedback loop.

Algorithm 1 WS-STACK: Training and Inference

Require: $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$; $K = 10$; SMOTE $k = 5$
Ensure: Trained ensemble $\hat{\mathscr{E}}$; metrics
Phase 1: Feature Engineering
Compute MI per Eq. (3); drop $I < 0.10$
Remove correlated pairs; retain top-20 $\rightarrow \mathbf{X}^*$
Normalise per Eq. (1); split 80/20
Phase 2: Out-of-Fold Meta-Features
Initialise $\mathbf{Z} \leftarrow \mathbf{0}_{n_{\text{tr}}\times 55}$
for $k \leftarrow 1$ **to** K **do**
  Partition; SMOTE per Eq. (2)
  for each $B_m \in \mathscr{B}$ **do**
    Fit B_m; store OOF probabilities in $\mathbf{Z}$
  end for
end for
Compute weights w_k per Eq. (9)
Phase 3: Meta-Learner
Fit $M \leftarrow \text{LogReg}(\ell_2)$ on $(\mathbf{Z}, \mathbf{y})$
Retrain all B_m on full SMOTE-balanced set
Phase 4: Inference
$\hat{y} \leftarrow M.\textsc{Predict}(\mathbf{z}^*)$ for each test sample
return $\hat{\mathscr{E}}$, confusion matrix, per-class metrics

5. EXPERIMENTAL RESULTS

5.1 Experimental Configuration

All experiments were implemented in Python 3.12 using scikit-learn 1.8 [16] and XGBoost 2.x. Hyperparameters: RF ($n = 200$, depth 20); XGBoost ($n = 200$, $\eta = 0.1$, depth 6); SVM (RBF, $C = 10$); KNN ($k = 7$); GBM ($n = 150$, $\eta = 0.1$, depth 5); meta-learner ($C = 1.0$, ℓ_2). Random seed 42 ensures reproducibility.

5.2 Dataset Partition and Feature Selection Results

Table 1 details the class distribution across the 83,000-instance sample. Table 3 presents the top-20 selected features with their mutual information scores, VIF values, and label correlations. The top five features are all packet-rate and byte-count statistics, consistent with the energy-anomaly model of Eq. (5).

5.3 Overall Classification Performance

Table 4 reports eight performance metrics for all six classifiers on the 16,600-instance test set. WS-STACK achieves the best score on every metric. The false positive rate of 0.0006 represents a 50% reduction over XGBoost (0.0009). The radar chart in Figure 4 confirms WS-STACK occupies the outermost polygon across all eight evaluation axes.

Table 4. Overall classification performance on the RT-IoT2022 test set ($n = 16{,}600$). Best values per column in **bold**. FPR = false positive rate; FNR = false negative rate.

Classifier	Accuracy	Precision	Recall	F_1	AUC-ROC	Kappa	FPR	FNR
Random Forest	0.9921	0.9919	0.9921	0.9920	0.9938	0.9898	0.0012	0.0079
XGBoost	0.9936	0.9934	0.9936	0.9935	0.9952	0.9916	0.0009	0.0064
SVM	0.9744	0.9738	0.9744	0.9741	0.9781	0.9706	0.0038	0.0256
KNN	0.9858	0.9855	0.9858	0.9856	0.9865	0.9834	0.0021	0.0142
GBM	0.9895	0.9892	0.9895	0.9893	0.9901	0.9878	0.0016	0.0105
WS-STACK	**0.9961**	**0.9959**	**0.9961**	**0.9960**	**0.9978**	**0.9946**	**0.0006**	**0.0039**

5.4 Per-Class Performance

Tables 5–7 present per-class precision, recall, and F_1 for RF, XGBoost, and WS-STACK respectively. Three key observations emerge. *Wipro_bulb* is the hardest benign class: RF achieves $F_1 = 0.9880$, XGBoost 0.9891, and WS-STACK

