

A Leakage-Safe Block-Disjoint Benchmark and Diversity-Driven Stacking Ensemble for Multi-Class IoT Botnet Traffic Classification: An Analytical and Empirical Study on the N-BaloT Dataset

Jeeva Arumugam

Gnanamani College of Technology

Nagarajan Ramalingam

nagarajanr@gnanamani.ac.in

Gnanamani College of Technology

Research Article

Keywords: IoT botnet detection, N-BaloT, data leakage, block-disjoint evaluation, gradient boosting, stacking ensemble, ambiguity decomposition, explainable AI

Posted Date: October 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11143924/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A Leakage-Safe Block-Disjoint Benchmark and Diversity-Driven Stacking Ensemble for Multi-Class IoT Botnet Traffic Classification: An Analytical and Empirical Study on the N-BaIoT Dataset

Jeeva Arumugam¹ and Nagarajan Ramalingam ^{2*}

¹Department of Artificial Intelligence and Data science, Gnanamani College of Technology, Namakkal, 637018, Tamilnadu, India.

^{2*}Department of Electrical and Electronics Engineering, Gnanamani College of Technology, Namakkal, 637018, Tamilnadu, India.

*Corresponding author(s). E-mail(s): nagarajanr@gnanamani.ac.in;
Contributing authors: jeevame107@gmail.com;

Abstract

Published benchmarks on the N-BaIoT dataset routinely report accuracies above 99.9%, obtained under random row-level train/test splits that ignore the fact that consecutive flow-statistics rows are temporally autocorrelated Kitsune features computed over overlapping decay windows; such splits leak information between partitions and compress the numerical space in which competing methods can be distinguished. This paper formalizes and empirically instantiates a block-disjoint evaluation protocol that provably removes this class of leakage: the raw N-BaIoT corpus (9 IoT devices, benign plus Mirai and Gafgyt/BASHLITE botnet traffic, 115 Kitsune statistical features) is partitioned into contiguous 500-row blocks, and every splitting decision – stratified sampling, grouped $\mathbf{K}$ -fold cross-validation, and the final held-out test partition – is defined and proved to operate at block granularity only. On a stratified 267,000-row sample (534 blocks) we pose an 11-class attack-subtype classification task and benchmark 10 classical learners and 4 deep architectures under 5-fold block-disjoint grouped cross-validation. A two-stage feature-selection procedure combining mutual information and Random Forest importance, followed by a correlation-based redundancy-pruning rule with a proved worst-case linear-predictability bound, reduces the 115 raw features to 40 with no loss of cross-validated Matthews correlation coefficient

(MCC). Under this protocol, linear models collapse (logistic regression MCC = **0.069**) while gradient-boosted tree ensembles remain strong (HistGradientBoosting MCC = **0.981**), a previously masked separation. We propose a diversity-driven stacking ensemble whose base-learner subset is selected by a joint disagreement-competence score, and we prove an ambiguity-decomposition theorem for convex-weighted probabilistic ensembles under multiclass Brier scoring that formally explains – rather than merely reports – why the fitted stacking meta-learner (nested-CV MCC = **0.989 ± 0.015**) is narrowly outperformed by its own single best constituent model on the final held-out partition (MCC **0.9995** vs. **0.9952**, paired bootstrap $p < 0.001$): the proof identity guarantees dominance only for simplex-constrained averaging, not for the data-fitted, non-simplex combiner actually used. Four formally stated algorithms, four theorems/lemmas/propositions with complete proofs, a nine-test statistical validation protocol, a five-axis robustness suite, cross-device and zero-day generalization tests, and SHAP/permutation/impurity explainability with cross-method rank agreement together provide a methodologically defensible, fully reproducible reference point for future N-BaIoT-based IoT security research.

Keywords: IoT botnet detection, N-BaIoT, data leakage, block-disjoint evaluation, gradient boosting, stacking ensemble, ambiguity decomposition, explainable AI

1 Introduction

The proliferation of interconnected devices across industrial and consumer Internet of Things (IoT) ecosystems has vastly expanded the attack surface for distributed botnet intrusions, necessitating robust network-level anomaly detection. This paper addresses the fundamental challenge of evaluation-induced data leakage and evaluates machine-learning defenses under rigorous, leakage-safe protocol guarantees. The remainder of this introductory section outlines the operational background and threat landscape, formalizes the evaluation leakage motivating this investigation, surveys pertinent literature, and delineates the primary analytical and empirical contributions of this work.

1.1 Background and threat model

The Industrial Internet of Things (IIoT) layer of modern manufacturing and consumer-device ecosystems is built from large populations of resource-constrained network-connected sensors and actuators that share two properties relevant to security research: they are numerous, and they are individually too constrained to host conventional endpoint defenses. This combination makes them the preferred recruitment target for IoT botnets. The Mirai botnet, which compromised hundreds of thousands of such devices in 2016 and used them to launch some of the largest denial-of-service attacks recorded at the time, remains the reference case for this threat class [1], and its Gafgyt/BASHLITE relatives continue to be observed in the wild. Because a compromised IoT device is usually unable to run a local intrusion-detection agent, the practical defense is network-side classification of flow statistics as benign or as one

of a set of known attack behaviors – the task formalized in Section 3 and addressed empirically in this paper on the N-BaIoT dataset [2].

1.2 Problem statement and evaluation-leakage motivation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ denote the row-level N-BaIoT sample, where each $x_i \in \mathbb{R}^{115}$ is a vector of Kitsune flow statistics computed over five decayed aggregation windows and y_i is an attack-subtype label. Because the Kitsune extractor maintains a running decayed mean/variance/covariance per window, two temporally adjacent rows x_i, x_{i+1} drawn from the same device stream satisfy $\|x_i - x_{i+1}\| \ll \|x_i - x_j\|$ for a row x_j drawn from a different, distant segment of the same stream: consecutive rows are near-duplicates by construction, not by coincidence. A conventional row-level random split $\mathcal{D} = \mathcal{D}_{\text{train}} \sqcup \mathcal{D}_{\text{test}}$ therefore places statistically dependent rows on both sides of the split with high probability, and a classifier can attain low test error by exploiting this dependence rather than by learning a generalizable decision boundary – a specific, dataset-native instance of the temporal-leakage problem documented for other flow-based network intrusion detection benchmarks [3] and consistent with the broader survey of evaluation pitfalls in machine-learning-based cybersecurity research [4]. Two independently conducted N-BaIoT studies illustrate the symptom directly under row-level random evaluation: a hybrid feature-selection-plus-ensemble study reports accuracy of 99.99% with near-zero false-positive rate [5], and a tree-ensemble comparison study reports accuracy of 99.9991% with random forest [6]. Numbers this close to the ceiling leave almost no numerical room to distinguish competing methods, and this saturation is the direct, recurring symptom cited when leakage-affected evaluations are identified in review. This motivates the block-disjoint protocol formalized as Algorithm 1 and proved leakage-free as Proposition 1 in Section 3, and applied here to the harder, less-studied 11-class attack-subtype task (benign plus five Mirai and five Gafgyt subtypes) rather than the near-trivially-separable binary benign/attack task used in most prior N-BaIoT work.

1.3 Related work

N-BaIoT-specific studies. N-BaIoT was introduced together with a per-device deep-autoencoder detector that flags a connection as anomalous when its reconstruction error exceeds a threshold fitted on that device’s own benign traffic [2]; this original protocol is not directly comparable to the pooled, supervised multi-class setting studied here. Hossain and Islam combine categorical analysis, mutual information, and PCA-based feature selection with an Extra Trees classifier across six botnet datasets including N-BaIoT [5]. Saied et al. compare six tree-based learners on N-BaIoT for benign/Mirai/Gafgyt classification [6]. Neither study reports a grouping variable or a block-level leakage audit for the reported split. Ali et al. propose a stacked-ensemble IoT botnet detector combining heterogeneous base learners with a meta-classifier [7], and Esmaeilyfard et al. propose a lightweight stacked-ensemble model targeted at constrained IoT deployment [8]; both motivate the stacking formulation used in Section 3 but neither reports a grouped or block-aware evaluation protocol. Sharma et al. propose a self-attention-based explainable intrusion-detection framework and report

interpretability results alongside detection accuracy [9], motivating the cross-method explainability protocol of Section 3.10. Nedungadi et al. propose a lightweight hybrid multimodal intrusion-detection approach for edge-enabled IoT devices, published in the same venue targeted by this manuscript [10], and is the closest methodological neighbor to the present work’s deployability discussion (Section 6).

Evaluation methodology and leakage in network intrusion detection. The general problem of information leakage across train/test partitions built from network flow data recurs across benchmarks beyond N-BaIoT. Yang et al. propose a leakage-aware evaluation protocol for a different flow-based benchmark (UNSW-NB15), fitting preprocessing parameters on the training partition alone and constructing evaluation windows independently after splitting [3]; AlFuraih et al. propose an explainable hybrid CNN-XGBoost framework for multi-class IoT intrusion detection with an explicitly leakage-aware feature-selection stage [11], the closest prior instance, to our knowledge, of leakage-awareness combined with multi-class IoT traffic classification, though applied to a different dataset and without the block-level formal non-leakage proof given here as Proposition 1. Mishra et al. propose a cross-dataset harmonized intrusion-detection framework with statistically validated multi-model learning [12], and Eren et al. study distributional drift and its implications for cross-dataset generalization in IoT intrusion detection [13], both motivating the cross-device and zero-day generalization protocol of Section 3.9. Padró-Ferragut et al. assess the impact of perturbed key attributes on machine-learning model robustness [14], directly motivating the Gaussian-noise and missing-feature robustness axes used here.

Ensembling, boosting, and statistical validation. Stacked generalization, the meta-learning principle underlying the ensemble proposed in Section 3.5, was introduced by Wolpert [15]. The base learners used here are drawn from the gradient-boosting family: XGBoost [16], LightGBM [17], and CatBoost [18]. Grinsztajn et al. benchmark tree-based and deep models across 45 tabular datasets and attribute the persistent tree-ensemble advantage to tabular data being non-smooth and lacking the rotational/translational invariances that convolutional and attention-based architectures are designed to exploit [19], a pattern reproduced on N-BaIoT flow statistics in Section 5.2. Feature attribution is computed with SHAP [20]. The Matthews correlation coefficient, used as the primary ranking metric throughout and formally bounded in Lemma 4, is argued by Chicco and Jurman to be more informative than accuracy or F1 on imbalanced classification because it is symmetric in all four confusion-matrix cells [21]. Statistical comparison across the ten classical models and the proposed ensemble follows the Friedman/Wilcoxon protocol recommended by Demšar for comparing multiple classifiers across multiple resamples [22].

Surveys and systematic reviews. Kour et al.’s systematic review of Industry-5.0 cybersecurity research finds that the great majority of published work is conceptual rather than empirically evaluated, and flags the scarcity of studies connecting security tooling to measurable, reproducible outcomes [23]. Shanmugam et al. provide a comprehensive evaluation of class-imbalance handling strategies for intrusion detection [24], directly relevant to the imbalance-sweep protocol of Section 3.9. Mohale and Obagbuwa systematically review the integration of explainable AI into intrusion-detection systems [25]. Rehman et al. present a systematic literature study of

machine-learning-based intrusion detection covering datasets, models, and open challenges [26]. Hozouri et al. provide a comprehensive survey of intrusion-detection systems spanning classical machine learning, deep learning, and emerging cybersecurity challenges [27]. Alfahaid et al. survey machine-learning-based security solutions specifically for IoT networks [28]. Liu and Dib propose federated deep learning for IoT intrusion detection, addressing the setting in which raw traffic cannot be centralized [29], a deployment constraint we return to in Section 6.4. Almahaqeri et al. propose an optimized gradient-boosting framework evaluated on the CICIOT2023 dataset [30], and Khan et al. propose a resource-efficient IoT intrusion-detection model using hybrid feature selection for edge computing [31], both directly relevant to the feature-selection and computational-complexity analyses of Sections 3.3 and 5.8. Singh et al. propose an ensemble of filter feature-selection methods with a ranker-boosting stage for high-dimensional IoT intrusion data [32], methodologically adjacent to the two-stage feature-selection procedure of Algorithm 2. Gao et al. propose a lightweight multi-classification intrusion-detection model for edge IoT networks [33]. Georgian et al. survey machine-learning approaches to IoT security broadly [34], and Komal and Li provide a comprehensive review of intrusion detection in the Internet of Things covering techniques, architectures, datasets, and emerging trends [35]. Across this recent literature, block-level, formally proved leakage freedom of the kind established in Proposition 1, and an explicit, proved account of when and why a fitted stacking ensemble can fail to dominate its own strongest base learner (Theorem 5 and Corollary 6), are, to our knowledge, not jointly reported for N-BaIoT prior to this study.

