PT Jurnal Cendekia Indonesia
Journal of Information Technology and Informatics Engineering
Homepage: https://journal.jci.co.id/jitie
Vol. 02 No. 04 (2026) Page: 104-116
E-ISSN: 3123-7827

TinyLLM-Assisted Real-Time MQTT Intrusion Detection with Conformal Alerting, Cross-Attack Generalization, and Evidence-Based Forensic Narratives

Logan Smith[1], Diana Qi[2], Riley Scott[3], Alan Zheng[4]

[1]Department of Computer Science, Clemson University, Clemson, SC, USA
[2]Department of Computer Science, Temple University, Philadelphia, PA, USA
[3]Department of Computer Science, Auburn University, Auburn, AL, USA
[4]Department of Computer Science, Brandeis University, Waltham, MA, USA

Email: [1*]l.smith_clemson@gmail.com

Abstract

MQTT intrusion detectors are often evaluated with random row partitions even when packet order and capture identifiers make those partitions optimistic. This study evaluates a leakage-aware, real-time pipeline on 222,567 labelled packets from four MQTTEEB-D acquisition loops. Nine packet fields were combined with strictly past context over 8- and 32-packet windows, while absolute time and message-order proxies were excluded. Model selection used an earlier validation interval; a separate calibration interval supported temperature scaling and split-conformal prediction sets and alert thresholds; the last 33,404 packets formed the test interval. The selected Causal HGB achieved macro-F1 0.761 (95% block-bootstrap interval 0.719–0.788) and attack-versus-legitimate AUROC 0.851. At a calibration budget of 25 false alerts per day, observed test recall was 4.3%; raw packet-level false alerts were 7.8 per day and 5-second-collapsed false incidents were 3.9 per day, normalized over the 6.1-hour test interval. Loop holdout, leave-one-attack-out, imbalance, field-corruption, drift, and leakage analyses exposed the gap between within-interval discrimination and deployment generalization. TinyLLM, defined here as a Tiny Language Layer for MQTT, paired a protocol-token SGD classifier with a deterministic evidence renderer. Across 600 audited alerts, exact slot fidelity was 100.0% with unsupported numeric-claim rate 0.0%. The resulting design joins measured detection, explicit uncertainty, and packet-verifiable forensic prose without free-form report generation.reorder itself correctly. If you want to try deleting this template content, please make sure to back it up first.

Keywords: MQTT, Intrusion Detection, Internet Of Things, Conformal Prediction, Open-Set Detection, Temporal Drift, Forensic Narratives, Tinyllm

Abstrak

Detektor intrusi MQTT sering kali dievaluasi menggunakan partisi baris acak, meskipun urutan paket dan pengenal penangkapan (capture identifiers) membuat partisi tersebut menjadi optimistik. Studi ini mengevaluasi alur kerja (pipeline) waktu nyata yang mempertimbangkan risiko kebocoran data (leakage-aware) menggunakan 222.567 paket berlabel dari empat siklus akuisisi MQTTEEB-D. Sembilan kolom paket dikombinasikan dengan konteks masa lalu (strictly past context) dalam jendela 8 dan 32 paket, sementara waktu absolut dan proksi urutan pesan tidak disertakan. Pemilihan model menggunakan interval validasi awal; interval kalibrasi terpisah mendukung temperature scaling serta set prediksi split-conformal dan ambang batas peringatan; sedangkan 33.404 paket terakhir membentuk interval pengujian. Model Causal HGB yang terpilih mencapai skor macro-F1 sebesar 0,761 (interval block-bootstrap 95%: 0,719–0,788) dan AUROC 0,851 untuk klasifikasi serangan versus lalu lintas sah. Dengan anggaran kalibrasi 25 peringatan palsu per hari, recall pengujian yang teramati adalah 4,3%; tingkat peringatan palsu mentah di tingkat paket adalah 7,8 per hari dan insiden palsu yang digabungkan dalam interval 5 detik adalah 3,9 per hari (dinormalisasi terhadap interval pengujian 6,1 jam). Analisis loop holdout, leave-one-attack-out, ketidakseimbangan data, kerusakan kolom (field-corruption), pergeseran data (drift), dan kebocoran data mengungkap kesenjangan antara kemampuan diskriminasi dalam interval dan generalisasi saat penerapan nyata. TinyLLM—yang didefinisikan di sini sebagai Tiny Language Layer untuk MQTT—memadukan pengklasifikasi SGD berbasis token protokol dengan penyaji bukti (evidence renderer) deterministik. Dari 600 peringatan yang diaudit, fidelitas slot yang tepat mencapai 100,0% dengan tingkat klaim numerik yang tidak didukung sebesar 0,0%. Desain yang dihasilkan menggabungkan deteksi terukur, ketidakpastian eksplisit, dan narasi forensik yang dapat diverifikasi melalui paket, tanpa memerlukan pembuatan laporan format bebas.

Received: 01-08-2026 | Revised: 09-08-2026 | Published Online: 10-08-2026

***Kata Kunci**: MQTT, Deteksi Intrusi, Internet Of Things, Prediksi Konformal*, Deteksi Open-Set, Pergeseran Temporal, Narasi Forensik, Tinyllm

A. INTRODUCTION

MQTT provides a small publish–subscribe surface for sensors, gateways, and constrained devices, but its operational simplicity also concentrates security decisions in brokers, topics, sessions, and compact control fields. The protocol standard defines stateful exchanges whose meaning depends on message sequence rather than on isolated bytes alone [3]. Flooding, repeated connection attempts, malformed control values, and deliberately slow sessions can therefore interfere with availability through different temporal signatures. A detector for this setting must classify packets quickly, remain useful as capture conditions change, and explain why an alert was emitted in terms an analyst can verify against the retained event.

MQTTEEB-D was released as a real MQTT cybersecurity corpus with four acquisition loops and five attack families alongside legitimate traffic [1]. Its companion data article documents the test environment and capture process [2]. The corpus is useful because it preserves acquisition order and includes DoS, SlowITe, malformed data, brute-force activity, and publish flooding in the same deployment. Those same properties make evaluation design consequential: the loops are not interchangeable independent samples, class frequencies are strongly unequal, and repeated protocol states can allow a random packet split to place nearly adjacent events on both sides of the train–test boundary.

Earlier MQTT corpora established complementary evaluation settings. MQTTset emphasized machine-learning traces for MQTT attacks [4], MQTT-IoT-IDS2020 paired flow evidence with an MQTT case study [5], and a dedicated DoS/DDoS-MQTT-IoT corpus focused on availability attacks [8]. Model studies have reported deep MQTT detectors [6], one-class classifiers [7], and transformer networks [9]. Feature engineering can materially alter MQTT classification [10], while XMID-MQTT shows the value of an explanation layer for analyst use [11]. These works motivate broad baselines, but results obtained within one random partition do not establish performance on a later loop or on a withheld attack mechanism.

Recent IoT-security work has begun to combine compact classifiers with language-assisted interpretation rather than treating detection and explanation as separate products. A TinyLLM-assisted RT-IoT2022 study paired conventional traffic classifiers with lightweight attack explanations [31], while language-guided feature selection on CICIDS2017 examined whether a semantic reading of feature names could reduce the representation without abandoning measurable detection [32]. Deep-autoencoder traffic modeling [33] and interpretable CloudTrail sequence labeling [34] broaden this pattern from flows to reconstruction and event chains. Together with MQTT-specific deep and feature-engineering studies [6], [10], these works motivate a detector whose language component is constrained by measured packet evidence.

At the response layer, predictive DDoS mitigation [35], multi-source microservice root-cause attribution [36], and system-call window localization with forensic narratives [37] show that a security decision is increasingly expected to be tied to an actionable incident unit rather than a row-level label. This operational view reinforces the closed-world warning [24] and unknown-attack comparison [23]: packet, sequence, and service-telemetry models answer different questions, so their outputs should not be collapsed into a single generalization claim.

The model comparison in this study spans sparse linear classification, random forests, extremely randomized trees, histogram gradient boosting, a compact multilayer perceptron, and a protocol-token text classifier. Gradient-boosted trees remain an efficient reference family in tabular learning [12], whereas autoencoder bottlenecks provide a classical route to low-dimensional representation learning [13]. The objective is not to crown one architecture in a closed contest. It is to measure how representation, order, calibration, and attack novelty change the conclusions drawn from the same captured evidence.

