

Source-Aware Evaluation of Tree-Based Intrusion Detection on MQTT-Associated IoT Devices

Bayu Setiaji, Haryoko, Windha Mega Pradnya Dhuhi, Eli Pujastuti, Lukman, and Afrig Aminudin

Abstract—*Packet-level intrusion-detection benchmarks can reward capture-specific signals and conceal differences in class coverage. This study audits a tree-model benchmark on Gotham Dataset 2025, treating MQTT association as device scope. Reconstruction identifies 56 source files containing 8,629,880 packets and reproduces the archived 112,845-row experiment. Removing 2,585 unresolved Unknown rows leaves 110,260 packets for revised evaluation. Seven fixed configurations are compared over five random and five source-disjoint splits with explicit class-coverage requirements. Multiclass macro F1 is 0.7295 ± 0.1150 under random splitting and 0.6455 ± 0.0211 under source separation for the ensemble, compared with 0.8247 ± 0.0094 and 0.6617 ± 0.0207 for XGBoost-200. XGBoost-200 therefore outperforms the ensemble in mean multiclass F1 under both protocols. Source scores use a fixed 15-label denominator. The original holdout score used a different averaging set and cannot establish a directly comparable generalization gap. Feature exclusions, uncapped testing, chronological evaluation, and probability diagnostics reveal further limitations. The contribution is a reproducible evaluation protocol; the results establish neither an ensemble accuracy-efficiency advantage nor operational edge-device performance.*

Index Terms— class coverage; dataset auditing, IoT intrusion detection, MQTT-associated devices; source-disjoint evaluation, tree models.

I. INTRODUCTION

MQTT supports publish-subscribe IoT communication [1]. MQTT-associated devices also exchange discovery, management, and other transport traffic. Generic flooding, scanning, brute force, and command-and-control labels therefore do not validate detection of malformed CONNECT messages, topic abuse, or broker-authentication attacks.

Gotham Dataset 2025 provides distributed device captures from an emulated IoT environment [2]. These captures allow

model comparisons across source devices, but random packet splitting can mix correlated observations. Removing addresses and filenames also leaves header attributes that may identify device configurations or overlap with annotation rules.

The submitted tree-ensemble study already favored XGBoost and the single LightGBM component in binary performance and computational cost. The contribution is procedural: (i) exact reconstruction of data scope and archived results; (ii) common-representation, unweighted model comparisons; (iii) repeated source-aware evaluation with explicit class coverage and uncertainty; and (iv) diagnostic checks of feature dependence, probability quality, and cost. No new classifier or explanation method is proposed.

II. RELATED WORK

MQTT-IoT-IDS2020 distinguishes packet and flow representations [3]. MQTT attack classification and feature-engineered ensembles are also established [4], [5]. Those protocol-focused settings differ from the device-scoped selection examined here. CICIoT2023, TON_IoT, N-BaIoT, Kitsune, and Bot-IoT cover other traffic-generation and detection settings [6], [7], [8], [9], [10]. Gotham builds on a reproducible testbed [11]. Numerical results across these datasets do not constitute a common leaderboard.

Recent transformer and hybrid approaches broaden the model space [12], [13], while LightGBM, XGBoost, Random Forest, and Extra Trees remain relevant tabular comparators [14], [15], [16], [17]. Security evaluation must account for dependence and artifacts [18]. TCP/IP window properties, flags, and TTL can support fingerprinting [19], while correlated predictors complicate permutation importance [20]. We use feature importance to audit predictive dependence, without claiming the explanation fidelity or analyst utility addressed by explainable-security research [21], [22].

¹Manuscript received July 25, 2026.

Bayu Setiaji is with the Department of Informatics, Faculty of Computer Science, Universitas Amikom Yogyakarta, Sleman, 55283, Indonesia; (corresponding author; e-mail: bayusetaji@amikom.ac.id).

Haryoko is with the Department of Information Technology, Faculty of Computer Science, Universitas Amikom Yogyakarta, Sleman, 55283, Indonesia; (e-mail: haryoko@amikom.ac.id).

Windha Mega Pradnya Dhuhi is with the Department of Informatics, Faculty of Computer Science, Universitas Amikom Yogyakarta, Sleman, 55283, Indonesia; (e-mail: windha@amikom.ac.id).

Eli Pujastuti is with the Department of Informatics, Faculty of Computer Science, Universitas Amikom Yogyakarta, Sleman, 55283, Indonesia; (e-mail: eli@amikom.ac.id).

Lukman is with the Department of Informatics Management, Faculty of Computer Science, Universitas Amikom Yogyakarta, Sleman, 55283, Indonesia; (e-mail: masman@amikom.ac.id).

Afrig Aminudin is with the Department of Information Systems, Faculty of Computer Science, Universitas Amikom Yogyakarta, Sleman, 55283, Indonesia; (e-mail: afrig@amikom.ac.id).

III. DATA AND EVALUATION PROTOCOL

This section defines the data scope, preprocessing, models, partitions, and metrics used to distinguish predictive performance from capture dependence. Figure 1 summarizes the separation between training and evaluation.

