Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MALGUARD-IoT: A COST-AWARE HIERARCHICAL AND EXPLAINABLE EDGE FRAMEWORK FOR MIRAI MALWARE DETECTION

HANI AL-BALASMEH[1]
[1]College of Engineering, University of Technology Bahrain (UTB), Bahrain
E-mail: [1]h.albalasmeh@utb.edu.bh

ABSTRACT

This study develops and evaluates MALGUARD-IoT, a cost-aware, hierarchical edge framework for detecting Mirai malware from identity-excluded network-flow statistics. The method selects 24 behavioral features, fits a histogram-gradient-boosting detector with five-fold isotonic calibration, and applies a prespecified decision threshold of 0.25 reflecting a 3:1 false-negative-to-false-positive cost ratio. A second classifier assigns detected malicious flows to reconnaissance, infection, or attack action. The experiments use 40,893 training and 9,335 held-out test flows and compare five baseline models. MALGUARD-IoT achieved test F1 = 1.000 and MCC = 1.000 (0 false positives and 0 false negatives), while the strongest uncalibrated boosting baseline missed three malware flows. Stage attribution achieved macro-F1 = 0.9873. The binary and stage artifacts required approximately 189.4 kB and 41.9 kB, with host-measured inference times of 5.93 and 10.64 ms per 1,000 flows, respectively. Robustness was conditional: F1 decreased to 0.9063 under 10% interquartile-range Gaussian perturbation and to 0.8338 under 20% feature dropout. These results support compact, auditable decisions on the evaluated Mirai benchmark but do not demonstrate cross-family or live-gateway generalization.

Keywords: *Internet Of Things, Mirai Malware, Edge Intrusion Detection, Probability Calibration, Explainable Learning.*

1. INTRODUCTION

The Internet of Things connects cameras, sensors, routers, meters, appliances, and cyber-physical controllers to networks whose security assumptions are often weaker than those of conventional computing platforms. Devices may use default credentials, receive irregular patches, expose remotely reachable services, and operate for years with limited memory and processing capacity. Mirai demonstrated how these weaknesses can be industrialized: compromised embedded devices were enrolled into a botnet that reached hundreds of thousands of infections and generated large distributed denial-of-service campaigns [1]. The continuing diversity of IoT deployments makes network-side detection at gateways and edge nodes an important complement to device hardening.

Machine-learning detectors can identify deviations in flow timing, packet length, flag combinations, and communication rates. Yet a high headline accuracy does not by itself establish operational value. Security data are imbalanced, false negatives and false positives have unequal consequences, raw model scores may not be calibrated probabilities, and models can exploit endpoint identities or dataset artifacts rather than behavioral evidence. Recent work has also highlighted data scarcity, temporal drift, and the need for lightweight models at the edge [7]-[11]. These concerns motivate an evaluation protocol that connects predictive performance to deployment decisions.

Terminology. An Internet of Things (IoT) gateway is a network-edge device aggregating traffic from connected equipment. Mirai is a botnet malware family that exploits vulnerable devices and can perform scanning, infection attempts, and distributed denial-of-service flooding. A bidirectional flow is a summarized exchange of packets between communicating endpoints; this work analyzes flow metadata rather than packet payloads. An intrusion detection system (IDS) flags potentially malicious behavior. Probability calibration evaluates whether estimated event probabilities agree with observed event frequencies. A false negative (FN) is a missed malicious flow, whereas a false positive (FP) is a benign flow flagged as malicious. The Matthews correlation coefficient (MCC) summarizes binary predictions using all four confusion-matrix cells; macro-F1 gives equal weight to the F1 score of each attack-stage class. Feature dropout represents missing

telemetry, while concept drift denotes changes in traffic or attack patterns over time.

1.1 Research Problem

Existing IoT malware studies frequently optimize either detection accuracy or malware-family classification. Fewer studies combine five properties in one reproducible system: identity-free flow features, calibrated malware probabilities, a threshold derived from explicit operational costs, hierarchical attack-stage attribution, and robustness tests under degraded telemetry. Without these properties, a detector may be accurate in a static benchmark but difficult to trust or operate at an edge gateway.

1.2 Research Questions And Hypotheses

The study addresses three research questions. RQ1 asks whether a compact, identity-free flow model can improve malware recall and MCC over classical and neural baselines on a fixed held-out test set. RQ2 asks whether calibrated probabilities and a predeclared cost threshold can reduce costly false negatives without increasing false alarms. RQ3 asks whether malicious flows can be attributed to reconnaissance, infection, or attack action with sufficient accuracy for differentiated response.

The hypotheses are: H1, MALGUARD-IoT produces a higher test MCC than every evaluated baseline; H2, the cost-aware threshold produces no more false negatives than the conventional 0.50 threshold; and H3, hierarchical stage attribution achieves macro-F1 above 0.90.

1.3 Contributions

- A compact hierarchical framework that separates malware detection from attack-stage attribution and maps outputs to risk-aware edge actions.
- An identity-free preprocessing protocol that excludes timestamps and IP addresses before learning, reducing direct memorization of endpoints.
- A calibrated, cost-sensitive decision rule whose threshold is fixed analytically from a 3:1 false-negative-to-false-positive cost ratio rather than optimized on the test set.
- A reproducible evaluation with five baselines, feature-count and calibration ablations, confusion matrices, MCC and PR-AUC, latency and model-size measurements, permutation explanations, chronological slices, and corruption stress tests.
- A complete supplementary package containing the source data files, cryptographic hashes, executable Python code, trained models, tables, and figures.

2. RELATED WORK

2.1 Network-Based IoT Malware Detection

N-BaIoT established a widely used network-based paradigm by extracting traffic statistics from nine commercial IoT devices infected with Mirai and BASHLITE and applying deep autoencoders [2]. Other public resources, including IoT-23, TON_IoT, and Edge-IIoTset, broadened the available malware and intrusion scenarios [3]-[5]. These datasets enabled direct comparison of classical machine learning, deep learning, and hybrid systems, but they also exposed evaluation risks: random splitting can mix closely related traffic across partitions, identifiers can leak device or campaign information, and class distributions can inflate accuracy.

2.2 Edge And Lightweight Detection

Edge-oriented systems seek to place inference close to the protected devices, thereby avoiding raw-data transfer to a remote cloud and shortening response time. Edge2Guard showed that compact anomaly detection can be deployed on constrained IoT gateways [6]. More recent boosting-based studies reported strong accuracy and favorable computational cost on large IoT traffic datasets [10]. Deep architectures based on BERT, recurrent networks, temporal convolution, graph neural networks, and multimodal binary analysis can also achieve high accuracy [7], [9], [12], but model complexity and preprocessing cost may constrain continuous edge use.

2.3 Reliability, Calibration, And Drift

Security-oriented machine learning requires more than random-split accuracy. Arp et al. documented common pitfalls such as temporal leakage, dataset artifacts, unrealistic base rates, weak baselines, and incomplete threat models [13]. Lyon et al. found that efficient tree models can remain competitive under limited training data while performance degrades as malware diversity and time separation increase [8]. Calibration methods such as isotonic regression can transform ranking scores into more meaningful probabilities [18], [19], while cost-sensitive thresholds can align decisions with operational priorities. MCC and PR-AUC are particularly useful when benign and malicious classes are imbalanced [21]-[23].