Operations-oriented retrieval systems add a second requirement: retrieved evidence must remain inspectable. Evidence-grounded question answering over private DevOps documents [38], bilingual cloud-incident triage [39], and word-level provenance analysis [40] each emphasize source anchoring rather than unsupported completion.

That principle also appears in CI failure diagnosis [41], retrieved normal prototypes for OpenStack logs [42], and deterministic HDFS failure narratives [43]. Self-supervised log anomaly modeling [44] and evidence-constrained incident visualization cards [45] further separate anomaly scoring from the evidence that an analyst sees.

Recent pipelines also join alert control and remediation safeguards. Conformal log alerting with evidence-grounded tickets [46], retrieval-augmented runbooks with command-safety checks [47], and cost-aware routing across parsing, diagnosis, and summarization tasks [48] treat operational cost, abstention, and action safety as first-class outputs. These studies complement the forensic traceability objective in [28]-[30], but they do not remove the need to audit packet-level support.

Three evaluation concerns guide the design. First, calibrated probabilities require separate assessment because high classification accuracy does not imply reliable confidence [14]. Second, macro-averaged and precision–recall measures are essential under severe class imbalance [15], [16]. Third, a deployment claim must survive temporal and semantic separation. Cross-dataset research has demonstrated substantial generalization loss [22], unknown-attack comparisons show different behavior from

supervised recognition [23], and the classic closed-world critique remains directly relevant to intrusion detection [24].

Methodological parallels outside network security reinforce the same evaluation discipline. Calibrated rejection in credit-risk tiering [49], cost-sensitive and one-class learning under extreme imbalance [50], multi-signal drift monitoring with rollback [51], and a model-risk workflow that combines calibration, distribution-free uncertainty, and SHAP [52] all separate discrimination from decision policy. Confidence-calibrated late fusion [53] and domain transfer with approximately shared features [54] likewise motivate explicit stress tests rather than assuming that one validation score transfers to a later acquisition loop. These studies are used as methodological precedents, not as evidence of MQTT performance.

Conformal prediction supplies a finite-sample mechanism for turning calibration scores into prediction sets or thresholds [17]. Standard marginal guarantees are distinct from conditional guarantees [18], and adaptation under distribution shift requires additional care [19]. Recent theory extends conformal reasoning beyond exact exchangeability [20] and toward risk-controlling sets [21]. Here, conformal methods are used as transparent calibration procedures: nominal levels and observed held-out behavior are both reported, with no assumption that one calibration interval removes loop drift.

For the language layer, evidence-based self-verification under adversarial prompts [55], abstention calibration for unanswerable retrieval [56], and behavior-level refusal with utility preservation [57] suggest that response policies should be evaluated separately from detector accuracy. Deterministic provenance explanations [58], continual red-teaming under nonstationary jailbreaks [59], and retrieval-quality prediction [60] sharpen the distinction between grounded evidence and fluent but unsupported prose.

Alert explanation is treated as an evidence-preservation problem. Feature attribution methods such as SHAP have made local explanations familiar in machine learning [25], while small language-model research shows that compact language components can be engineered for constrained devices [26], [27]. Digital-forensics guidance, however, places traceability and verification ahead of fluency [28]. Experiments with local language models for forensic reports [29] and retrieval-assisted incident timelines [30] further suggest that prose is most useful when every claim remains anchored to accessible evidence.

Human-facing presentation and deployment add a final layer. Prompt-injection risk cards [61], scientific-search evidence cards [62], and infrastructure risk-brief cards [63] organize confidence, provenance, and action cues for review; human-aligned design critique [64] illustrates why interface quality should be audited rather than presumed. Hybrid edge-cloud deployment [65], trajectory reliability prediction for tool-using agents [66], and confidence-gated, time-aware memory updating [67] also show that low-latency components need explicit routing, failure monitoring, and stale-evidence control. These adjacent systems are design precedents for the bounded Tiny Language Layer, not substitutes for MQTT-specific validation.

This study contributes an end-to-end empirical account with four linked parts: a causal feature policy that removes absolute time and message-order proxies; a validation–calibration–test chronology; cross-loop, cross-attack, imbalance, corruption, and drift tests; and a deterministic forensic language layer. The term TinyLLM means Tiny Language Layer for MQTT. It combines the measured protocol-token SGD classifier with a closed-slot renderer and is a task-specific language interface, not a pretrained language model. This definition keeps the language claim commensurate with the executed system while allowing alert text, fidelity, size, and latency to be audited directly.

B. METHODS

Dataset Audit And Temporal Partitioning

The analysis read the four raw loop CSV files identified by the MQTTEEB-D release [1], [2]. All 222,813 packet rows participated in causal state updates. The 246 rows without a valid class label were excluded only as prediction targets, leaving 222,567 labelled observations. This choice retained packet timing for later rows while avoiding an inferred class. The labelled set covered 20.27 hours from 2024-09-09T03:17:53.839303017+00:00 through 2024-09-09T23:34:12.429866076+00:00. Attacks outnumbered legitimate packets by 11.68:1, so accuracy was never used as the sole selection statistic.

The processed variants were audited before model fitting. All_Processed and Missing_Values_outliers retained 222,813 rows but assigned all 246 missing targets to malformed. The SMOTE, normalized, and standardized variants each contained 634,644 balanced rows—105,774 per class—so the latter two were not measurement-only transforms. To avoid label imputation, resampling leakage, and duplicate analytical populations, fitting and scoring used only the four raw loops.

Table 1 Raw Acquisition-Loop Inventory

Capture	Raw rows	Labelled	Missing label	Span (h)	UTC range
Loop 1	137,319	137,147	172	8.43	03:17:53–11:43:39
Loop 2	42,470	42,418	52	2.60	11:43:39–14:19:43
Loop 3	21,501	21,485	16	1.43	16:40:13–18:06:17
Loop 4	21,523	21,517	6	2.07	21:29:57–23:34:12
All loops	222,813	222,567	246	20.27	03:17:53–23:34:12

Table 2 Class Counts In Each Acquisition Loop

Capture	Legitimate	Bruteforce	DoS	Flood	Malformed	SlowITe
Loop 1	11,158	16,124	26,340	6,417	64,343	12,765
Loop 2	3,208	5,532	7,717	2,115	20,183	3,663

Capture	Legitimate	Bruteforce	DoS	Flood	Malformed	SlowITe
Loop 3	2,815	2,028	3,857	1,155	9,829	1,801
Loop 4	376	3,693	3,888	526	11,173	1,861
All	17,557	27,377	41,802	10,213	105,528	20,090

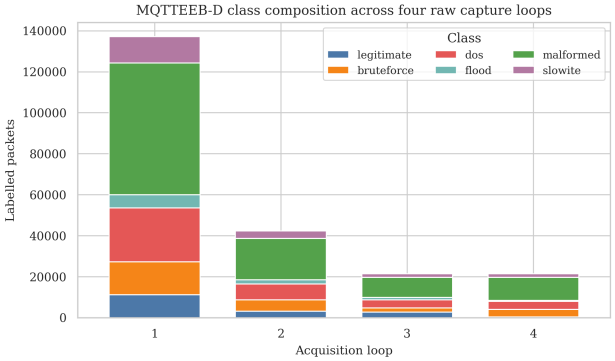


Figure 1 Class Composition Across The Four Raw Capture Loops.

The partition followed one global stable chronological ordering, with equal timestamps kept together. The first 60% formed training data, the next 10% was used only for model selection, the next 15% was reserved for temperature and conformal calibration, and the final 15% was evaluated once. The calibration interval was not included when fitting classifiers. This separation prevents threshold choice from consuming the final interval and keeps the reported test metrics tied to later captured traffic.

