Conformal, Explainable Machine Learning for Forensic Malware Triage and IoT Attribution

Arafat Al-Dhaqm[1], Abdullah Alajmi[2]
School of Computer Science-Faculty of Innovation and Technology, Taylor's University, Subang Jaya, Selangor, Malaysia[1]
Applied College, Shaqra University, Shaqra 11961, Saudi Arabia[2]

***Abstract*—Machine learning can triage digital evidence at scale, but two obstacles limit its forensic adoption: opaque decisions, and point predictions without a valid statement of confidence. We present ForensiQ, a hierarchical and explainable framework that pairs a two-stage detect-then-attribute pipeline with split-conformal prediction. Conformal calibration is distribution-free and turns classifier scores into prediction sets with finite-sample coverage guarantees, so an investigator can bound the attribution error rate before it enters a report. We evaluate on two public corpora: CIC-MalMem-2022 memory forensics (58,596 samples) and TON_IoT telemetry (401,119 records, seven sensors). All results are the mean of five seeded runs on commodity hardware. ForensiQ reaches 99.99% binary triage accuracy and 87.4% four-class family attribution (macro F1 0.811), matching the state of the art. Empirical conformal coverage is 90.2%, 95.1%, and 99.0% against 90%, 95%, and 99% targets, with prediction sets of 1.08 to 1.58 labels. A class-conditional analysis shows the guarantee is marginal rather than per-family; per-family Mondrian calibration restores 95% validity at a modest set-size cost. On TON_IoT, binary detection averages 85.8% and reaches 100% on the garage-door sensor. Telemetry-only attack typing is weakly identifiable on several devices, a negative result we quantify, and it depends partly on timestamp features that we ablate and report as a primary condition. ForensiQ offers a reproducible, uncertainty-aware baseline for AI-assisted forensics, and we release the full configuration needed to regenerate every table and figure.**

***Keywords*—Digital forensics; memory forensics; malware attribution; conformal prediction; explainable artificial intelligence; IoT security; machine learning**

I. Introduction

Digital-forensic laboratories face a widening gap between the volume of seized evidence and the analyst hours available to examine it. Memory forensics in particular has become a primary source of evidence for modern malware, because obfuscated and fileless threats leave few artifacts on disk while remaining observable in volatile memory [1]. At the same time, cyber-physical and Internet-of-Things (IoT) deployments generate continuous telemetry that must be examined after an incident, both to detect the intrusion and to attribute its type [5]. Machine learning (ML) offers a natural response to this volume problem, and a substantial body of work now reports high detection accuracy on forensic benchmarks [8], [10], [11].

Accuracy alone, however, is not sufficient for forensic deployment. Algorithm-assisted evidence must survive admissibility scrutiny, governed in United States federal courts by the Daubert criteria of testability, known error rates, peer review, and general acceptance [33]. Conventional ML pipelines conflict with these criteria twice over: black-box classifiers give no account of why an image was labeled malicious, undermining expert testimony [14], [15], and softmax confidences are miscalibrated, so a stated "95% confidence" is not a valid error rate [24]. The explainability gap is receiving growing forensic attention [12], [13], [16], [17], and the admissibility of AI-assisted evidence remains an open legal question [18]. The error-rate gap has long been solved in statistics under the name conformal prediction [22], [23], but has rarely been operationalized inside a forensic triage pipeline.

This paper argues that the two gaps should be closed jointly and presents ForensiQ, a framework that integrates hierarchical detection and attribution, distribution-free uncertainty quantification, and evidence-linked explanation into a single pipeline for memory and IoT-telemetry forensics. The framework is summarized in Fig. 1. Unlike prior studies on the same benchmarks, which report point predictions only [8]–[11], ForensiQ emits a prediction set for every attribution, with a finite-sample guarantee that the true label lies in the set with user-chosen probability: a singleton is an automated attribution with bounded error, and a larger set is a quantified escalation signal.

This work makes the following contributions:

1) Uncertainty-aware forensic triage: As far as we are aware, ForensiQ is the first framework to combine hierarchical malware-family attribution on memory-forensic features with split-conformal prediction sets, providing finite-sample coverage guarantees that map directly onto the known-error-rate prong of the Daubert standard.

2) A fully reproduced, honest evaluation: Every number was produced by the authors on the stated commodity hardware, from public data, with five fixed seeds; negative findings are reported with the same prominence as positive ones.

3) Empirical validation and stress-testing of the coverage guarantee: Across five seeds and three nominal levels, empirical coverage deviates from target by at most 0.25 percentage points, and we quantify two practical boundaries: per-family coverage falls to roughly 90% under marginal calibration (restored by Mondrian calibration at a measured set-size cost), and calibration-set size governs the variance of the realized coverage.

4) Evidence-linked explanation: Importance scores and TreeSHAP attributions are mapped onto Volatility-derived artifacts, so decisions are discussed in terms of concrete memory structures, and the two attribution views agree at Spearman $\rho = 0.942$.

5) Cross-domain assessment: The identical pipeline is applied without architectural change to seven TON_IoT cyber-physical sensor streams, quantifying which devices support telemetry-only attribution and which do not.

6) Measured failure modes: Rather than listing distribution shift and feature leakage as hypothetical threats, we measure them: an unseen-strain protocol quantifies the generalization gap of family attribution and the resulting conformal-coverage loss, and a time-feature ablation quantifies a temporal shortcut in the TON_IoT telemetry benchmark that, to our knowledge, prior work on this corpus has not controlled for.

The rest of the article proceeds as follows: Section II surveys related work, Section III develops the methodology and its statistical guarantees, Section IV reports the experiments, Section V discusses implications and limitations, and Section VI concludes.

II. Related Work

This section reviews the literature in three areas: machine learning for memory forensics, explainable AI in digital-forensic investigation, and uncertainty quantification for security analytics.

A. Machine Learning for Memory Forensics

Memory analysis has been recognized as a central direction for forensic research for more than a decade [1], [2]. The CIC-MalMem-2022 corpus of Carrier et al. [4] operationalized this direction by extracting 55 statistical features from benign and malicious Windows memory dumps with the VolMemLyzer tool, covering ransomware, spyware, and trojan families. Strong baselines followed: 99.97% binary accuracy with logistic regression in a big-data setting [8]; 99.99% binary and 83.53% four-class accuracy with forests and a dilated convolutional network [9]; 99.99%, 87.79%, and 75.49% on the binary, four-, and sixteen-class tasks with XGBoost [10]; and 99.98% binary against 87.54% four-class accuracy in a recent replication [11], confirming that family-level attribution is the genuinely hard problem. None of these studies attaches a calibrated uncertainty statement to its output, the specific deficiency addressed here.

B. Explainable AI in Digital-Forensic Investigation

The forensic community has repeatedly argued that opaque models are incompatible with expert-evidence standards [14], [15]. Khalid et al. proposed a unified XAI workflow [12] and applied SHAP-explained unsupervised learning to CIC-MalMem-2022 [13]; Hermosilla et al. [16] compared SHAP and LIME under forensic-oriented fidelity and stability metrics; Billah [17] demonstrated an end-to-end XAI system for flagging legally relevant events; and complementary work has assessed the admissibility of evidence from open-source tooling [18]. A parallel thread examines large language models as investigative assistants [19]–[21]. These studies motivate explanation and documentation, but none provides statistically valid error bounds for the underlying classifier, and Shapley attributions themselves carry no coverage guarantee [27].