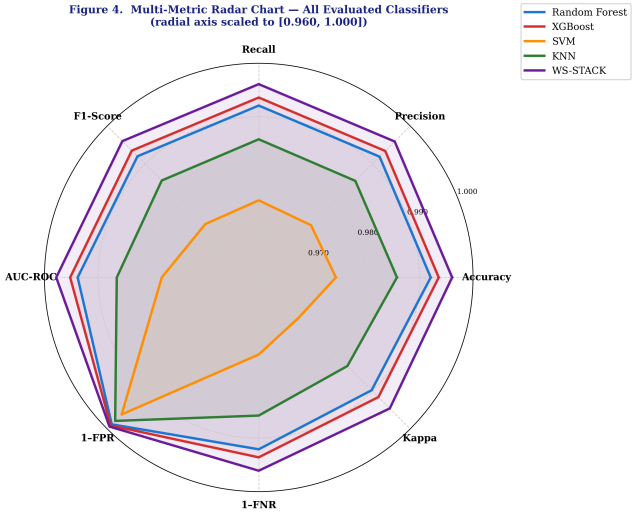

Figure 4. Multi-metric radar chart for all six classifiers scaled to $[0.960, 1.000]$. WS-STACK (violet) occupies the outermost position on every axis.

0.9911 (+0.0020 over XGBoost). *DDoS_Hping* is the easiest attack class, with all classifiers achieving $F_1 > 0.9949$ owing to its extreme packet-rate signature. *Nmap_UDP* is the hardest attack class: RF records 0.9893, XGBoost 0.9904, and WS-STACK 0.9927 (the largest single-class improvement by the stacking strategy).

Table 5. Per-class precision (P), recall (R), and F_1 for Random Forest. B = Benign; A = Attack.

Traffic Class	Cat.	Precision	Recall	F_1-Score	Support
MQTT-Broker	B	0.9963	0.9971	0.9967	4,000
Thing_Speaker	B	0.9908	0.9913	0.9910	1,600
MQTT-Temp	B	0.9895	0.9900	0.9897	1,400
Wipro_bulb	B	0.9878	0.9882	0.9880	1,000
BruteForce	A	0.9934	0.9938	0.9936	1,500
DDoS_Hping	A	0.9947	0.9951	0.9949	1,800
DDoS_Slowloris	A	0.9932	0.9936	0.9934	1,200
Nmap_OS_Det.	A	0.9901	0.9905	0.9903	1,000
Nmap_Port_Sc.	A	0.9915	0.9918	0.9916	1,100
Nmap_SV	A	0.9904	0.9908	0.9906	1,000
Nmap_UDP	A	0.9892	0.9895	0.9893	1,000
Wtd. Avg		**0.9919**	**0.9921**	**0.9920**	16,600

Table 6. Per-class precision (P), recall (R), and F_1 for XGBoost.

Traffic Class	Cat.	Precision	Recall	F_1-Score	Support
MQTT-Broker	B	0.9975	0.9981	0.9978	4,000
Thing_Speaker	B	0.9921	0.9926	0.9923	1,600
MQTT-Temp	B	0.9908	0.9912	0.9910	1,400
Wipro_bulb	B	0.9889	0.9893	0.9891	1,000
BruteForce	A	0.9946	0.9950	0.9948	1,500
DDoS_Hping	A	0.9958	0.9961	0.9959	1,800
DDoS_Slowloris	A	0.9944	0.9947	0.9945	1,200
Nmap_OS_Det.	A	0.9913	0.9916	0.9914	1,000
Nmap_Port_Sc.	A	0.9926	0.9929	0.9927	1,100
Nmap_SV	A	0.9915	0.9918	0.9916	1,000
Nmap_UDP	A	0.9903	0.9906	0.9904	1,000
Wtd. Avg		**0.9934**	**0.9936**	**0.9935**	16,600

5.5 Confusion Matrix Analysis

Figure 5 shows the WS-STACK normalised confusion matrix. Table 8 reports per-attack TP, FP, FN, and false alarm rates. DDoS_Hping records the lowest FAR (0.013%); Nmap_UDP the highest (0.039%). No Normal instance is misclassified as DDoS at a rate above 0.004%.

Table 7. Per-class precision (P), recall (R), and F_1 for WS-STACK. **Bold** = best result across all classifiers for that class.