Table 3 Fixed Chronological Development, Calibration, And Test Intervals

Interval	Rows	Share	Start UTC	End UTC	Loops
Train	133,517	60.0%	03:17:53	11:18:57	1
Validation	22,252	10.0%	11:18:57	12:44:27	1,2
Calibration	33,394	15.0%	12:44:27	17:27:09	2,3
Test	33,404	15.0%	17:27:09	23:34:12	3,4

Leakage-Aware Causal Representation

The packet representation contained 9 contemporaneous protocol fields: tcp_flags, tcp_len, mqtt_conack_flags, mqtt_conflag_cleansess, mqtt_conflags, mqtt_dupflag, mqtt_hdrflags, mqtt_kalive, mqtt_qos. Absolute timestamp, the identically valued tcp_time_delta field, and the absolute message sequence were excluded as acquisition-order proxies. Timestamp was used only to derive relative inter-arrival time within a loop; message identity was reduced to presence and change indicators. For windows of 8 and 32 packets, 21 causal context variables summarized strictly earlier packets: inter-arrival mean and dispersion, length mean and dispersion, packet and byte rates, nonzero QoS, duplicate flag, message presence, and message change. The current row never entered its own rolling summary.

Table 4 Feature Policy And Availability

Role	Count	Fields or construction	Use
Current packet	9	tcp_flags, tcp_len, mqtt_conack_flags, mqtt_conflag_cleansess, mqtt_conflags, mqtt_dupflag, mqtt_hdrflags, mqtt_kalive, mqtt_qos	Primary static and causal models
Past context	21	Past-8/32 timing, length, rate, QoS, duplicate, and message indicators	Causal temporal models
Excluded proxies	3	timestamp, tcp_time_delta, mqtt_msg	Sensitivity diagnostic only

Models, Selection, And Uncertainty

Eight supervised pipelines were fitted with median imputation where required. Static baselines used nine packet fields; causal tree ensembles used the same fields plus past-8/32 context. Logistic regression supplied a linear reference, random forest and ExtraTrees represented bagged trees, histogram gradient boosting represented the boosting family, and the 48–24-unit MLP represented a small neural detector under a fixed 35-epoch budget rather than convergence-based stopping. TinyLLM tokenized explicit field–value descriptions into a 4,096-feature sparse vocabulary and fitted a class-weighted SGD log-loss classifier. Validation macro-F1 selected one pipeline, with validation negative log-likelihood breaking a tie. Every pipeline was then refitted on training plus validation, leaving calibration untouched. This matched-study comparison also avoids attributing performance gains to language branding alone: lightweight IoT detection [31], semantic feature selection [32], and reconstruction-based traffic modeling [33] motivate representation diversity, but selection remains validation-only here.

Table 5 Validation-Only Model Selection

Model	Features	Macro-F1	Bal. acc.	Attack AUROC	NLL	Fit (s)
Causal HGB	30	0.844	0.857	0.919	0.369	8.49
Causal ExtraTrees	30	0.843	0.846	0.898	0.372	3.57
Random forest	9	0.570	0.622	0.819	0.976	2.51
Static HGB	9	0.553	0.613	0.821	0.954	3.35
Static ExtraTrees	9	0.537	0.497	0.785	1.294	1.31
Tiny MLP	9	0.512	0.562	0.784	1.213	6.52
TinyLLM token SGD	9	0.476	0.495	0.717	1.159	2.92
Logistic regression	9	0.381	0.371	0.607	1.486	5.61

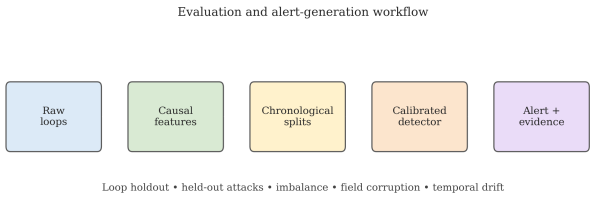


Figure 2 Evaluation And Alert-Generation Workflow.

Temperature scaling optimized one scalar on the calibration interval and was evaluated on the test interval, following the distinction between discrimination and confidence calibration [14]. Test uncertainty was measured with 95% moving-block bootstrap intervals using 256-packet blocks. Accuracy, balanced accuracy, macro-precision, macro-recall, macro-F1, weighted F1, Matthews correlation, negative log-likelihood, multiclass Brier score, 15-bin expected calibration error, one-vs-rest macro-AUROC, and attack-versus-legitimate AUROC and AUPRC were computed. Macro-F1 was primary because the class distribution was unequal [15], and AUPRC was retained because it is sensitive to rare positive outcomes [16]. Reporting therefore separates discrimination, calibration, and operating policy, consistent with calibrated rejection [49], drift-aware rollback [51], and distribution-free model-risk workflows [52].

Conformal Alerting And Generalization Tests

Multiclass split-conformal scores were 1 minus the scaled probability assigned to the true class. Quantiles at $\alpha = 0.05$, 0.10, and 0.20 produced test prediction sets [17]. A second calibration used only the 2,525 legitimate calibration packets. Attack score was one minus legitimate probability. The observed benign packet rate over calibration wall time converted daily false-alert budgets of 10, 25, and 50 into finite-sample upper quantiles. The resulting thresholds were fixed before application to the final interval. Raw false-positive packets and incidents collapsed across gaps shorter than 5 seconds were both normalized by test wall time. Because capture loops alter both class mix and protocol states, empirical coverage and false-alert rates are reported alongside nominal values rather than read as conditional guarantees [18]–[21]. The benign-only alert budget is especially close in spirit to conformal log alert control [46], although its unit of evidence here is a packet and collapsed incident rather than a log window.

Attack-episode analysis required consecutive raw rows with the same attack label and inter-packet gaps below 5 seconds. A label transition, intervening row, or gap of at least 5 seconds began a new episode. Detection delay was measured from episode start to its first alert; a missed episode had no delay value.

Generalization was examined in five ways. A loop holdout trained on loops 1–3 and tested on loop 4. Leave-one-attack-out trials removed one attack family from supervised training and tested that attack with loop-4 legitimate traffic; benign-only Isolation Forest and PCA reconstruction supplied open-set comparators [23]. Imbalance trials retained 100%, 50%, 25%, or 10% of complete flood and SlowITe episodes in five seeded samples. Each sample fitted the same static-field HGB with 55 boosting iterations, 15 leaf nodes, balanced class weights, and an unchanged natural test set. Field-corruption trials replaced 0%–30% of current packet cells with values sampled from empirical training marginals. Four consecutive test windows measured observed drift. A fixed ExtraTrees sensitivity analysis compared static fields, causal context, and absolute time/order proxies so representation changed while the estimator family remained constant. These tests directly operationalize the cross-domain and closed-world concerns in [22], [24]. The distinction between reusable and environment-specific structure is aligned with shared-feature domain transfer [54], while trajectory-level reliability analysis [66] motivates reporting failure modes rather than only aggregate success.

Evidence-Based Forensic Language Layer

For each selected alert, the detector stored event time, loop and row identity, predicted attack, attack score, threshold, and two packet features with the largest robust deviation from the legitimate development median. A closed-slot renderer inserted only those stored values into a fixed sentence. The resulting Tiny Language Layer for MQTT was audited on stratified eligible alerts for exact class, score, threshold, and evidence slots; unsupported numeric and entity claims; rendering latency; and serialized size. This preserves the verification priority of incident-response forensics [28] while avoiding the evidence drift that can occur in unconstrained report writing [29], [30]. Permutation importance was computed separately as the macro-F1 decrease after 3 feature shuffles on 3,600 class-stratified test packets; it is a global sensitivity summary rather than a claim of causation. Its evidence binding follows source-anchored AIOps and deterministic failure narratives [38], [40], [43]. The audit fields also reflect incident-card and self-verification designs [45], [55], [61], [62]..

C. RESULTS AND DISCUSSION

Chronological Detection Performance

Validation selected Causal HGB with macro-F1 0.844. After refitting on training plus validation, its raw chronological-test macro-F1 was 0.761, balanced accuracy 0.797, and attack AUROC 0.851. Temperature scaling left the argmax classification unchanged, so scaled macro-F1 remained 0.761; it changed probability quality, including ECE from 0.068 to 0.070. The block-bootstrap interval for macro-F1 was 0.719–0.788.