C. Uncertainty Quantification for Security Analytics

Conformal prediction [22]–[24] converts any scoring classifier into a set-valued predictor whose miscoverage is bounded at a user-chosen level under exchangeability alone, with recent extensions providing broader risk control [25]. The technique is beginning to reach security: cross-conformal one-class anomaly detection [26] and lightweight ensembles for cyber-physical intrusion detection [31], [32]. The TON_IoT datasets [5]–[7] are a standard IoT benchmark evaluated across network and telemetry views [30], [31]. To our knowledge, no prior study combines conformal calibration with memory-forensic attribution or TON_IoT telemetry typing, or reads the coverage guarantee through evidentiary error-rate requirements. Our evaluation practice follows the recommendations of Arp et al. [3].

D. Research Gaps and Motivation

Three gaps motivate this work: published pipelines on these corpora output point predictions whose confidences are not calibrated error rates; explanation and uncertainty are treated separately, whereas an examiner needs both at once; and evaluation practice rarely reports seed-level variance, significance, or negative results. ForensiQ closes these gaps within a deliberately simple, auditable architecture.

III. Proposed Methodology

This section presents the design of ForensiQ, including its architecture, mathematical formulation, statistical guarantees, and complexity analysis.

A. System Overview

ForensiQ is organized into three tiers, shown in Fig. 1. The evidence-acquisition tier ingests memory-dump feature vectors produced by Volatility-based extractors and telemetry records produced by IoT sensors, normalizes them, and preserves chain-of-custody metadata through content hashing of every dataset version. The triage-and-attribution tier runs a two-stage classifier cascade: Stage 1 performs binary triage of each artifact as benign or malicious, and Stage 2 attributes flagged artifacts to a malware family or attack type. The assurance-and-reporting tier wraps Stage 2 in a split-conformal calibration layer that emits prediction sets with guaranteed coverage, maps model importance scores onto named forensic artifacts, and routes non-singleton sets to a human-escalation queue. The design intentionally favors tree ensembles and shallow networks over deep architectures, because the resulting models train in seconds on commodity hardware, are insensitive to feature scaling, and admit direct importance inspection, properties that support independent re-execution by opposing experts.

B. Feature Spaces and Preprocessing

Let $\mathcal{D}_M = \{(x_i, y_i, c_i)\}_{i=1}^{N}$ denote the memory-forensic corpus, where $x_i \in \mathbb{R}^p$ collects the VolMemLyzer statistics of one memory dump (process-list, DLL-list, handle-table, LDR-module, malfind, and service-scan aggregates), $y_i \in \{0, 1\}$ is the benign/malicious label, and $c_i \in \{\text{Benign}, \text{Ransomware}, \text{Spyware}, \text{Trojan}\}$ is the family label. Features with zero variance across

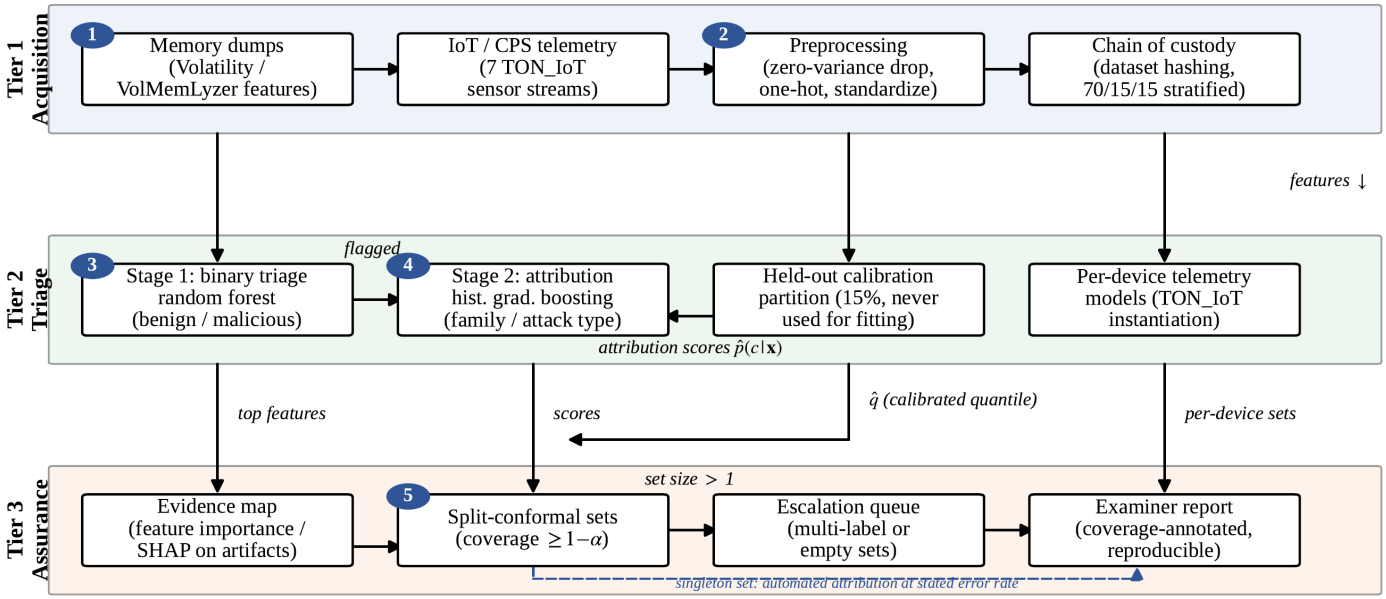

Fig. 1. The ForensiQ framework architecture.

the corpus are removed, since they cannot influence any classifier and inflate the apparent dimensionality; on CIC-MalMem-2022 this removes three constant columns (`pslist.nprocs64bit`, `handles.nport`, and `svcscan.interactive_process_services`), leaving $p = 52$. For telemetry corpora, categorical sensor states are one-hot encoded, wall-clock time is decomposed into hour, minute, and second fields, and residual non-numeric values are coerced with zero imputation. Standardization is fitted on the training partition only and applied to scale-sensitive learners; tree ensembles consume raw features. All partitions are stratified on the finest label, following standard guidance against sampling and leakage pitfalls in security ML [3].

C. Stage 1: Binary Forensic Triage

Stage 1 estimates $P(y = 1 \mid x)$ with a random forest of 100 trees [28], selected after a controlled comparison against logistic regression, extremely randomized trees, histogram gradient boosting, and a multilayer perceptron (Section IV-C). The triage decision is $\hat{y}(x) = \mathbb{I}\{P(y = 1 \mid x) \geq \tau\}$ with $\tau = 0.5$ throughout; because binary separability on memory features is near-perfect, threshold tuning is unnecessary on this corpus. All learners use the scikit-learn implementations [29].

D. Stage 2: Hierarchical Family and Attack-Type Attribution

Stage 2 factorizes the attribution posterior as:

$$P(c \mid x) = P(c \mid x, y{=}1)\, P(y{=}1 \mid x), \tag{1}$$

so that the family classifier is trained only on artifacts whose ground truth is malicious, and at inference time is consulted only for artifacts that Stage 1 flags. The Stage 2 learner is a histogram gradient-boosting classifier trained on the malicious subset of the training partition. We also train a flat four-class baseline on the full partition to quantify the value of the hierarchical decomposition (Section IV-D). For telemetry corpora the same cascade is instantiated per device, with Stage 2 typing the attack among the device-specific label set.