Recent context. Reviews published in 2024–2026 consistently identify dataset representativeness, test-protocol comparability, imbalance, inference cost, and model transfer as central difficulties for IoT intrusion detection [27]–[30]. A 2024 study explicitly investigates class imbalance and concept drift in heterogeneous IoT-edge botnet detection

[31], while a 2025 feature-selection ensemble evaluates Mirai and BASHLITE and illustrates that extremely high within-dataset scores are achievable without establishing cross-site transfer [32]. MALGUARD-IoT addresses a different, narrower question: whether cost-calibrated detection, coarse attack-stage attribution, and telemetry-stress evidence can be combined in a compact flow-based pipeline evaluated on the supplied Mirai split. This comparison is conceptual, not a head-to-head benchmark, because data and experimental protocols differ.

2.4 Research Gap

Table 1 positions MALGUARD-IoT against representative studies. Prior work offers important advances in deep sequence modeling, cross-architecture binary analysis, or high-performance boosting. The unresolved gap is an integrated edge pipeline that explicitly combines identity exclusion, calibration, predeclared error costs, stage attribution, global explanation, and telemetry-degradation tests in one reproducible experiment.

Interpretive comparison. Earlier N-BaIoT work established device-level behavior modeling [2], whereas recent surveys emphasize deployment realism and variation across datasets [27]–[30]. The present framework should therefore be understood as an integrated decision pipeline rather than a claim that its classifier universally surpasses previously published models. The source corpus, possible common collection conditions, and absence of leave-device-out validation constrain direct generalization claims.

Table 1. Comparison with representative IoT malware studies

Study	Method	Scope	Calibration	Stage	Robustness	Distinguishing focus
Meidan et al. [2]	Flow statistics + autoencoders	Mirai/BASHLITE	No	No	No	Device-specific anomaly models
Prajapati et al. [7]	BERT, TCN, GNN, RNN	IoT malicious traffic	No	No	Partial	High-performing deep models; heavier training
Lyon et al. [8]	RF, LightGBM, LR, MLP	IoT-23 malware	No	No	Drift	Data scarcity and drift benchmarking
Abd-Ellah et al. [9]	CNN + feature engineering	IoT malware	No	No	No	Multiple deep preprocessing variants
Yakovyna and Fadieiev [10]	Boosting ensembles	IoT-23	No	No	No	Lightweight high-performance IDS
Xia et al. [11]	Multimodal DL	Cross-architecture binaries	No	No	No	Static malware, EV charging context
MALGUARD-IoT	Calibrated HGB + hierarchy	Mirai flows	Yes	Yes	Noise/dropout	Identity-free, cost-aware edge response

3. MALGUARD-IOT FRAMEWORK

MALGUARD-IoT is designed for a gateway or edge node that observes bidirectional network-flow summaries. It does not inspect executable binaries or encrypted payload contents. The framework receives a flow vector, removes identity-bearing fields, applies the training median imputer and a fixed 24-feature mask, produces a calibrated malware probability, and compares that probability with a cost-derived threshold. Malicious flows are attributed to one of three stages and forwarded to a response policy. Figure 1 summarizes the online and offline pipeline, while Algorithms 1 and 2 formalize model construction and edge inference.

Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

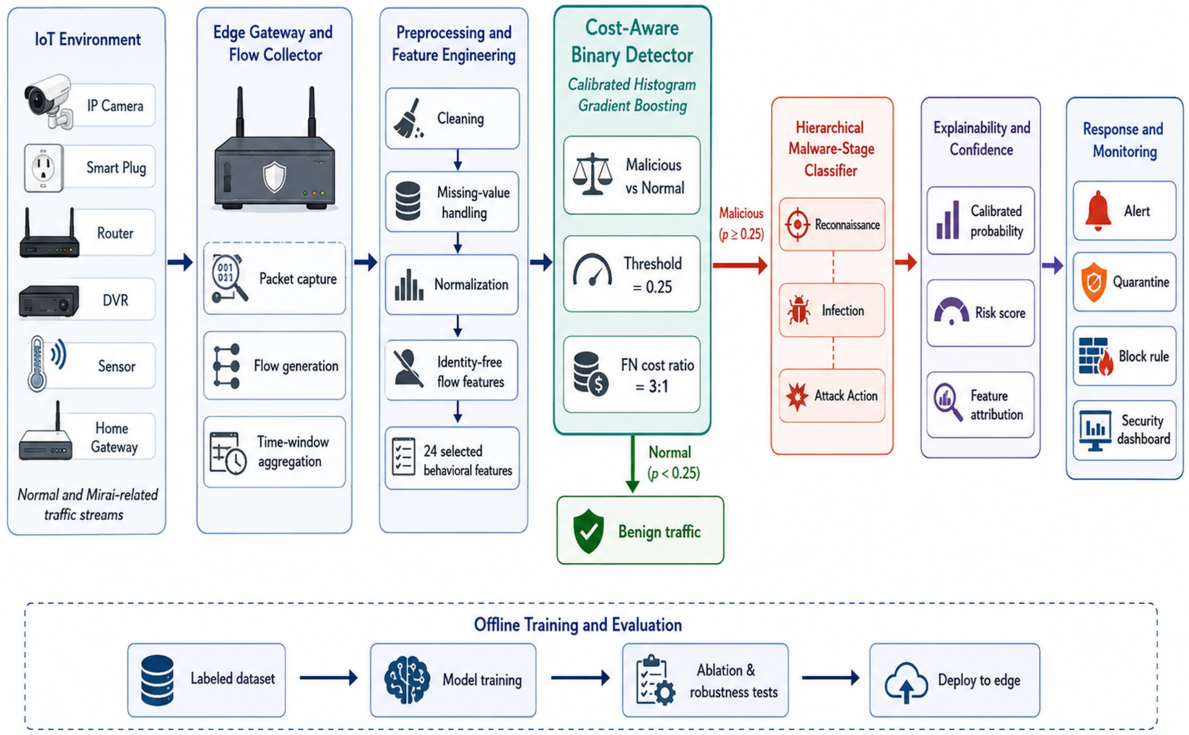


Figure 1. Architecture of the proposed MALGUARD-IoT framework

3.1 Threat Model

The adversary controls an IoT endpoint or has obtained credentials that permit scanning, brute-force infection attempts, or UDP flooding. The gateway and its model artifacts are trusted, while network telemetry may be incomplete or noisy. The framework seeks behavioral detection of the represented Mirai stages. It does not protect the model from direct file-system tampering, guarantee attribution to a particular malware binary, or claim detection of unseen families. Encrypted traffic is compatible because only flow-level metadata are required.

3.2 Identity-Free Behavioral Representation

The source files contain 44 candidate numeric flow features after removing the two time fields, two endpoint addresses, and three label fields. IP addresses and timestamps are excluded to discourage direct endpoint and campaign memorization. Median imputation is learned on the training set. Mutual information ranks features according to dependency with the binary class label [20], and the highest-ranked 24 features define the deployable representation.

$$I(X_j;Y) = \Sigma_x \Sigma_y p(x,y) \log[p(x,y)/(p(x)p(y))] \quad (1)$$

3.3 Calibrated Binary Detector

The binary learner is a histogram-based gradient boosting classifier with 120 iterations, 19 maximum leaves, learning rate 0.08, and L2 regularization 0.3. Gradient boosting constructs an additive decision function by fitting successive weak learners to residual error [14]. Five-fold isotonic calibration is then fitted through out-of-fold predictions, limiting the use of overconfident raw scores [18], [19], [26].

$$F_M(x) = F_{(M-1)}(x) + \eta\, h_M(x) \quad (2)$$

$$p_hat(x) = g(F_M(x)), \text{ where } g \text{ is monotone isotonic regression} \quad (3)$$

3.4 Cost-Aware Threshold And Risk Policy

In edge security, a missed malware flow can be more costly than a benign flow temporarily rate-limited for inspection. The experiment fixes the false-negative cost at three times the false-positive cost before examining test results. For calibrated probabilities, the Bayes decision threshold is therefore:

$$\tau = C_{FP} / (C_{FP} + C_{FN}) = 1/(1+3) = 0.25 \quad (4)$$

Flows below τ are permitted while telemetry continues. Flows at or above τ enter stage

Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

attribution. Reconnaissance can trigger observation and connection-rate limits; infection can trigger credential protection and temporary isolation; and attack action can trigger immediate quarantine and upstream filtering. These actions are policy examples rather than evaluated mitigation outcomes.