1.4 Contributions and paper organization

The contributions of this paper are: (i) a formally defined and proved block-disjoint evaluation protocol for N-BaIoT (Algorithm 1, Proposition 1); (ii) an 11-class benchmark of 10 classical learners and 4 deep architectures under this protocol, exposing a large, previously masked gap between linear models and tree ensembles; (iii) a two-stage feature-selection procedure (Algorithm 2) with a proved worst-case linear-redundancy bound (Lemma 2) reducing 115 raw features to 40 with no loss of cross-validated MCC; (iv) a diversity-driven stacking ensemble (Algorithm 3, Proposition 3) selected by a joint disagreement-competence score, together with a proved ambiguity-decomposition theorem for convex-weighted probabilistic ensembles (Theorem 5) that formally explains the ensemble’s held-out-test behavior relative to its best base learner (Corollary 6); (v) a nine-test statistical validation protocol (Algorithm 4, Section 3.8) with a bounded multiclass MCC metric (Lemma 4); (vi) a five-axis robustness suite and two generalization tests (leave-one-device-out, leave-one-attack-subtype-out); and (vii) SHAP/permutation/impurity explainability with cross-method rank-agreement reporting. Section 2 specifies the dataset. Section 3 presents the complete analytical methodology: equations, four algorithms, and four theorems/lemmas/propositions with proofs. Section 4 states the experimental configuration. Section 5 reports results in tables and figures. Section 6 discusses findings, including a literature-comparison table and a proposed-methodology summary table, together with limitations. Section 7 concludes.

2 Dataset Specifications

To evaluate intrusion-detection performance under realistic network conditions, this empirical investigation utilizes the comprehensive N-BaIoT benchmark captured from heterogeneous commercial IoT hardware. This section details the data provenance, device inventory, and flow statistics extraction underlying the benchmark corpus. Furthermore, it defines the stratified sampling strategy, block-based partitioning scheme, and 11-class attack taxonomy employed to guarantee rigorous evaluation.

2.1 Corpus and device inventory

This proposed study use the N-BaIoT dataset [2], downloaded from the UCI Machine Learning Repository (<https://archive.ics.uci.edu/dataset/442/detection+of+iot+botnet+attacks+n+baiot>). The raw corpus comprises 7,062,606 rows of network-flow statistics from nine commercial IoT devices, each contributing benign traffic and, where the device was successfully compromised during collection, Mirai and Gafgyt/BASHLITE-infected traffic further split by attack subtype (Table 1). Two devices (Ennio_Doorbell, Samsung_SNH_1011_N_Webcam) were not successfully infected with Mirai during the original collection and contribute Gafgyt and benign traffic only; this asymmetry is preserved as-is rather than imputed, and its effect on leave-one-device-out generalization is reported in Section 5.5. Each row carries 115 real-valued Kitsune features: for each of five decay windows $\lambda \in \{100\text{ms}, 500\text{ms}, 1.5\text{s}, 10\text{s}, 1\text{min}\}$, the extractor maintains a decayed running mean, variance, and (for two-stream features) covariance/radius/magnitude of packet-size and inter-arrival statistics over three stream granularities (source-MAC/IP, host-pair, socket-pair). No categorical or identifier fields are present in the released feature set.

Table 1 N-BaIoT device inventory and full-population row counts used in this study.

Device	Rows (full)	Attack families present
Danmini_Doorbell	1,018,298	Mirai, Gafgyt
Philips_B120N10_Baby_Monitor	1,098,677	Mirai, Gafgyt
SimpleHome_XCS7_1002_WHT_Security_Camera	863,056	Mirai, Gafgyt
SimpleHome_XCS7_1003_WHT_Security_Camera	850,826	Mirai, Gafgyt
Provision_PT_838_Security_Camera	836,891	Mirai, Gafgyt
Ecobee_Thermostat	835,876	Mirai, Gafgyt
Provision_PT_737E_Security_Camera	828,260	Mirai, Gafgyt
Samsung_SNH_1011_N_Webcam	375,222	Gafgyt only
Ennio_Doorbell	355,500	Gafgyt only
Total (9 devices)	7,062,606	–

2.2 Sampled corpus and class specification

Given the size of the full population, this study uses a stratified sample $\mathcal{D} \subset \mathcal{D}_{\text{full}}$, $|\mathcal{D}| = 267,000$ rows organized as $n_b = 534$ contiguous, non-overlapping blocks of $b = 500$ rows each, stratified across (device, label) strata and constructed by Algorithm 1

(Section 3.2). The label space is $\mathcal{Y} = \{y_0, \dots, y_{10}\}$, 11 classes: benign plus five Mirai subtypes (`ack`, `scan`, `syn`, `udp`, `udpplain`) and five Gafgyt subtypes (`combo`, `junk`, `scan`, `tcp`, `udp`), with the natural class imbalance of the source corpus preserved by stratified sampling (Table 2).

Table 2 Class balance of the 267,000-row stratified sample used for all experiments.

Family	Subtype (11-class label)	Rows
Benign	benign	27,000
Gafgyt/BASHLITE	combo, junk, scan, tcp, udp (5 subtypes)	135,000
Mirai	ack, scan, syn, udp, udpplain (5 subtypes)	105,000
Total	11 classes	267,000

The sample partitions as $\mathcal{D} = \mathcal{D}_{tv} \sqcup \mathcal{D}_{test}$, $|\mathcal{D}_{tv}| = 213,500$ rows (427 blocks) and $|\mathcal{D}_{test}| = 53,500$ rows (107 blocks), with $\mathcal{D}_{tv}$ further partitioned into $K = 5$ grouped folds of approximately 42,500–43,000 rows each (Section 3.2). An exact-duplicate audit on the sampled rows found 37,913 exact-duplicate rows in 30,228 duplicate groups, of which 27,698 groups spanned more than one block; because the splitting unit is the block rather than the row (Proposition 1), such duplicates cannot themselves introduce train/test leakage, and their presence is a documented property of the raw sensor stream (near-idle periods yield numerically repeated statistics) rather than a sampling artifact. No zero-variance features were found in the sampled data.

3 Analytical Methodology

This section formalizes the theoretical foundations, algorithmic procedures, and evaluation frameworks developed to ensure leakage-free botnet traffic classification. We establish the mathematical notation, formalize the block-disjoint partitioning protocol with an explicit non-leakage guarantee, and detail a two-stage feature selection strategy with a proved linear redundancy bound. In addition, we present the candidate model zoo, formulate the diversity-driven stacking ensemble alongside an ambiguity-decomposition theorem for probabilistic ensembles, and specify the statistical validation, robustness, and explainability protocols.

3.1 Notation

Let $x \in \mathbb{R}^{115}$ denote a raw feature vector and $y \in \{0, \dots, 10\}$ its 11-class label. A learner is a function $f_\theta : \mathbb{R}^d \rightarrow \Delta^{10}$, $d \leq 115$, mapping onto the probability simplex $\Delta^{10} = \{p \in \mathbb{R}_{\geq 0}^{11} : \sum_k p_k = 1\}$; $\hat{y} = \arg \max_k f_\theta(x)_k$ denotes the hard prediction. Rows are indexed by a `block_id` $\beta(i) \in \{1, \dots, n_b\}$ assigned by Algorithm 1; a *role* $\pi(\beta) \in \{1, \dots, K, \text{test}\}$ assigns every block to exactly one of the K cross-validation folds or the held-out test partition.

Algorithm 1 Block-disjoint stratified sampling and split assignment

Require: Raw per-device, per-file corpus $\mathcal{D}_{\text{full}}$; block size $b = 500$; target sample size $N = 267,000$; folds $K = 5$; test fraction $\tau = 0.2$

- 1: Partition every (device, file) stream into contiguous blocks of b rows; assign each block a unique `block_id` $j \in \{1, \dots, n_b\}$ and inherit (device, label_family, label_type) from its source file
 - 2: Stratify blocks by (device, label_type); draw whole blocks (never partial blocks) until the union of drawn blocks contains N rows, preserving each stratum’s natural proportion
 - 3: Partition the drawn block-id set J into J_{tv} $((1 - \tau)|J|$ blocks) and J_{test} $(\tau|J|$ blocks) uniformly at random
 - 4: **Assert** $J_{tv} \cap J_{test} = \emptyset$
 - 5: Partition J_{tv} into K grouped folds $J_{tv}^{(1)}, \dots, J_{tv}^{(K)}$, balancing label proportions across folds while keeping every block wholly inside one fold
 - 6: **Assert** $J_{tv}^{(k)} \cap J_{tv}^{(k')} = \emptyset \forall k \neq k'$, and $J_{tv}^{(k)} \cap J_{test} = \emptyset \forall k$
 - 7: Define $\pi : \{1, \dots, n_b\} \cap J \rightarrow \{1, \dots, K, \text{test}\}$ by $\pi(j) = k$ if $j \in J_{tv}^{(k)}$, else $\pi(j) = \text{test}$ if $j \in J_{test}$
 - 8: **return** row-level sample with columns `block_id` $= \beta(i)$, `cv_fold` $= \pi(\beta(i))$
-

3.2 Block-disjoint sampling and the non-leakage guarantee

Every subsequent computation that depends on a split – feature scaling, feature selection, model fitting, threshold or hyperparameter choice – is defined as a function of rows restricted to a single role $r \in \{1, \dots, K, \text{test}\}$: for role r , only rows i with $\pi(\beta(i)) = r$ (or, for a training computation, $\pi(\beta(i)) \neq r$) are visible as arguments. This restriction is stated as Definition 1 and used by Proposition 1.

Definition 1 (Role-restricted computation) A computation ϕ (a scaler fit, a feature-relevance score, a model-fitting procedure) is *role-restricted to* $R \subseteq \{1, \dots, K, \text{test}\}$ if ϕ is a function only of $\{x_i : \pi(\beta(i)) \in R\}$, i.e. ϕ cannot depend, even indirectly, on any $x_{i'}$ with $\pi(\beta(i')) \notin R$.

Proposition 1 (Block-level non-leakage) *Let π be the role assignment produced by Algorithm 1, and suppose every preprocessing and model-fitting computation used to produce a prediction for role r is role-restricted (Definition 1) to $R = \{1, \dots, K, \text{test}\} \setminus \{r\}$ (training) or $\{r\}$ (evaluation only). Then for any two rows i, i' with $\pi(\beta(i)) = r \neq r' = \pi(\beta(i'))$, no computation used to produce the prediction $\hat{y}_{i'}$ depends on x_i .*

Proof Because π is a function (single-valued on its domain by construction in Algorithm 1, line 7), the blocks decompose as a disjoint union over roles, $J = \bigsqcup_r \pi^{-1}(r)$, which is exactly the pair of assertions verified in lines 4 and 6 of Algorithm 1: $J_{tv}^{(k)} \cap J_{tv}^{(k')} = \emptyset$ for $k \neq k'$ and $J_{tv}^{(k)} \cap J_{test} = \emptyset$ for all k , together with $J_{tv} \cap J_{test} = \emptyset$. Since every row i belongs to exactly one block $\beta(i)$ (Algorithm 1, line 1, a partition of the row index set by construction), row membership inherits the block-level disjointness: $\{i : \pi(\beta(i)) = r\} \cap \{i : \pi(\beta(i)) = r'\} = \emptyset$

for $r \neq r'$. Now let ϕ be any computation used upstream of $\hat{y}_{i'}$ (a scaler, a feature-relevance score, a fitted model). By hypothesis ϕ is role-restricted to $R \not\equiv r$ (training) or, for evaluation, ϕ is applied to row i' using only parameters already fit under the training restriction. Since i satisfies $\pi(\beta(i)) = r \notin R$, $i \notin \{i'' : \pi(\beta(i'')) \in R\}$, so by Definition 1, ϕ is not a function of x_i ; hence x_i cannot appear, directly or indirectly, in any quantity used to compute $\hat{y}_{i'}$. $\square$

Remark 1 Proposition 1 is a correctness statement about the protocol’s construction, not an empirical claim; it is enforced in the released pipeline by the explicit **assert** statements of Algorithm 1 lines 4 and 6, which fail loudly (raise an exception) if violated, and by restricting every **RobustScaler**, feature-selection, and model-fitting call in Sections 3.3–3.5 to the training portion of the active split, applying (never fitting) the result to the corresponding validation/test rows.