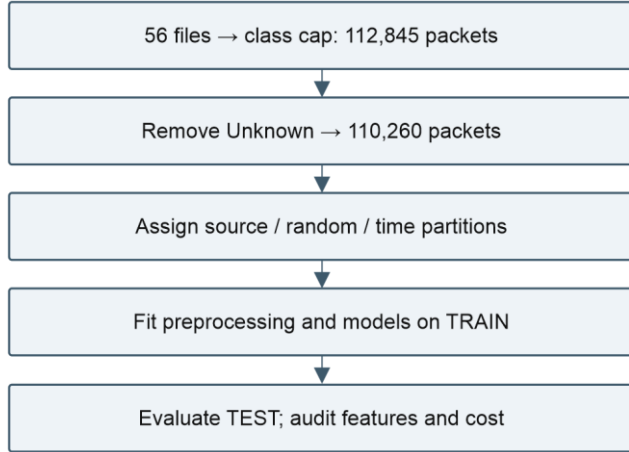


Fig. 1. Source-Aware Evaluation Workflow

Source identities and timestamps determine partitions but are excluded from predictors. Transformations are fitted on training data; the uncapped check reuses the fitted source-split models.

A. Scope, Sampling, and Labels

The processed CSV archive is identified by Zenodo DOI 10.5281/zenodo.14502760, issued on 5 February 2025 [2]; the dataset paper appeared in 2026 [23]. The original loader sorts unique absolute matching paths, lowercases them, replaces hyphens with underscores, and retains paths containing mqtt, air_quality, building_monitor, cooler_motor, domotic_monitor, hydraulic_system, or predictive_maintenance. It retains all packets in those files. This selects 56 files and 8,629,880 packets, of which 162,123 (1.88%) explicitly contain MQTT in their protocol strings. The earlier broader profiler counted 60 files and is not the modelling denominator. Selection pseudocode, all filenames, hashes, and class/family counts are supplied in Supplement S1.

Historical sampling normalizes labels, samples each class without replacement to at most 10,000 packets with seed 42, retains smaller classes, and sorts selected global row indices. This fixed sample contains 112,845 packets, including 10,000 benign packets (8.86%). The cap controls computational cost and does not estimate operational prevalence.

The inspected upstream labeller initializes unmatched packets as Unknown [24]. Because this is an unresolved annotation residual, its 2,585 rows are retained only for historical verification and removed before every revised fit and test. The revised sample contains 110,260 packets, 15 named classes, and 10,000 benign packets (9.07%). A named

label is not independent verification of an attack. The exact labeller commit used for the downloaded archive is unknown; the inspected implementation is pinned in the supplement. Binary classification distinguishes benign traffic from named attacks.

B. Features and Preprocessing

Eight raw predictors are retained: frame length; IP flags, TTL, and protocol; TCP flags; TCP window value and scale factor; and TCP-PDU size. Addresses, ports, timestamps, protocol strings, source identifiers, and checksums are excluded. Training-fitted preprocessing applies median imputation and standardization to numeric fields and most-frequent imputation with one-hot encoding to categorical fields. Revised fits remove training-constant columns. Unseen categories are ignored.

The archived representation has 16 encoded columns: a request for the top 20 therefore retains only 16. Historical feature ranking uses mutual information without an explicit redundancy term; it is not mRMR [25]. The selected binary ensemble uses ten columns, and its multiclass counterpart uses all 16. Revised base models share eight raw fields and 16 encoded columns across all accepted source fits, without feature-count selection; per-fit counts accompany Supplement S2.

C. Models and Comparison Fairness

The ensemble combines Extra Trees and Random Forest, each with 60 trees, depth 8, and minimum leaf size 4; a Decision Tree with depth 6 and minimum leaf size 8; and LightGBM with 100 iterations, depth 5, 23 leaves, and learning rate 0.05. Prediction is

$$\hat{y}(x) = \arg \max_{c \in C} \frac{\sum_{m=1}^4 w_m p_m(c | x)}{\sum_{m=1}^4 w_m} \quad (1)$$

where x is the feature vector, $\hat{y}(x)$ the predicted label, C the class set, and c a candidate class. For member m , $p_m(c|x)$ is its class probability and w_m its weight. Members 1–4 are Extra Trees, Random Forest, Decision Tree, and LightGBM, with weights (3, 2, 1, 4). The denominator normalizes the weights; $\arg \max$ selects the largest weighted probability. These heuristic weights are neither learned nor calibrated, and LightGBM is required to reproduce this four-member configuration.

The archived ensemble selected among six combinations of requested feature count $k \in \{10, 15, 20\}$ and balanced/unweighted fitting using validation macro F1, then refitted on training plus validation data. Baselines used fixed settings and all 16 columns. Random Forest used balanced_subsample; Extra Trees, LightGBM-200, and histogram gradient boosting used balanced weights; XGBoost was unweighted. This historical comparison was asymmetric.

Seven unweighted configurations are fixed without search: the ensemble, its LightGBM-100 component, XGBoost-200, LightGBM-200, Random Forest-200, Extra Trees-200, and histogram gradient boosting. The forests use 200 full-depth trees. XGBoost uses depth 6, learning rate 0.05, row/column subsampling 0.9, and histogram construction. LightGBM-200

uses 63 leaves, learning rate 0.05, and column subsampling 0.9; `subsample_freq=0` leaves row bagging inactive. Histogram boosting uses 160 iterations, 31 leaves, learning rate 0.05, and L2 regularization 0.01. Complete estimator settings are supplied in Supplement S2.