3.5 Hierarchical Attack-Stage Attribution

A second histogram-based gradient boosting model is trained only on malicious training flows. Class-balanced sample weights address the rarity of infection records relative to reconnaissance. The output labels are reconnaissance, infection, and action. Separating binary detection from stage attribution prevents benign dominance from suppressing the smaller malware subclasses and yields an interpretable response label.

Algorithm 1. Offline training and model selection for MALGUARD-IoT

Input: Labeled IoT flow dataset D.
Output: Binary detector M, probability calibrator C, malware-stage classifier S, and deployment package P.
1. Remove timestamps, endpoint identities, labels, and non-finite values; learn the training-set median imputer.
2. Retain the candidate bidirectional flow statistics and construct time-window summaries.
3. Rank candidate features by mutual information estimated on the training partition only.
4. Retain the 24 highest-ranked identity-free behavioral features.
5. Preserve the supplied train/test separation and fit every transformation on training records only.
6. Train the classical and neural baselines and the histogram-gradient-boosting detector.
7. Fit five-fold isotonic probability calibration on the training partition.
8. Fix the decision threshold at tau = 0.25 from the predeclared 3:1 false-negative cost.
9. Train the class-balanced malware-stage classifier using malicious training flows.
10. Evaluate discrimination, calibration, ablation, robustness, explanation, temporal stability, latency, and size.
11. Serialize the feature mask, imputer, calibrated detector, threshold, and stage classifier.

Goal: *Maximize defensible detection quality while preserving compact deployment and interpretable decisions.*

Algorithm 2. Online inference, risk scoring, and automated response

Input: Incoming bidirectional IoT flow x_t.
Output: Binary label y, calibrated malware probability p_mal, stage s, explanation e, and response action a.
1. Capture the completed bidirectional flow summary at the trusted edge gateway.
2. Apply the stored median imputer and construct the 24 selected behavioral features.
3. Compute the calibrated malware probability p_mal = M(x_t).
4. If p_mal < 0.25, label the flow benign, record the decision, and continue monitoring.
5. Otherwise, invoke S(x_t) to assign reconnaissance, infection, or attack action.
6. Generate feature-attribution evidence and combine probability and stage severity for policy review.
7. Map the stage and local policy to alert, rate-limit, quarantine, or block actions.
8. Write the model version, probability, stage, evidence, and response to the security log.

Goal: *Prioritize missed-malware reduction while retaining stage awareness, traceability, and human override.*

4. MATERIALS AND METHODS

4.1 Dataset And Partition

The experiment uses public preprocessed Mirai bidirectional flows distributed by the comsyssec datasets repository [24]. The supplied training and test files are kept intact rather than recombined and randomly repartitioned. The training file contains 40,893 flows, of which 2,307 are malicious. The test file contains 9,335 flows, of which 748 are malicious. Malicious labels represent port-scanning reconnaissance, brute-force infection, and UDP-flooding action. Table 2 reports the distribution and the package records SHA-256 hashes for exact file verification.

Table 2. Flow distribution by partition and attack stage

Partition	Benign	Recon.	Infection	Action	Total
Training	38,586	1,617	109	581	40,893
Held-out test	8,587	324	199	225	9,335
Total	47,173	1,941	308	806	50,228

Training SHA-256: a238e922b62a095bda853374c80b4f47281599a852bb7708bda0bce99814932b. Test SHA-256: 6fd1e539e3e57dc68d1ea0a668b0627e584e9b722d95cb60111810b9410e0335.

4.2 Experimental Protocol

All preprocessing transformations are fitted on training records and applied unchanged to the test set. Mutual-information estimation uses every malicious training record and a seeded sample of up to 10,000 benign records to reduce ranking cost. Baseline models use a stratified 78:22 split of the training file for fitting consistency; the proposed calibrated model uses five-fold calibration across the full training file. The independent test file is used once for final comparison, ablation, explanation, and stress evaluation. Random seed 42 is fixed throughout.

4.3 Baselines

Five baselines represent linear, bagging, randomized-tree, boosting, and neural alternatives

[14]-[17]: class-balanced logistic regression; random forest with 120 trees; extremely randomized trees with 120 trees; uncalibrated histogram gradient boosting; and a shallow multilayer perceptron with hidden layers of 24 and 12 neurons. Robust scaling is applied only to logistic regression and the multilayer perceptron. Default binary decisions use threshold 0.50.

4.4 Evaluation Measures

Accuracy is reported for completeness, but conclusions prioritize precision, recall, F1, MCC, ROC-AUC, and PR-AUC. MCC summarizes all four cells of the confusion matrix and remains informative under imbalance. Brier score and ten-bin expected calibration error assess probability quality. The formulas for F1 and MCC are:

$$F1 = 2TP / (2TP + FP + FN) \quad (5)$$

$$MCC = (TP \cdot TN - FP \cdot FN) / \sqrt{[(TP+FP)(TP+FN)(TN+FP)(TN+FN)]} \quad (6)$$

4.5 Robustness, Explainability, And Efficiency

Permutation importance measures the F1 decrease after each selected feature is shuffled eight times. Robustness is tested through ten repetitions of Gaussian perturbation at 1%, 5%, and 10% of each feature's training interquartile range and through random replacement of 5%, 10%, and 20% of feature values with training medians. The test file is also sorted by window start and divided into five equal chronological slices. Inference time is measured for probability prediction and normalized to milliseconds per 1,000 flows; serialized artifact sizes are recorded.

4.6 Reproducibility

The supplementary package includes the two CSV files, Python source, requirements, generated tables, PNG figures, and compressed joblib artifacts. File hashes, seed, feature list, model hyperparameters, threshold, and all evaluation transformations are encoded in the script. The implementation uses Python, pandas, NumPy, scikit-learn, matplotlib, and joblib. Exact wall-clock values may vary by processor and software version, so the included latency values should be interpreted as host-specific rather than universal edge benchmarks.

4.7 Computational Structure

For each completed flow, preprocessing is linear in the 24 deployed features. Histogram-based gradient boosting then evaluates a bounded sequence of decision-tree paths; the binary model is executed for every flow, whereas the stage model is invoked only when the calibrated probability crosses the malware threshold. This hierarchical gating limits average stage-classification cost when the alert rate is low. The implementation stores neither payload bytes nor endpoint identifiers, which reduces memory and privacy exposure relative to payload-based or identity-dependent approaches.

5. RESULTS

5.1 Comparative Binary Malware Detection

Table 3 and Figure 2 compare MALGUARD-IoT with five baselines on the untouched test file. The proposed system classified all 8,587 benign and 748 malicious flows correctly at the predeclared threshold. The strongest baseline, uncalibrated histogram gradient boosting, achieved F1 = 0.9980 and MCC = 0.9978 but missed three malware flows. Logistic regression recovered every malicious flow, yet generated 41 false positives; random forest and Extra Trees produced 217 and 206 false negatives, respectively; and the shallow multilayer perceptron missed 246 malicious flows. Thus, the proposed configuration improved the best baseline by eliminating the remaining false negatives rather than by a large change in aggregate accuracy.