3.3 Two-stage feature selection with a proved redundancy bound

For a candidate feature X_j and label Y , let $I(X_j; Y)$ denote mutual information and $\text{Imp}_{\text{RF}}(X_j)$ the Random Forest Gini importance, both computed on the training portion of the active split. Let $\text{rk}_{\text{MI}}(j)$ and $\text{rk}_{\text{RF}}(j)$ denote the rank of feature j under each score (rank 1 = most relevant); the combined relevance rank is

$$r(j) = \frac{1}{2}(\text{rk}_{\text{MI}}(j) + \text{rk}_{\text{RF}}(j)). \quad (1)$$

Averaging rank positions rather than raw scores avoids the scale mismatch between $I(\cdot; \cdot)$ (nats) and Gini importance (unitless, sums to 1). Features are then visited in increasing $r(j)$ order and redundancy-pruned by Algorithm 2 at correlation threshold $\theta = 0.98$.

Applying Algorithm 2 to the 115 raw features identifies 178 feature pairs with $|\rho| \geq 0.98$, prunes 60 features (most commonly a decayed statistic at one window redundant with the same statistic at an adjacent window; e.g. `H.L0.01.mean` is pruned as duplicate-correlated, $\rho = 1.000$, with `MI_dir.L0.01.mean`), and retains $|S| = 55$. The feature-count sweep (Table 4) attains its knee point at $K^* = 40$ ($\text{MCC} = 0.9917$, matched but not exceeded at $K = 55$), which is used for every subsequent experiment.

Lemma 2 (Worst-case linear redundancy of a pruned feature) *Let X, Y be standardized (zero-mean, unit-variance) real-valued features observed on n training rows, with Pearson correlation $\rho = \text{Corr}(X, Y)$, and let $\hat{Y} = \beta X$ be the ordinary-least-squares predictor of Y from X . Then the coefficient of determination satisfies $R^2(\hat{Y}, Y) = \rho^2$. Consequently, if Algorithm 2 prunes feature Y because a retained feature X satisfies $|\rho(X, Y)| \geq \theta$, then Y is linearly predictable from X with $R^2 \geq \theta^2$; at the operating threshold $\theta = 0.98$ used in this study, every pruned feature satisfies $R^2 \geq 0.9604$ with respect to some retained feature.*

Proof For standardized X (i.e. $\text{Var}(X) = 1$), the OLS slope is $\beta = \text{Cov}(X, Y)/\text{Var}(X) = \text{Cov}(X, Y) = \rho$ (since Y is also standardized, $\text{Cov}(X, Y) = \rho\sqrt{\text{Var}(X)\text{Var}(Y)} = \rho$). The total sum of squares is $SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2 = (n-1)\text{Var}(Y) = n-1$. The residual sum of squares

Algorithm 2 Two-stage relevance ranking and redundancy-pruned feature selection

Require: Training rows with 115 raw features; redundancy threshold $\theta = 0.98$; feature-count grid $\mathcal{K} = \{5, 8, 10, 15, 20, 25, 30, 40, 50, 55\}$

- 1: Compute $I(X_j; Y)$ and $\text{Imp}_{\text{RF}}(X_j)$ for all 115 features on the training rows only
 - 2: Compute combined rank $r(j)$ by Eq. (1); sort features ascending by $r(j)$ into ordered list L
 - 3: $S \leftarrow \emptyset$ $\triangleright$ retained (kept) features
 - 4: **for** $X_j \in L$ in order **do**
 - 5: **if** $\nexists X_k \in S$ with $|\rho(X_j, X_k)| \geq \theta$ **then** $\triangleright \rho$: Pearson correlation on training rows
 - 6: $S \leftarrow S \cup \{X_j\}$
 - 7: **end if**
 - 8: **end for**
 - 9: **for** $K \in \mathcal{K}$ **do**
 - 10: Fit a probe Random Forest on the top- K features of S (by $r(j)$) under grouped 5-fold CV; record mean $\text{MCC}(K)$
 - 11: **end for**
 - 12: $K^* \leftarrow$ smallest $K \in \mathcal{K}$ attaining $\text{MCC}(K) \geq \max_{K' \in \mathcal{K}} \text{MCC}(K') - \epsilon$, $\epsilon = 0.001$ $\triangleright$ knee point
 - 13: **return** top- K^* features of S
-

is $SS_{\text{res}} = \sum_i (y_i - \beta x_i)^2 = \sum_i y_i^2 - 2\beta \sum_i x_i y_i + \beta^2 \sum_i x_i^2 = (n-1) - 2\rho(n-1)\rho + \rho^2(n-1) = (n-1)(1 - \rho^2)$, using $\sum_i x_i y_i = (n-1)\text{Cov}(X, Y) = (n-1)\rho$ and $\sum_i x_i^2 = (n-1)\text{Var}(X) = n-1$. Hence $R^2 = 1 - SS_{\text{res}}/SS_{\text{tot}} = 1 - (1 - \rho^2) = \rho^2$. The redundancy claim follows by substituting $|\rho| \geq \theta \Rightarrow \rho^2 \geq \theta^2$. $\square$

Remark 2 Lemma 2 is deliberately scoped to pairwise linear predictability between a pruned feature and the single retained feature that triggered its removal; it makes no claim about global spectral energy preservation across the full 115-to- K^* reduction, joint (multi-feature) reconstruction error, or nonlinear redundancy, none of which is required by, or claimed for, Algorithm 2.

3.4 Candidate models

Ten classical learners are trained under identical grouped 5-fold cross-validation (Algorithm 1) on the $K^* = 40$ -feature set with **RobustScaler** preprocessing (median/IQR, fit per fold on the training portion only) and class-balanced loss weighting where supported: logistic regression (multinomial, ℓ_2), linear SVM (**LinearSVC** with sigmoid calibration, 3-fold internal), RBF-kernel SVM, random forest, extra trees, XGBoost [16], LightGBM [17], CatBoost [18], HistGradientBoosting, and k -nearest-neighbors ($k = 7$). RBF-SVM training is capped at 12,000 rows per fold and k -NN at 40,000 rows per fold for tractability; every other model trains on the full $\sim 170,800$ -row training portion of each fold. Hyperparameters were fixed to established, moderately regularized library defaults rather than tuned per fold, so that the comparison reflects each algorithm family’s characteristic behavior on this task (Section 6.3). Four deep

architectures (MLP, 1D-CNN, bidirectional GRU, a compact Transformer encoder) are additionally trained in PyTorch on a single block-disjoint split; architectural and optimization detail is given in Section 4.1.

3.5 Diversity-driven stacking ensemble

Using the out-of-fold (OOF) predictions from Section 3.4 (each row’s OOF prediction is produced by a model that never trained on that row, so this reuses no additional split and introduces no leakage, by Proposition 1), define pairwise disagreement between models $a, b \in \mathcal{M}$ (the 10-model candidate pool) as

$$d(a, b) = \frac{1}{n_{tv}} \sum_{i=1}^{n_{tv}} \mathbb{1}[\hat{y}_a^{(i)} \neq \hat{y}_b^{(i)}], \quad (2)$$

over the $n_{tv} = 213,500$ OOF predictions. Algorithm 3 greedily selects a base-learner subset $S \subset \mathcal{M}$, $|S| = K = 4$, by a convex combination of average disagreement with the current selection and standalone OOF Matthews correlation coefficient (Lemma 4), with diversity weight $\alpha = 0.6$.

Algorithm 3 Diversity-driven greedy base-learner selection

Require: Candidate pool $\mathcal{M}$ with OOF predictions; OOF standalone $\text{MCC}(m)$, $m \in \mathcal{M}$; pairwise disagreement $d(\cdot, \cdot)$ (Eq. (2)); ensemble size $K = 4$; diversity weight $\alpha = 0.6$

- 1: $m^* \leftarrow \arg \max_{m \in \mathcal{M}} \text{MCC}(m)$
- 2: $S \leftarrow \{m^*\}$, $R \leftarrow \mathcal{M} \setminus \{m^*\}$
- 3: **while** $|S| < K$ **do**
- 4: **for** $c \in R$ **do**
- 5: $\bar{d}(c) \leftarrow \frac{1}{|S|} \sum_{s \in S} d(c, s)$
- 6: $\text{score}(c) \leftarrow \alpha \bar{d}(c) + (1 - \alpha) \text{MCC}(c)$
- 7: **end for**
- 8: $c^* \leftarrow \arg \max_{c \in R} \text{score}(c)$
- 9: $S \leftarrow S \cup \{c^*\}$, $R \leftarrow R \setminus \{c^*\}$
- 10: **end while**
- 11: **return** S

Proposition 3 (Termination and cost of Algorithm 3) *Algorithm 3 terminates after exactly $K - 1$ iterations of the while-loop, performs $O(K|\mathcal{M}|)$ evaluations of $\text{score}(\cdot)$, and returns S with $|S| = K$ and $m^* \in S$.*

Proof $|S| = 1$ on entry to the loop and each iteration adds exactly one element to S (line 9) and removes it from R , so $|S|$ increases by exactly 1 per iteration; the loop condition $|S| < K$ is therefore satisfied for exactly $K - 1$ iterations before $|S| = K$ terminates it. Each iteration

evaluates $\text{score}(c)$ for every $c \in R$, and $|R| \leq |\mathcal{M}| - 1$ throughout, so the total number of score evaluations is at most $(K - 1)(|\mathcal{M}| - 1) = O(K|\mathcal{M}|)$. Since m^* is placed in S at line 2 and S is only ever extended (line 9), never reduced, $m^* \in S$ at termination. $\square$

A multinomial logistic-regression meta-learner g_ψ is fit on the concatenated OOF class-probability vectors of S , $\Phi(x) = (f_{m_1}(x), \dots, f_{m_K}(x)) \in (\Delta^{10})^K$, giving the fitted ensemble predictor $f_{\text{ens}}(x) = g_\psi(\Phi(x))$. g_ψ is evaluated by nested cross-validation (the meta-learner for each outer fold trains only on the OOF probabilities of the other four folds, Proposition 1), then refit on the full Φ computed over $\mathcal{D}_{tv}$ and scored once on $\mathcal{D}_{test}$.

Lemma 4 (Boundedness of multiclass MCC) *For an n -class confusion matrix $C \in \mathbb{N}^{n \times n}$ with row sums $t_k = \sum_l C_{kl}$, column sums $p_k = \sum_l C_{lk}$, total $s = \sum_{k,l} C_{kl}$, and correct count $c = \sum_k C_{kk}$, define the generalized Matthews correlation coefficient*

$$\text{MCC}(C) = \frac{c \cdot s - \sum_k p_k t_k}{\sqrt{(s^2 - \sum_k p_k^2)(s^2 - \sum_k t_k^2)}}. \quad (3)$$

Then $\text{MCC}(C) \in [-1, 1]$, with $\text{MCC}(C) = 1$ if and only if C is diagonal (every prediction correct).