D. Partitions and Class Coverage

For seeds 42–46, GroupShuffleSplit proposes partitions with 14 test and 42 training files. The first proposal containing benign traffic and at least ten named multiclass labels on both sides is accepted. This rule was introduced after auditing unrestricted splits and before revised model comparison; it uses labels, never model scores. Unrestricted seeds 43, 44, and 46 yield benign-only tests and cannot assess attack detection. Those pilot results remain in the audit supplement.

TABLE I
SOURCE-DISJOINT PARTITIONS AND CLASS COVERAGE

SEED	PROPOSAL	TRAIN/TEST	CLASSES	UNSEEN
		ROWS	TRAIN/TEST	ROWS
42	3	47,002 / 63,258	13 / 15	10,015
43	7	70,078 / 40,182	15 / 13	0
44	2	47,181 / 63,079	13 / 15	10,015
45	8	43,233 / 67,027	13 / 15	10,015
46	2	47,255 / 63,005	13 / 15	10,015

Note. Each split has 42 training and 14 test files. Class counts are multiclass; binary fits reuse these assignments. Unseen rows have a target label absent from training. Proposal is the first coverage-eligible candidate.

Table I exposes unequal packet/class coverage: four splits include 10,015 unseen-label test packets. Files identify distinct emulated devices, while families and scenarios may remain shared. These overlapping group holdouts are not k-fold cross-validation or independent datasets; inference concerns eligible source transfers.

The same seeds generate task-stratified random 70%/15%/15% partitions. Revised fixed models fit the combined first two partitions (85%); the middle partition is not used for tuning. Model seeds follow split seeds, while the capped sample remains fixed at seed 42. Variation therefore combines splitting and fitting variation conditional on one sample.

An uncapped check evaluates every named-label packet in the accepted seed-42 test files using the capped-training models without refitting. A chronological diagnostic sorts the confirmed sample by UTC time and places the last 15% in testing, moving all boundary ties to the test side. Training timestamps strictly precede test timestamps. This diagnostic also changes scenario and label composition.

E. Metrics, Uncertainty, and Cost

Macro F1 over a declared label set is

$$F_{1,c} = \frac{1}{|C|} \sum_{c \in C} \frac{2TP_c}{2TP_c + FP_c + FN_c} \quad (2)$$

where C is the included label set, $|C|$ its size, and TP_c , FP_c , and FN_c the true-positive, false-positive, and false-negative counts for class c . Every class receives equal weight; a zero denominator contributes zero. Fixed-label F1 uses 15 multiclass or two binary labels. Supported-label F1 includes

only true test labels. The former penalizes absent labels, while the latter changes its denominator with coverage. Historical default-union scores are preserved separately.

Attack is the positive binary class. ROC-AUC and non-interpolated average precision (AP) summarize ranking [26].

$FPR = \frac{FP}{FP+TN}$ is the fraction of benign packets flagged as

attacks, and $FNR = \frac{FN}{FN+TP}$ is the fraction of attacks missed.

Here TN denotes correctly rejected benign packets; the other counts follow their binary meanings. Metrics use the classifier's default decision rule. Binary Brier score averages $(p_{\text{attack}} - y)^2$, where p_{attack} is the attack probability and y is 1 for attack and 0 for benign. Log loss and ten-bin equal-width expected calibration error (ECE) supplement this probability-error measure; none guarantees calibration after deployment.

We report five-run means and sample standard deviations. Conditional 95% source-cluster intervals use 2,000 resamples of held-out files, retaining the fixed class denominator. Archived random-test intervals use 2,000 class-stratified packet resamples, which omit packet dependence. Paired model differences use 5,000 identical source-cluster resamples. These intervals condition on fitted models and cannot quantify new-scenario or retraining uncertainty; inclusion of zero does not establish equivalence.

For assumed attack prevalence π , expected alert precision is

$$\text{Precision}_\pi = \frac{\pi(1 - FNR)}{\pi(1 - FNR) + (1 - \pi)FPR} \quad (3)$$

where π is the attack proportion and FPR/FNR retain the definitions above. The numerator is the proportion of correctly detected attacks and the denominator the proportion of all alerts; precision is undefined if no alerts occur. Expected false alerts per million packets are $10^6(1 - \pi)FPR$, where $1 - \pi$ is the benign proportion. Scenarios $\pi \in \{0.001, 0.01, 0.05\}$ assume unchanged class-conditional performance and decision rules; they are not measured deployment priors.

Archived models are timed together on the first 1,000 test rows, with three warm-ups and 15 timed predictions using `time.perf_counter_ns`. We report median and first/third quartiles of amortized milliseconds per packet; preprocessing is included, while CSV parsing, capture, and feature extraction are excluded. Storage is compressed model size, not RAM. The workstation has Windows 11, eight logical CPUs, 31.95 GB RAM, Python 3.12.10, scikit-learn 1.8.0, LightGBM 4.6.0, and XGBoost 3.3.0. Revised fitting uses four computational threads and sequential ensemble members. No gateway, energy, or live-stream measurements are available.

IV. RESULTS AND DISCUSSION

The results first compare source-disjoint performance, then audit historical scores, retained features, decision errors, and alternative partitions. Scores are interpreted with class support and measurement scope.