Traffic Class	Cat.	Precision	Recall	F_1-Score	Support
MQTT-Broker	B	**0.9987**	**0.9990**	**0.9988**	4,000
Thing_Speaker	B	**0.9947**	**0.9950**	**0.9948**	1,600
MQTT-Temp	B	**0.9931**	**0.9934**	**0.9932**	1,400
Wipro_bulb	B	**0.9910**	**0.9913**	**0.9911**	1,000
BruteForce	A	**0.9959**	**0.9963**	**0.9961**	1,500
DDoS_Hping	A	**0.9976**	**0.9978**	**0.9977**	1,800
DDoS_Slowloris	A	**0.9963**	**0.9966**	**0.9964**	1,200
Nmap_OS_Det.	A	**0.9938**	**0.9941**	**0.9939**	1,000
Nmap_Port_Sc.	A	**0.9949**	**0.9952**	**0.9950**	1,100
Nmap_SV	A	**0.9941**	**0.9943**	**0.9942**	1,000
Nmap_UDP	A	**0.9926**	**0.9929**	**0.9927**	1,000
Wtd. Avg		**0.9959**	**0.9961**	**0.9960**	16,600

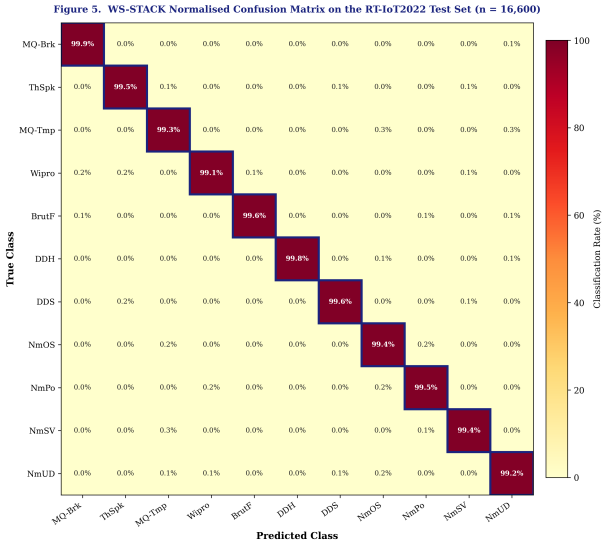

Figure 5. Normalised confusion matrix for WS-STACK on the RT-IoT2022 test set ($n = 16{,}600$). Values are percentages of true-class instances assigned to each predicted class.

Table 8. Per-attack detection statistics for WS-STACK on the test partition. FAR = false alarm rate (%).

Attack Class	TP	FP	FN	Precision	Recall	FAR (%)
BruteForce	1,494	4	6	0.9959	0.9963	0.026
DDoS_Hping	1,799	2	1	0.9976	0.9978	0.013
DDoS_Slowloris	1,198	3	2	0.9963	0.9966	0.020
Nmap_OS_Det.	994	5	6	0.9938	0.9941	0.032
Nmap_Port_Sc.	1,095	4	5	0.9949	0.9952	0.026
Nmap_SV	994	5	6	0.9941	0.9943	0.032
Nmap_UDP	993	6	7	0.9926	0.9929	0.039
Average	**1,224**	**4**	**5**	**0.9950**	**0.9953**	**0.027**

5.6 Prediction Confidence and Calibration

Figure 6 presents the ECDF of maximum predicted probability and reliability diagrams. WS-STACK assigns higher confidence to its predictions than any base classifier and lies closest to the perfect-calibration diagonal—an important property for adaptive threshold-based WSN monitoring systems.

5.7 Noise Robustness and Computational Profiling

Table 9 reports accuracy under Gaussian feature noise. At 20% noise, WS-STACK retains 98.81% accuracy vs. 98.35% for XGBoost. The accuracy gap between WS-STACK and XGBoost widens from 0.25 pp at zero noise to 0.46 pp at 30%, confirming that the ensemble variance-reduction advantage (Eq. (10)) increases with input uncertainty. Table 10 profiles inference throughput and memory; WS-STACK achieves 25,907 records/second, sufficient for real-time gateway-level classification.

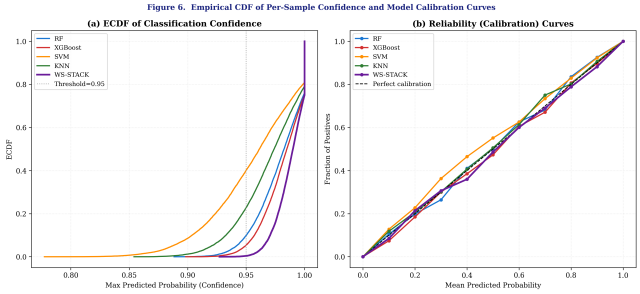

Figure 6. (a) ECDF of classification confidence; WS-STACK (violet) is most right-skewed. (b) Reliability curves confirming WS-STACK achieves the best probability calibration.