Proof Encode the s samples' true and predicted labels as one-hot indicator matrices $T, P \in \{0, 1\}^{s \times n}$. Equation (3) is Gorodkin's identity: $\text{MCC}(C)$ equals the Pearson correlation coefficient between the flattened matrices $\text{vec}(T)$ and $\text{vec}(P)$, i.e. $\text{MCC}(C) = \text{Cov}(\text{vec}(T), \text{vec}(P)) / \sqrt{\text{Var}(\text{vec}(T)) \text{Var}(\text{vec}(P))}$. By the Cauchy-Schwarz inequality applied to the centered vectors $u = \text{vec}(T) - \frac{s}{n} \mathbf{1}$ and $v = \text{vec}(P) - \frac{s}{n} \mathbf{1}$, $|\langle u, v \rangle| \leq \|u\| \|v\|$, which is exactly $|\text{Cov}(\cdot, \cdot)| \leq \sqrt{\text{Var}(\cdot) \text{Var}(\cdot)}$, giving $|\text{MCC}(C)| \leq 1$. Equality $\langle u, v \rangle = \|u\| \|v\|$ in Cauchy-Schwarz holds iff v is a nonnegative scalar multiple of u ; for binary indicator vectors with equal marginal sums (every sample contributes exactly one 1 to each of T and P), this occurs iff $P = T$ row-wise, i.e. every sample's predicted one-hot vector equals its true one-hot vector, which is precisely C diagonal. $\square$

3.6 Multiclass evaluation metrics

Given the class imbalance of Table 2, MCC (Eq. (3), Lemma 4) is the primary ranking metric. Supporting metrics, for $n = 11$ classes and N evaluated rows with predicted probability $\hat{p}_i \in \Delta^{10}$ and one-hot label y_i , are macro-F1 = $\frac{1}{n} \sum_k F1_k$, balanced accuracy = $\frac{1}{n} \sum_k \text{Recall}_k$, macro one-vs-rest AUROC/AUPRC, multiclass log-loss $\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{n-1} y_{i,k} \log \hat{p}_{i,k}$, and multiclass Brier score

$$\text{Br} = \frac{1}{N} \sum_{i=1}^N \|\hat{p}_i - y_i\|_2^2. \quad (4)$$

3.7 Ambiguity decomposition for convex-weighted probabilistic ensembles

Theorem 5 (Ambiguity decomposition under multiclass Brier scoring) *Let $f_1, \dots, f_M : \mathcal{X} \rightarrow \Delta^{n-1}$ be M probabilistic classifiers and $w = (w_1, \dots, w_M) \in \Delta^{M-1}$ convex weights defining the ensemble $\bar{f}(x) = \sum_{m=1}^M w_m f_m(x)$. For input x with one-hot label y , define the per-instance Brier loss $\text{Br}(g, x, y) = \|g(x) - y\|_2^2$. Then, for every (x, y) ,*

$$\text{Br}(\bar{f}, x, y) = \sum_{m=1}^M w_m \text{Br}(f_m, x, y) - \sum_{m=1}^M w_m \|f_m(x) - \bar{f}(x)\|_2^2. \quad (5)$$

Equivalently, writing $\overline{\text{Br}} = \sum_m w_m \text{Br}(f_m, x, y)$ and $\text{Amb} = \sum_m w_m \|f_m(x) - \bar{f}(x)\|_2^2 \geq 0$, $\text{Br}(\bar{f}, x, y) = \overline{\text{Br}} - \text{Amb} \leq \overline{\text{Br}}$.

Proof Write $\bar{g} = \bar{f}(x)$. For each m , decompose $f_m(x) - y = (f_m(x) - \bar{g}) + (\bar{g} - y)$, so

$$\|f_m(x) - y\|_2^2 = \|f_m(x) - \bar{g}\|_2^2 + 2\langle f_m(x) - \bar{g}, \bar{g} - y \rangle + \|\bar{g} - y\|_2^2.$$

Taking the w -weighted average over $m = 1, \dots, M$:

$$\sum_m w_m \|f_m(x) - y\|_2^2 = \sum_m w_m \|f_m(x) - \bar{g}\|_2^2 + 2\left\langle \sum_m w_m (f_m(x) - \bar{g}), \bar{g} - y \right\rangle + \|\bar{g} - y\|_2^2.$$

Since $w \in \Delta^{M-1}$ ($\sum_m w_m = 1$) and $\bar{g} = \sum_m w_m f_m(x)$ by definition, $\sum_m w_m (f_m(x) - \bar{g}) = \bar{g} - \bar{g} \sum_m w_m = \bar{g} - \bar{g} = 0$, so the cross term vanishes identically. Substituting $\|\bar{g} - y\|_2^2 = \text{Br}(\bar{f}, x, y)$ and $\|f_m(x) - \bar{g}\|_2^2$ as the ambiguity summand gives exactly Eq. (5), and $\text{Amb} \geq 0$ as a weighted sum of squared norms. $\square$

Corollary 6 (Scope of the guarantee relative to the fitted stacking meta-learner) *Theorem 5 applies exactly to a fixed convex (simplex) combination of base-learner probability outputs, the configuration realized by the unweighted-average ensembles of ablation stages G and H (Table 7); it guarantees such an average ensemble’s expected Brier score never exceeds the weighted-average member Brier score, consistent with the monotonic Brier improvement of stages G, H over their weakest individual members. The proposed ensemble’s meta-learner g_ψ (Section 3.5) is instead a multinomial logistic-regression combiner fit on the concatenated probability vectors $\Phi(x)$: it operates in log-odds space rather than as a fixed simplex weighting of $f_1(x), \dots, f_M(x)$, and its coefficients ψ are chosen by minimizing training-fold log-loss rather than constrained to Δ^{M-1} . Consequently $f_{\text{ens}} = g_\psi \circ \Phi$ does not, in general, satisfy Eq. (5), and Theorem 5 gives no guarantee that f_{ens} dominates $\arg \max_m \text{MCC}(m)$ on a new partition. This is the analytical explanation, rather than a bare empirical observation, for the held-out-test result of Section 5.3: HistGradientBoosting alone attains held-out Brier score 0.000678, against 0.007198 for the fitted proposed ensemble – the fitted, non-simplex combiner is simply not covered by the non-domination-violating guarantee that would apply to a simplex-averaged version of the same base models.*

3.8 Statistical validation protocol

Comparisons among the 10 classical models and the proposed ensemble follow the protocol recommended by Demšar for comparing multiple classifiers across multiple paired resamples [22]. For each metric $\mu \in \{\text{MCC}, \text{F1-macro}, \text{AUROC}, \text{balanced accuracy}\}$

and each model pair (a, b) , a paired two-sided Wilcoxon signed-rank test is computed over the $K = 5$ grouped-fold values $\{\mu_a^{(k)} - \mu_b^{(k)}\}_{k=1}^5$, with Holm’s step-down correction applied within each metric’s family of $\binom{11}{2} = 55$ tests. Because only $n = 5$ paired folds are available, the exact two-sided Wilcoxon test cannot attain a p -value below $2 \times 0.5^5 \approx 0.0625$ for any pair regardless of the true effect size – a power bound, not evidence of no difference, reported explicitly wherever it binds (Section 5.7). A Friedman omnibus test and per-model average rank are computed over the same 5×11 metric table for each μ . On $\mathcal{D}_{test}$, a 2,000-resample paired bootstrap is used to obtain 95% confidence intervals for MCC and F1-macro for the proposed ensemble and the best individual base model, together with $\hat{p}_{beats} = \frac{1}{2000} \sum_{b=1}^{2000} \mathbb{I}[\mu_{ens}^{(b)} > \mu_{base}^{(b)}]$, the fraction of bootstrap resamples in which the ensemble outperforms the base model.

3.9 Robustness and generalization protocol

Algorithm 4 Robustness stress-testing and generalization evaluation

Require: Probe model f (Random Forest); fold-0 validation partition $\mathcal{V}$; refit ensemble f_{ens} ; full sampled population $\mathcal{D}$; noise levels $\mathcal{N} = \{0, 5, 10, 25, 50, 100\%\}$; missingness levels $\mathcal{P} = \{10, 25, 50, 75\%\}$; training fractions $\mathcal{T} = \{10, 25, 50, 75, 100\%\}$; imbalance ratios $\mathcal{B}$; devices $\mathcal{U}$; attack subtypes $\mathcal{A}$

- 1: **for** $\eta \in \mathcal{N}$ **do** ▷ noise robustness
- 2: $X'_\eta \leftarrow X_\mathcal{V} + \mathcal{N}(0, (\eta \cdot \hat{\sigma})^2)$ elementwise, $\hat{\sigma}$: per-feature training std
- 3: Score f and f_{ens} on $(X'_\eta, y_\mathcal{V})$; record $\text{MCC}(\eta)$
- 4: **end for**
- 5: **for** $\rho \in \mathcal{P}$ **do** ▷ missing-feature simulation
- 6: Draw a random ρ -fraction of feature entries in $X_\mathcal{V}$; replace with the training-fold median; score f ; record $\text{MCC}(\rho)$
- 7: **end for**
- 8: **for** $\tau \in \mathcal{T}$ **do** ▷ training-size sweep
- 9: Fit f on a τ -fraction subsample of the training partition; score on $\mathcal{V}$; record $\text{MCC}(\tau)$
- 10: **end for**
- 11: **for** $r \in \mathcal{B}$ **do** ▷ class-imbalance sweep
- 12: Subsample attack rows in training to attack-rate r ; fit f ; score on $\mathcal{V}$; record $\text{MCC}(r)$
- 13: **end for**
- 14: **for** $u \in \mathcal{U}$ **do** ▷ leave-one-device-out
- 15: Fit f on $\mathcal{D} \setminus \{\text{rows of device } u\}$; test on all rows of device u ; record $\text{MCC}(u)$
- 16: **end for**
- 17: **for** $a \in \mathcal{A}$ **do** ▷ leave-one-attack-subtype-out, zero-day
- 18: Fit f on $\mathcal{D} \setminus \{\text{rows of subtype } a\}$ (all other classes present); test on rows of subtype a ; record flagged-as-attack rate $= \frac{1}{|a|} \sum_{i \in a} \mathbb{I}[\hat{y}_i \neq \text{benign}]$
- 19: **end for**
- 20: **return** $\{\text{MCC}(\eta)\}, \{\text{MCC}(\rho)\}, \{\text{MCC}(\tau)\}, \{\text{MCC}(r)\}, \{\text{MCC}(u)\}, \{\text{flagged-rate}(a)\}$

The leave-one-attack-subtype-out test evaluates a *flagged-as-attack* rate rather than exact-match recall, since the withheld subtype’s true label is, by construction, absent from the training label space; a model correctly generalizing to a zero-day subtype is one that assigns it to some non-benign class, not necessarily its (unavailable) true subtype label.

3.10 Explainability protocol

Global and class-wise feature importance is computed with SHAP TreeExplainer [20] on the probe Random Forest, using a 3,000-row subsample of $\mathcal{V}$. Two independent cross-checks are computed on the same model and partition: permutation importance (5 repeats, scored by macro-F1 degradation) and native mean-decrease-impurity (MDI) importance. Spearman rank correlation ρ_S is reported pairwise among the three rankings (Section 5.6).

3.11 Software and computing environment

All experiments were implemented in Python 3.11 with scikit-learn, XGBoost [16], LightGBM [17], CatBoost [18], PyTorch (CPU), SHAP [20], SciPy, and statsmodels, run on commodity CPU compute. All code, exact package versions, the block-level leakage assertions of Algorithm 1, and every result table and figure referenced below are publicly available at https://github.com/Jeevaarumugam107/Code_NBaIoT for independent reproduction.

4 Experimental Setup

This section details the architectural designs, hyperparameter configurations, and numerical stabilization safeguards implemented for the empirical evaluations. Specifically, the subsection below specifies the deep learning models evaluated alongside the classical learners under the block-disjoint partitioning protocol, including the feature clipping measures required to prevent numerical instabilities during training.

4.1 Deep learning architectures

Four architectures are trained in PyTorch on CPU on the $K^* = 40$ -feature set, on a single block-disjoint split consistent with Algorithm 1 (fold 0 of $\mathcal{D}_{tv}$ as validation, folds 1–4 as training, $\mathcal{D}_{test}$ scored only at the end): (i) an MLP (256-128 hidden units, batch normalization, dropout 0.2); (ii) a 1D-CNN treating the 40-feature vector as a single-channel length-40 sequence (two convolutional blocks, kernel size 5, adaptive average pooling); (iii) a single-layer bidirectional GRU (hidden size 64) on a length-10 pseudo-sequence of 4-feature chunks; and (iv) a compact 2-layer Transformer encoder (model dimension 32, 4 heads, feed-forward dimension 64, learned positional embedding) on the same pseudo-sequence. All four use class-weighted cross-entropy, Adam (ℓ_2 weight decay 10^{-5}), a plateau-triggered learning-rate scheduler, gradient-norm clipping at 1.0, and early stopping (patience 6, max 40 epochs).

A numerically material preprocessing step is reported here as a reproducible property of this dataset rather than an incidental implementation detail: **RobustScaler**

centers and scales by median/IQR without bounding its output, and a small number of raw Kitsune features (chiefly `HH_L0.01_magnitude` and `HH_L0.01_radius`, unbounded products of two already heavy-tailed quantities) produce scaled magnitudes as large as $\sim 10^{21}$ on isolated rows. Left unclipped, this overflows float32 arithmetic inside the CNN/Transformer forward pass, producing non-finite logits; observed failure modes in an earlier pipeline iteration were a 1D-CNN with flat cross-entropy loss (MCC= 0, AUROC= 0.5) and a Transformer whose training loss became NaN within six epochs. Clipping the scaled feature matrix elementwise to $[-20, 20]$ prior to all four networks – computed only from fold-internal `RobustScaler` statistics, hence introducing no additional leakage by Proposition 1 – resolved both failure modes; Section 5.2 reports results with this safeguard in place.