Table 3. Held-out binary malware-detection performance

Model	Prec.	Recall	F1	MCC	PR-AUC	FP	FN
Logistic Regression	0.9480	1.0000	0.9733	0.9713	0.9921	41	0
Random Forest	1.0000	0.7099	0.8303	0.8321	0.9938	0	217
Extra Trees	0.9766	0.7246	0.8319	0.8302	0.9637	13	206
HistGradientBoosting	1.0000	0.9960	0.9980	0.9978	1.0000	0	3
Shallow MLP	0.9078	0.6711	0.7717	0.7650	0.6881	51	246
MALGUARD-IoT (proposed)	1.0000	1.0000	1.0000	1.0000	1.0000	0	0

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

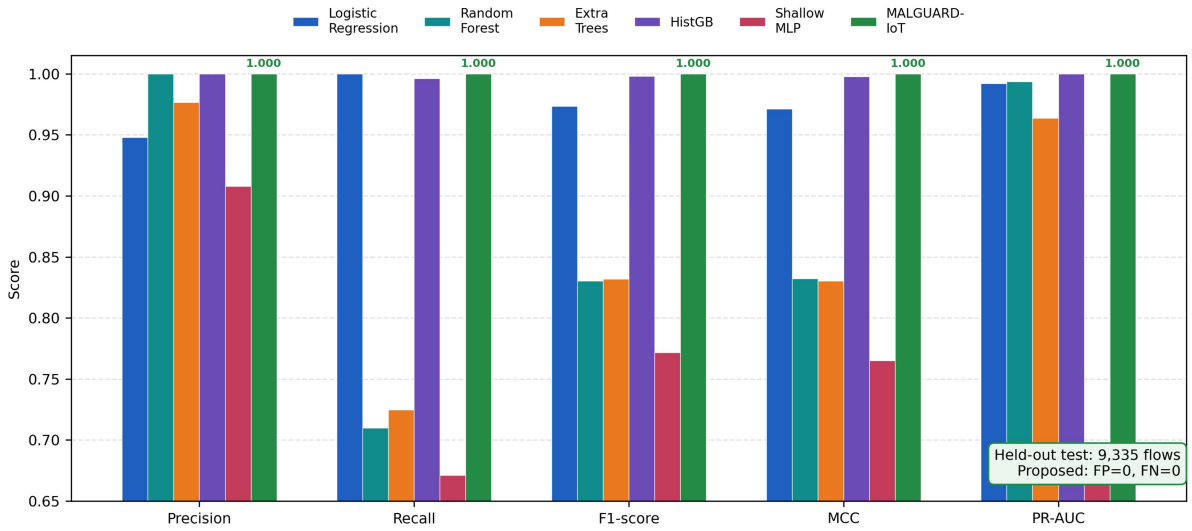


Figure 2. Comparative held-out performance of the binary malware detectors

The perfect class decisions are a property of this fixed benchmark, not evidence of universal malware detection. The train and test files originate from the same curated Mirai collection, and the selected behavioral variables create a wide score margin for the represented attacks. Accordingly, the study treats the result as benchmark-specific evidence and evaluates calibration, threshold sensitivity, robustness, temporal slices, and limitations separately.

5.2 Feature, Calibration, And Threshold Ablation

The ablation results in Table 4 and Figure 3 show that feature count and decision policy affect different failure modes. With 12 and 16 features, recall was only 0.6832 and 0.6939, producing 237 and 229 false negatives. Expanding to 24 features increased F1 to 0.9946 and reduced false negatives to eight; using all 44 identity-free candidates reduced them to three. The calibrated 24-feature model at the conventional 0.50 threshold produced ten false negatives. Applying the analytically derived threshold of 0.25 recovered all ten without creating a false positive, thereby supporting H2.

Error-analysis interpretation. On the held-out test, malware prevalence was 748/9,335 = 8.01%. The uncalibrated boosting baseline missed 3/748 = 0.40% of malicious flows and the raw 24-feature ablation missed 8/748 = 1.07%; the calibrated 0.50-threshold model missed 10/748 = 1.34%. These percentages are derived from the published confusion counts, not from new training runs. Logistic regression attained full recall at the cost of 41/8,587 = 0.48% benign-flow false-positive rate. Since the evaluated set contains only 748 malicious cases from one curated source, zero observed misses cannot guarantee zero future miss probability. In particular, if these cases were independent and identically distributed, the conventional rule-of-three upper 95% bound on the unseen miss probability would be approximately 3/748 = 0.40%; the assumption is optimistic if flows are temporally or device correlated.

Table 4. Feature, calibration, and threshold ablation

Configuration	Prec.	Recall	F1	MCC	FP	FN
HGB, 12 features, threshold 0.50	0.9808	0.6832	0.8054	0.8066	10	237
HGB, 16 features, threshold 0.50	1.0000	0.6939	0.8193	0.8221	0	229
HGB, 24 features, threshold 0.50	1.0000	0.9893	0.9946	0.9942	0	8
HGB, 44 features, threshold 0.50	1.0000	0.9960	0.9980	0.9978	0	3
MI-24 + isotonic, threshold 0.50	1.0000	0.9866	0.9933	0.9927	0	10
Full MALGUARD-IoT, cost threshold 0.25	1.0000	1.0000	1.0000	1.0000	0	0

Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

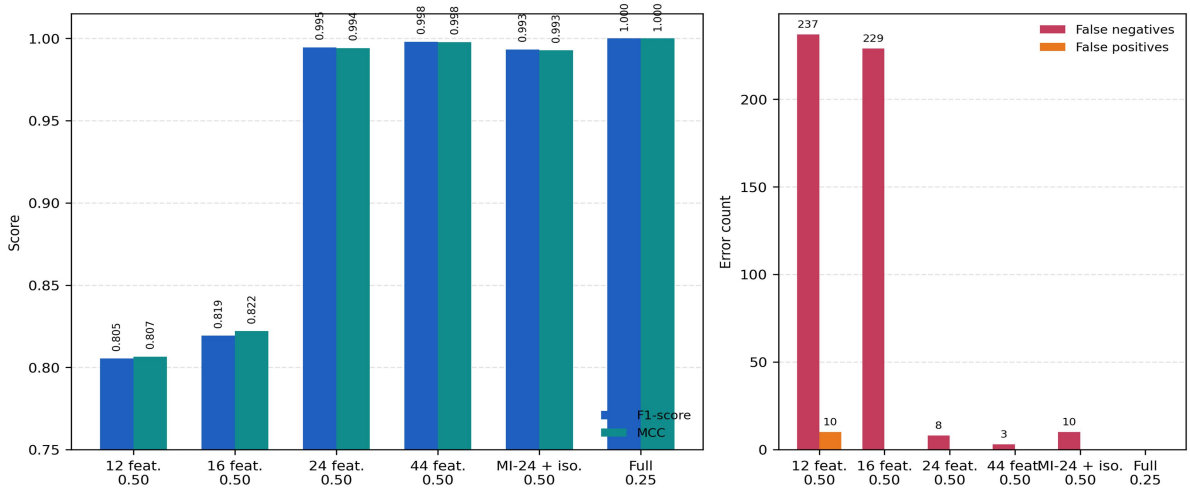


Figure 3. Ablation analysis of feature count, calibration, and decision threshold

A further caution is that isotonic calibration did not improve every test-set probability metric: the calibrated proposal had Brier score 0.00363 and ECE 0.01022, whereas the raw HistGradientBoosting baseline achieved lower values on this particular test file. The benefit claimed here is therefore not that calibration universally outperformed the raw scores, but that it provided an explicit probability scale on which a predeclared cost-sensitive rule could be applied and audited.