A. Primary Source-Disjoint Comparison

Table II gives XGBoost-200 the highest mean fixed-label source multiclass F1, 0.6617 ± 0.0207 , versus 0.6455 ± 0.0211 for the ensemble. Source weighted F1 is 0.7517 and 0.7455, respectively, illustrating frequent-label influence. Random multiclass and source binary scores also favor XGBoost. Rankings remain conditional on the accepted captures.

TABLE II
REPEATED PREDICTIVE PERFORMANCE (MEAN $\pm$ SD)

MODEL	RANDOM MC	SOURCE MC	SOURCE SUPPORTED	SOURCE BINARY
Ensemble	0.7295 $\pm$ 0.1150	0.6455 $\pm$ 0.0211	0.6666 $\pm$ 0.0680	0.9335 $\pm$ 0.0317
LGBM-100	0.6946 $\pm$ 0.2315	0.6610 $\pm$ 0.0170	0.6822 $\pm$ 0.0645	0.9750 $\pm$ 0.0090
XGB-200	0.8247 $\pm$ 0.0094	0.6617 $\pm$ 0.0207	0.6832 $\pm$ 0.0688	0.9817 $\pm$ 0.0050
LGBM-200	0.2800 $\pm$ 0.3026	0.5559 $\pm$ 0.2361	0.5772 $\pm$ 0.2553	0.9821 $\pm$ 0.0054
RF-200	0.8258 $\pm$ 0.0099	0.6438 $\pm$ 0.0142	0.6644 $\pm$ 0.0602	0.9820 $\pm$ 0.0053
ET-200	0.8257 $\pm$ 0.0099	0.6476 $\pm$ 0.0219	0.6687 $\pm$ 0.0692	0.9511 $\pm$ 0.0647
HistGB	0.8231 $\pm$ 0.0091	0.6515 $\pm$ 0.0280	0.6727 $\pm$ 0.0714	0.9639 $\pm$ 0.0355

Note. Five seeds per entry. MC uses 15 fixed labels; supported MC uses true test labels. Binary uses two labels. LGBM: LightGBM; XGB: XGBoost; RF: Random Forest; ET: Extra Trees; HistGB: histogram gradient boosting. Suffixes denote tree/iteration counts.

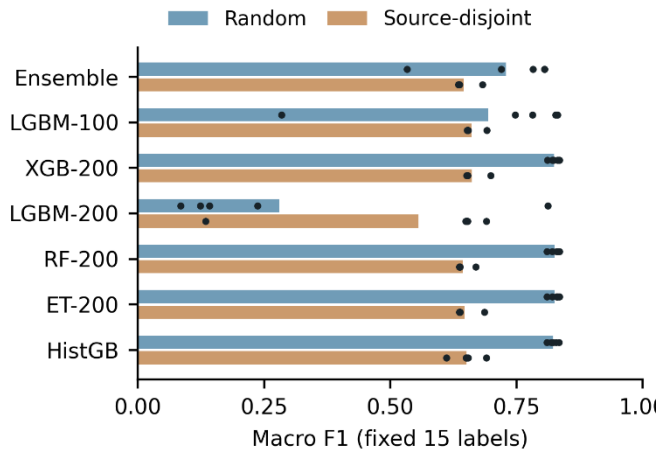


Fig. 2. Multiclass performance across evaluation protocols.

Paired bars show five-run means using all 15 labels; dots show individual runs. The spread reveals configuration sensitivity that means alone conceal.

LightGBM-200 random multiclass F1 ranges from 0.0851 to 0.8119; the ensemble ranges from 0.5336 to 0.8058 (Figure 2). A fresh fit reproduces the poor LightGBM-200 source seed-44 score of 0.134645 with finite, normalized probabilities. No poor run is removed. The check verifies reproducibility, not the cause of instability. Source intervals are broad: seed-42 fixed-label multiclass F1 is 0.6356 [0.0633, 0.6374] for the ensemble and 0.6522 [0.0637, 0.6529] for

XGBoost. Few attack-bearing clusters and resampled class coverage constrain inference (Supplement S2).

B. Historical Accuracy and Computational Cost

All 42 archived random-split scores are recovered from saved models on reconstructed test rows, agreeing to floating-point precision. Table III retains Unknown and historical weighting; its numbers must not be mixed with the revised 15-label analysis. The original comparison and ablation tables timed models separately and retained only summary latency; their differing values cannot establish measurement dispersion. We replace both with the common-session measurements defined in Section III-E. Complete component removals and feature-count variants are supplied in Supplement S3.

TABLE III
ARCHIVED PREDICTION QUALITY AND MODEL COST

TASK/MODEL	MACRO F1	MS/PACKET [Q1,Q3]	MODEL KiB
Binary / ensemble	0.9598	0.110 [0.110, 0.112]	297.1
Binary / XGB-200	0.9865	0.018 [0.018, 0.020]	90.7
Binary / LGBM-only	0.9849	0.021 [0.019, 0.023]	92.4
MC / ensemble	0.8302	0.132 [0.129, 0.138]	2,051.7
MC / XGB-200	0.8312	0.040 [0.038, 0.068]	1,088.8
MC / LGBM-only	0.8304	0.050 [0.048, 0.054]	1,519.8

Note. Unknown is retained. Latency is median [first, third quartile] amortized batch cost including preprocessing. KiB=1,024 bytes of compressed storage; neither cost measures operational edge performance.