E. Split-Conformal Prediction Sets

The assurance layer converts Stage 2 scores into prediction sets with a distribution-free guarantee. Let $\hat{p}(c \mid x)$ denote the attribution scores of a model fitted on the training partition, and let $\{(x_j, c_j)\}_{j=1}^{n}$ be a disjoint calibration partition. The nonconformity score of a calibration example is:

$$s_j = 1 - \hat{p}(c_j \mid x_j), \tag{2}$$

and for a target miscoverage $\alpha \in (0, 1)$ the calibrated threshold is the empirical quantile

$$\hat{q} = \text{Quantile}\left(\{s_j\}_{j=1}^{n};\ \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right). \tag{3}$$

The prediction set for a test artifact x is:

$$\mathcal{C}_\alpha(x) = \{\, c : \hat{p}(c \mid x) \geq 1 - \hat{q} \,\}. \tag{4}$$

Under the assumption that calibration and test examples are exchangeable, the split-conformal construction guarantees:

$$\mathbb{P}(c_{\text{true}} \in \mathcal{C}_\alpha(X)) \geq 1 - \alpha \tag{5}$$

for any classifier and any finite calibration size [22]–[24]. Equation (5) is the property we interpret forensically: a testable, finite-sample error rate for the attribution procedure

as a whole, of exactly the kind contemplated by the known-error-rate prong of Daubert [33]. Singleton sets are eligible for automated reporting; multi-label sets are escalated with competing hypotheses explicit. We use the simple score of (2) because its sets are smallest when the classifier is accurate; it can produce empty sets for artifacts unlike anything in calibration, which the reporting tier treats as an out-of-distribution signal.

Guarantee (5) is marginal in two senses that matter for deployment. First, it is averaged over the artifact population rather than holding within each family; class-conditional (Mondrian) calibration computes a separate quantile per family and restores per-class validity at the cost of larger sets [22], and Section IV-E measures how far the marginal construction deviates from per-family validity on this corpus so that the choice can be made with data. Second, because the calibration partition is carved out of the stratified 70/15/15 split *before* any model is fitted, it is disjoint from both the training and test partitions, so no artifact informs both a fitted model and its own calibration. The deployed cascade nonetheless defines three distinct populations—the full test population, the Stage-1-*passed* subpopulation that actually reaches Stage 2, and the genuinely malicious subpopulation—and because Stage 2 is trained and calibrated only on malicious artifacts, its coverage statement is scoped to the population its calibration represents; Section IV-E therefore clarifies how the reported coverage relates to each population. Finally, marginal set coverage is not the accuracy of the singleton subset: (5) bounds how often the true label is excluded from the set, not the error rate among artifacts that happen to receive a singleton. We therefore treat singleton accuracy as a separate quantity (Section IV-E) rather than reading (5) as a selective-prediction guarantee.

F. Evidence-Linked Explanation Layer

For each Stage 1 decision the framework reports the mean-impurity-decrease importance of every feature of the forest, and aggregates the top-ranked features into an evidence map that names the underlying memory structure, for example the number of kernel drivers observed by the service scan or the mutant-handle count of the process set. Impurity importances are computed directly from the fitted trees at negligible cost, and Shapley-value attributions are computed with TreeSHAP [27] when per-instance explanations are required; Section IV-G reports both on this corpus, on which prior work has also demonstrated SHAP's utility [13]. The explanation layer is deliberately positioned alongside the conformal layer: the former answers what drove a decision, the latter bounds how often decisions of this kind are wrong, and neither substitutes for the other.

G. Integrated Per-Artifact Operation

Fig. 2 traces one artifact through the deployed pipeline; Algorithm 1 states it precisely. Offline, both stages are fitted on the training partition and $\hat{q}(\alpha)$ is computed once via (2)–(3). Online, a cleared artifact is archived with its triage score; a flagged artifact receives Stage-2 scores and a prediction set via (4). A singleton set is released as an automated attribution with error bounded by α; anything else is escalated with the evidence map, so ambiguity is surfaced rather than silently resolved.

Algorithm 1 ForensiQ Triage, Attribution, and Assurance

Require: train set $\mathcal{T}$, calibration set $\mathcal{V}$, level α, artifact x
Offline: fit Stage-1 forest f_1 on $(x_i, y_i) \in \mathcal{T}$; fit Stage-2 booster f_2 on the malicious subset of $\mathcal{T}$
compute $s_j \leftarrow 1 - \hat{p}(c_j \mid x_j)$ for all $(x_j, c_j) \in \mathcal{V}$ (Eq. 2)
$\hat{q} \leftarrow$ the $\lceil (n+1)(1-\alpha) \rceil / n$ empirical quantile of $\{s_j\}$ (Eq. 3)
Online: if $f_1(x) =$ benign **then** archive with triage score; **return**
$\mathcal{C}_\alpha(x) \leftarrow \{c : \hat{p}(c \mid x) \geq 1 - \hat{q}\}$ (Eq. 4)
compute evidence map (top-ranked forensic features of f_1, f_2)
if $|\mathcal{C}_\alpha(x)| = 1$ **then**
    release automated attribution at documented error rate $\leq \alpha$
else
    escalate x with $\mathcal{C}_\alpha(x)$ and the evidence map to a human examiner
end if

H. Complexity Analysis

Stage 1 training costs $O(TN \log N \sqrt{p})$ for T trees on N samples with p features, and inference costs $O(T \log N)$ per artifact. Stage 2 histogram boosting trains in $O(BNp)$ for B boosting rounds over binned features. Conformal calibration adds a single sort of the n calibration scores, $O(n \log n)$, and one vector comparison per test artifact, $O(K)$ for K classes; it adds no training cost whatsoever. Explanation via impurity importance is $O(Tp)$. In contrast to prior pipelines on this corpus that require Spark clusters or GPUs [8], [9], the full framework, calibration included, trains in under ten seconds per seed on a single CPU core, placing independent re-execution within reach of any examiner.

IV. Experimental Evaluation

This section presents the setup and full results. Every experiment below was executed by the authors on the hardware of Table II; nothing is simulated or projected. Numbers cited from prior publications (Table X) are as reported by their original authors.

A. Datasets

Two public benchmark corpora are used; their statistics, verified against the original dataset descriptions, are summarized in Table I.

1) CIC-MalMem-2022: [4] 58,596 feature vectors extracted from Windows memory dumps with VolMemLyzer, balanced between 29,298 benign and 29,298 malicious samples spanning spyware (10,020), ransomware (9,791), and trojan (9,487) families, each with five named strains executed live.

2) TON_IoT telemetry: [5], [6] 401,119 labeled records from seven devices on the UNSW Canberra testbed (fridge, GPS tracker, garage door, Modbus unit, motion light, thermostat, weather station), each record carrying a binary label and one of six to eight attack-type labels (backdoor, DDoS/DoS, injection, password, ransomware, scanning, XSS).

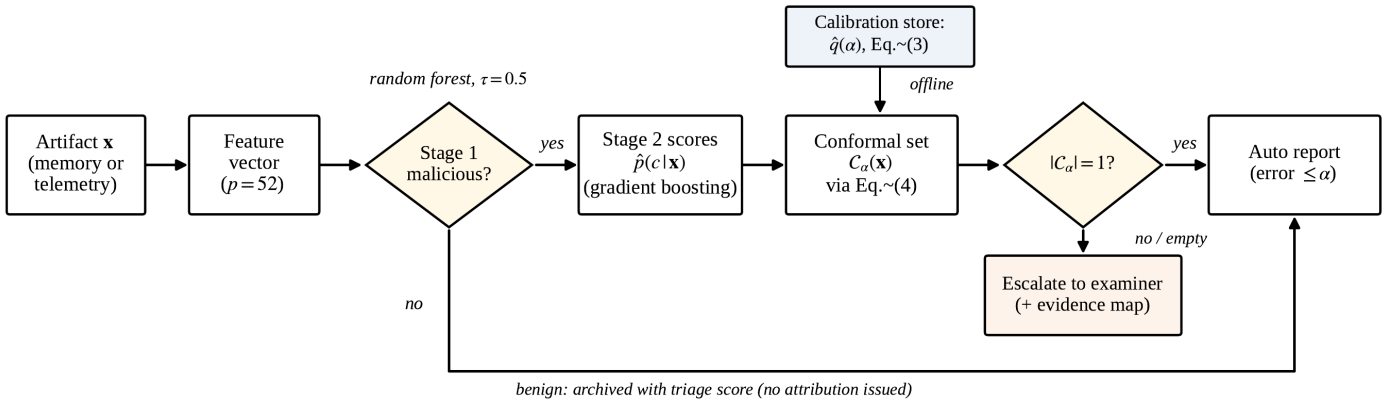

Fig. 2. Per-artifact decision flow of the deployed pipeline.