Table 6 Chronological Test Performance And Operating Cost

Model	Acc.	Bal. acc.	Macro-F1	AUROC	AUPRC	ECE	µs/pkt	Size
Causal Extra Trees	0.832	0.807	0.775	0.896	0.990	0.057	5.3	147.76 MB
Causal HGB	0.821	0.797	0.761	0.851	0.985	0.068	9.3	1.21 MB
Tiny MLP	0.666	0.569	0.532	0.862	0.985	0.239	0.3	59.8 kB
Static Extra Trees	0.647	0.525	0.530	0.708	0.965	0.307	3.6	19.22 MB
Random forest	0.530	0.587	0.497	0.729	0.965	0.181	3.1	8.29 MB
Static HGB	0.521	0.581	0.495	0.839	0.983	0.162	7.3	673.5 kB
TinyLLM token SGD	0.626	0.463	0.435	0.767	0.969	0.244	11.6	8.0 kB
Logistic regression	0.547	0.402	0.403	0.663	0.961	0.284	0.4	1.9 kB

Validation-selected model. AUROC and AUPRC treat every attack label as positive; ECE uses 15 bins.

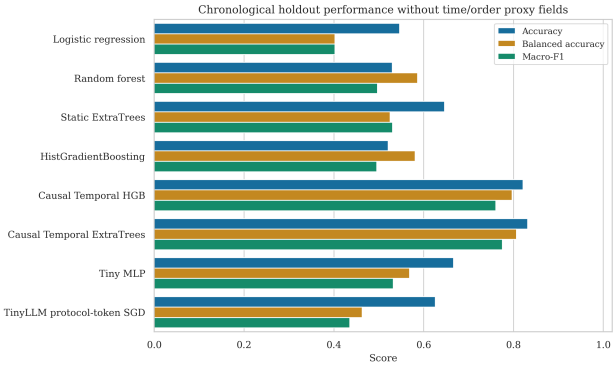


Figure 3 Model performance on the final chronological test interval.

Table 7 Per-Class Test Performance Of The Selected Detector

Class	Precision	Recall	F1	Support
Legitimate	0.499	0.607	0.548	2,403
Brute force	0.876	0.805	0.839	5,721
DoS	0.595	0.843	0.698	3,888
Flood	0.994	0.951	0.972	1,681
Malformed	0.979	0.848	0.909	17,850
SlowITe	0.510	0.726	0.599	1,861

Classwise results explain why aggregate accuracy is insufficient. Flood had the highest F1 (0.972), whereas Legitimate had the lowest (0.548). The confusion matrix in Fig. 4 shows the associated error flows rather than hiding them in a single attack-versus-benign score. This pattern is consistent with shared packet signatures and changing attack phases: causal windows improve context, but they do not make different attack mechanisms perfectly separable.

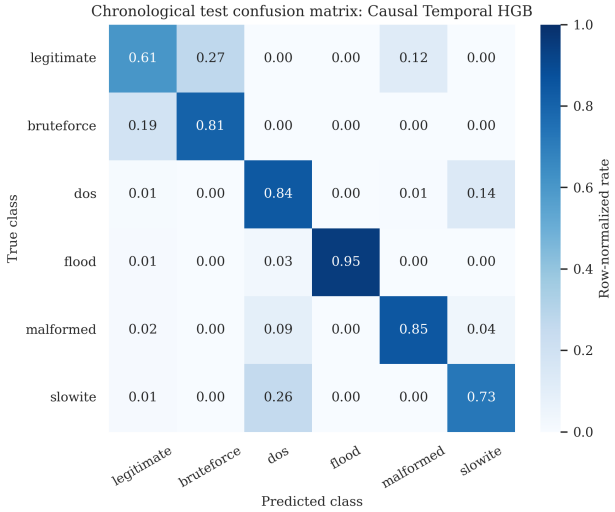


Figure 4 Confusion Matrix For The Validation-Selected Detector.

Calibration And Conformal Alert Budgets

Table 8 Probability Calibration And Multiclass Conformal Prediction Sets

Analysis	Parameter	Measure 1	Measure 2	Measure 3
Calibration: Raw	T=1.000	ECE 0.068	NLL 0.504	Brier 0.282
Calibration: Temperature-scaled	T=1.050	ECE 0.070	NLL 0.499	Brier 0.282
Prediction set: α=0.05	$\hat{q}$=0.838	Coverage 91.6%	Mean size 1.46	Singletons 56.6%
Prediction set: α=0.10	$\hat{q}$=0.731	Coverage 87.4%	Mean size 1.26	Singletons 74.3%
Prediction set: α=0.20	$\hat{q}$=0.522	Coverage 81.7%	Mean size 0.98	Singletons 96.3%

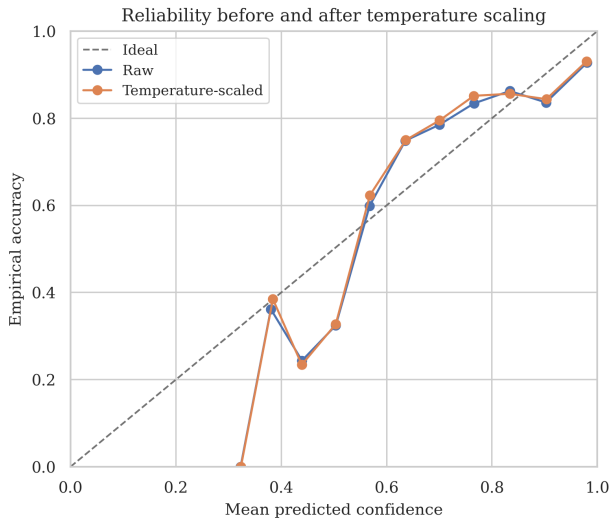


Figure 5 Reliability Before And After Temperature Scaling.

Table 9 Daily-Budget Conformal Thresholds And Observed Test Behavior

Target/day	Threshold	Raw FP/day	Incident FP/day	Recall	Episode miss	Median delay (s)
10	0.9998	7.8	3.9	4.3 %	90.5 %	0.10
25	0.9998	7.8	3.9	4.3 %	90.5 %	0.10
50	0.9997	7.8	3.9	4.6 %	89.2 %	0.11

Table 10 Attack-Episode Detection At The 25-False-Alerts-Per-Day Calibration Budget

Attack	Episodes	Missed	Miss rate	Median delay (s)	P95 delay (s)
Brute force	88	84	95.5%	1.56	3.17
DoS	60	60	100.0%	—	—
Flood	34	16	47.1%	0.02	3.52
Malformed	52	52	100.0%	—	—
SlowITe	71	64	90.1%	3.61	5.36

At α = 0.10, the multiclass set achieved 87.4% empirical coverage against a 90% nominal target, with mean size 1.26 and singleton rate 74.3%. For attack alerts, the 25-per-day calibration budget yielded 7.8 raw false-positive packets and 3.9 collapsed false incidents per normalized test day, with recall 4.3%. These daily values normalize counts observed during the 6.1-hour final interval; they are not estimates from repeated days. Across attack families at this operating point, DoS had an episode miss rate of 100.0%; no episode was detected, so first-alert delay was undefined. The difference between target and observation is the deployment-relevant result: calibration quantiles are finite-sample procedures, while later loop composition can depart from the calibration exchangeability assumption [19], [20]. The same separation of calibrated scores from downstream action appears in evidence-grounded log alerting [46]; cross-domain studies of risk tiering and fusion similarly treat rejection or calibration as operating layers rather than substitutes for discrimination [49], [53].

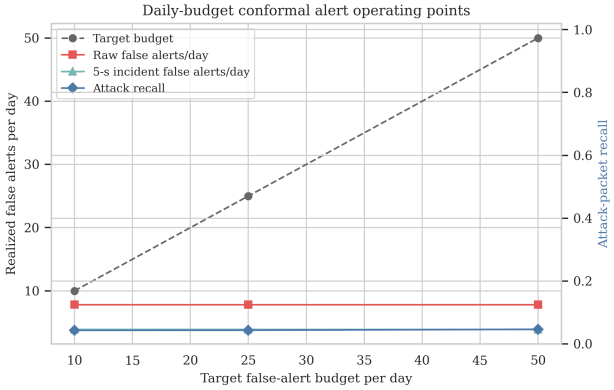


Figure 6 Daily-Budget Conformal Alert Operating Points.