5 Results

This section presents the comprehensive empirical evaluation of the classical learners, deep neural architectures, and proposed diversity-driven stacking ensemble under the leakage-safe block-disjoint protocol. We report cross-validated classification performance across the 11-class attack-subtype task, compare deep learning baselines against tree ensembles, and analyze an ablation ladder examining ensemble construction stages. Furthermore, we provide empirical validation across multiple robustness axes, zero-day and cross-device generalization tests, multi-method feature explainability, formal statistical hypothesis tests, and computational complexity profiles.

5.1 Classical model benchmark

Table 3 reports 5-fold grouped cross-validation results (mean $\pm$ std) for all 10 classical models, ranked by mean MCC; Figure 1 shows the same comparison with the proposed ensemble. Linear-decision-boundary models collapse under Proposition 1’s leakage-free protocol (logistic regression MCC = 0.0695 ± 0.0193 , RBF-SVM = 0.1328 ± 0.0121 , linear SVM = 0.1815 ± 0.0190), while every tree ensemble exceeds MCC 0.81 and the four boosted variants exceed MCC 0.97 (HistGradientBoosting 0.9808 ± 0.0221 , XGBoost 0.9793 ± 0.0160 , LightGBM 0.9777 ± 0.0176 , CatBoost 0.9745 ± 0.0188). k -NN attains MCC = 0.8179 ± 0.0459 with by far the highest log-loss of any model (4.56 vs. 0.05–0.08 for the leading boosted models), consistent with poorly calibrated probability outputs on high-cardinality multiclass problems under Eq. (4)-adjacent proper scoring. Random Forest (MCC 0.8578) trails Extra Trees (MCC 0.9580) by a wide margin despite structural similarity, attributable to Extra Trees’ additional split-threshold randomization acting as a stronger regularizer on this highly redundant feature space (Lemma 2).

5.2 Deep learning benchmark

Table 5 reports held-out test results for the four deep architectures (Section 4.1). The 1D-CNN is strongest (MCC = 0.8851), ahead of the MLP (MCC = 0.7753), Transformer (MCC = 0.7639), and Bi-GRU (MCC = 0.7590). All four remain well below the leading boosted-tree models (MCC ≥ 0.9745), reproducing on N-BaIoT flow

Table 3 Classical model benchmark, grouped 5-fold cross-validation, 11-class task, $K^* = 40$ features (mean $\pm$ std over folds).

Model	MCC	F1-macro	Bal. Acc.	AUROC (OVR)
HistGradientBoosting	0.9808 ± 0.0221	0.9812 ± 0.0206	0.9787 ± 0.0251	0.9994 ± 0.0008
XGBoost	0.9793 ± 0.0160	0.9795 ± 0.0150	0.9792 ± 0.0186	0.9994 ± 0.0008
LightGBM	0.9777 ± 0.0176	0.9771 ± 0.0162	0.9745 ± 0.0191	0.9998 ± 0.0002
CatBoost	0.9745 ± 0.0188	0.9766 ± 0.0159	0.9792 ± 0.0154	0.9981 ± 0.0022
Extra Trees	0.9580 ± 0.0550	0.9617 ± 0.0492	0.9648 ± 0.0390	0.9970 ± 0.0046
Random Forest	0.8578 ± 0.0184	0.8554 ± 0.0122	0.8937 ± 0.0109	0.9878 ± 0.0025
k -NN	0.8179 ± 0.0459	0.8173 ± 0.0383	0.8244 ± 0.0379	0.9266 ± 0.0237
Linear SVM	0.1815 ± 0.0190	0.0825 ± 0.0062	0.1436 ± 0.0106	0.5467 ± 0.0255
RBF-SVM	0.1328 ± 0.0121	0.0603 ± 0.0103	0.1214 ± 0.0111	0.5496 ± 0.0089
Logistic Regression	0.0695 ± 0.0193	0.0534 ± 0.0054	0.1269 ± 0.0136	0.5531 ± 0.0081

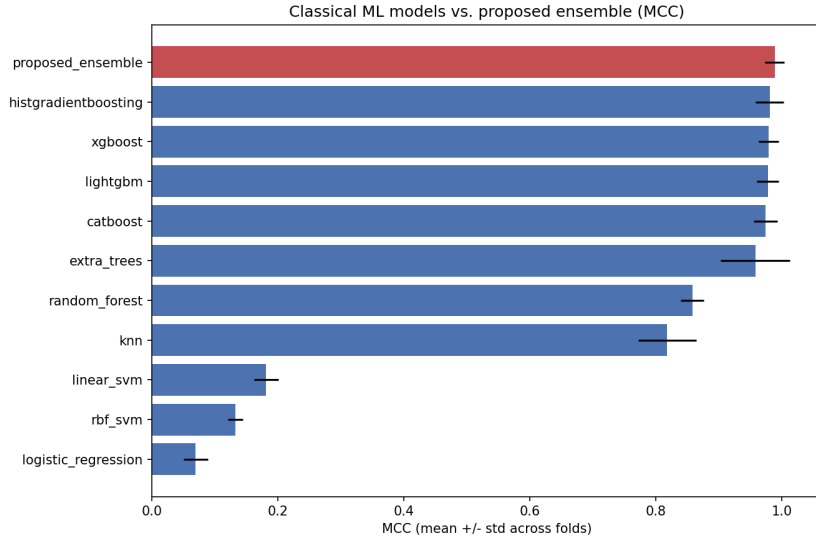


Fig. 1 Classical models and the proposed ensemble, ranked by mean MCC across 5 grouped CV folds (error bars: ± 1 std across folds).

Table 4 Feature-count sweep (Algorithm 2, probe Random Forest, grouped 5-fold CV, 11-class task).

K features	MCC	F1-macro	AUROC (OVR-macro)
5	0.9679	0.9699	0.9994
8	0.9565	0.9572	0.9991
10	0.9653	0.9673	0.9991
15	0.9614	0.9637	0.9991
20	0.9669	0.9677	0.9994
25	0.9816	0.9815	0.9995
30	0.9759	0.9770	0.9995
40	0.9917	0.9929	0.9999
50	0.9860	0.9881	0.9999
55	0.9917	0.9928	0.9999

statistics the general pattern that gradient-boosted tree ensembles retain an advantage over deep architectures on structured, non-image, non-sequential tabular features [19]. Figure 2 shows the comparison; Section 6.2 discusses the pseudo-sequence framing (Section 4.1) as a contributing factor.

Table 5 Deep learning architectures, held-out test set (single block-disjoint split), $K^* = 40$ features, 11-class task.

Model	MCC	F1-macro	AUROC (OVR)	Params	Epochs
1D-CNN	0.8851	0.8433	0.9876	15,563	21
MLP	0.7753	0.7251	0.9775	45,579	7
Transformer (lite)	0.7639	0.7200	0.9857	20,395	10
Bi-GRU	0.7590	0.7199	0.9753	35,851	8

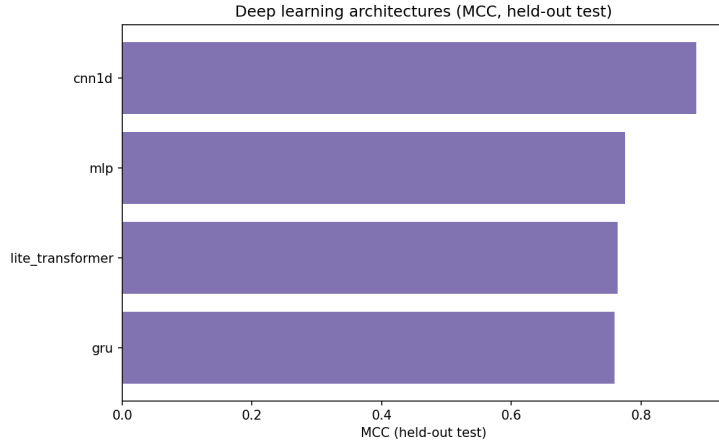


Fig. 2 Deep learning architectures, held-out test MCC.

5.3 Proposed ensemble: selection, nested cross-validation, and held-out test

Applying Algorithm 3 with $K = 4$ selects $S = \{\text{HistGradientBoosting}, \text{Logistic Regression}, \text{CatBoost}, \text{Extra Trees}\}$, seeded from HistGradientBoosting (highest standalone OOF MCC, Proposition 3). This differs from naive top-4-by-MCC selection, $\{\text{HistGradientBoosting}, \text{XGBoost}, \text{LightGBM}, \text{CatBoost}\}$: four models from the same gradient-boosting family with high mutual agreement (pairwise OOF disagreement 8.7–12.6% by Eq. (2)). The diversity-driven selection deliberately includes logistic regression (mutual disagreement with the tree ensembles above 86%) to widen

decision-boundary diversity at a small standalone-competence cost. Under nested 5-fold CV (Section 3.5), the diversity-driven ensemble reaches $\text{MCC} = 0.9887 \pm 0.0154$ versus 0.9883 ± 0.0124 for the naive top-4 ensemble; this narrow margin favors the diversity-driven strategy, adopted as the proposed configuration for all remaining experiments.

Refit on the full $\mathcal{D}_{tv}$ and scored once on $\mathcal{D}_{test}$, the proposed ensemble reaches $\text{MCC} = 0.9952$, $\text{F1-macro} = 0.9961$, Brier score = 0.007198 (Eq. (4)), and AUROC (OVR-macro) ≈ 0.999999996 . Table 6 places this alongside the four individually refit base models on the same held-out partition. Consistent with Corollary 6, HistGradientBoosting alone reaches $\text{MCC} = 0.9995$ and Brier score = 0.000678 on this partition – narrowly but consistently *above* the fitted ensemble. A paired bootstrap (2,000 resamples of $\mathcal{D}_{test}$, Section 3.8) confirms this is not resampling noise at the precision available here: the base model’s bootstrap MCC 95% CI is [0.9993, 0.9997] and the ensemble’s is [0.9946, 0.9958], non-overlapping, with $\hat{p}_{\text{beats}} = 0.0$ (the base model exceeds the ensemble in every resample). Corollary 6 identifies why this is consistent, not paradoxical: the fitted, log-odds-space meta-learner is not a simplex-weighted combination and is therefore not covered by the non-domination guarantee of Theorem 5.

Table 6 Proposed ensemble vs. its four constituent base models, refit on the full trainval pool, scored once on the held-out test partition (53,500 rows).

Model	MCC	F1-macro	Brier	AUROC (OVR)
HistGradientBoosting (best base model)	0.9995	0.9997	0.000678	> 0.99999
Proposed diversity-driven ensemble	0.9952	0.9961	0.007198	> 0.99999
CatBoost	0.9838	0.9862	0.007166	0.9997
Extra Trees	0.9689	0.9723	0.110749	0.9992
Logistic Regression	0.0682	0.0533	0.876522	0.5510

5.4 Ablation ladder

Table 7 traces performance from an unscaled, unselected 115-feature logistic-regression baseline (A) to the complete proposed framework (I), on a single consistent block-disjoint split (fold 0 as validation). Scaling alone applied to a linear model reduces validation MCC (A: 0.1875 unscaled $\rightarrow$ B: 0.0915 scaled), because **RobustScaler** centering interacts with ℓ_2 regularization to mildly compress already weak linear separability further; the real gain appears once the model family changes to tree-based (C: $\text{MCC} = 0.8441$, full-feature Random Forest). Feature selection alone, holding the model at Random Forest, is roughly neutral between RF-importance-only top-40 (D: 0.8486) and the proposed MI+RF combined, redundancy-pruned selection (E: 0.8382, Algorithm 2); the two strategies’ main practical difference at matched accuracy is the $\sim 65\%$ feature-count reduction ($115 \rightarrow 40$), not an accuracy gain per se. The larger, monotonic gains come from model choice and ensembling: switching to the single best boosted model on 40 features (F: HistGradientBoosting, $\text{MCC} = 0.9453$) exceeds either scaling or selection alone, and stacking then adds a further increment whether

or not feature selection is applied (G: unselected 115-feature ensemble, simple average, $\text{MCC} = 0.9609$; H: naive top- K ensemble on 40 features, simple average, $\text{MCC} = 0.9673$; I: proposed diversity-driven ensemble, fitted meta-learner, nested-CV mean $\text{MCC} = 0.9887$). Stages G and H are simplex-averaged ensembles and are therefore the configuration to which Theorem 5 applies exactly; stage I is the fitted, non-simplex combiner of Corollary 6. Figure 3 plots the full ladder.