XGBoost and the LightGBM-only model outperform the binary ensemble while using less stored space and lower measured inference time. Multiclass XGBoost-ensemble macro F1 is 0.0010, with paired source-cluster 95% interval $[-0.0009, 0.0031]$. The binary difference is 0.0267 [0.0065, 0.0685]. The multiclass interval does not establish equivalence. Removing the ensemble's Decision Tree yields 0.8304 multiclass F1, also undermining a claim that every member is necessary. The evidence supports reconsidering ensemble complexity, not an accuracy-efficiency advantage.

C. Why the Original Holdout was Not Comparable

The original unrestricted holdout trains on 106,217 packets and tests on 6,628 from 14 files: 1,657 benign, 3,071 Telnet brute force, 934 TCP scan, 109 ingress transfer, and 857 Unknown. Air-quality-1 supplies 5,050 test packets and all non-benign cases; the remaining files contain only benign traffic. The full filename/count table appears in Supplement S3. Its reported multiclass F1=0.5949 averages over seven labels in the union of targets and predictions. The same predictions score 0.2603 over all 16 historical labels and 0.8329 over the five supported labels. Comparing 0.5949 directly with the 16-label random score of 0.8302 confounds transfer with the averaging denominator and class coverage.

The historical binary holdout scores 0.9667, compared with random F1=0.9598, with FPR falling from 0.1373 to 0.0966 while FNR remains zero. Its benign share is 25%, versus 8.86% in the sampled population, and no flooding class occurs. This different attack mixture and lower observed FPR explain why source separation need not lower aggregate F1 on this particular test.

D. Residual Feature Dependence

Removing IP protocol and TCP flags reduces the ensemble's source multiclass F1 from 0.6455 ± 0.0211 to 0.4768 ± 0.1162 . Additionally removing TTL, IP flags, and TCP window fields leaves frame and TCP-PDU lengths and reduces it to 0.4051 ± 0.0066 . Figure 3 compares the same exclusions for the three focal models. Scores near 0.40 under the strict exclusion indicate dependence on retained header information, without proving that the remaining features are artifact-free.

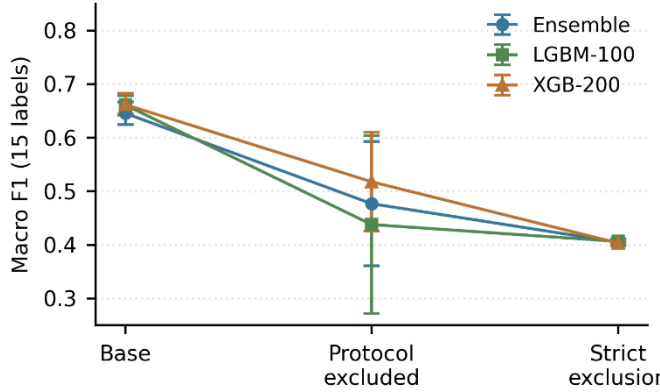


Fig. 3. Source-disjoint feature-exclusion sensitivity

Markers show mean fixed-label multiclass F1; error bars show sample SD over five seeds. Protocol exclusion removes IP protocol/TCP flags; strict exclusion also removes TTL, IP flags, and TCP window fields.

An identifier-retention diagnostic raises seed-42 LightGBM-100 random multiclass F1 from 0.7474 to 1.0000 (rounded), while expanding 16 encoded columns to 3,099. This is a shortcut warning, not an improved detector. Archived permutation importance places window size, TTL, protocol, and frame length among the strongest dependencies. Correlated predictors can substitute for one another, and shuffling can create unrealistic packets [20]. Neither permutation nor feature exclusion provides causal explanation or evidence of analyst utility. The former one-field baseline-substitution procedure is an ad hoc diagnostic; its local-explanation claim is withdrawn, and no SHAP/LIME result is claimed.

E. Binary Errors and Probability Quality

TABLE IV

REVISED BINARY METRICS UNDER SOURCE SEPARATION

MODEL	ACC.	W-F1	ROC	AP	FPR	FNR
Ensemble	0.9865	0.9855	0.9946	0.9997	0.2124	0.0004
LGBM-100	0.9947	0.9945	0.9965	0.9998	0.0787	0.0006
XGB-200	0.9960	0.9959	0.9991	0.9999	0.0467	0.0012
LGBM-200	0.9960	0.9960	0.9996	1.0000	0.0468	0.0011
RF-200	0.9960	0.9960	0.9919	0.9995	0.0466	0.0012
ET-200	0.9883	0.9894	0.9911	0.9994	0.0466	0.0092
HistGB	0.9921	0.9925	0.9984	0.9999	0.0467	0.0052

Note. Mean per-split scores over five seeds; Acc.=accuracy, W-F1=weighted F1, ROC=ROC-AUC, AP=non-interpolated average precision. FPR/FNR use the unchanged decision rule. SD and probability metrics are supplied in Supplement S2.

Table IV shows high weighted F1/AP alongside mean FPR=0.2124 for the ensemble versus 0.0467 for XGBoost. Mean Brier scores are 0.0171 and 0.0046. Supplement S2 supplies all metric SDs and per-run random/source comparisons with sizes, class support, weighted F1, intervals, and per-class recall.