Table 11 Loop Holdout: Train On Loops 1–3 And Test On Loop 4

Model	Accuracy	Bal. acc.	Macro-F1	Attack AUROC	Attack AUPRC
Causal ExtraTrees	0.815	0.848	0.745	0.997	1.000
Causal HGB	0.804	0.841	0.737	0.990	1.000
Random forest	0.565	0.753	0.653	0.989	1.000
Static ExtraTrees	0.696	0.720	0.614	0.980	1.000
Static HGB	0.555	0.741	0.614	0.990	1.000
Tiny MLP	0.623	0.684	0.543	0.990	1.000
TinyLLM token SGD	0.637	0.626	0.522	0.986	1.000
Logistic regression	0.551	0.569	0.462	0.901	0.998

When all of loops 1–3 were available for fitting and loop 4 was held out, Causal ExtraTrees ranked first by macro-F1 at 0.745; its attack AUROC was 0.997. The ranking need not match the global chronological test because the latter begins in loop 3 and continues through loop 4, whereas the loop experiment removes loop 4 completely from training. The comparison makes acquisition shift visible instead of averaging it away.

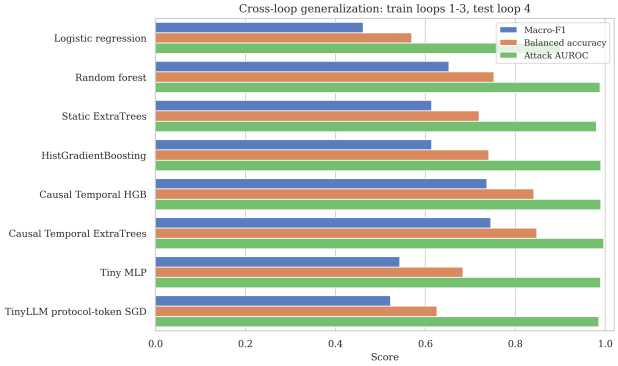


Figure 7 Generalization From Loops 1–3 To Loop 4.

Fig. 7.

Table 12 Leave-One-Attack-Out Performance Averaged Across Five Held Attacks

Method	Mean recall	Minimum recall	Mean benign FPR	Mean AUROC
Benign-only Isolation Forest	0.219	0.016	0.000	0.614
Benign-only PCA reconstruction	0.180	0.000	0.000	0.382
Seen-attack HGB	0.038	0.000	0.000	0.605

Across attacks omitted from supervised training, Benign-only Isolation Forest had the highest mean recall (0.219), but its mean loop-4 benign false-positive rate was 0.000. Minimum recall across the five omitted attacks was 0.016. The full bars in Fig. 8 show that an average conceals attack-specific failure. Benign-only reconstruction or isolation can recognize deviations without attack labels, whereas a supervised binary model can score an unseen mechanism as attack only if it resembles patterns learned from the remaining families. This is the practical distinction between

recognizing a known class and detecting an unknown attack [23], [24]. Sequence-localized host intrusion analysis [37] and CloudTrail attack-chain staging [34] illustrate why the incident unit and event order must be specified before comparing detection rates.

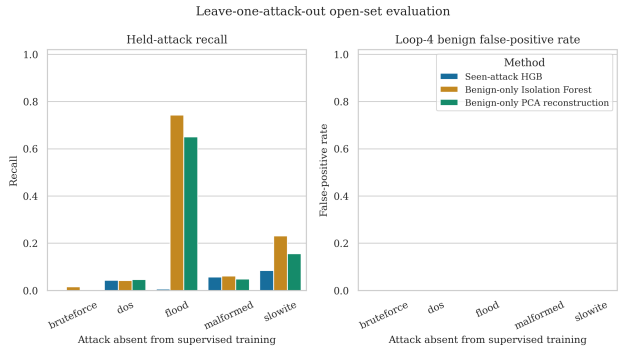


Figure 8 Detection Of Attacks Absent From Supervised Training.

Imbalance, Corruption, Drift, And Leakage

Table 13 Robustness Under Empirical Imbalance, Field Corruption, And Chronological Drift

Stress	Setting	Outcome 1	Outcome 2	Outcome 3
Episode retention	100%; n=5	Macro-F1 0.494±0.019	Flood R 0.909	SlowITe R 0.707
Episode retention	50%; n=5	Macro-F1 0.492±0.022	Flood R 0.889	SlowITe R 0.698
Episode retention	25%; n=5	Macro-F1 0.472±0.019	Flood R 0.733	SlowITe R 0.695
Episode retention	10%; n=5	Macro-F1 0.472±0.020	Flood R 0.749	SlowITe R 0.687
Field corruption	0%	Macro-F1 0.761	Bal. acc. 0.797	AUROC 0.851
Field corruption	5%	Macro-F1 0.749	Bal. acc. 0.788	AUROC 0.846
Field corruption	10%	Macro-F1 0.739	Bal. acc. 0.780	AUROC 0.842
Field corruption	20%	Macro-F1 0.721	Bal. acc. 0.766	AUROC 0.830
Field corruption	30%	Macro-F1 0.705	Bal. acc. 0.752	AUROC 0.818
Test drift	Window 1	Macro-F1 0.381	PSI 0.019	JS 0.137
Test drift	Window 2	Macro-F1 0.685	PSI 0.064	JS 0.209
Test drift	Window 3	Macro-F1 0.372	PSI 0.118	JS 0.171
Test drift	Window 4	Macro-F1 0.482	PSI 0.087	JS 0.156

At 30% field-cell corruption, macro-F1 changed from 0.761 to 0.705, and attack AUROC changed from 0.851 to 0.818. Across the four chronological windows, macro-F1 ranged from 0.372 in window 3 to 0.685 in window 2. These values connect performance change to the traffic actually observed, rather than to synthetic Gaussian noise or shuffled labels. Reducing complete rare-attack episodes from 100% to 10% changed mean macro-F1 from 0.494 to 0.472; mean flood/SlowITe recall changed from 0.808 to 0.718. Whole-episode sampling prevents fragments of the same attack episode from appearing in both retained and removed training subsets. The observed sensitivity is therefore interpreted alongside imbalance-aware [50], drift-monitoring [51], and model-risk [52] precedents rather than as a universal robustness guarantee.

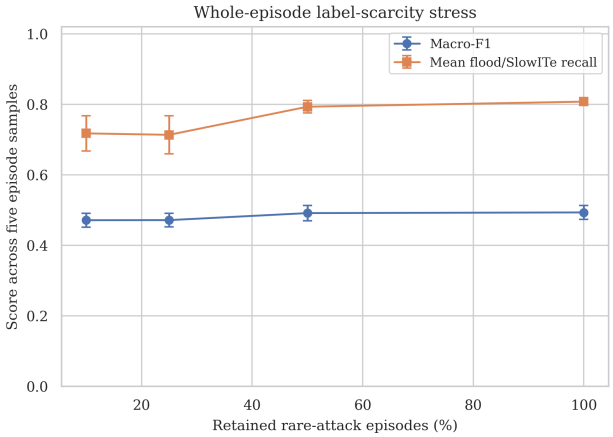


Figure 9 Whole-Episode Rare-Class Retention Stress Across Five Samples.

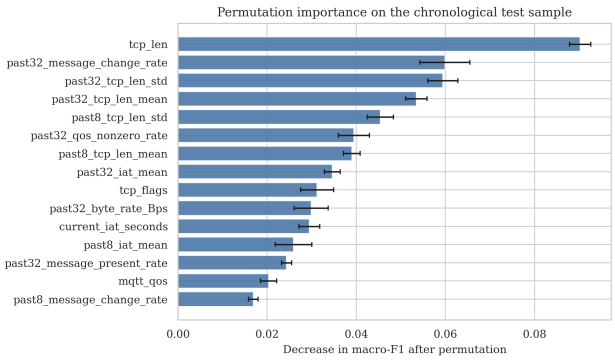
Table 14 Fixed-Extratrees Sensitivity To Causal Context And Acquisition Proxies

Feature set	Role	Count	Macro-F1	Bal. acc.	ECE
Static protocol fields (9)	packet-only ablation	9	0.530	0.525	0.307
Causal protocol + past 8/32 context	primary causal representation	30	0.775	0.807	0.057
All fields including time/order proxies	absolute proxy diagnostic only	12	0.317	0.382	0.412

Within the fixed ExtraTrees comparison, static packet fields produced macro-F1 0.530, while the causal representation produced 0.775. Allowing absolute time and order proxies produced 0.317. These ablation values are distinct from the selected Causal HGB headline because the estimator is deliberately held to ExtraTrees. The proxy result is intentionally not a headline model: even when a monotone capture field raises a score, it identifies where a packet occurred in this acquisition rather than a reusable protocol mechanism. Conversely, a lower proxy score would not vindicate random splitting. The central safeguard is semantic—only information available at the packet's arrival and summaries of earlier packets enter the deployed representation.