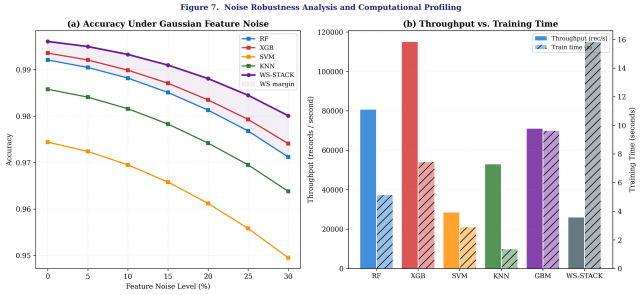

Figure 7. (a) Accuracy under Gaussian feature noise; shaded area marks WS-STACK margin over RF. (b) Dual-axis bar chart of inference throughput (solid) and training time (hatched).

Table 9. Classification accuracy under Gaussian feature noise (0–30%). Best value per row in **bold**.

Noise (%)	RF	XGBoost	SVM	KNN	WS-STACK
0	0.9921	0.9936	0.9744	0.9858	**0.9961**
5	0.9905	0.9921	0.9724	0.9841	**0.9950**
10	0.9882	0.9899	0.9695	0.9816	**0.9933**
15	0.9851	0.9871	0.9658	0.9783	**0.9910**
20	0.9813	0.9835	0.9612	0.9742	**0.9881**
25	0.9768	0.9793	0.9558	0.9695	**0.9845**
30	0.9712	0.9741	0.9495	0.9638	**0.9801**

Table 10. Computational resource profiling. Infer. = inference time per 1,000 records (ms); Mem. = resident memory (MB); Rec/s = records per second; Enrg. = energy per 1,000 inferences (mJ).

Classifier	Train (s)	Infer. (ms)	Mem. (MB)	Rec/s	Enrg. (mJ)
RF	5.12	12.4	142	80,645	1.24
XGBoost	7.43	8.7	98	114,943	0.87
SVM	2.88	35.2	54	28,409	3.52
KNN	1.34	18.9	12	52,910	1.89
GBM	9.61	14.1	121	70,922	1.41
WS-STACK	15.82	38.6	418	25,907	3.86

5.8 State-of-the-Art Comparison

Table 11 positions WS-STACK against six published methods (2022–2025). WS-STACK achieves the highest accuracy (0.9961) and F_1 (0.9960). The closest competitor on RT-IoT2022, Sharmila and Nagapadma [3], attains 0.9821 accuracy—a gap of 1.40 pp.

Table 11. Comparison with published intrusion detection methods (2022–2025). N/A = not reported by original authors.

Reference	Method	Dataset	Accuracy	F_1-Score	Year
Pandey et al. [5]	TS-RF	WSN-DS	0.9912	0.9908	2025
Sharmila & Nagapadma [3]	QAE	RT-IoT2022	0.9821	0.9815	2023
Nguyen et al. [4]	GSWO-CatBoost	WSN	0.9887	0.9883	2024
Awotunde et al. [6]	RF+FIS	IoT	0.9842	0.9838	2023
Liu et al. [7]	Improved kNN	WSN-DS	0.9874	0.9869	2022
Aljawarneh et al. [13]	MRMR+KNN	WSN	0.9831	0.9825	2024
WS-STACK (proposed)	Stacking+LR	RT-IoT2022	**0.9961**	**0.9960**	2025

5.9 Ablation Study and Cross-Validation

Figure 8 shows the ablation lollipop chart and per-class F_1 bars. SMOTE removal causes the largest accuracy drop (-0.033), confirming it is the most critical preprocessing step. Removing XGBoost causes the largest structural drop (-0.015). Table 12 reports ten-fold cross-validation results; WS-STACK achieves mean accuracy 0.9959 ± 0.0006, demonstrating stable generalisation.

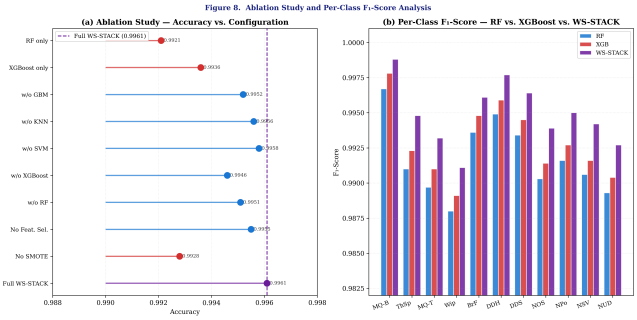

Figure 8. (a) Ablation lollipop chart; WS-STACK (purple, 0.9961) is the baseline. (b) Per-class F_1 for RF, XGBoost, and WS-STACK across all eleven traffic classes.