Calibration qualification. Calibrating a score does not guarantee improved Brier score or expected calibration error under dataset shift; indeed, the uncalibrated HGB scored better on these reported test metrics. The theoretical Bayes threshold of 0.25 is optimal only if probabilities reflect deployment prevalence and costs are correctly specified. A prevalence shift or a different cost ratio can change the operationally appropriate threshold. The threshold sweep is a retrospective sensitivity analysis, not an independent demonstration of its prospective performance.

5.3 Hierarchical Malware-Stage Attribution

The stage classifier achieved accuracy 0.9866, macro-F1 0.9873, MCC 0.9796, and log loss 0.0205. It correctly classified all 225 action flows, 197 of 199 infection flows, and 316 of 324 reconnaissance flows. The ten errors were structured rather than diffuse: eight reconnaissance flows were assigned to action, and two infection flows were assigned to reconnaissance. Figure 4 shows that action recall was 1.000, infection precision was 1.000, and reconnaissance F1 was 0.9844. The macro-F1 exceeds the prespecified 0.90 criterion, supporting H3.

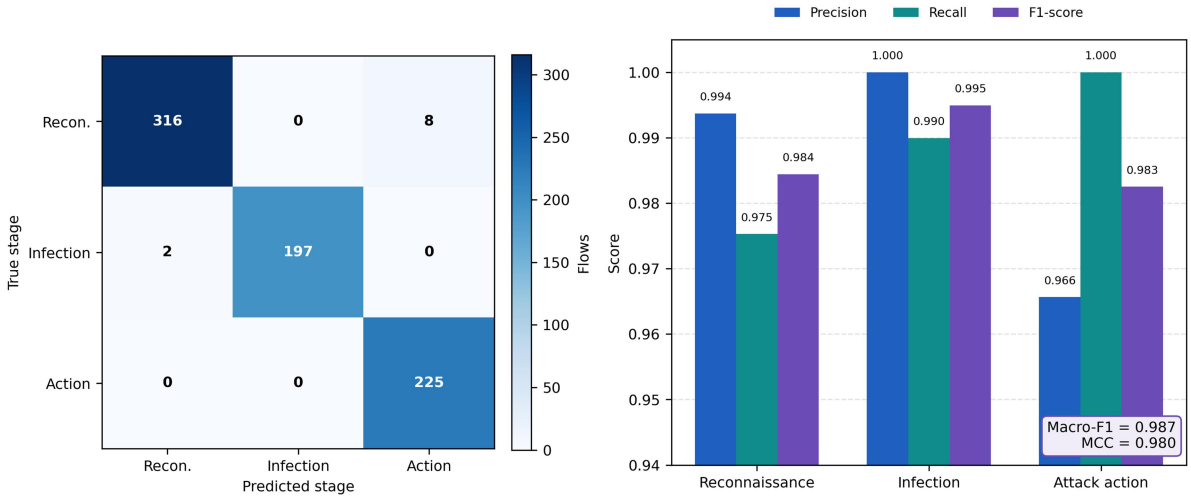


Figure 4. Confusion matrix and per-class results for hierarchical malware-stage attribution

The hierarchy adds operational meaning to a binary alarm. A reconnaissance result can be monitored or rate-limited, an infection result can trigger credential protection and temporary isolation, and an action-stage result can justify rapid containment. These response mappings remain policy recommendations; the experiment evaluates classification, not the downstream effectiveness or safety of automated mitigation.

5.4 Robustness And Temporal Stability

Figure 5 and Table 5 summarize repeated corruption tests. Gaussian perturbation of 1% of the training interquartile range reduced mean F1 to 0.9096, and the value remained near 0.91 at 5% and 10% noise. Random feature dropout was less damaging at moderate levels: 5% dropout retained F1 = 0.9595 and MCC = 0.9564; 10% dropout retained F1 = 0.9198; and 20% dropout reduced F1 to 0.8338. The pattern indicates that median replacement can tolerate sparse missing telemetry better than broad numerical distortion, but performance still deteriorates materially under severe corruption.

Table 5. Mean corruption-test performance

Condition	Precision	Recall	F1	MCC
clean	1.0000	1.0000	1.0000	1.0000
Gaussian noise 1% IQR	0.9910	0.8406	0.9096	0.9060
Gaussian noise 5% IQR	0.9874	0.8413	0.9085	0.9046
Gaussian noise 10% IQR	0.9825	0.8412	0.9063	0.9020
Feature dropout 5%	0.9854	0.9349	0.9595	0.9564
Feature dropout 10%	0.9723	0.8727	0.9198	0.9148
Feature dropout 20%	0.9389	0.7501	0.8338	0.8272

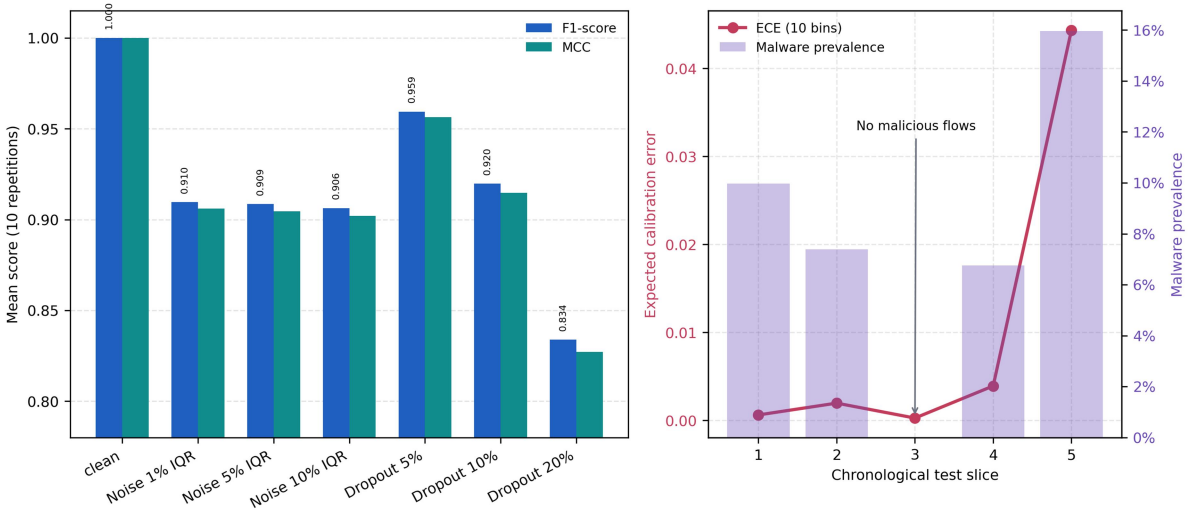


Figure 5. Robustness under telemetry corruption and calibration across chronological test slices

Stage-error analysis. Of 748 held-out malicious flows, 10 were assigned the wrong stage, an error rate of 1.34%. Eight of 324 reconnaissance flows (2.47%) were promoted to attack action; these errors can result in unnecessarily restrictive response if the stage output is directly coupled to automatic containment. Two of 199 infection flows (1.01%) were labeled reconnaissance, potentially delaying a high-priority intervention. These asymmetric consequences support applying conservative policy checks and human review to stage-specific response, which was not experimentally assessed.

All five chronological slices retained perfect class decisions, but their probability quality was not constant. ECE increased from 0.00062 in the first slice to 0.04435 in the fifth slice, which also had the highest malware prevalence. The third slice contained no malicious flows, so positive-class F1 and AUC are not informative there. This separation between classification stability and calibration drift supports monitoring probability distributions and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ECE after deployment, even when headline accuracy remains unchanged.