Evidence Narratives And Operational Interpretation

The five largest permutation macro-F1 decreases were tcp_len (0.090), past32_message_change_rate (0.060), past32_tcp_len_std (0.059), past32_tcp_len_mean (0.053), past8_tcp_len_std (0.045). Negative or near-zero decreases are interpreted as redundancy under this fitted model, not as proof that the underlying protocol field is irrelevant. Fig. 10 includes the top 15 features with repeat variability, making the ranking's uncertainty visible.



XS

Table 15 Tinyllm Language-Layer Fidelity, Latency, And Footprint

Component or audit	Measured result	Interpretation
Implementation	ProtocolTextSGD + deterministic closed-slot renderer	Sparse protocol-token classifier plus closed slots
Audited alerts	600	Stratified eligible test alerts
All slots exact	100.0%	Class, score, threshold, and two evidence fields
Unsupported numeric claims	0.0%	Claims not present in allowed slots
Unsupported entity claims	0.0%	Attack entities outside the predicted slot
Rendering latency	mean 2.63 µs; p95 2.76 µs	Closed-slot sentence rendering
Combined footprint	8.1 kB	Serialized classifier and renderer template

Table 16 Packet-Verifiable Evidence Records Emitted For Test Alerts

Event	UTC time	Class	Confidence	Evidence fields
L4-R20363	2024-09-09T23:20:00.267589091+00:00	Bruteforce	100.0%	past32_iat_std; past32_iat_mean
L4-R19735	2024-09-09T23:15:40.461132050+00:00	Bruteforce	100.0%	past32_iat_std; past32_iat_mean
L3-R19064	2024-09-09T17:41:59.757746935+00:00	Bruteforce	100.0%	past32_iat_std; past32_iat_mean

One emitted evidence narrative stated: “The detector assigned bruteforce with 100.0% confidence. past32_iat_std=3.87 was above the legitimate median (0.0107) by 395.32 robust-IQR units; past32_iat_mean=1.84 was above the legitimate median (0.01) by 82.58 robust-IQR units. The alert preserves the packet fields and model score for analyst verification.” Its event identifier and timestamp point back to the retained packet, while the two feature values and reference medians can be recomputed. Across 600 audited alerts, all required slots were exact in 100.0%; unsupported numeric and entity claim rates were 0.0% and 0.0%, respectively. The measured combined footprint was 8.1 kB, and p95 rendering latency was 2.76 µs. These results characterize a bounded reporting layer, not an autonomous forensic conclusion. This bounded form is closer to retrieved normal-prototype explanations [42], deterministic HDFS narratives [43], and evidence-grounded incident tickets [46] than to unconstrained report generation.

The findings have four operational implications. First, model ranking is conditional on when and where packets are tested; loop holdout and chronological partitions answer different deployment questions. Second, a conformal budget is useful as a control interface only when nominal and observed rates are shown together. Third, causal rolling context is a defensible way to capture rate attacks, provided state resets and past-only construction are explicit. Fourth, forensic language should expose evidence slots and preserve identifiers rather than convert a classification score into a causal allegation. Safe runbook checking [47], cost-aware AIOps routing [48], and explicit risk-card presentation [61], [63] further motivate keeping recommendations, model routing, and evidence display outside the classifier's unverified prose channel. Hybrid placement and stale-memory controls [65], [67] are deployment concerns for future broker-scale studies.

The scope is bounded by the corpus. All packets came from one deployment day, the class mixture was attack-heavy, and loop 4 contained substantially fewer legitimate packets than attack packets. Daily false-alert rates were normalized from the 6.1-hour final test span, so they describe operating intensity rather than evidence accumulated across repeated days. Marginal split-conformal calibration can also be stressed by loop drift, so the measured final-interval coverage and false-positive rate are the relevant outcomes. The results establish reproducible behavior for the observed MQTTEEB-D acquisition, not universal performance across brokers, firmware stacks, or organizations. Reliability forecasts for tool-using agents [66] reinforce the need to report such scope limitations as observable failure conditions, not as footnotes to a single score.

D. CONCLUSIONS

A leakage-aware chronological evaluation on 222,567 labelled MQTTEEB-D packets showed that real-time MQTT intrusion detection must be assessed as a coupled problem of representation, temporal generalization, calibration, and evidence retention. The validation-selected Causal HGB reached scaled-test macro-F1 0.761 and attack AUROC 0.851, while loop holdout, omitted attacks, imbalance, corruption, and drift exposed where those headline values did not transfer. Conformal prediction sets and benign-calibrated alert thresholds made uncertainty and alert budgets measurable, but their observed final-interval behavior remained essential under nonexchangeable capture dynamics [20]. TinyLLM—the Tiny Language Layer for MQTT—converted stored detector evidence into closed-slot narratives with measured fidelity, latency, and footprint. The resulting pipeline answers the closed-world warning in [24] and the forensic traceability objective in [28] with an auditable design: later traffic is kept later, unseen attacks are tested as unseen, uncertainty is reported numerically, and every narrative value points back to a

retained packet record. The broader literature on lightweight IoT detection [31], conformal incident alerting [46], and evidence-constrained operations interfaces [45], [61] supports this separation of detector, uncertainty controller, and human-facing evidence layer.

E. References

[1] A. Aqachtoul, K. Karam, A. Elamrani, M. Najib, N. Rafalia, and M. Bakhouya, "MQTTEEB-D: A Real-World IoT Cybersecurity Dataset for AI-Powered Threat Detection in MQTT Networks," Mendeley Data, ver. 1, Mar. 20, 2025, doi: 10.17632/jfttfjn6tr.1.

[2] A. Aqachtoul, K. Karam, A. Elamrani, M. Najib, N. Rafalia, and M. Bakhouya, "MQTTEEB-D: A Real-World IoT Cybersecurity Dataset for AI-Powered Threat Detection in MQTT Networks," Data in Brief, vol. 62, Art. no. 111897, 2025, doi: 10.1016/j.dib.2025.111897.

[3] A. Banks and R. Gupta, Eds., "MQTT Version 3.1.1," OASIS Standard, Oct. 29, 2014. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. Accessed: Aug. 5, 2026.

[4] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "MQTTset, a New Dataset for Machine Learning Techniques on MQTT," Sensors, vol. 20, no. 22, Art. no. 6578, 2020, doi: 10.3390/s20226578.

[5] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset)," in Selected Papers from the 12th International Networking Conference, LNNS, vol. 180. Cham, Switzerland: Springer, 2021, pp. 73-84, doi: 10.1007/978-3-030-64758-2_6.

[6] M. A. Khan, M. A. Khan, S. U. Jan, J. Ahmad, S. S. Jamal, A. A. Shah, N. Pitropakis, and W. J. Buchanan, "A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT," Sensors, vol. 21, no. 21, Art. no. 7016, 2021, doi: 10.3390/s21217016.

[7] E. Jove, J. Aveleira-Mata, H. Alaiz-Moreton, J.-L. Casteleiro-Roca, D. Y. Marcos del Blanco, F. Zayas-Gato, H. Quintian, and J. L. Calvo-Rolle, "Intelligent One-Class Classifiers for the Development of an Intrusion Detection System: The MQTT Case Study," Electronics, vol. 11, no. 3, Art. no. 422, 2022, doi: 10.3390/electronics11030422.