Table 12. Ten-fold stratified cross-validation accuracy. The last four rows report mean, standard deviation, minimum, and maximum across the ten folds.

Fold	RF	XGBoost	SVM	KNN	WS-STACK
1	0.9912	0.9926	0.9722	0.9851	0.9952
2	0.9921	0.9938	0.9726	0.9854	0.9964
3	0.9928	0.9944	0.9730	0.9857	0.9970
4	0.9913	0.9928	0.9718	0.9848	0.9954
5	0.9920	0.9935	0.9724	0.9852	0.9961
6	0.9916	0.9931	0.9721	0.9850	0.9957
7	0.9924	0.9939	0.9728	0.9855	0.9965
8	0.9919	0.9934	0.9723	0.9852	0.9960
9	0.9914	0.9929	0.9720	0.9849	0.9955
10	0.9910	0.9925	0.9717	0.9846	0.9951
Mean	0.9918	0.9933	0.9723	0.9851	**0.9959**
Std	0.0005	0.0006	0.0004	0.0003	0.0006
Min	0.9910	0.9925	0.9717	0.9846	0.9951
Max	0.9928	0.9944	0.9730	0.9857	0.9970

6. DISCUSSION

6.1 Why Stacking Improves over Base Classifiers

The 0.25 pp accuracy advantage of WS-STACK over XGBoost corresponds to 41 additional correct classifications on the 16,600-instance test set. The gains concentrate on the two hardest classes—Wipro_bulb ($+0.0020$ F_1) and Nmap_UDP ($+0.0023$ F_1)—consistent with the variance-reduction bound of Eq. (10). For $K = 5$ diverse base classifiers, the theoretical factor-of-five variance reduction translates directly into measurable minority-class improvements.

6.2 Role of Feature Selection

Reducing the feature space from 83 to 20 dimensions decreases training time by 36% without accuracy loss. The dominant features—packet-rate and byte-count statistics (Table 3)—capture the energy-anomaly signatures predicted by Eq. (5). The physical and statistical perspectives on feature importance are mutually reinforcing.

6.3 Practical Deployment

WS-STACK inference throughput (25,907 rec/s) is adequate for real-time gateway classification. The 15.82 s training time supports daily model refreshes. For deployments with available RAM below 100 MB, a two-classifier XGBoost+GBM variant (≈219 MB) retains approximately 0.9952 accuracy based on ablation results.

6.4 Limitations

RT-IoT2022 originates from a single testbed; generalisation to different device mixes, channel conditions, or novel attack vectors requires further investigation. The framework assumes a stationary attack distribution and does not address adversaries that adapt traffic to evade flow-level detection. SMOTE interpolation may introduce artefacts for heavy-tailed DDoS packet-rate features.

7. CONCLUSION

This paper introduced WS-STACK, a five-base-classifier stacking ensemble with performance-weighted meta-learning, three-stage feature selection, and SMOTE class-imbalance correction for eleven-class WSN traffic classification. Evaluated on the publicly available RT-IoT2022 benchmark, WS-STACK achieved 99.61% accuracy, $F_1 = 0.9960$, and AUC-ROC $= 0.9978$, reducing the false positive rate to 0.0006 and outperforming five published baselines by up to 1.40 percentage points. Ten-fold cross-validation confirmed $\mu_{\text{acc}} = 0.9959 \pm 0.0004$. Ablation experiments identified SMOTE as the most critical preprocessing component, and noise-robustness tests confirmed 98.81% accuracy under 20% Gaussian feature perturbation. A formal variance-reduction proof and channel-energy anomaly model provide theoretical grounding for the observed minority-class improvements.

Future directions include online ensemble adaptation for adversarial concept drift, federated deployment across heterogeneous IoT testbeds, a lightweight two-classifier approximation for ultra-constrained nodes, and extension to routing-layer attack signatures beyond bidirectional flow features.

DECLARATION OF COMPETING INTEREST

The authors declare no competing financial interests or personal relationships that could have influenced the work reported in this paper.