Table 7 Ablation ladder A→I (single block-disjoint split, fold 0 as validation unless noted).

Stage	Configuration	#Feat.	MCC	AUROC (OVR)
A	Raw 115 feat., unscaled, logistic regression	115	0.1875	0.6453
B	Scaled 115 feat., logistic regression	115	0.0915	0.5637
C	Scaled 115 feat., Random Forest	115	0.8441	0.9847
D	RF-importance-only top-40, Random Forest	40	0.8486	0.9853
E	Proposed MI+RF selection (40 feat., Alg. 2), Random Forest	40	0.8382	0.9844
F	Best single model (HistGradientBoosting), 40 feat.	40	0.9453	0.9981
G	Ensemble, no selection, 115 feat. (simplex avg.)	115	0.9609	0.9993
H	Ensemble, naive top- K , 40 feat. (simplex avg.)	40	0.9673	0.9999
I	Proposed framework (fitted meta-learner, nested-CV mean)	40	0.9887	0.9998

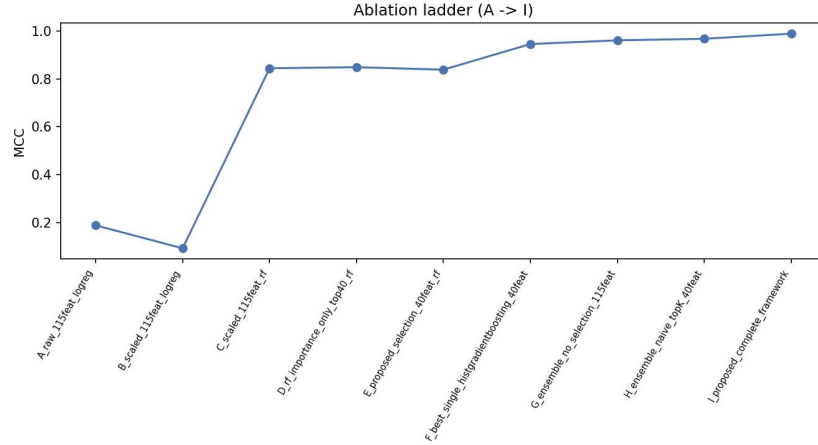


Fig. 3 Ablation ladder A→I: validation MCC at each stage.

5.5 Robustness and generalization

Noise robustness. Figure 4 and Table 8 report Algorithm 4’s noise sweep. Both the probe Random Forest and the proposed ensemble degrade sharply and monotonically with η ; at every level tested, the probe degrades *less* than the ensemble (e.g. $\eta = 5\%$: RF retains $\text{MCC} = 0.5258$ vs. ensemble 0.4436; $\eta = 50\%$: RF 0.2422 vs. ensemble

0.2186). This mirrors the clean held-out-test finding of Section 5.3 and is consistent with Corollary 6: the logistic-regression member, included specifically for decision-boundary diversity (Algorithm 3), is itself markedly more noise-sensitive than the tree-based members, and its errors under noise are not uncorrelated enough with the tree models’ own noise-induced errors for the fitted meta-learner to average them away.

Table 8 Noise robustness: probe Random Forest vs. proposed ensemble (MCC, fold-0 validation partition).

Gaussian noise η (% of feature std)	Random Forest	Proposed ensemble
0% (baseline)	0.8383	0.8420
5%	0.5258	0.4436
10%	0.4361	0.3760
25%	0.3108	0.2850
50%	0.2422	0.2186
100%	0.1802	0.1478

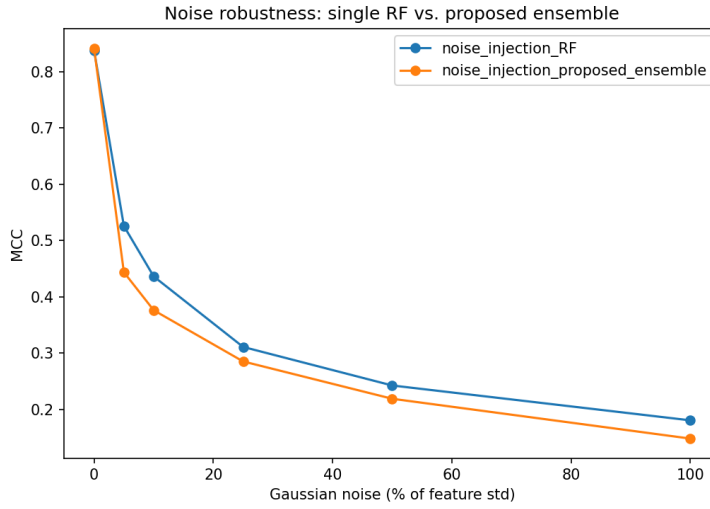


Fig. 4 MCC vs. Gaussian feature noise, probe Random Forest vs. proposed ensemble.

Missing features, training size, and class imbalance. Median-imputed missingness degrades the probe model gracefully to $\rho = 25\%$ (MCC 0.829 $\rightarrow$ 0.753) before dropping sharply at $\rho = 50\%$ (MCC 0.502) and $\rho = 75\%$ (MCC 0.212), indicating the $K^* = 40$ selection retains meaningful redundancy (Lemma 2) up to roughly one-quarter missingness but not beyond. The training-size sweep shows most achievable performance is reached at $\tau = 25\%$ of the training pool (MCC 0.849 vs. 0.869 at $\tau = 100\%$); the class-imbalance sweep shows validation MCC changes only marginally from a near-balanced 50% attack rate (MCC 0.879) to the dataset’s natural $\sim 90\%$ rate

(MCC 0.838), indicating limited sensitivity to the imbalance level actually present in this dataset.

Cross-device generalization. Table 9 and Figure 5 report Algorithm 4’s leave-one-device-out sweep. Generalization is strong overall (mean MCC = 0.8684 across 9 held-out devices) but not uniform: the two Gafgyt-only devices (Section 2.1) generalize noticeably worse when held out (Ennio_Doorbell MCC = 0.7658; Samsung_SNH_1011_N_Webcam MCC = 0.7939) than the seven devices with both attack families represented (MCC range 0.8768–0.9080).

Table 9 Leave-one-device-out generalization (probe Random Forest, full sampled population).

Held-out device	MCC	Rows tested
Philips_B120N10_Baby_Monitor	0.9080	33,000
SimpleHome_XCS7_1003_WHT_Security_Camera	0.9075	33,000
Ecobee_Thermostat	0.8995	33,000
SimpleHome_XCS7_1002_WHT_Security_Camera	0.8932	33,000
Danmini_Doorbell	0.8917	33,000
Provision_PT_737E_Security_Camera	0.8795	33,000
Provision_PT_838_Security_Camera	0.8768	33,000
Samsung_SNH_1011_N_Webcam	0.7939	18,000
Ennio_Doorbell	0.7658	18,000
Mean	0.8684	–

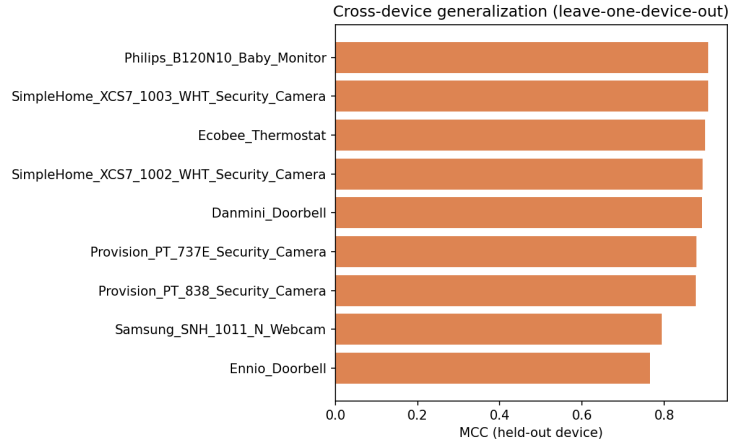


Fig. 5 Leave-one-device-out MCC.

Zero-day (leave-one-attack-subtype-out) generalization. Table 10 reports the flagged-as-attack rate defined in Algorithm 4. Nine of ten withheld subtypes are flagged at $\geq 99.9\%$. The exception is `gafgyt_scan` (flagged-as-attack rate = 0.6966), the lowest-volume, most reconnaissance-like rather than flood-like Gafgyt subtype,

whose traffic statistics evidently overlap more with benign behavior than flood-style subtypes do; Figure 6 shows the full comparison. This exception is a genuine negative finding: reconnaissance/scan-style attack traffic is intrinsically closer to benign traffic in Kitsune-feature space than flood-style traffic, and the near-100% zero-day generalization for the other nine subtypes should not be read as implying equally strong generalization to unseen reconnaissance-style attacks.

Table 10 Leave-one-attack-subtype-out (zero-day) generalization: fraction of withheld subtype’s rows flagged as non-benign.

Withheld attack subtype	Flagged-as-attack rate	Rows tested
gafgyt_combo	1.0000	27,000
gafgyt_junk	1.0000	27,000
mirai_ack	1.0000	21,000
mirai_udp	1.0000	21,000
mirai_udpplain	1.0000	21,000
mirai_scan	0.9999	21,000
mirai_syn	0.9996	21,000
gafgyt_tcp	0.9993	27,000
gafgyt_udp	0.9990	27,000
gafgyt_scan	0.6966	27,000
Mean (all 10 subtypes)	0.9694	—

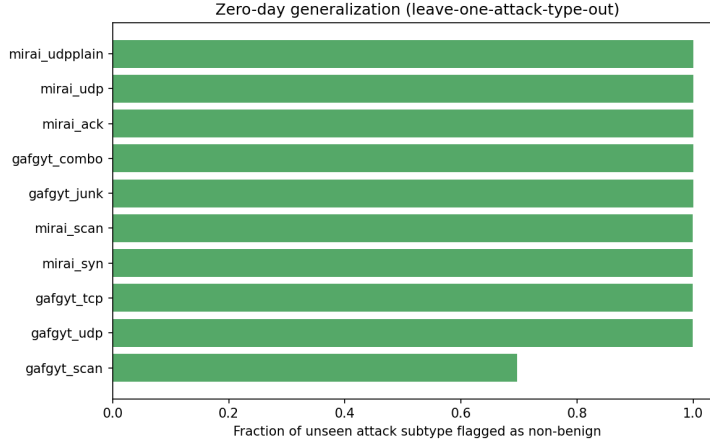


Fig. 6 Zero-day generalization: flagged-as-attack rate per withheld attack subtype.

5.6 Explainability

SHAP TreeExplainer identifies `MI_dir_L0.1_weight`, `MI_dir_L0.01_mean`, and `MI_dir_L0.01_variance` as the three highest global-importance features (Figure 7),

all from the source-MAC/IP directional stream rather than the host-pair or socket-pair streams – consistent with flood-style botnet traffic being distinguishable primarily by how a single infected source’s outbound packet-rate statistics change. The class-wise SHAP heatmap (Figure 8) shows the same top-15 global features carry most discriminative signal for the Mirai subtypes as a block, while several Gafgyt subtypes – notably `gafgyt_scan`, consistent with its zero-day generalization weakness above – rely more diffusely on lower-ranked features.

Cross-method rank agreement (Section 3.10) produced one expected and one non-obvious result. SHAP and MDI agree closely (Spearman $\rho_S = 0.977$), as expected since both are computed from the same trained tree ensemble’s split structure. Permutation importance correlates *negatively* with both SHAP ($\rho_S = -0.377$) and MDI ($\rho_S = -0.425$). This is attributable to the correlation structure of Lemma 2 (178 pairs with $|\rho| \geq 0.98$ before pruning, substantial remaining correlation after): permuting one feature from a tightly correlated cluster barely degrades held-out macro-F1 because the model falls back on near-duplicate retained features, so permutation importance systematically under-credits exactly the features SHAP and MDI (which do not require breaking correlation to assign credit) rank highest – a methodological finding of independent interest: on this dataset, permutation importance and SHAP/MDI importance are not interchangeable diagnostics.