5.5 Explainability And Feature Dependence

Figure 6 contrasts training-set mutual information with held-out permutation importance. Mutual information ranked forward packet length, total forward length, backward header length, packet rate, and inter-arrival statistics highly. On the held-out test set, total backward header length produced the largest F1 decrease when permuted, followed by destination port, mean forward packet length, ACK count, and minimum forward packet length. The difference between the two rankings is expected: mutual information measures marginal training association, whereas permutation importance measures the predictive dependence of the fitted model in the presence of correlated features.

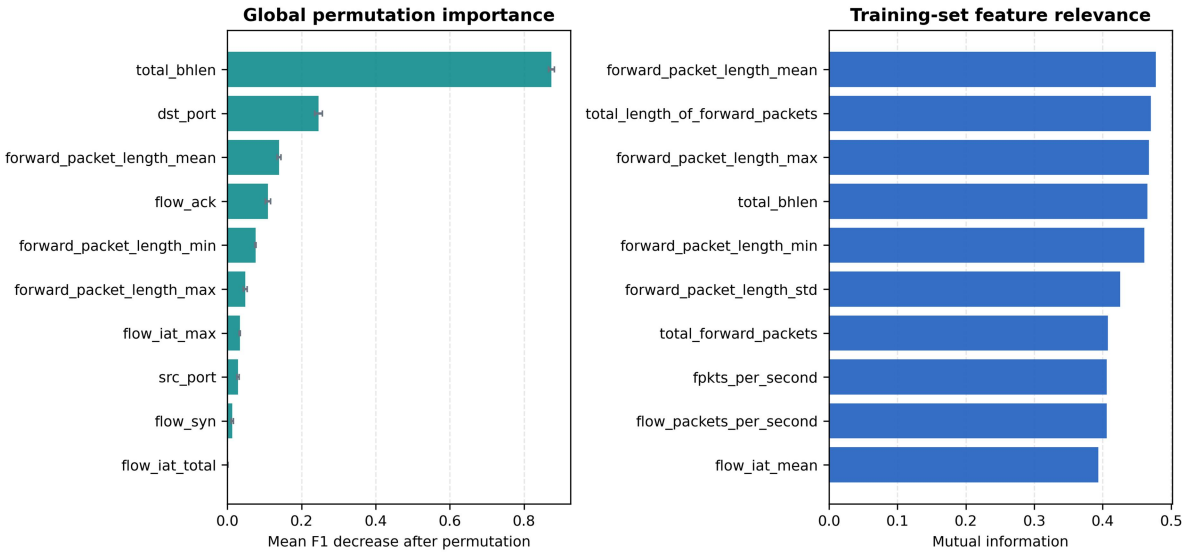


Figure 6. Global Permutation Importance And Training-Set Mutual-Information Ranking

Robustness interpretation. The 10% IQR noise condition reduced recall from 1.0000 to 0.8412, corresponding to about 15.9 percentage points of lost sensitivity; 20% dropout reduced it to 0.7501, a 25.0-point loss. Thus, the word "robust" must be interpreted comparatively within the tested perturbation types, not as a general security guarantee. The observed 0.04435 fifth-slice ECE also motivates distinct monitoring of score reliability and binary error even when the chosen threshold yields perfect class decisions. These are descriptive comparisons of the existing experiments, not tests of statistical significance.

The explanation is global and does not establish causality. Port and packet-shape features can be manipulated by an adaptive adversary, and correlated traffic statistics can redistribute importance. Operational use should therefore log local evidence, monitor feature drift, and combine the model with signatures, inventory controls, and analyst review rather than interpreting the ranking as a permanent malware signature.

5.6 Probability Calibration And Threshold Sensitivity

Figure 7 reports the calibrated score distributions and a threshold sweep computed from the saved model. The Brier score was 0.00363 and ten-bin ECE was 0.01022. The threshold sweep found a zero-error interval from approximately 0.12 to 0.35 on this test set. The predeclared threshold 0.25 lies inside that interval and gives cost FP + 3FN = 0. At 0.40, three false negatives remained; at 0.50, ten false negatives produced a cost of 30; and at 0.60, 23 false negatives produced a cost of 69. Thresholds below 0.12 began to introduce false positives.

Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

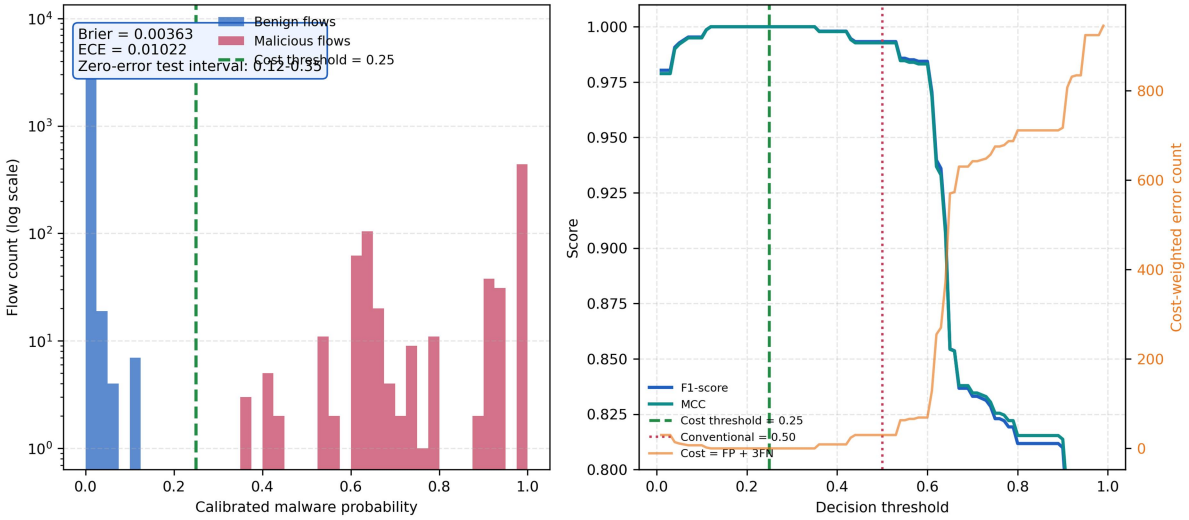


Figure 7. Calibrated Score Distributions And Sensitivity Of F1, MCC, And Cost-Weighted Error To The Decision Threshold

The zero-error interval is not a threshold guarantee for future traffic. Its importance is diagnostic: the cost-based threshold was not tuned to a single isolated test optimum and remained separated from the first observed errors. In deployment, the threshold should be reviewed when class prevalence, calibration, feature distributions, or analyst-confirmed costs change, using a validation stream that is independent of the final reporting set.

5.7 Edge Efficiency And Deployment Footprint

The calibrated binary artifact occupies 189.4 kB and the stage artifact 41.9 kB, for a combined serialized footprint of approximately 231.3 kB. On the evaluation host, binary probability inference required 5.93 ms per 1,000 flows, while stage inference required 10.64 ms per 1,000 malicious flows. The second model is conditionally invoked only after a positive binary decision; therefore, the average pipeline cost depends on the deployment alert rate rather than equaling the sum for every flow.