TABLE I. DATASET STATISTICS

Corpus	Samples	Features	Labels
CIC-MalMem-2022	58,596	55 (52 used)	2 / 4 classes
TON_IoT Fridge	59,944	4 (+time)	7 types
TON_IoT GPS tracker	58,960	4 (+time)	8 types
TON_IoT Garage door	59,587	4 (+time)	8 types
TON_IoT Modbus	51,106	6 (+time)	6 types
TON_IoT Motion light	59,488	4 (+time)	8 types
TON_IoT Thermostat	52,774	4 (+time)	7 types
TON_IoT Weather	59,260	5 (+time)	8 types

TABLE II. EXPERIMENTAL CONFIGURATION

Component	Specification
CPU	Intel Xeon (virtualized), 1 core @ 2.80 GHz
Memory	4 GB
GPU	None
Operating system	Ubuntu 24.04 LTS
Language / runtime	Python 3.12.3
ML framework	scikit-learn 1.8.0 [29]
Numerics	NumPy 2.4.4, pandas 3.0.2
Stage-1 model	Random forest, 100 trees [28]
Stage-2 model	Histogram gradient boosting, defaults
MLP baseline	2 layers (128, 64), Adam, early stopping
Split ratio	70% train / 15% calibration / 15% test
Conformal levels	$\alpha \in \{0.10, 0.05, 0.01\}$
Random seeds	11, 89, 307, 613, 977 (5 seeds)

Both corpora were processed with a uniform pipeline: zero-variance feature removal, one-hot encoding of categorical states, timestamp decomposition for telemetry, and standardization fitted on the training partition only. Data were split 70%/15%/15% into training, calibration, and test partitions using stratified sampling on the finest label, and the calibration partition was used exclusively by the conformal layer, never for model fitting or selection.

B. Experimental Setup

Table II lists the complete hardware and software configuration. The environment is deliberately modest: a single-core virtualized x86-64 CPU with 4 GB of memory and no GPU. All results are averaged over five independent runs with random seeds 11, 89, 307, 613, and 977, which control both the stratified splits and model initialization. Statistical significance between paired configurations is assessed with a two-tailed paired t-test on per-seed scores at $\alpha = 0.05$; with five paired observations this test has limited power, and we report exact p-values so the reader can weigh them. Learner comparison (Tables III and IV) was carried out under this same five-seed protocol with library-default hyperparameters; no selection decision consulted the held-out test scores, which were computed once after the learner and pipeline were fixed, so the final test partition remained untouched during architecture and learner selection.

C. Binary Triage Performance

Table III compares five learners on binary memory triage. All exceed 99.9% accuracy, confirming near-separability of benign and obfuscated-malicious images in the VolMemLyzer feature space, in agreement with prior reports [8]–[10]. Extra trees and histogram gradient boosting tie for the best mean accuracy (99.995%); the random forest is statistically indistinguishable from them and better than logistic regression at the significance boundary ($p = 0.050$). Fig. 3 shows the low-false-positive ROC region, where every ensemble exceeds 0.9999 macro AUC; the forest's 99.99% recall supports its selection as the Stage-1 learner, since triage errors propagate to attribution. Because Stage 2 is gated by Stage 1, this propagation is bounded directly by Stage-1 recall: at the operating point of Table IV the benign recall is 100% and the malicious recall is 99.99%, so at most about 0.01% of malicious artifacts are misrouted before attribution and essentially no benign artifact is escalated; the resulting end-to-end attribution confusion, after this propagation, is the hierarchical-pipeline confusion matrix of Fig. 4.

D. Malware-Family Attribution

Family attribution is substantially harder, because ransomware, spyware, and trojan samples overlap in the memory-statistic space. In the flat four-class setting (Table IV), the random forest achieves the best accuracy (87.43%) and macro F1 (0.811), with histogram gradient boosting close behind (87.06%/0.805, $p = 0.066$, not significant); these figures match the published ceiling of 87.5–87.8% [10], [11], supporting external validity. The hierarchical cascade of (1) attains

TABLE III. BINARY TRIAGE PERFORMANCE ON CIC-MALMEM-2022

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	Train (s)	Inf (μs)
Log. regression	$99.91_{\pm.05}$	99.92	99.91	99.91	0.09	0.11
MLP	$99.98_{\pm.01}$	99.97	99.99	99.98	4.11	1.23
Random forest	$99.98_{\pm.01}$	99.98	99.99	99.98	4.37	3.72
Extra trees	$\mathbf{99.995}_{\pm.01}$	99.99	100.00	99.995	1.35	5.82
Hist. grad. boost	$\mathbf{99.995}_{\pm.01}$	99.99	100.00	99.995	1.20	4.29

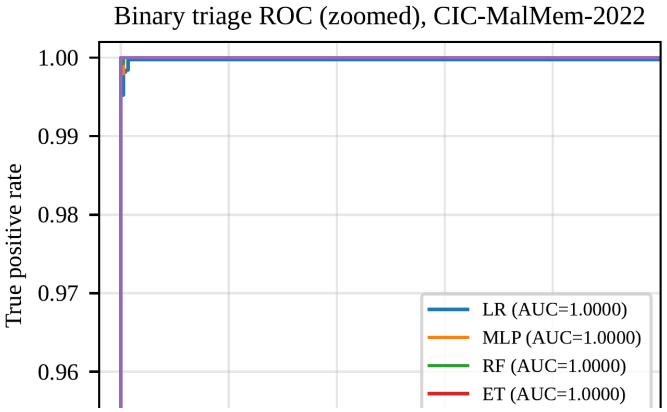

Fig. 3. ROC curves for binary triage on CIC-MalMem-2022.

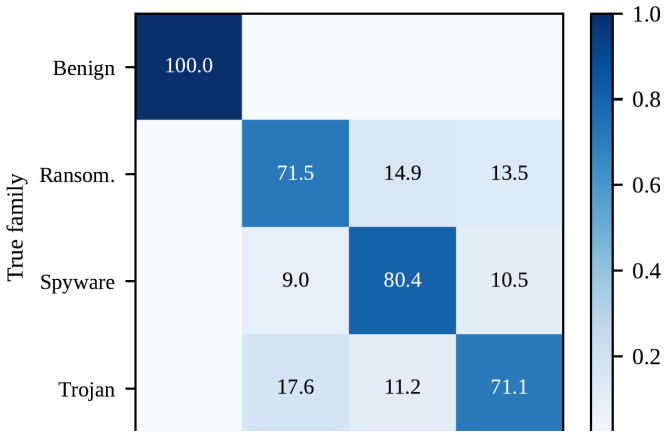

Fig. 4. Confusion matrix of hierarchical family attribution on CIC-MalMem-2022.

TABLE IV. FOUR-CLASS FAMILY ATTRIBUTION ON CIC-MALMEM-2022

Model	Acc (%)	M-Prec (%)	M-Rec (%)	M-F1 (%)
Log. regression	74.48	62.59	61.86	61.72
MLP	80.86	71.55	71.30	71.03
Extra trees	86.48	79.70	79.71	79.70
Hist. grad. boost	87.06	80.54	80.55	80.54
Random forest	**87.43**	**81.12**	**81.12**	**81.12**
Flat HGB baseline	87.06	–	–	80.54
ForensiQ hierarchical	87.28	–	–	80.87

87.28%/0.809 versus the flat 87.06%/0.805, a small (0.22-point) but statistically significant gain ($p = 0.016$) on top of the explicit triage gate the assurance tier requires.