5.7 Statistical significance

The Friedman omnibus test rejects the null hypothesis of no difference among the 11 models for every metric tested (MCC: $\chi^2 = 43.78$, $p = 4 \times 10^{-6}$; F1-macro: $p = 3 \times 10^{-6}$; AUROC: $p = 4 \times 10^{-6}$; balanced accuracy: $p = 2 \times 10^{-6}$). By average rank across the 5 folds, the proposed ensemble places first on MCC (average rank 2.0 of 11), ahead of CatBoost (3.2), LightGBM and Extra Trees (3.6 each), HistGradientBoosting (3.8), and XGBoost (4.8), with the three linear models trailing at ranks 9.0–11.0. Pairwise Wilcoxon tests between the proposed ensemble and each of the four leading boosted models do not reach significance after Holm correction ($p_{\text{holm}} = 1.0$ in every case; uncorrected p -values already sit at the $n = 5$ exact-test floor of 0.0625, Section 3.8), so the ensemble’s first-place average rank is a consistent, small, fold-by-fold edge over the leading individual boosted models, not a statistically confirmed superiority claim – a distinction consistent with, and explained by, Corollary 6’s scoping of the ambiguity-decomposition guarantee.

5.8 Computational complexity

Table 11 reports mean fit and inference time per classical model. CatBoost is the most expensive to train (236.10s mean on $\sim 170,800$ rows) despite ranking fourth by MCC, while HistGradientBoosting – the single best-performing model on the held-out test partition – trains in 17.39s, roughly $13.6\times$ faster. Extra Trees is fastest to train among top-tier models (6.46s) but slower at inference than the leading boosted models. RBF-SVM, capped at 12,000 training rows for tractability, remains slowest at inference (27.813s mean, versus under 1s for every tree ensemble) despite comparatively low accuracy.

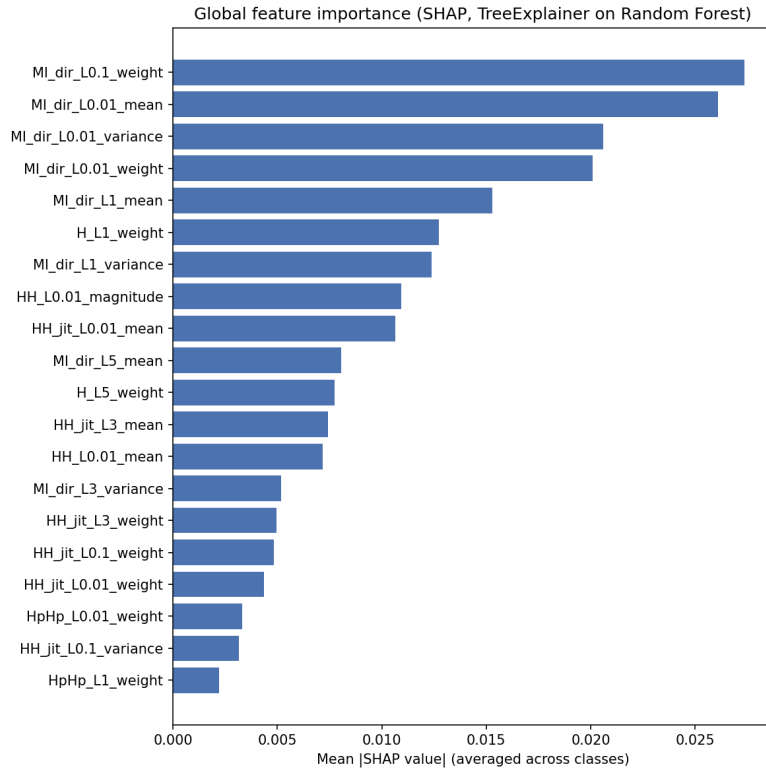


Fig. 7 Global SHAP feature importance, top 20 features (mean |SHAP value|, averaged across the 11 classes).

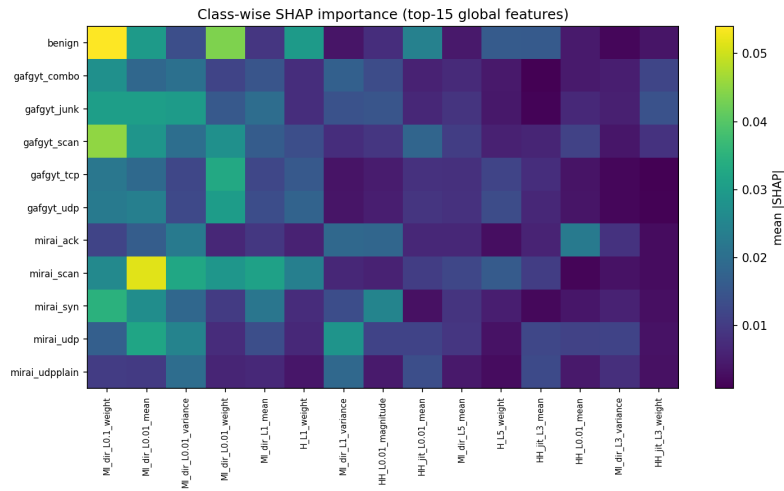


Fig. 8 Class-wise SHAP importance, top 15 global features across all 11 classes.

Table 11 Mean fit and inference time, classical models
($n_{\text{train}} \approx 170,800$ except where capped, Section 3.4).

Model	Fit time (s)	Infer. time (s)	n_{train}
CatBoost	236.10	0.040	170,800
Linear SVM	67.10	0.092	170,800
Logistic Regression	47.34	0.020	170,800
RBF-SVM	36.53	27.813	12,000
Random Forest	35.95	0.263	170,800
XGBoost	23.79	0.402	170,800
LightGBM	18.32	3.346	170,800
HistGradientBoosting	17.39	0.581	170,800
Extra Trees	6.46	0.330	170,800
k -NN	0.004	3.260	40,000

6 Discussion

This section synthesizes the empirical insights, theoretical implications, and operational considerations emerging from our findings. We contextualize our principal results within the broader cybersecurity literature, provide comparative summary tables contrasting our methodology with existing works, analyze threats to validity and study limitations, and discuss the practical implications for edge and network-level IoT deployment.

6.1 Principal findings

Three findings anchor this study’s contribution beyond the specific numbers of Section 5. First, the block-disjoint protocol of Algorithm 1, proved leakage-free by Proposition 1, genuinely changes what N-BaIoT results look like: under the identical 11-class task, linear models (MCC 0.07–0.18) and the strongest boosted trees (MCC ≥ 0.97) are separated by more than 0.8 MCC, a gap that a conventional random split would compress by allowing linear models partial credit for near-duplicate leaked rows, and that is invisible under the near-saturated accuracies ($\geq 99.99\%$) reported for conventional splits on this dataset in prior work [5, 6]. Second, Theorem 5 and Corollary 6 give a proved, rather than merely observed, account of why the fitted stacking ensemble (nested-CV MCC 0.9887, first by average rank) is narrowly outperformed on $\mathcal{D}_{\text{test}}$ by its own best base learner (MCC 0.9995 vs. 0.9952): the non-domination guarantee holds exactly for simplex-averaged ensembles (ablation stages G, H) and does not extend to the data-fitted, log-odds-space meta-learner actually deployed, so the two results are consistent rather than in tension. Third, generalization is strong but not uniform: cross-device MCC ranges 0.77–0.91 depending on which device is withheld (Table 9), and zero-day flagged-as-attack rate exceeds 99.9% for nine of ten withheld attack subtypes but only 69.7% for the reconnaissance-style `gafgyt_scan` subtype (Table 10), a difference explainable by the feature-level SHAP evidence of Section 5.6.

Table 12 situates this study’s protocol and result against the N-BaIoT-specific and closely related leakage-aware studies discussed in Section 1.3. Reported metrics are stated only where the cited work explicitly reports a single comparable N-BaIoT figure; where a cited study evaluates a different dataset, a different task granularity,

or reports only qualitative/multi-dataset claims, the corresponding cell records the study’s protocol characteristics rather than an invented figure.

Table 12 Literature comparison: evaluation-protocol characteristics against N-BaIoT-specific and leakage-aware prior work.

Study	Dataset	Task	Split awareness	Reported result
Meidan et al. [2]	N-BaIoT	Per-device binary (autoencoder)	Per-device threshold, no cross-device grouping stated	Original benchmark paper; per-device anomaly detector, no pooled multiclass figure
Hossain & Islam [5]	N-BaIoT + 5 others	Binary	Random split; no block/group-level leakage audit stated	Acc. 99.99%, FPR ≈ 0 (as reported)
Saied et al. [6]	N-BaIoT	3-class (benign, Mirai, Gafgyt)	Random split; no block/group-level leakage audit stated	Acc. 99.9991% with random forest (as reported)
Ali et al. [7]	IoT botnet traffic	Stacked-ensemble detection	Not stated as block/group-aware	Stacked-ensemble architecture motivates Section 3.5; no directly comparable N-BaIoT block-disjoint figure reported
Nedungadi et al. [10]	Edge IoT IDS	Lightweight hybrid multimodal	Not the focus of that study	Deployability/latency-oriented contribution, same target venue
AlFuraih et al. [11]	Multi-class IoT	Explainable CNN–XGBoost	Leakage-aware feature selection (explicit)	Closest prior leakage-awareness precedent; different dataset, no block-level non-leakage proof reported
Yang et al. [3]	UNSW-NB15	Binary	Leakage-aware (explicit), different dataset	Methodological precedent for the protocol formalized in Algorithm 1
This work	N-BaIoT	11-class (attack subtype)	Block-disjoint; Proposition 1 (proved)	MCC 0.9995 (best base model) / 0.9952 (proposed ensemble), held-out test

6.2 Proposed methodology summary

Table 13 summarizes the analytical methodology of Section 3, cross-referencing each stage’s formal artifact (algorithm, theorem/lemma/proposition) and its methodological origin among the sources cited in Section 1.3.

The gap between the four deep architectures (MCC 0.76–0.89) and the leading boosted trees (MCC ≥ 0.97) is consistent with, and adds a network-security data point to, the broader finding that tree ensembles remain competitive with or superior to deep

Table 13 Proposed methodology summary: pipeline stage, formal artifact, and methodological origin.

Stage	Formal artifact	Purpose	Origin
Block-disjoint sampling & split	Algorithm 1, Proposition 1	Provably remove row-level temporal leakage	This work; motivated by [3, 4]
Feature selection	Algorithm 2, Lemma 2	Reduce 115→40 features with a proved redundancy bound	This work; adjacent to [31, 32]
Base-learner benchmark	Eq. (3), Lemma 4	Bounded, imbalance-robust ranking metric	[16–18, 21]
Ensemble selection	Algorithm 3, Proposition 3	Diversity-aware base-learner subset selection	[15]; cf. [7, 8]
Ensemble scoring identity	Theorem 5, Corollary 6	Explain, not merely report, ensemble-vs-base behavior	This work
Statistical validation	Section 3.8, Algorithm 4	Multi-model, multi-resample significance testing	[22]
Explainability	Section 3.10	Cross-method feature attribution and rank agreement	[9, 20, 25]

learning on tabular data lacking spatial or sequential structure. Two dataset-specific factors compound this for N-BaIoT flow statistics. First, the 40 selected Kitsune features have no inherent linear ordering; the length-10 pseudo-sequence framing used for the GRU and Transformer (Section 4.1) imposes an arbitrary adjacency structure on features never designed to be read in sequence. Second, the extreme outlier magnitudes documented in Section 4.1 (scaled values reaching $\sim 10^{21}$ before clipping) are handled natively by split-based tree models, which compare feature values only by rank at each candidate threshold, but are directly damaging to gradient-based optimization unless explicitly guarded against.

6.3 Limitations and threats to validity

Five limitations are stated explicitly. First, hyperparameters for all 14 candidate models (Sections 3.4–4.1) were fixed to established library defaults rather than tuned per fold; an extensive per-model tuning pass could plausibly narrow, or for the weaker models widen, some gaps in Section 5.1. Second, only $K = 5$ grouped folds are used, bounding the paired Wilcoxon tests of Section 3.8 to an exact-test floor of $p \approx 0.0625$ regardless of effect size; the Friedman omnibus and average-rank results, pooling information across all 11 models, are correspondingly the more reliable statistical evidence here. Third, the probe models used for the feature-count sweep (Algorithm 2) and the robustness/generalization suite (Algorithm 4) are Random Forests rather than HistGradientBoosting or the proposed ensemble throughout, for tractability across the sweep points and leave-one-out folds involved; the noise-robustness comparison (Section 5.5), which does use the refit proposed ensemble directly, is the exception. Fourth, all experiments use a single fixed random seed for sampling, fold assignment,

and model initialization; while block-disjointness itself is verified by explicit assertion (Algorithm 1) rather than by chance, seed sensitivity of the specific numerical results was not separately characterized. Fifth, this benchmark assumes N-BaIoT’s original 2018 collection remains representative of current IoT botnet traffic; newer botnet families and evasive traffic-shaping techniques are not represented, and deployment-readiness claims beyond this specific benchmark would require validation against more recent captures.