[8] A. Alatram, L. F. Sikos, M. Johnstone, P. Szewczyk, and J. J. Kang, "DoS/DDoS-MQTT-IoT: A Dataset for Evaluating Intrusions in IoT Networks Using the MQTT Protocol," Computer Networks, vol. 231, Art. no. 109809, 2023, doi: 10.1016/j.comnet.2023.109809.

[9] S. Ullah, J. Ahmad, M. A. Khan, M. S. Alshehri, W. Boulila, A. Koubaa, S. U. Jan, and M. M. Iqbal Ch, "TNN-IDS: Transformer Neural Network-Based Intrusion Detection System for MQTT-Enabled IoT Networks," Computer Networks, vol. 237, Art. no. 110072, 2023, doi: 10.1016/j.comnet.2023.110072.

[10] A. Al Hanif and M. Ilyas, "Effective Feature Engineering Framework for Securing MQTT Protocol in IoT Environments," Sensors, vol. 24, no. 6, Art. no. 1782, 2024, doi: 10.3390/s24061782.

[11] H. Zeghida, M. Boulaiche, R. Chikh, A. Patel, A. L. B. Barros, and A. M. Bamhdi, "XMID-MQTT: Explaining Machine Learning-Based Intrusion Detection System for MQTT Protocol in IoT Environment," International Journal of Information Security, vol. 24, no. 3, Art. no. 128, 2025, doi: 10.1007/s10207-025-01036-w.

[12] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in Advances in Neural Information Processing Systems 30, 2017, pp. 3146-3154.

[13] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," Science, vol. 313, no. 5786, pp. 504-507, 2006, doi: 10.1126/science.1127647.

[14] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in Proc. 34th International Conference on Machine Learning, PMLR, vol. 70, 2017, pp. 1321-1330.

[15] H. He and E. A. Garcia, "Learning from Imbalanced Data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, 2009, doi: 10.1109/TKDE.2008.239.

[16] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," PLOS ONE, vol. 10, no. 3, Art. no. e0118432, 2015, doi: 10.1371/journal.pone.0118432.

[17] A. N. Angelopoulos and S. Bates, "Conformal Prediction: A Gentle Introduction," Foundations and Trends in Machine Learning, vol. 16, no. 4, pp. 494-591, 2023, doi: 10.1561/2200000101.

[18] V. Vovk, "Conditional Validity of Inductive Conformal Predictors," in Proc. Asian Conference on Machine Learning, PMLR, vol. 25, 2012, pp. 475-490.

[19] I. Gibbs and E. Candès, "Adaptive Conformal Inference Under Distribution Shift," in Advances in Neural Information Processing Systems 34, 2021, pp. 1660-1672.

[20] R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani, "Conformal Prediction Beyond Exchangeability," Annals of Statistics, vol. 51, no. 2, pp. 816-845, 2023, doi: 10.1214/23-AOS2276.

[21] S. Bates, A. Angelopoulos, L. Lei, J. Malik, and M. I. Jordan, "Distribution-Free, Risk-Controlling Prediction Sets," Journal of the ACM, vol. 68, no. 6, Art. no. 43, 2021, doi: 10.1145/3478535.

[22] M. Cantone, C. Marrocco, and A. Bria, "Machine Learning in Network Intrusion Detection: A Cross-Dataset Generalization Study," IEEE Access, vol.

12, pp. 144489-144508, 2024, doi: 10.1109/ACCESS.2024.3472907.

[23] T. Zoppi, A. Ceccarelli, T. Puccetti, and A. Bondavalli, "Which Algorithm Can Detect Unknown Attacks? Comparison of Supervised, Unsupervised and Meta-Learning Algorithms for Intrusion Detection," Computers & Security, vol. 127, Art. no. 103107, 2023, doi: 10.1016/j.cose.2023.103107.

[24] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in Proc. IEEE Symposium on Security and Privacy, 2010, pp. 305-316, doi: 10.1109/SP.2010.25.

[25] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems 30, 2017, pp. 4768-4777, doi: 10.5555/3295222.3295230.

[26] P. Zhang, G. Zeng, T. Wang, and W. Lu, "TinyLlama: An Open-Source Small Language Model," arXiv:2401.02385, 2024, doi: 10.48550/arXiv.2401.02385.

[27] Z. Liu, C. Zhao, F. Iandola, C. Lai, Y. Tian, I. Fedorov, Y. Xiong, E. Chang, Y. Shi, R. Krishnamoorthi, L. Lai, and V. Chandra, "MobileLLM: Optimizing Sub-Billion Parameter Language Models for On-Device Use Cases," in Proc. 41st International Conference on Machine Learning, PMLR, vol. 235, 2024, pp. 32431-32454.

[28] K. Kent, S. Chevalier, T. Grance, and H. Dang, "Guide to Integrating Forensic Techniques into Incident Response," NIST Special Publication 800-86, Aug. 2006, doi: 10.6028/NIST.SP.800-86.

[29] 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 Science International: Digital Investigation, vol. 48, Art. no. 301683, 2024, doi: 10.1016/j.fsidi.2023.301683.

[30] F. Y. Loumachi, M. C. Ghanem, and M. A. Ferrag, "GenDFIR: Advancing Cyber Incident Timeline Analysis Through Retrieval-Augmented Generation and Large Language Models," Computers, vol. 14, no. 2, Art. no. 67, 2025, doi: 10.3390/computers14020067.

[31] S. Lu and D. Zhou, "TinyLLM-Assisted Intrusion Detection for Real-Time IoT Networks," J. Adv. Comput. Syst., vol. 4, no. 8, pp. 72-87, Aug. 2024, doi: 10.69987/JACS.2024.40809.

[32] Y. Li and S. Lu, "Language-Guided Feature Selection for DDoS and Intrusion Detection on CICIDS2017," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 284-305, Apr. 2025, doi: 10.51903/jtie.v4i1.531.

[33] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, "IoT Traffic Classification and Anomaly Detection Method Based on Deep Autoencoders," Appl. Comput. Eng., vol. 69, pp. 64-70, Jul. 2024, doi: 10.54254/2755-2721/69/20241511.

[34] J. Bai, S. Chen, D. Zheng, and M.-J. Kuo, "Interpretable Attack-Chain Stage Detection from AWS CloudTrail Event Sequences via Linear Models and HMM Smoothing," Informatics, Electr. Electron. Eng. (Infotron), vol. 6, no. 1, pp. 28-43, May 2026, doi: 10.33474/infotron.v6i1.24923.

[35] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, "Predictive Optimization of DDoS Attack Mitigation in Distributed Systems Using Machine Learning," Appl. Comput. Eng., vol. 64, pp. 94-99, May 2024, doi: 10.54254/2755-2721/64/20241350.

[36] G. Liu, S. He, and I. Liu, "LLM-Augmented Multi-Source Root Cause Attribution for CPU and Network Faults in Microservices," J. Adv. Comput. Syst., vol. 3, no. 6, pp. 39-57, Jun. 2023, doi: 10.69987/JACS.2023.30604.

[37] Q. Xin, "Host-Based Intrusion Detection with System Call Sequences: Window Localization and Forensic Narratives," AVITEC, vol. 8, no. 2, pp. 325-334, Jun. 2026, doi: 10.28989/avitec.v8i2.3973.

[38] B. Zhang, H. Rao, and D. Zhao, "Evidence-Grounded RAG for Cloud-Native DevOps: Hallucination-Resistant AIOps Question Answering over Private Operations Documents," J. Adv. Comput. Syst., vol. 4, no. 3, pp. 109-125, Mar. 2024, doi: 10.69987/JACS.2024.40308.

[39] G. Liu, C. Li, and E. Zhang, "OpsLLM for Cloud Incident Triage: Bilingual RAG-Based Root Cause Analysis and Alert Summarization for AI Infrastructure Operations," J. Adv. Comput. Syst., vol. 4, no. 4, pp. 97-111, Apr. 2024, doi: 10.69987/JACS.2024.40408.

[40] C. Li, J. Bai, and S. Wang, "Evidence-Chain Reliable RAG: Word-Level Hallucination Detection, Source Attribution, and Provenance Explanation for LLM Applications," J. Adv. Comput. Syst., vol. 4, no. 2, pp. 76-92, Feb. 2024, doi: 10.69987/JACS.2024.40207.