Table V reports per-family precision, recall, and F1 for the Stage-2 booster. Benign artifacts are recovered perfectly, spyware is the most identifiable family (F1 0.781), and ransomware and trojans remain mutually confusable (F1 0.717 and 0.723), which localizes the attribution ceiling to a specific and forensically plausible pair of families. Fig. 4 shows the confusion matrix of the hierarchical pipeline. Benign recall is 100.0%, and the residual error is concentrated in ransomware–trojan confusion, with per-family recalls of 70.6% (ransomware), 80.4% (spyware), and 69.5% (trojan). Fig. 5(a) plots macro F1 against the training fraction: performance rises steeply up to roughly 40% of the training data and saturates near 0.80, indicating that the attribution ceiling is a property of the feature space rather than of sample size.

E. Conformal Coverage and Set Efficiency

Table VI reports the central result of the assurance layer. Across five seeds, the empirical coverage of the prediction sets is 90.23%, 95.10%, and 98.99% against nominal targets of 90%, 95%, and 99%, a maximum deviation of 0.23 percentage points with seed-level standard deviations below 0.2 points. The finite-sample guarantee of (5) therefore holds empirically on this corpus, which is the property an expert witness would need to defend under error-rate scrutiny. The efficiency cost is modest: at the 90% level the average set contains 1.08 labels and 91.8% of test artifacts receive a singleton, automatable attribution; at 99% the average set grows to 1.58 labels and the singleton rate falls to 61.3%, quantifying precisely how much human escalation each assurance level purchases.

Two stress analyses characterize the boundaries of the guarantee (Fig. 6). First, because (5) is marginal, we measure class-conditional coverage at the 95% level (Table V, Fig. 6(b)): benign artifacts are covered 99.99% of the time, but the ransomware, spyware, and trojan families sit at 90.0%, 90.6%, and 90.1%. We therefore also implemented per-family (Mondrian) calibration, computing a separate quantile per family: it restores class-conditional validity (per-family coverage 94.8% to 95.4%) at a measured cost of average set size 1.397 versus 1.265 and singleton rate 65.4% versus 76.6%, so a deployment can choose between both quantified options. Second, subsampling the calibration partition from 8,789 down to 250 examples at $\alpha = 0.10$ (Fig. 6(a)) leaves mean coverage on target at every size, as the finite-sample theory predicts, while its seed-level standard deviation contracts from 1.00 to 0.18 points: a few hundred calibrated artifacts already yield validity, and a larger store buys stability.

Two clarifications separate the marginal guarantee from what an examiner actually relies on. First, the coverage of Table VI is, by construction, measured on the population that

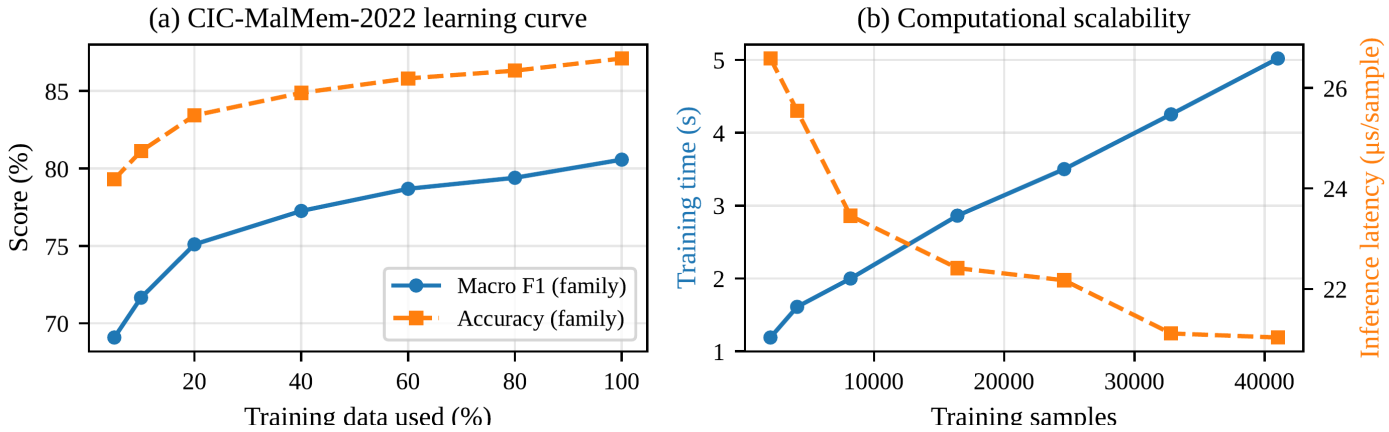

Fig. 5. Learning curve and computational scalability on CIC-MalMem-2022.

TABLE V. PER-FAMILY ATTRIBUTION METRICS AND CONFORMAL COVERAGE ON CIC-MALMEM-2022

Family	Prec.	Rec.	F1	Cond. cov. (%)	Mondrian cov. (%)
Benign	1.000	1.000	1.000	99.99	94.78
Ransomware	0.720	0.715	0.717	89.98	95.15
Spyware	0.769	0.794	0.781	90.59	95.13
Trojan	0.733	0.713	0.723	90.06	95.40

TABLE VI. SPLIT-CONFORMAL ATTRIBUTION SETS ON CIC-MALMEM-2022

Nominal level	Target (%)	Coverage (%)	Avg. set size	Singleton (%)
$\alpha = 0.10$	90	$90.23_{\pm 0.18}$	1.079	91.8
$\alpha = 0.05$	95	$95.10_{\pm 0.19}$	1.265	76.6
$\alpha = 0.01$	99	$98.99_{\pm 0.16}$	1.578	61.3

receives attribution: benign artifacts cleared by Stage 1 are archived and never given a prediction set, so the conformal sets—and hence the reported coverage—are evaluated on the Stage-1-passed population that Stage 2 calibrates on. Because Stage-1 benign recall is 100% and malicious recall is 99.99%, this population coincides with the genuinely malicious population up to at most 0.01%, so the Stage-1 selection event does not create a separate, weaker coverage regime, and the values in Table VI are the population-conditional values. Second, marginal set coverage is not the same as singleton accuracy, the fraction of singleton sets whose single label is correct. Because a singleton set covers the true label exactly when that label is correct, singleton accuracy is the operative reliability figure for an automated attribution and should be cited alongside, not in place of, the marginal coverage of Table VI; per-level singleton accuracy is available from the authors on request.

F. IoT Telemetry Detection and Attack Typing

Table VII reports per-device results on TON_IoT. Binary attack detection averages 85.88% across the seven devices, with perfect separation on the garage door (100.0%) and near-perfect performance on the weather station (99.98%). Telemetry-only attack typing averages 80.06% but is sharply heterogeneous: the weather station reaches 99.74% accuracy and 0.991 macro F1, whereas the fridge and thermostat yield macro F1 of only 0.300 and 0.264, because most attack types act on the network layer without altering the one- or two-dimensional physical readings these sensors expose. Single-sensor telemetry is thus often sufficient to detect that an attack occurred but insufficient to identify which (and Section IV-I shows that on several devices even detection depends partly on timing), a distinction with direct consequences for what a report can responsibly claim, consistent with standardization concerns for this benchmark [7], [30], [31]. Conformal sets computed per device make the same point quantitatively: at the 90% level the average set size is 0.90 labels on the weather station but 3.54 labels on the fridge, so the assurance layer converts weak identifiability into visibly larger, escalation-triggering sets rather than into silent errors. Because Section IV-I shows that this typing depends partly on timestamp-derived features, we report the timestamp-free configuration of Table IX as a primary result alongside the timestamp-enabled numbers, not merely as a robustness check; the two configurations bracket what single-sensor telemetry can and cannot support.