Historically, ensemble FPR/FNR is 0.137333/0 versus XGBoost's 0.036667/0.001167. Ensemble/XGBoost log loss is 0.0429/0.0124, Brier 0.0092/0.0034, and ECE 0.0214/0.0020 (Supplement S3). Neither historical nor revised ranking scores establish a usable alert threshold.

At an assumed 1% attack prior, Equation (3) projects ensemble precision of 6.85% and 135,960 false alerts per million packets, compared with XGBoost's 21.58% and 36,300. FPR remains unchanged under this prior-only reweighting. Some balanced historical baselines observe zero false positives among 1,500 benign packets; the one-sided 95% binomial upper bound is approximately 0.20%, even before accounting for packet dependence. A zero observed FPR is not a population guarantee. Calibration and threshold selection require a separate training/validation procedure and deployment-specific costs.

F. Class Errors, Uncapped Testing, and Chronology

The historical confusion matrix in Figure 4 exposes zero recall for C&C Communication (0/25), recall of 0.5370 for Mirai C&C Communication (29/54), and 0.6250 for Reporting (15/24). Mirai TCP and UDP Flooding recalls are 0.86 and 0.70. In particular, 210 Mirai TCP and 449 Mirai UDP packets are assigned to the corresponding Merlin flooding classes. Preserving transport type while confusing botnet family is an attribution limitation.

TABLE V
UNCAPPED AND CHRONOLOGICAL MULTICLASS F1

MODEL	UNCAPPED MC	CHRONOLOGICAL MC	CHRONOLOGICAL SUPPORTED
Ensemble	0.5398	0.0739	0.2771
LGBM-100	0.5633	0.0056	0.0209
XGB-200	0.5631	0.0552	0.2070

Note. MC uses 15 fixed labels. The uncapped test supports all 15; the chronological test supports four. These checks use the same testbed and different label compositions.

Table V contrasts uncapped and chronological testing. The uncapped seed-42 holdout has 5,861,157 packets from 14 files, 94.58% attacks, and 10,015 unseen-label rows. Its binary F1 is 0.9289, 0.9715, and 0.9834 for the ensemble, LightGBM-100, and XGBoost, respectively. Multiclass weighted F1 is 0.9012, 0.9024, and 0.9024, substantially above macro F1. Removing the cap restores the captured distribution, not an operational prior.

Chronological splitting trains on 93,721 rows and tests on 16,539 at 2025-01-25 16:31:49.099116 UTC. Only four multiclass labels occur in testing, and 10,000 test rows have a label absent from training. Binary F1 ranges from 0.4735 to 0.4759. These results combine temporal/scenario change and unseen labels; they do not isolate elapsed time. Full supports and metrics are supplied in Supplement S4.

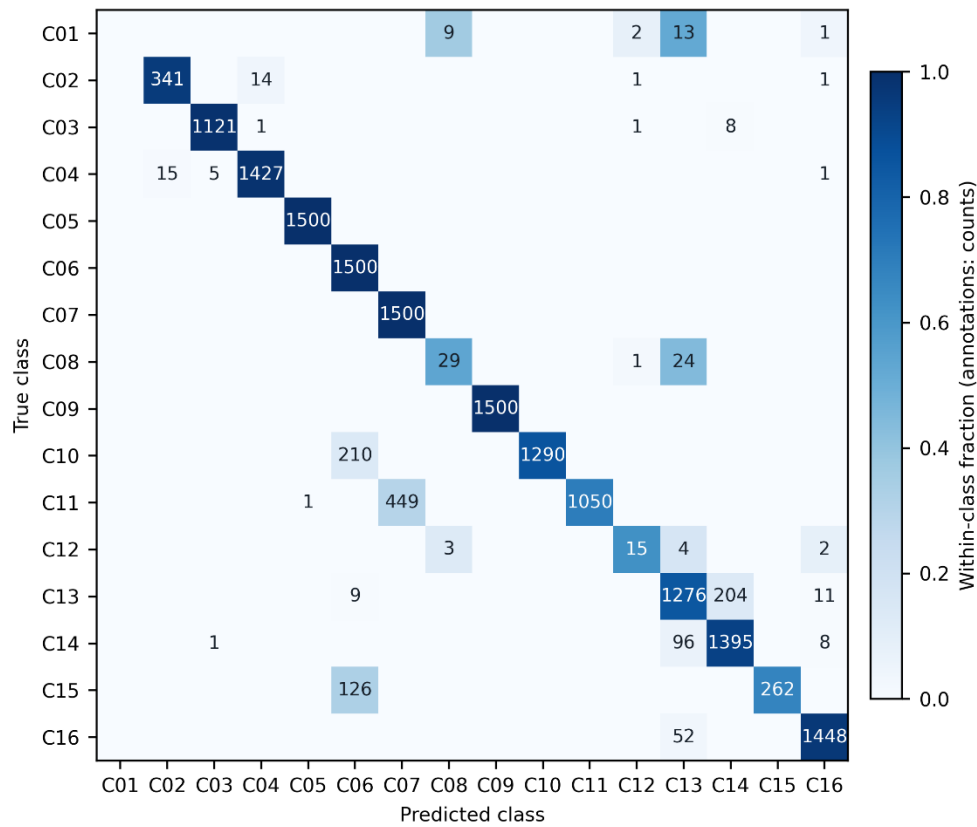


Fig. 4. Class-level errors in the archived multiclass ensemble.