Table 6. Deployment Footprint And Host-Measured Inference Time

Component	Features	Size (kB)	Inference ms/1,000	Invocation
Binary calibrated detector	24	189.4	5.93	All flows
Stage attribution model	24	41.9	10.64	Malware-positive flows only
Combined serialized package	-	231.3	Workload-dependent	Hierarchical invocation

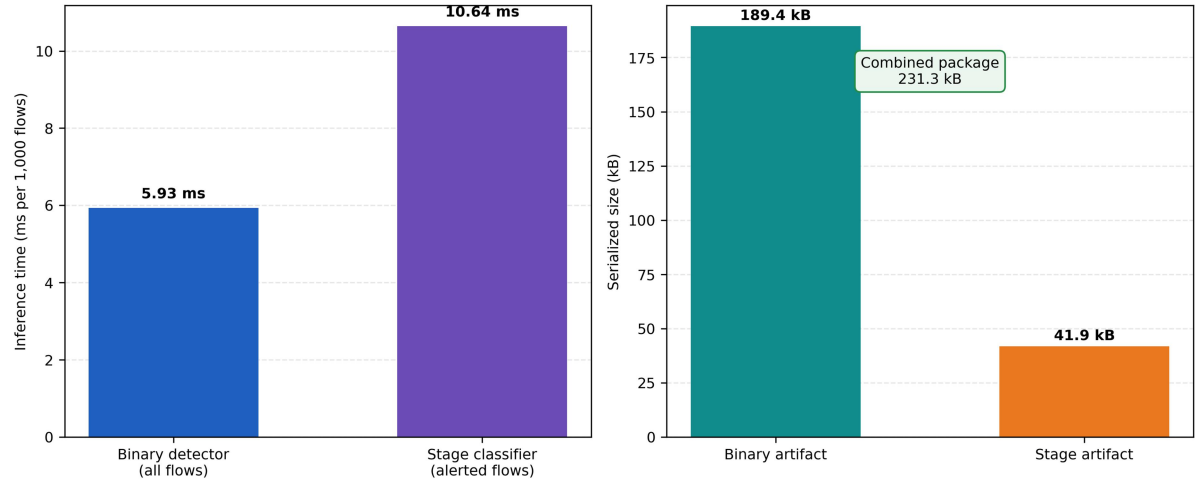


Figure 8. Host-measured inference time and serialized model footprint

The measurements support software compactness but do not establish real-time performance on every gateway. CPU type, concurrent packet processing, flow-export delay, memory pressure, and model-loading strategy can dominate end-to-end latency. A production claim therefore requires direct measurement on representative ARM or x86 edge hardware under realistic traffic and response workloads.

6. DISCUSSION

6.1 Interpretation Of Findings

The main contribution is the architecture surrounding the classifier rather than the perfect binary score in isolation. The strongest raw baseline was already close to the ceiling, with three false negatives. MALGUARD-IoT combines identity exclusion, a 24-feature deployment mask, calibrated scores, an independently justified cost threshold, and a second-stage attribution model. This combination changes the output from an opaque binary prediction into an auditable edge decision containing a probability, threshold, malware stage, explanatory evidence, and policy action.

The study also demonstrates why several evaluation views are necessary. Aggregate classification was perfect, yet calibration varied across chronological slices and corruption sharply reduced recall. The threshold analysis showed a wide clean margin on the reported test file, while the probability metrics showed that the raw HistGradientBoosting scores were better calibrated on that same file. These apparently conflicting results are informative: a model can classify a curated benchmark perfectly without being universally calibrated, robust, or transferable.

Classical boosting remained more competitive than the evaluated shallow neural model for compact tabular flows. This is consistent with evidence that tree ensembles can be strong, efficient baselines for IoT telemetry [8], [10]. Deep sequence, graph, and binary models remain appropriate when richer structures are available [7], [9], [11], but they should be justified through cross-device transfer, temporal evaluation, uncertainty, and edge-resource measurements rather than architectural novelty alone.

Critical synthesis against recent work. Recent IoT-IDS surveys argue that reliable evaluation requires diverse datasets, well specified traffic collection, computational profiling, and representative deployment conditions [27]–[30]. Compared with those desiderata, this study offers explicit cost assumptions, preserved train/test separation, stage-specific consequences, model-size measurement, and controlled corruption stress tests. It does not offer external-dataset replication, adversarially optimized attacks, uncertainty intervals from device-level resampling, or physical-gateway throughput measurements. Its strongest contribution is thus the specification and auditability of a multi-stage detection decision, rather than evidence of statistically significant superiority to all alternatives.

6.2 Operational Use And Monitoring

A gateway can batch completed flow summaries, apply the binary detector continuously, and invoke stage attribution only for alerts. Each record should preserve the model and feature versions, calibrated probability, threshold, stage, principal feature contributions, and response outcome. Policy should separate observation, rate limiting, credential protection, isolation, and blocking rather than applying one irreversible response to every positive flow. Human override is especially important when the score is near the threshold or when the observed feature distribution is outside the training range.

Post-deployment monitoring should track malware prevalence, false-alarm confirmation, missed incidents, ECE, score histograms, feature drift, missing-data rates, stage distribution, and response latency. The increase in fifth-slice ECE illustrates why stable classification alone is not sufficient. Retraining should be triggered by independently labeled evidence and evaluated on a later holdout, not by repeatedly optimizing against the same benchmark.

6.3 Security, Privacy, And Failure Modes

Removing IP addresses and timestamps reduces direct endpoint memorization and the amount of identifying information retained, but flow telemetry is not anonymous. Ports, packet lengths, timing, and connection rates can reveal device behavior. Access control, encryption, retention limits, audit logging, and documented lawful purposes remain necessary. The model artifacts and feature-extraction code must also be integrity-protected because direct modification at the trusted gateway is outside the evaluated threat model.

Adaptive attackers may shift ports, fragment flows, throttle scans, alter packet sizes, create benign-looking cover traffic, or exploit feature-extraction gaps. Robust deployment should therefore combine MALGUARD-IoT with device inventory, signature intelligence, secure update mechanisms, segmentation, rate controls, and incident-response procedures [25]. The classifier

should be treated as one detection layer rather than a complete security boundary.

7. LIMITATIONS AND THREATS TO VALIDITY

First, the experiment represents one Mirai flow collection with three attack stages. Perfect binary decisions can reflect dataset separability and do not establish zero-day, BASHLITE, ransomware, cross-device, or cross-network performance. Second, the public files are already flow-extracted, so this study does not validate packet capture, windowing, ground-truth construction, or online feature latency. Third, although identities and timestamps are excluded, ports and traffic-shape features may still encode scenario-specific artifacts. Fourth, the chronological slices come from the same held-out file and are not a long-term concept-drift study. Fifth, the Gaussian-noise and feature-dropout tests are synthetic; adaptive adversarial evasion was not evaluated. Sixth, latency was measured on the execution host rather than Raspberry Pi, router, or industrial gateway hardware. Seventh, the cost ratio of 3:1 is an explicit design assumption, not an empirically estimated economic cost. Organizations should recalculate the threshold for their own incident and false-alarm consequences.

Validity and statistical interpretation. All percentages in the expanded error analyses are arithmetic functions of the counts already reported in Tables 2–5 and should not be mistaken for new experimental results. Test flows may share device, capture session, or campaign structure; their dependence may invalidate simple binomial confidence statements. Furthermore, the supplied train/test files are described as temporally separated, but the degree of device or campaign separation cannot be established solely from the manuscript. The apparent near-ceiling separation should be checked using independent collection sites, leave-device/campaign-out tests, probability reliability diagrams, and temporally blocked confidence intervals before a production readiness claim.

8. CONCLUSION AND FUTURE WORK

This paper presented MALGUARD-IoT, a compact, hierarchical, calibrated, cost-aware, and explainable framework for flow-based Mirai malware detection at the network edge. On a fixed public train/test partition, the proposed binary model produced no test errors at a predeclared 0.25 threshold and the stage model achieved 0.987 macro-F1. The package records model size, inference time, calibration, feature importance, ablations, chronological slices, and robustness under degraded telemetry. These results support the proposed architecture while also showing that performance declines when telemetry is corrupted.