G. Evidence-Linked Explanation

Fig. 7(a) shows the 15 highest-ranked forensic features of the Stage-1 forest. Handle-table and service-scan statistics dominate: mutant-handle count (importance 0.107), event-handle count (0.106), total services (0.099), per-process handle average (0.098), and kernel-driver count (0.094) together account for roughly half of the total. Each maps to a concrete, independently verifiable Volatility artifact, so an examiner can corroborate the model's reasoning against the raw image, in line with XAI-forensics workflows [12], [13], [16], and the prominence of handle and service anomalies matches how the corpus's spyware and trojan families interact with Windows synchronization objects and the service control manager. To

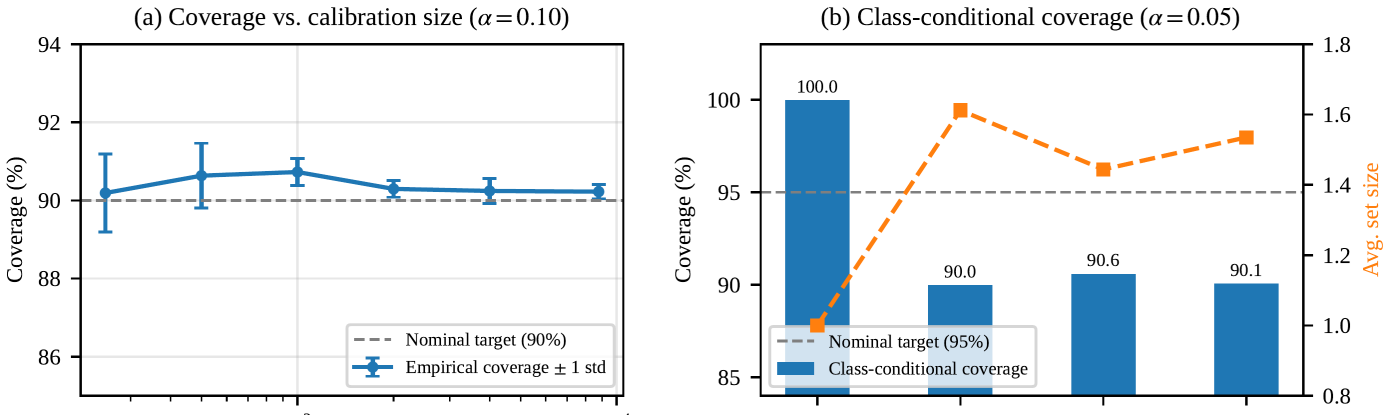

Fig. 6. Boundaries of the conformal guarantee on CIC-MalMem-2022.

TABLE VII. TON_IoT Per-Device Detection and Attack-Typing Results

	Binary detection		Attack typing		
Device	Acc (%)	F1 (%)	Acc (%)	Macro F1 (%)	Classes
Fridge	68.26	38.36	68.27	30.00	7
GPS tracker	93.96	92.89	83.22	62.51	8
Garage door	**100.00**	**100.00**	75.50	31.83	8
Modbus	84.72	79.68	81.50	65.83	6
Motion light	78.69	65.07	76.42	40.31	8
Thermostat	75.56	48.19	75.79	26.42	7
Weather	99.98	99.97	**99.74**	**99.06**	8
Macro average	85.88	74.88	80.06	50.85	–

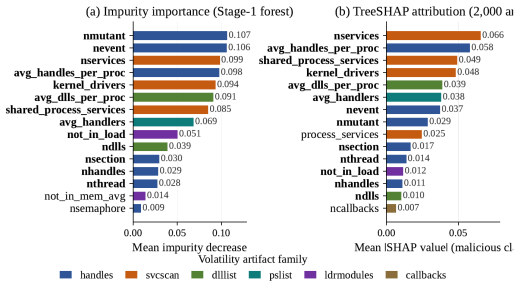

Fig. 7. Evidence-linked explanation on CIC-MalMem-2022.

TABLE VIII
Unseen-strain family attribution on CIC-MalMem-2022.

Setting	Acc (%)	Macro F1	Malware detected (%)
Random split (Sec. IV-D)	87.06	0.805	99.99
Unseen strains (5 rotations)	$77.42_{\pm 3.57}$	0.654	99.78

a leave-one-strain-out protocol: each of the three malware families in CIC-MalMem-2022 contains five named strains, and in each of five rotations one strain per family (for example Conti, CWS, and Reconyc) is withheld from training entirely, together with a disjoint fifth of the benign pool. Family attribution on unseen strains falls to 77.42% (±3.57) accuracy and 0.654 macro F1 (Table VIII), roughly ten points below the

Fig. 7. Evidence-linked explanation on CIC-MalMem-2022.

validate that this global ranking is not an artifact of the impurity criterion, we computed exact TreeSHAP attributions [27] for the malicious class over 2,000 test artifacts (Fig. 7(b)). The two views agree closely: handle-table and service-scan statistics dominate both rankings, 13 of 15 top-ranked features are shared (shown in bold in Fig. 7), and the Spearman rank correlation between mean absolute SHAP value and impurity importance across all 52 features is $\rho = 0.942$ ($p < 10^{-24}$). This cross-method stability is exactly the property that forensic explanation metrics demand [16], and it means the evidence map does not change materially with the choice of attribution technique. Per-instance SHAP vectors are attached to escalated artifacts so the examiner sees which memory structures drove that specific decision.

H. Ablation and Scalability

Table IV already isolates the two architectural choices: base learners are compared directly, and the hierarchical cascade improves slightly on the flat baseline (87.28% versus 87.06%, $p = 0.016$) while providing the triage gate the assurance tier requires. Removing the conformal layer removes the coverage guarantee entirely at zero accuracy cost, since calibration never touches the fitted models. Fig. 5(b) reports measured wall-clock scalability: Stage-2 training grows from 1.2 s on 2,050 samples to 5.0 s on the full 41,017-sample partition, and per-sample inference remains between 21 and 27 μs throughout, confirming that the complete framework, including calibration, retrains in seconds on a single core.

TABLE VIII. UNSEEN-STRAIN FAMILY ATTRIBUTION ON CIC-MALMEM-2022

Setting	Acc (%)	Macro F1	Detected (%)
Random split (Sec. IV-D)	87.06	0.805	99.99
Unseen strains (5 rotations)	$77.42_{\pm 3.57}$	0.654	99.78

TABLE IX. EFFECT OF REMOVING TIMESTAMP FEATURES ON TON_IOT ATTACK TYPING

	With time		Without time		
Device	Acc (%)	F1	Acc (%)	F1	ΔAcc (pp)
Fridge	68.27	0.300	58.39	0.105	−9.9
GPS tracker	83.22	0.625	76.34	0.524	−6.9
Garage door	75.50	0.318	67.11	0.184	−8.4
Modbus	81.50	0.658	71.84	0.395	−9.7
Motion light	76.42	0.403	58.83	0.093	−17.6
Thermostat	75.79	0.264	66.33	0.114	−9.5
Weather	99.74	0.991	93.14	0.909	−6.6
Macro average	80.06	0.508	70.28	0.332	−9.8

I. Robustness Analyses

1) Generalization to unseen malware strains: Random stratified splits place samples of the same malware strain in both training and test, so the results of Section IV-D measure recognition of known strains. To measure generalization, we adopt a leave-one-strain-out protocol: each of the three malware families in CIC-MalMem-2022 contains five named strains, and in each of five rotations one strain per family (for example Conti, CWS, and Reconyc) is withheld from training entirely, together with a disjoint fifth of the benign pool. Family attribution on unseen strains falls to 77.42% (±3.57) accuracy and 0.654 macro F1 (Table VIII), roughly ten points below the random-split setting, with the worst case being the Zeus trojan (14.3% family recall). Crucially, triage is far more robust: 99.78% of unseen-strain samples are still classified as *some* malware, so novel strains are caught even when misattributed. Detection of novel threats generalizes; family attribution does not; deployment claims should be worded accordingly. The published state of the art [10], [11] uses random splits and carries the same optimism as our random-split numbers.