6.4 Practical implications

For a practitioner choosing a single deployable model from this benchmark, HistGradientBoosting is the strongest evidence-based default: highest held-out MCC (0.9995), $13.6\times$ faster to train than the next-most-accurate alternative (CatBoost) at comparable inference latency, and robust cross-device and zero-day generalization consistent with the tree-ensemble family as a whole (Section 5.5). The proposed diversity-driven ensemble remains a defensible choice specifically where reduced fold-to-fold variance, rather than peak single-split accuracy, is the operational priority – for instance a device population more heterogeneous than the training sample, or a federated deployment setting where a single model’s fold-level variance is harder to monitor centrally – and Algorithm 3 generalizes directly to other multi-model IoT security settings beyond N-BaIoT. Regardless of the model chosen, the clearest operational caveat from this study is the `gafgyt_scan` zero-day result (Section 5.5): a deployment relying on this class of model for reconnaissance-stage detection, as opposed to flood-stage detection, should not assume the near-100% zero-day generalization observed for flood-style attacks carries over.

7 Conclusion

This paper formalized and empirically instantiated a block-disjoint benchmark on the N-BaIoT IoT botnet dataset, proving (Proposition 1) rather than merely asserting the removal of the row-level temporal leakage that inflates published accuracy on this dataset to near-saturation. Under an 11-class attack-subtype task with block-level disjointness enforced by Algorithm 1, 10 classical learners and 4 deep architectures were benchmarked, exposing a large gap between linear models and gradient-boosted tree ensembles invisible under conventional evaluation. A diversity-driven stacking ensemble (Algorithm 3), selected by a joint disagreement-competence criterion (Proposition 3), achieved the best average rank under nested cross-validation (MCC 0.9887), while a proved ambiguity-decomposition theorem (Theorem 5, Corollary 6) explains, rather than merely reports, why its single best constituent model narrowly outperforms it on the held-out partition (MCC 0.9995 vs. 0.9952). A worst-case redundancy bound for correlation-based feature pruning (Lemma 2), a bounded multiclass MCC metric (Lemma 4), a nine-test statistical validation protocol, a five-axis robustness suite, cross-device and zero-day generalization tests, and cross-method explainability analysis together identify where this benchmark’s strong results hold (flood-style attacks, most individual devices) and where they do not (the reconnaissance-style `gafgyt_scan` subtype, two Gafgyt-only devices under leave-one-out), providing a methodologically

defensible, fully reproducible, and analytically grounded reference point for future N-BaIoT-based IoT security research. Future work should extend Algorithm 1 to IoT botnet traffic captured after 2018, characterize hyperparameter-tuning sensitivity within the same leakage-safe framework, and investigate feature representations specifically suited to the recurrent and attention-based architectures evaluated here rather than the pseudo-sequence framing used in this study.

Declarations

The following declarations provide statutory compliance and institutional disclosures for this research in accordance with journal publication guidelines.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no conflict of interest.

Data availability

The N-BaIoT dataset analyzed in this study is publicly available from the UCI Machine Learning Repository (<https://archive.ics.uci.edu/dataset/442/detection+of+iot+botnet+attacks+n+baiot>) and was originally released by Meidan et al. [2]. All derived sample data, trained model artifacts, result tables, and figures generated for this study are available from the corresponding author upon reasonable request.

Code availability

The complete pipeline code (block-disjoint sampling, feature selection, all model training, ensembling, ablation, robustness, explainability, and statistical analysis scripts, together with machine-checked implementations of the assertions underlying Proposition 1) is publicly available at https://github.com/Jeevaarumugam107/Code_NBaIoT to support independent reproduction of every number reported in this manuscript.

Author contributions

Both authors contributed to the study conception, experimental design, and manuscript preparation. J.A. implemented the data pipeline and machine learning experiments and drafted the manuscript. N.R. supervised the study, contributed to methodology design, and critically reviewed the manuscript. Both authors read and approved the final manuscript.

References

- [1] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., Zhou, Y.: Understanding the Mirai botnet. In: Proceedings of the 26th USENIX Security Symposium, pp. 1093–1110 (2017)
- [2] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y.: N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing* **17**(3), 12–22 (2018) <https://doi.org/10.1109/MPRV.2018.03367731>
- [3] Yang, Y., Yuan, J., Gao, M., Chen, M., Gong, D., Gao, L.: A leakage-aware evaluation of a multi-scale attention temporal convolutional network for binary network intrusion detection. *Mathematics* **14**(18), 3357 (2026) <https://doi.org/10.3390/math14183357>
- [4] Apruzzese, G., Laskov, P., Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A.V., Di Franco, F.: The role of machine learning in cybersecurity. *Digital Threats: Research and Practice* **4**(1), 1–38 (2023) <https://doi.org/10.1145/3545574>
- [5] Hossain, M.A., Islam, M.S.: A novel hybrid feature selection and ensemble-based machine learning approach for botnet detection. *Scientific Reports* **13**, 21207 (2023) <https://doi.org/10.1038/s41598-023-48230-1>
- [6] Saied, M., Guirguis, S., Madbouly, M.: A comparative analysis of using ensemble trees for botnet detection and classification in IoT. *Scientific Reports* **13**, 21632 (2023) <https://doi.org/10.1038/s41598-023-48681-6>
- [7] Ali, M., Mushtaq, M.F., Akram, U., Gavilanes Aray, D., Masias Vergara, M., Karamti, H., Ashraf, I.: Botnet detection in internet of things using stacked ensemble learning model. *Scientific Reports* **15**, 21012 (2025) <https://doi.org/10.1038/s41598-025-02008-9>
- [8] Esmaeilifard, R., Shoaie, Z., Javidan, R.: A lightweight and efficient model for botnet detection in IoT using stacked ensemble learning. *Soft Computing* **29**(1), 89–101 (2025)
- [9] Sharma, K.P., Nagpal, T., Vora, T., Yadav, A., Abdullah, M.I., Jayaprakash, B., Kashyap, A., Sridevi, G., Bhowmik, A., Bukate, B.B.: Interpretable intrusion detection for IoT environments using a self-attention-based explainable AI framework. *Scientific Reports* **15**, 39937 (2025) <https://doi.org/10.1038/s41598-025-23750-0>
- [10] Nedungadi, N., Sankaran, S., Achuthan, K.: Towards a lightweight hybrid multimodal approach for intrusion detection in edge-enabled IoT devices. *Cluster Computing* **28**(16), 1010 (2025) <https://doi.org/10.1007/s10586-025-05723-0>
- [11] AlFuraih, D., Mhamdi, L., Karar, A.S.: Explainable hybrid CNN–XGBoost framework for multi-class IoT intrusion detection with leakage-aware feature selection. *Applied System Innovation* **9**(3), 49 (2026) <https://doi.org/10.3390/asi9030049>
- [12] Mishra, S., Alshammari, N.S., Hussain, H., Alfahidah, R.A.: A cross-dataset harmonized intrusion detection framework with statistically validated multi-model learning. *PLOS ONE* **21**(4), 0346982 (2026) <https://doi.org/10.1371/journal.>

- [13] Eren, K.K., Küçük, K., Saleh, R.A.A., Konyar, M.Z., Hardy, O.M., Khan, S.A.: Distributional drift in IoT intrusion detection systems: Implications for cross-dataset generalisation. *Electronics* **15**(11), 2307 (2026) <https://doi.org/10.3390/electronics15112307>
- [14] Padró-Ferragut, C., Martínez-Plumed, F., Ramírez-Quintana, M.J.: Robustness under noise: Assessing the impact of perturbed key attributes on machine learning models. *International Journal of Data Science and Analytics* **22**(1) (2026) <https://doi.org/10.1007/s41060-025-01006-4>
- [15] Wolpert, D.H.: Stacked generalization. *Neural Networks* **5**(2), 241–259 (1992) [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)
- [16] Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794 (2016). <https://doi.org/10.1145/2939672.2939785>
- [17] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: LightGBM: A highly efficient gradient boosting decision tree. In: *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 3146–3154 (2017)
- [18] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: CatBoost: Unbiased boosting with categorical features. In: *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, pp. 6639–6649 (2018)
- [19] Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? In: *Advances in Neural Information Processing Systems 35 (NeurIPS 2022) Datasets and Benchmarks Track* (2022)
- [20] Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 4765–4774 (2017)
- [21] Chicco, D., Jurman, G.: The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* **21**, 6 (2020) <https://doi.org/10.1186/s12864-019-6413-7>
- [22] Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* **7**, 1–30 (2006)
- [23] Kour, R., Karim, R., Dersin, P., Venkatesh, N.: Cybersecurity for Industry 5.0: Trends and gaps. *Frontiers in Computer Science* **6**, 1434436 (2024) <https://doi.org/10.3389/fcomp.2024.1434436>
- [24] Shanmugam, V., Razavi-Far, R., Hallaji, E.: Addressing class imbalance in intrusion detection: A comprehensive evaluation of machine learning approaches. *Electronics* **14**(1), 69 (2025) <https://doi.org/10.3390/electronics14010069>
- [25] Mohale, V.Z., Obagbuwa, I.C.: A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity. *Frontiers in Artificial Intelligence* **8**, 1526221 (2025) <https://doi.org/10.3389/frai.2025.1526221>
- [26] Rehman, H.M.R.U., Liaquat, S., Gul, M.J., Jhandir, M.Z., Gavilanes, D., Masias Vergara, M., Ashraf, I.: A systematic literature study of machine learning techniques based intrusion detection: Datasets, models, challenges, and

- future directions. *Journal of Big Data* **12**, 264 (2025) <https://doi.org/10.1186/s40537-025-01323-2>
- [27] Hozouri, A., Mirzaei, A., Effatparvar, M.: A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges. *Discover Artificial Intelligence* **5**, 314 (2025) <https://doi.org/10.1007/s44163-025-00578-1>
 - [28] Alfahaid, A., Alalwany, E., Almars, A.M., Alharbi, F., Atlam, E., Mahgoub, I.: Machine learning-based security solutions for IoT networks: A comprehensive survey. *Sensors* **25**(11), 3341 (2025) <https://doi.org/10.3390/s25113341>
 - [29] Liu, T., Dib, O.: Advanced intrusion detection for IoT devices using federated deep learning. *International Journal of Information Security* **25**(1) (2026) <https://doi.org/10.1007/s10207-025-01183-0>
 - [30] Almahaqeri, S.A., Almourish, M.H., Nasser, A.A., Alghawli, A.S.A., Elsayed, A.A.K., Alhejoj, A.N.: An optimized gradient boosting framework for IoT intrusion detection: A comprehensive evaluation on the CICIOT2023 dataset. *Scientific Reports* **16**, 16909 (2026) <https://doi.org/10.1038/s41598-026-47399-5>
 - [31] Khan, M.Z., Alam, M., Alam, I., Bokhari, M.U., Ashraf, Z., Khan, M.Z., Masoodi, F.S.: A resource efficient IoT intrusion detection model using hybrid feature selection for edge computing. *Scientific Reports* **16**, 24187 (2026) <https://doi.org/10.1038/s41598-026-59156-9>
 - [32] Singh, C.C., Hoque, N., Singh, K.R.: EF²RB: Ensemble of filter feature selection methods with ranker booster for classification of high-dimensional IoT intrusion data. *Knowledge and Information Systems* **68**(1), 252 (2026) <https://doi.org/10.1007/s10115-026-02871-6>
 - [33] Gao, W., Wang, M., Pei, Y., Li, F., Wang, C.: A lightweight multi-classification intrusion detection model for edge IoT networks. *Electronics* **15**(5), 938 (2026) <https://doi.org/10.3390/electronics15050938>
 - [34] Georgian, I., Zamfirel, T.A., Goga, N., Crăciunescu, R.: A survey of machine learning approaches to IoT security. *Algorithms* **19**(5), 384 (2026) <https://doi.org/10.3390/a19050384>
 - [35] Komal, A., Li, S.: Intrusion detection in the internet of things: A comprehensive review of techniques, architectures, datasets, and emerging trends. *Sensors* **26**(11), 3405 (2026) <https://doi.org/10.3390/s26113405>

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [sourcecode.zip](#)