Class key. C01: C&C Communication; C02: File Download; C03: Ingress Tool Transfer; C04: Merlin C&C Communication; C05: Merlin ICMP Flooding; C06: Merlin TCP Flooding; C07: Merlin UDP Flooding; C08: Mirai C&C Communication; C09: Mirai GRE Flooding; C10: Mirai TCP Flooding; C11: Mirai UDP Flooding; C12: Reporting; C13: TCP Scan; C14: Telnet Brute Force; C15: Unknown; C16: benign.

Rows represent true classes and columns predicted classes. Annotations are packet counts; colour shows the fraction within each true class, so darker diagonal cells indicate higher recall. Unknown is retained only for this historical reconstruction.

V. THREATS TO VALIDITY

All evaluations use one emulated testbed. Source/device separation does not separate every attack execution, host image, or network-wide scenario. Few devices carry most attacks, and no split can place a class confined to one device on both sides. The coverage rule also conditions the eligible test population. Five overlapping holdouts and bootstrap intervals over only 14 test files cannot establish a universal model ranking.

Class capping and Unknown removal change the target distribution. Named labels are not independently adjudicated, and conditional drift could invalidate the prior-sensitivity calculations. Header exclusions reduce some shortcut opportunities without proving their absence. The revised untuned comparison improves symmetry but does not estimate each model family's best achievable performance. Fixed weights and uncalibrated probabilities limit decision claims. Workstation batch timing and compressed storage measure neither end-to-end edge latency nor working memory, energy, or broker-load throughput.

VI. CONCLUSION

Reconstruction corrects source scope, feature counts, voting weights, residual labels, and the interpretation of the original holdout denominator. Revised source multiclass F1 averages 0.6455 for the ensemble versus 0.6617 for XGBoost. Feature and partition diagnostics expose capture dependence and motivate simpler model selection. Independent captures, operational priors, training-separated calibration, and target-hardware measurements are required before deployment claims.

DATA AND CODE AVAILABILITY

The original data are available under DOI 10.5281/zenodo.14502760 [2]. This study's code, configurations, source/split manifests, saved models, complete results, figures, and Supplementary Materials S1–S4 are publicly archived in Zenodo at <https://doi.org/10.5281/zenodo.22706220>.

ACKNOWLEDGMENT

The authors acknowledge the dataset and scientific Python developers. AI assistance supported revision code, output checking, and text preparation. The authors remain responsible for verification and interpretation.

REFERENCES

- [1] OASIS, "MQTT Version 5.0," OASIS Standard, Mar. 7, 2019. Accessed: Sep. 10, 2026. [Online]. Available: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>
- [2] O. Belarbi, T. Spyridopoulos, E. Anthi, O. Rana, P. Carnelli, and A. Khan, "A Device-Level IoT Network Traffic Dataset with Distributed Capture and Non-IID Characteristics," Zenodo, dataset record 14502760, issued Feb. 5, 2025, doi: 10.5281/zenodo.14502760. Accessed: Sep. 10, 2026.
- [3] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset)," Selected Papers from the 12th International Networking Conference, pp. 73-84, 2021, doi: 10.1007/978-3-030-64758-2_6.
- [4] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. Garcia, and C. Benavides, "Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol," Complexity, vol. 2019, no. 1, Art. 6516253, 2019, doi: 10.1155/2019/6516253.
- [5] A. A. Hanif and M. Ilyas, "Enhance the Detection of DoS and Brute Force Attacks within the MQTT Environment Through Feature Engineering and Employing an Ensemble Technique," International Journal of Artificial Intelligence & Applications, vol. 15, no. 4, pp. 01-19, 2024, doi: 10.5121/ijaa.2024.15401.
- [6] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," Sensors, vol. 23, no. 13, Art. 5941, 2023, doi: 10.3390/s23135941.
- [7] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," Sustainable Cities and Society, vol. 72, Art. 102994, 2021, doi: 10.1016/j.scs.2021.102994.
- [8] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," IEEE Pervasive Computing, vol. 17, no. 3, pp. 12-22, 2018, doi: 10.1109/mpcv.2018.03367731.
- [9] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," Proceedings 2018 Network and Distributed System Security Symposium, 2018, doi: 10.14722/ndss.2018.23204.
- [10] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," Future Generation Computer Systems, vol. 100, pp. 779-796, 2019, doi: 10.1016/j.future.2019.05.041.
- [11] X. Sáez-de-Cámara, J. L. Flores, C. Arellano, A. Urbietta, and U. Zurutuza, "Gotham Testbed: A Reproducible IoT Testbed for Security Experiments and Dataset Generation," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 1, pp. 186-203, 2024, doi: 10.1109/tdsc.2023.3247166.
- [12] M. A. Ferrag, M. Ndhlovu, N. Tihanyi, L. C. Cordeiro, M. Debbah, T. Lestable, and N. S. Thandi, "Revolutionizing Cyber Threat Detection With Large Language Models: A Privacy-Preserving BERT-Based Lightweight Model for IoT/IIoT Devices," IEEE Access, vol. 12, pp. 23733-23750, 2024, doi: 10.1109/access.2024.3363469.
- [13] A. Bensaoud and J. Kalita, "Optimized detection of cyber-attacks on IoT networks via hybrid deep learning models," Ad Hoc Networks, vol. 170, Art. 103770, 2025, doi: 10.1016/j.adhoc.2025.103770.
- [14] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," Advances in Neural Information Processing Systems, vol. 30, 2017. Accessed: Sep. 10, 2026. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bd9eb6b76fa-Abstract.html>
- [15] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, 2016, doi: 10.1145/2939672.2939785.