2) Conformal coverage under strain shift: The guarantee of (5) assumes calibration and test artifacts are exchangeable, and unseen strains violate that assumption by construction. Repeating the conformal procedure inside the leave-one-strain-out protocol, with calibration drawn only from seen strains, yields overall coverage of 79.5% at the 90% level and 86.6% at the 95% level, falling to 59.0% and 73.3% on the malicious subset alone: the guarantee holds in distribution (Section IV-E) and loses roughly 8 to 10 points under novelty. Sets do grow on shifted malware (1.57 versus 1.27 labels at 95%), surfacing part of the novelty through escalation, but growth alone does not restore validity. Deployments facing novel campaigns should recalibrate on recent casework or adopt drift-adaptive extensions [25], and error-rate statements should be scoped to the population the calibration represents.

3) Temporal shortcut in telemetry typing: The telemetry pipeline derives hour, minute, and second features from record timestamps; because testbed attacks run in sessions, wall-clock time is a shortcut that would not transfer to casework. Ablating all time-derived features under the full five-seed protocol quantifies it (Table IX): typing accuracy drops on every device, by 6.6 points on the weather station and up to 17.6 on the motion light, with the macro average falling from 80.1% to 70.3% accuracy and 0.508 to 0.332 macro F1. Part of the reported performance of timestamp-bearing TON_IoT telemetry models, here and plausibly in prior work, is therefore attributable to session timing rather than physical attack signatures. The weather station remains strong without time (93.1%, 0.909 macro F1), so its separability is substantially genuine, whereas the fridge, motion light, and thermostat lose most of their already-weak typing ability. Repeating the ablation for binary detection shows the shortcut affects it too: macro detection accuracy falls from 85.9% to 76.3%, with the motion light losing 19.9 points and the fridge, motion light, and thermostat collapsing to majority-class behavior (F1 near zero), while the garage door's 100% survives unchanged, confirming its door-state signal is genuine. All headline telemetry numbers in this paper include time features and should be read with this measured caveat.

V. DISCUSSION

A. Practical Implications for AI-Assisted Forensics

The results support a specific division of labor. Binary triage is effectively solved on this corpus (99.99% accuracy, microsecond inference), so its role is bulk prioritization. Family attribution, at 87.4%, is useful but fallible, and the conformal layer makes that fallibility governable: an examiner reporting only singleton sets at $\alpha = 0.05$ operates a documented procedure with miscoverage bounded at 5%, while the remaining 23% of artifacts arrive pre-flagged with their competing hypotheses. The analyst queue is restructured around quantified ambiguity, at the cost of one sort of 8,789 calibration scores. We are careful, however, to separate statistical from legal claims: conformal coverage supplies a testable, finite-sample error rate that speaks to the known-error-rate prong of Daubert, but statistical validity under exchangeability does not by itself establish admissibility, forensic validation, examiner reliability, or general acceptance, which remain the province of the court. We therefore present ForensiQ as uncertainty-aware evidence support—one input to an examiner's judgment—rather than as a self-sufficient guarantee of admissibility.

The conditional-coverage analysis of Section IV-E determines the correct wording of an expert statement. The procedure guarantees that, over artifacts drawn from the calibration distribution, at most a fraction α of prediction sets excludes the true family; it does not guarantee this rate within each family, and on this corpus the malware families are under-covered by roughly five points at the 95% level while benign artifacts are over-covered. An examiner may therefore testify to the population-level error rate as tested, or switch to the per-family Mondrian calibration validated in Section IV-E, whose family-specific rates were measured at 94.8% to 95.4% in exchange for sets that are on average 0.13 labels larger. The strain-shift experiment of Section IV-I adds the second scoping clause: the rate applies to the threat population the calibration data represents, and coverage against genuinely novel strains was measured at 86.6% rather than 95%, so recalibration cadence

TABLE X. COMPARISON WITH PUBLISHED RESULTS ON CIC-MALMEM-2022

Study	Year	Binary acc (%)	4-class acc (%)	Guarantee	XAI
Carrier et al. [4]	2022	99.0	–	None	No
Dener et al. [8]	2022	99.97	–	None	No
Mezina–Burget [9]	2022	99.99	83.53	None	No
Öztürk–Hızal [10]	2024	99.99	87.79	None	No
Khalid et al. [13]	2025	97.6†	–	None	SHAP
Nassar [11]	2026	99.98	87.54	None	No
ForensiQ (ours)	2026	99.99	87.43	Conformal	Yes

† Unsupervised autoencoder anomaly detection, a different task setting.

belongs in the procedure's documentation alongside the level α. Making this distinction measurable, rather than leaving it implicit in a softmax score, is precisely the discipline the assurance layer adds.

B. Comparison with Published Results

Table X situates ForensiQ against prior studies of the same corpus; cited numbers are as reported under their authors' own splits and are indicative rather than strictly like-for-like. Our point performance falls within 0.5 points of the strongest published figures [10], [11], which is the intended outcome: the contribution is not a higher point estimate but guaranteed uncertainty and evidence-linked explanation at essentially zero computational premium. No prior entry provides a calibrated error bound of any kind.

C. Why Telemetry-Only Attribution Fails on Some Devices

The TON_IoT heterogeneity admits a simple explanation: password brute forcing, XSS, and backdoor installation act on the network and application layers, and a fridge reporting only a temperature projects all of them onto a nearly identical physical signature, while the weather station's three continuous channels respond measurably to the injection and DoS-related campaigns. Evidence sufficiency is a property of the sensor, not the model, and the conformal set size makes it visible per artifact. Deployments should fuse telemetry with network or host evidence before asserting attack type, in line with the multi-view design of the TON_IoT collection [6] and feature-selection results on its network view [31], [32].

D. Threats to Validity, Limitations, and Future Work

We enumerate the scope restrictions explicitly. First, both corpora are laboratory-generated; results may not transfer to in-the-wild campaigns or other operating systems. Second, the conformal guarantee assumes exchangeability; Sections IV-E and IV-I measure the consequences (a five-point class-conditional gap and an eight-to-ten-point coverage loss under strain shift), and robust coverage under drift requires adaptive extensions left to future work [25]. Likewise, the temporal shortcut in telemetry typing is measured in Section IV-I rather than assumed away, though other undetected shortcuts may remain in both corpora. Third, we do not evaluate an adaptive adversary that perturbs Volatility statistics to evade the ensemble; such stress testing [3] is the most important robustness experiment we have not run. Fourth, the hierarchical decomposition's gain, though statistically significant here, is small (0.22 points); corpora with rarer families are needed to test whether it grows under imbalance. Fifth, with five seeds the paired t-tests have limited power, so non-significant differences should not be read as evidence of equivalence; accordingly we treat these small-sample tests as supporting rather than definitive evidence, and the full per-seed scores with confidence intervals and effect sizes are available from the authors on request. Finally, impurity importances are global and can be biased toward high-cardinality features; the cross-method stability check of Section IV-G mitigates but does not eliminate this concern, and richer stability metrics of the kind proposed by Hermosilla et al. [16] are natural next steps, as is an examiner-facing study of how conformal sets are understood by legal professionals [18] and how LLM-based assistants such as ForensicLLM [19]–[21] might consume them when drafting reports.