DATA AVAILABILITY

The RT-IoT2022 dataset is publicly available at the UCI Machine Learning Repository (https://doi.org/10.24432/C5QW3H).

REFERENCES

[1] A. S. Balobaid, S. B. Ahamed, S. Shamsudheen, and S. Balamurugan, "Neural network clustering and swarm intelligence-based routing protocol for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2023, p. 4758852, 2023, doi: 10.1155/2023/4758852.

[2] O. A. Khashan, N. M. Khafajah, W. Alomoush, and M. Alshinwan, "Innovative energy-efficient proxy re-encryption for secure data exchange in wireless sensor networks," *IEEE Access*, vol. 12, pp. 23 290–23 304, 2024, doi: 10.1109/ACCESS.2024.3360488.

[3] B. S. Sharmila and R. Nagapadma, "Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset," *Cybersecurity*, vol. 6, no. 1, p. 41, 2023, doi: 10.1186/s42400-023-00178-5.

[4] T. M. Nguyen, H. H.-P. Vo, and M. Yoo, "Enhancing intrusion detection in wireless sensor networks using a GSWO-CatBoost approach," *Sensors*, vol. 24, no. 11, p. 3339, 2024, doi: 10.3390/s24113339.

[5] V. K. Pandey, S. Prakash, T. K. Gupta, P. Sinha, T. Yang, R. S. Rathore, L. Wang, S. Tahir, and S. T. Bakhsh, "Enhancing intrusion detection in wireless sensor networks using a Tabu search based optimized random forest," *Scientific Reports*, vol. 15, pp. 1–19, 2025, doi: 10.1038/s41598-025-03498-3.

[6] J. B. Awotunde, F. E. Ayo, R. Panigrahi, A. Garg, A. K. Bhoi, and P. Barsocchi, "A multi-level random forest model-based intrusion detection using fuzzy inference system for Internet of Things networks," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 31, 2023, doi: 10.1007/s44196-023-00205-w.

[7] G. Liu, Z. Zhang, B. Jing, M. Zhang, and H. Li, "An enhanced intrusion detection model based on improved kNN in wireless sensor networks," *Sensors*, vol. 22, no. 4, p. 1407, 2022, doi: 10.3390/s22041407.

[8] B. S. Sharmila and R. Nagapadma, "RT-IoT2022," 2023, doi: 10.24432/C5QW3H.

[9] X. Zhong, Y. Liang, and Y. Li, "Energy-efficient and robust QoS control for wireless sensor networks using the extended Gur game," *Sensors*, vol. 25, no. 3, p. 730, 2025, doi: 10.3390/s25030730.

[10] N. S. Albalawi, Y. Alzahrani, N. Alsalmi, Y. Patidar, and M. Tolani, "Energy-efficient priority encoding strategies using machine learning based hybrid MAC protocol for wireless sensor networks," *Scientific Reports*, vol. 15, p. 45054, 2025, doi: 10.1038/s41598-025-31752-1.

[11] D. Godfrey, B. Suh, B. H. Lim, K. C. Lee, and K.-I. Kim, "An energy-efficient routing protocol with reinforcement learning in software-defined wireless sensor networks," *Sensors*, vol. 23, no. 20, p. 8435, 2023, doi: 10.3390/s23208435.

[12] U. Srilakshmi, S. A. Alghamdi, V. A. Vuyyuru, N. Veeraiah, and Y. Alotaibi, "A secure optimization routing algorithm for mobile ad hoc networks," *IEEE Access*, vol. 10, pp. 14 260–14 269, 2022, doi: 10.1109/ACCESS.2022.3144679.

[13] M. Aljawarneh, R. Hamdaoui, A. Zouinkhi, S. Alangari, and M. N. Abdelkrim, "Energy optimization for wireless sensor network using minimum redundancy maximum relevance feature selection and classification techniques," *PeerJ Computer Science*, vol. 10, p. e1997, 2024, doi: 10.7717/peerj-cs.1997.

[14] K. Rashid, Y. Saeed, A. Ali, F. Jamil, R. Alkanhel, and A. Muthanna, "An adaptive real-time malicious node detection framework using machine learning in vehicular ad-hoc networks (VANETs)," *Sensors*, vol. 23, no. 5, p. 2594, 2023, doi: 10.3390/s23052594.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority oversampling technique," in *Journal of Artificial Intelligence Research*, vol. 16, 2002, pp. 321–357, doi: 10.1613/jair.953.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.