[41] H. Zhang, "LLM-Driven CI Failure Diagnosis and Automated Repair: From GitHub Actions Logs to Patch Recommendation," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 190-214, Apr. 2025, doi: 10.51903/jtie.v4i1.484.

[42] [42] Q. Xin, "Explaining OpenStack Failure-Injection Log Anomalies with Retrieved Normal Prototypes," Emerg. Inf. Sci. Technol., vol. 6, no. 2, pp. 125-146, Nov. 2025, doi: 10.18196/eist.v6i2.31232.

[43] X. Sun, Z. S. Zhong, and Q. Wu, "Retrieval-Grounded HDFS Log Anomaly Detection and Deterministic Failure Narrative Generation," J. Comput. Syst. Appl., vol. 3, no. 1, pp. 15-30, Jun. 2026, doi: 10.64229/j6d7fr94.

[44] [44] Q. Xin, "Self-Supervised Log Anomaly Detection with LogBERT-Style Transformers: Full Empirical Evaluation on a Reproducible SynHDFS Benchmark," J. Electr. Eng. Comput. Sci., vol. 11, no. 1, pp. 23-35, May 2026, doi: 10.54732/jeecs.v11i1.3.

[45] [45] J. Nie, G. Liu, C. Li, and T. Zou, "Evidence-Constrained Incident Visualization Cards for Distributed Cloud Logs: A UI/UX Framework for Turning Hadoop, OpenStack, and ZooKeeper Logs into Actionable SRE Design Interfaces," Int. J. Graph. Des., vol. 4, no. 1, pp. 179-185, Apr. 2026, doi: 10.51903/ijgd.v4i1.3703.

[46] Q. Xin, "Log Anomaly Detection with Conformal Alert Control and Evidence-Grounded Incident Ticket Generation," AVITEC, vol. 8, no. 2, pp. 247-264, May 2026, doi: 10.28989/avitec.v8i2.3974.

[47] B. Zhang, X. Sun, G. Liu, and B. Zhou, "LLM-Style DevOps Copilot for Cloud-Native Troubleshooting: Retrieval-Augmented Runbook Generation and Command-Safety Evaluation," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 104-118, Aug. 2026, doi: 10.51903/jtie.v5i2.534.

[48] C. Li, G. Liu, and Z. Zhao, "Cost-Aware LLM-Style Routing for AIOps Log Analysis: Log Parsing, Anomaly Detection, Fault Diagnosis, and Incident Summarization on LogEval Task Files," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 91-103, Jun. 2026, doi: 10.51903/jtie.v5i2.538.

[49] Y. Chen, Y. Zhang, D. Chau, and M. Sherman, "Credit Card Default Risk Tiering with Probability Calibration and Uncertainty-Driven Rejection: A Reproducible Study on the UCI Credit Card Clients Dataset," J. Adv. Comput. Syst., vol. 3, no. 4, pp. 31-47, Apr. 2023, doi: 10.69987/JACS.2023.30403.

[50] J. Jin, T. Huang, and S. Lu, "Cost-Sensitive Learning, Simulated PU Learning, and One-Class Autoencoding for Extreme-Imbalance Credit Card Fraud Detection," J. Adv. Comput. Syst., vol. 4, no. 6, pp. 64-73, Jun. 2024, doi: 10.69987/JACS.2024.40605.

[51] H. Zhang, "DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models," J. Adv. Comput. Syst., vol. 3, no. 7, pp. 24-40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[52] J. Jin, T. Huang, and S. Lu, "A Model-Risk-Friendly Probability of Default Workflow: Calibration, Distribution-Free Uncertainty Quantification, and SHAP Explanations on the UCI Credit Card Default Dataset," J. Adv. Comput. Syst., vol. 4, no. 6, pp. 74-85, Jun. 2024, doi: 10.69987/JACS.2024.40606.

[53] Q. Xin, "Uncertainty-Aware Late Fusion for 3D Perception (Confidence Calibration + Fusion Rule Learning)," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 215-238, Apr. 2025, doi: 10.51903/jtie.v4i1.485.

[54] Z. S. Zhong, X. Pan, and Q. Lei, "Bridging Domains with Approximately Shared Features," in Proc. 28th Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR, vol. 258, 2025, pp. 559-567.

[55] [55] D. Zheng, B. Zhang, and J. Geibel, "VerifySafe: Toxicity-Safe Agent Responses under Adversarial Prompts with Evidence-Based Self-Verification," J. Adv. Comput. Syst., vol. 4, no. 1, pp. 67-82, Jan. 2024, doi: 10.69987/JACS.2024.40106.

[56] Z. S. Zhong, J. Chen, E. Zhong, and X. Sun, "Evidence-Calibrated RAG for Unanswerable Question Answering: Retrieval Coverage, Abstention Calibration, and Hallucination-Proxy Analysis on SQuAD 2.0," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 502-520, Aug. 2025, doi: 10.51903/jtie.v4i2.536.

[57] D. Zheng and C. Li, "Behavior-Level Jailbreak Resistance via Multi-Stage Refusal + Utility Preservation," J. Adv. Comput. Syst., vol. 4, no. 1, pp. 83-99, Jan. 2024, doi: 10.69987/JACS.2024.40107.

[58] J. Jin, "Evidence-Chain Reliable RAG: Hallucination Detection, Source Attribution, and Deterministic Provenance Explanations," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 520-533, Aug. 2025, doi: 10.51903/jtie.v4i2.535.

[59] D. Zheng, C. Li, and H. Davidson, "Continual Red-Teaming for In-the-Wild Jailbreaks via Online Guardrail Updates and Guardrail Distillation," J. Adv. Comput. Syst., vol. 3, no. 2, pp. 35-49, Feb. 2023, doi: 10.69987/JACS.2023.30203.

[60] R. Zhang, Z. Wen, C. Wang, C. Tang, P. Xu, and Y. Jiang, "Quality Analysis and Evaluation Prediction of RAG Retrieval Based on Machine Learning Algorithms," arXiv:2511.19481, 2025, doi: 10.48550/arXiv.2511.19481.

[61] W. Su, H. Rao, and E. Ma, "Privacy and Data-Integrity Risk Cards for LLM Agents: A UI/UX Design Framework for Secure Human Oversight under Prompt-Injection Attacks," Int. J. Graph. Des., vol. 4, no. 1, pp. 186-191, Apr. 2026, doi: 10.51903/ijgd.v4i1.3699.

[62] J. Jin, "LLM-Style Evidence Cards for Scientific Search Interfaces: A UI/UX Design Framework for Retrieval Transparency, Ranking Trust, and Visual Evidence Hierarchy," Int. J. Graph. Des., vol. 3, no. 2, pp. 397-414, Oct. 2025, doi: 10.51903/ijgd.v3i2.3698.

[63] G. Liu, S. He, and H. Wong, "LLM-Compatible Visual Brief Cards for AI Infrastructure Capacity Dashboards: A UI/UX Framework for Turning Forecast Risk into Graphic Design Decisions," Int. J. Graph. Des., vol. 3, no. 1, pp. 196-213, May 2025, doi: 10.51903/ijgd.v3i1.3723.

[64] Y. Li, S. Lu, and L. Zhao, "LLM-as-Design-Critic: Aligning AI-Generated UI Feedback with Human Graphic Design Judgment," Int. J. Graph. Des., vol. 3, no. 1, pp. 196-215, May 2025, doi: 10.51903/ijgd.v3i1.3661.

[65] Q. Xin, "Hybrid Cloud Architecture for Efficient and Cost-Effective Large Language Model Deployment," J. Inf. Syst. Informatics, vol. 7, no. 3, pp. 2182-2195, Sep. 2025, doi: 10.51519/journalisi.v7i3.1170.

[66] Z. S. Zhong, C. Li, and H. Rao, "Trajectory Reliability Prediction for Generalist AI Agents:

Tool-Use Failure Analysis and Success Forecasting on ZClawBench," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 341-360, Apr. 2026, doi: 10.51903/jtie.v5i1.539.

[67] X. Sun, Y. Lu, and J. Chen, "Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting," J. Adv. Comput. Syst., vol. 3, no. 8, pp. 9-24, Aug. 2023, doi: 10.69987/JACS.2023.30802.