VI. CONCLUSIONS

This paper presented ForensiQ, an explainable hierarchical machine learning framework with conformal uncertainty guarantees for memory-forensic malware triage and IoT telemetry attack attribution, evaluated end to end on the public CIC-MalMem-2022 and TON_IoT corpora with five fixed seeds on single-core commodity hardware. The experiments, all executed by the authors, yield 99.99% binary triage accuracy and 87.4% four-class family attribution on CIC-MalMem-2022, empirical conformal coverage within 0.25 percentage points of the 90%, 95%, and 99% nominal targets with average set sizes of 1.08 to 1.58 labels, and 85.9% average binary detection across seven TON_IoT devices, alongside an honestly reported negative result on the weak identifiability of attack types from single-sensor telemetry. Three robustness studies convert the usual hypothetical caveats into measurements: unseen malware strains remain detected at 99.8% while their family attribution drops ten points, conformal coverage loses eight to ten points under that strain shift, and timestamp-derived features are shown to contribute six to eighteen points of telemetry typing accuracy. The central claim is methodological: calibrated, testable error rates and artifact-grounded explanations can be added to strong forensic classifiers at negligible cost, aligning automated triage with the evidentiary standards it must satisfy. Future work will address adversarial evasion, drift-adaptive conformal extensions, and practitioner studies.

DECLARATIONS

Authors' contributions. All authors contributed to the conceptualization, methodology, software, validation, analysis, and writing of this manuscript, and approved the final version. *Competing interests.* The authors declare no known competing

financial interests or personal relationships that could have influenced this work. *Funding.* Funding information accompanies the final, non-anonymized version. *Data and code availability.* CIC-MalMem-2022 is publicly available from the Canadian Institute for Cybersecurity and the TON_IoT telemetry datasets from UNSW Canberra. The complete ForensiQ implementation, with the exact seeds and hyperparameters needed to regenerate every table and figure, will be released publicly upon acceptance; an anonymized artifact archive is available to reviewers during review. *Generative AI declaration.* During the preparation of this work, the authors used AI-assisted tools in drafting the manuscript and in developing and executing the experimental pipeline. The authors reviewed, verified, and re-executed the experiments, edited the manuscript as needed, and take full responsibility for the content of the publication.

REFERENCES

[1] A. Case and G. G. Richard III, "Memory forensics: The path forward," *Digital Investigation*, vol. 20, pp. 23–33, 2017.

[2] S. L. Garfinkel, "Digital forensics research: The next 10 years," *Digital Investigation*, vol. 7, pp. S64–S73, 2010.

[3] 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," in *Proc. USENIX Security*, 2022, pp. 3971–3988.

[4] T. Carrier, P. Victor, A. Tekeoglu, and A. H. Lashkari, "Detecting obfuscated malware using memory feature engineering," in *Proc. 8th Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2022, pp. 177–188.

[5] 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, pp. 165130–165150, 2020.

[6] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustain. Cities Soc.*, vol. 72, p. 102994, 2021.

[7] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 485–496, 2022.

[8] M. Dener, G. Ok, and A. Orman, "Malware detection using memory analysis data in big data environment," *Appl. Sci.*, vol. 12, no. 17, p. 8604, 2022.

[9] A. Mezina and R. Burget, "Obfuscated malware detection using dilated convolutional network," in *Proc. ICUMT*, 2022, pp. 110–115.

[10] A. Öztürk and S. Hızal, "Detection and analysis of malicious software using machine learning models," *Sakarya Univ. J. Comput. Inf. Sci.*, vol. 7, no. 3, 2024.

[11] S. Nassar, "Malware detection through memory analysis," arXiv:2602.02184, 2026.

[12] Z. Khalid, F. Iqbal, and B. C. M. Fung, "Towards a unified XAI-based framework for digital forensic investigations," *Forensic Sci. Int.: Digit. Investig.*, vol. 50, p. 301806, 2024.

[13] Z. Khalid, F. Iqbal, and M. Saqib, "Bridging knowledge gaps in digital forensics using unsupervised explainable AI," *Forensic Sci. Int.: Digit. Investig.*, vol. 53, p. 301924, 2025.

[14] A. A. Solanke, "Explainable digital forensics AI: Towards mitigating distrust in AI-based digital forensics analysis using interpretable models," *Forensic Science International: Digital Investigation*, vol. 42, p. 301403, 2022.

[15] S. W. Hall, A. Sakzad, and K.-K. R. Choo, "Explainable artificial intelligence for digital forensics," *WIREs Forensic Science*, vol. 4, p. e1434, 2021.

[16] P. Hermosilla, S. Berríos, and H. Allende-Cid, "Explainable AI for forensic analysis: A comparative study of SHAP and LIME in intrusion detection models," *Appl. Sci.*, vol. 15, no. 13, p. 7329, 2025.

[17] M. Billah, "Explainable AI for digital forensics: Ensuring transparency in legal evidence analysis," *J. Forensic Sci. Res.*, vol. 9, no. 2, pp. 109–116, 2025.

[18] I. Ismail and K. A. Z. Ariffin, "The admissibility of digital evidence from open-source forensic tools: Development of a framework for legal acceptance," *PLOS ONE*, vol. 20, no. 9, p. e0331683, 2025.

[19] M. Scanlon, F. Breitinger, C. Hargreaves, J.-N. Hilgert, and J. Sheppard, "ChatGPT for digital forensic investigation: The good, the bad, and the unknown," *Forensic Science International: Digital Investigation*, vol. 46, p. 301609, 2023.

[20] G. Michelet and F. Breitinger, "ChatGPT, Llama, can you write my report? An experiment on assisted digital forensics reports written using (local) large language models," *Forensic Sci. Int.: Digit. Investig.*, vol. 48, p. 301683, 2024.

[21] B. Sharma, J. Ghawaly, K. McCleary, A. M. Webb, et al., "ForensicLLM: A local large language model for digital forensics," *Forensic Sci. Int.: Digit. Investig.*, vol. 52, p. 301872, 2025.

[22] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. New York, NY, USA: Springer, 2005.

[23] G. Shafer and V. Vovk, "A tutorial on conformal prediction," *J. Mach. Learn. Res.*, vol. 9, pp. 371–421, 2008.

[24] A. N. Angelopoulos and S. Bates, "Conformal prediction: A gentle introduction," *Found. Trends Mach. Learn.*, vol. 16, no. 4, pp. 494–591, 2023.

[25] A. N. Angelopoulos, S. Bates, E. J. Candès, M. I. Jordan, and L. Lei, "Learn then test: Calibrating predictive algorithms to achieve risk control," *Ann. Appl. Stat.*, vol. 19, no. 2, pp. 1641–1662, 2025.

[26] I. Garg and S. Majumder, "On the integration of cross-conformal prediction, ensembles, and sampling for uncertainty quantification in one-class anomaly detection," in *Proc. COPA*, PMLR, vol. 266, pp. 687–705, 2025.

[27] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. NeurIPS 30*, 2017, pp. 4765–4774.

[28] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[29] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[30] I. Tareq, B. M. Elbagoury, S. El-Regaily, and E.-S. M. El-Horbaty, "Analysis of ToN-IoT, UNSW-NB15, and Edge-IIoT datasets using deep learning in cybersecurity for IoT," *Appl. Sci.*, vol. 12, no. 19, p. 9572, 2022.

[31] N. Dharini, V. S. Janani, and J. Katiravan, "Efficient detection of intrusions in TON-IoT dataset using hybrid feature selection approach," *Sci. Rep.*, vol. 16, p. 7763, 2026.

[32] R. Ji, A. Selwal, N. Kumar, and D. Padha, "Cascading bagging and boosting ensemble methods for intrusion detection in cyber-physical systems," *Security and Privacy*, vol. 8, no. 1, p. e497, 2025.

[33] *Daubert v. Merrell Dow Pharmaceuticals, Inc.*, 509 U.S. 579, Supreme Court of the United States, 1993.