- [16] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001, doi: 10.1023/a:1010933404324.
- [17] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3-42, 2006, doi: 10.1007/s10994-006-6226-1.
- [18] F. Ceschin, M. Botacin, A. Bifet, B. Pfahringer, L. S. Oliveira, H. M. Gomes, and A. Grégio, "Machine Learning (In) Security: A Stream of Problems," *Digital Threats: Research and Practice*, vol. 5, no. 1, pp. 1-32, 2024, doi: 10.1145/3617897.
- [19] Nmap Project, "TCP/IP Fingerprinting Methods Supported by Nmap," *Nmap Network Scanning*. Accessed: Sep. 10, 2026. [Online]. Available: <https://nmap.org/book/osdetect-methods.html>
- [20] Scikit-learn developers, "Permutation feature importance," user guide, subsection on strongly correlated features. Accessed: Sep. 10, 2026. [Online]. Available: https://scikit-learn.org/stable/modules/permutation_importance.html
- [21] G. Rjoub, J. Bentahar, O. Abdel Wahab, R. Mizouni, A. Song, R. Cohen, H. Otrouk, and A. Mourad, "A Survey on Explainable Artificial Intelligence for Cybersecurity," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 5115-5140, 2023, doi: 10.1109/tnsm.2023.3282740.
- [22] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017. Accessed: Sep. 10, 2026. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html>
- [23] O. Belarbi, T. Spyridopoulos, E. Anthi, O. Rana, P. Carnelli, and A. Khan, "A Device-Level IoT Network Traffic Dataset with Distributed Capture and Non-IID Characteristics," *Data*, vol. 11, no. 8, Art. 207, 2026, doi: 10.3390/data11080207.
- [24] O. Belarbi et al., "Gotham network packet labeller," `src/helpers/labeller.py`, commit cad1575bc8fd65f9e30d5cc88b04d7ba137c980d. Accessed: Sep. 10, 2026. [Online]. Available: <https://github.com/othmbela/gotham-network-packet-labeller/blob/cad1575bc8fd65f9e30d5cc88b04d7ba137c980d/src/helpers/labeller.py>
- [25] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226-1238, 2005, doi: 10.1109/tpami.2005.159.
- [26] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," *PLOS ONE*, vol. 10, no. 3, Art. e0118432, 2015, doi: 10.1371/journal.pone.0118432.

Bayu Setiaji obtained his Bachelor's degree in Informatics from Universitas Amikom Yogyakarta in 2006. He obtained his Master's degree in Informatics from Universitas Amikom Yogyakarta in 2012. He is currently a Lecturer in the S1 Informatics Study Program, Faculty of Computer Science, Universitas Amikom Yogyakarta, where he also serves as the Coordinator for Learning Curriculum Development and Internships. His research interests include the Internet of Things (IoT), Unmanned Aerial Vehicles (UAV), artificial intelligence (AI), and generative art.

Haryoko obtained his Bachelor's degree in Information Systems from Universitas Amikom Yogyakarta in 2010. He obtained his Master's degree in Computer Science from the Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, in 2016. He is currently a Lecturer in the Information Technology Study Program, Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia. His research interests include computer networks, the Internet of Things, network security, image processing, and machine learning.

Windha Mega Pradnya Dhuhita is a lecturer at Amikom Yogyakarta since 2008. Her academic interests focus on databases and big data. Throughout her research and development activities, she has consistently maintained a strong awareness of the ethical implications of technology usage, particularly in the context of big data and artificial intelligence. She has contributed to projects exploring database optimization techniques to enhance performance and efficiency in handling large volumes of data. Her research

interests include Database Systems, Data Structures, Calculus, Artificial Intelligence, and Expert Systems. She can be contacted at email: windha@amikom.ac.id.

Eli Pujastuti is an Assistant Professor and the Head of the Informatics Study Program, Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia. Her research interests include human-computer interaction, user-interface evaluation, digital entrepreneurship, educational technology and automation.

Lukman obtained his Bachelor's degree in Informatics from Universitas Amikom Yogyakarta in 2006. He obtained his Master's degree in Informatics from Universitas Amikom Yogyakarta in 2014. He is currently a Lecturer in the D3 Information Management Study Program, Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia. His research interests include web programming, e-commerce, project management, computer networks, and system analysis and design.

Afrig Aminudin obtained his Bachelor's degree in Informatics Engineering from Universitas Amikom Yogyakarta in 2014. He obtained his Master's degree in Information Technology from Universitas Gadjah Mada in 2017. He obtained his Ph.D. degree in Computer Science from Universiti Malaysia Pahang in 2025. Since 2017, he has been a Lecturer at Universitas Amikom Yogyakarta. His research interests include image processing, image watermarking, image authentication, and machine learning.