Future work will evaluate cross-dataset and leave-device-out transfer using N-BaIoT, IoT-23, TON_IoT, and Edge-IIoTset; add uncertainty-based abstention; test adversarial rate and port manipulation; benchmark on representative gateways; estimate incident costs with operators; introduce online calibration and drift detection; and conduct a live deployment in which alert quality, resource consumption, and response outcomes are measured over time.

Further analysis required for deployment. A future confirmatory experiment should measure per-device and per-campaign error with blocked bootstrap uncertainty, compare calibration under target prevalence reweighting, estimate alert volumes per gateway-day at realistic benign base rates, report memory peaks and end-to-end latency on representative ARM hardware, and evaluate evasion by adaptive port and rate manipulation. No numerical outcomes are claimed for these unperformed studies.

DATA AND CODE AVAILABILITY

The complete reproducibility package accompanying this manuscript contains the public train/test CSV files used in the experiment, their SHA-256 hashes, the Python experiment script, dependency list, trained joblib artifacts, generated result tables, and figures. The original source repository is cited in [24].

CONFLICT OF INTEREST

The author declares no conflict of interest.

This study analyzes an existing public network-flow dataset; no participants were recruited and no live-device intervention was conducted. Funding information must be confirmed by the author before submission.

REFERENCES

[1] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J.A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, 'Understanding the Mirai Botnet,' 26th USENIX Security Symposium, 2017, pp. 1093-1110.

[2] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici, 'N-BaIoT: Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders,' IEEE Pervasive Computing, Vol. 17, No. 3, 2018, pp. 12-22.

[3] S. Garcia, A. Parmisano, and M.J. Erquiaga, 'IoT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic,' Zenodo, 2020, doi:10.5281/zenodo.4743746.

[4] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, 'TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems,' IEEE Access, Vol. 8, 2020, pp. 165130-165150.

[5] M.A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, 'Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications,' IEEE Access, Vol. 10, 2022, pp. 40281-40306.

[6] B. Sudharsan, J.G. Breslin, and M.I. Ali, 'Edge2Guard: Botnet Detection at the Edge Using Deep Learning,' IEEE International Conference on Pervasive Computing and Communications Workshops, 2021, pp. 680-685.

[7] N. Prajapati, B. Karki, S. Gopali, and A.S. Namin, 'IoT Malware Network Traffic Detection Using Deep Learning and GraphSAGE Models,' IEEE 49th Annual Computers, Software, and Applications Conference, 2025, pp. 1841-1846, doi:10.1109/COMPSAC65507.2025.00252.

[8] J. Lyon, E. Saeedizade, and S. Sengupta, 'Benchmarking Machine Learning Models for IoT Malware Detection under Data Scarcity and Drift,' IEEE 23rd Consumer Communications and Networking Conference, 2026; arXiv:2601.18736.

[9] M.K. Abd-Ellah et al., 'Malware Detection in IoT Networks with CNNs and Integrated Feature Engineering,' Scientific Reports, Vol. 16, 2026, doi:10.1038/s41598-026-47389-7.

[10] V. Yakovyna and A. Fadieiev, 'Enhancing Internet of Things Intrusion Detection through High-Performance Boosting Ensemble Learning,' Scientific Reports, Vol. 16, Article 19249, 2026, doi:10.1038/s41598-026-50089-x.

[11] L. Xia, Y. Chen, and L. Han, 'A Deep Learning-Based IoT Malware Detection Approach for Electric Vehicle Charging Stations,' Scientific Reports, Vol. 16, Article 10607, 2026, doi:10.1038/s41598-026-45220-x.

[12] M. Maghanaki, S. Keramati, F.F. Chen, and M. Shahin, 'Systematic Evaluation of Machine Learning and Deep Learning Models for IoT Malware Detection,' Sensors, Vol. 26, No. 6, Article 1750, 2026.

[13] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, 'Dos and Don'ts of Machine Learning in Computer Security,' 31st USENIX Security Symposium, 2022, pp. 3971-3988.

[14] J.H. Friedman, 'Greedy Function Approximation: A Gradient Boosting Machine,' Annals of Statistics, Vol. 29, No. 5, 2001, pp. 1189-1232.

[15] L. Breiman, 'Random Forests,' Machine Learning, Vol. 45, 2001, pp. 5-32.

[16] P. Geurts, D. Ernst, and L. Wehenkel, 'Extremely Randomized Trees,' Machine Learning, Vol. 63, 2006, pp. 3-42.

[17] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.

[18] A. Niculescu-Mizil and R. Caruana, 'Predicting Good Probabilities with Supervised Learning,' 22nd International Conference on Machine Learning, 2005, pp. 625-632.

[19] B. Zadrozny and C. Elkan, 'Transforming Classifier Scores into Accurate Multiclass Probability Estimates,' 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 694-699.

[20] R. Battiti, 'Using Mutual Information for Selecting Features in Supervised Neural Net Learning,' IEEE Transactions on Neural Networks, Vol. 5, No. 4, 1994, pp. 537-550.

[21] T. Fawcett, 'An Introduction to ROC Analysis,' Pattern Recognition Letters, Vol. 27, No. 8, 2006, pp. 861-874.

[22] 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, 2015, e0118432.

[23] B.W. Matthews, 'Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme,' Biochimica et Biophysica Acta, Vol. 405, No. 2, 1975, pp. 442-451.

[24] Communication Systems Security Group, 'Datasets: Mirai Botnet Flow Dataset,' GitHub repository, accessed 6 August 2026.

[25] National Institute of Standards and Technology, IoT Device Cybersecurity Guidance for the Federal Government: Establishing IoT Device Cybersecurity Requirements, NIST SP 800-213, 2021.

[26] C. Guo, G. Pleiss, Y. Sun, and K.Q. Weinberger, 'On Calibration of Modern Neural Networks,' 34th International Conference on Machine Learning, 2017, pp. 1321-1330.

[27] B.R. Kikissagbe and M. Adda, "Machine Learning-Based Intrusion Detection Methods in IoT Systems: A Comprehensive Review," Electronics, Vol. 13, No. 18, 2024, Article 3601, doi:10.3390/electronics13183601.

[28] M.M. Rahman, S.A. Shakil, and M.R. Mustakim, "A Survey on Intrusion Detection System in IoT Networks," Cyber Security and Applications, Vol. 3, 2025, Article 100082, doi:10.1016/j.csa.2024.100082.

[29] Q. Abu Al-Haija and A. Droos, "A Comprehensive Survey on Deep Learning-Based Intrusion Detection Systems in Internet of Things (IoT)," Expert Systems, 2025, doi:10.1111/exsy.13726.

[30] "Landscape of Learning Techniques for Intrusion Detection System in IoT: A Systematic Literature Review," Computers & Electrical Engineering, Vol. 120, Part B, 2024, Article 109725, doi:10.1016/j.compeleceng.2024.109725.

[31] "Class Imbalance and Concept Drift Invariant Online Botnet Threat Detection Framework for Heterogeneous IoT Edge," Computers & Security, Vol. 141, 2024, Article 103820, doi:10.1016/j.cose.2024.103820.

[32] "A Novel Feature Selection-Driven Ensemble Learning Approach for Accurate Botnet Attack Detection," Alexandria Engineering Journal, Vol. 118, 2025, pp. 261–277, doi:10.1016/j.aej.2025.01.042.