

Cost and Confidence Tiered Explainable and Generative Triage for SOC/CSIRT Operations over Network Telemetry Alerts/Events

^{1st} Mohammad Makchudul Alam

Cyber Threat Intelligence
BGD e-GOV CIRT
DNS Research Lab
Dhaka, Bangladesh
engrarifce@gmail.com

^{2nd} Md. Redowan Z. Anik

Cyber Threat Intelligence
BGD e-GOV CIRT
Dhaka, Bangladesh
redowanzanik@gmail.com

^{3rd} Sabrein S. E. el Bodour

Cyber Threat Intelligence
BGD e-GOV CIRT
Dhaka, Bangladesh
sabreinsr@gmail.com

^{4th} Anca Delia Jurcut

School of Computer Science
University College Dublin
Dublin, Ireland
anca.jurcut@ucd.ie

Abstract—Security Operations Centers (SOCs) and national Computer Security Incident Response Teams (CSIRTs) triage third-party security observations, such as Shadowserver reports of sinkhole contacts and exposed services, at volumes far beyond analyst capacity, where opaque automation is unacceptable. We present CSIRTNova’s deployed triage subsystem, which composes explainable and generative AI in a confidence-gated cascade: a decision cache, a calibrated XGBoost classifier emitting exact TreeSHAP attributions, an LLM, and a human analyst. On its own distribution the cascade is cheap, calibrated (expected calibration error 0.004, 1.7 ms per alert), fully explained, and its two cheap tiers carry 90.7% of automated verdicts. Ablating it on 1,070 later alerts, scored against disposition labels built from evidence the classifier never sees, maps its operating envelope: the 0.90 gate admits every alert, so the cascade resolves at its classifier (0.508 on the stratified sample, 0.726 re-weighted to the deployed mix), cache reuse inherits that accuracy, and three generative models, one local and two hosted, do not improve on it, because what decides these dispositions reaches neither the features nor the prompt. We distill design principles—abstention-capable tiers, applicability-aware gates, one provenance-carrying decision record, human-only feedback—and map them to EU AI Act transparency and oversight duties.

Index Terms—Explainable AI, SHAP, security alert triage, generative AI, large language models, human-in-the-loop, SOC/CSIRT, Shadowserver, EU AI Act.

I. INTRODUCTION

A *Security Operations Center* (SOC) is the team that continuously monitors an organization’s systems and triages the security alerts they raise. A *Computer Security Incident Response Team* (CSIRT) receives, analyzes, and coordinates the response to incidents on behalf of a *constituency*, the set of organizations and networks it serves. National and sectoral CSIRTs rarely have sensors inside their constituents’ networks. Instead they rely on third-party security observations, most prominently the free daily reports of the Shadowserver Foundation [1]. These list, for each network owner, the addresses seen contacting a botnet *sinkhole* (a server that has taken over a botnet’s command-and-control domain, so infected hosts reveal themselves by connecting to it), attacking *honeypots* (decoy systems), or exposing vulnerable or openly accessible services to Internet-wide scans. They summarize observations made elsewhere; they are not packet or flow traffic captured by the CSIRT. A platform parses them into events and condenses

the events into alerts. Throughout this paper, and in its title, *network telemetry alerts/events* means alerts and events derived from such third-party observation reports, not traffic the CSIRT captures itself.

The problem is volume. In our deployment one seven-week window holds 24.4 million such events, condensed into 158k alerts for 156 constituencies. Each report type carries a published definition [2], so the observation is rarely in doubt: a sinkhole record means that address contacted a sinkholed controller. What the team decides is the *disposition*—whether the indicator is attributable to a constituent and corroborated enough to act on (*notify*), attributable to none and therefore telemetry (*roll up*), or attributable but too thin (*review*). Analysts already report overwhelming false-positive rates and alert fatigue [5], which makes triage the step where automation is most valuable. It is also where automation is most dangerous: a false dismissal leaves a compromised host unattended, an opaque escalation erodes the trust a team depends on, and regulation increasingly requires automated decisions to be transparent and subject to human oversight [6].

Two AI paradigms can help. Explainable AI (XAI) offers fast, faithful per-decision attributions for tree models [3], [4]; generative AI (GenAI) offers flexible reasoning over unstructured evidence. Neither suffices alone. A classifier over a fixed feature space cannot reason about evidence outside that space, while a large language model (LLM) is too slow and costly to run on every alert, and its free text is not a faithful attribution. The question is therefore how to *compose* heterogeneous deciders so that each alert is handled by the least expensive mechanism that can handle it correctly, while every automated verdict remains explainable and reversible. We study this question in CSIRTNova, a deployed SOC/CSIRT platform, and contribute:

- a four-tier, confidence-gated **triage cascade** (cache → calibrated XGBoost with exact TreeSHAP → LLM → analyst) and six **design principles** for ordering tiers, setting gates, preserving provenance, handling failure, and routing feedback (Sec. IV);
- an **explanation layer** that stores attributions and generative rationales in one decision record, mapped to EU AI Act Articles 13 and 14 (Secs. IV and V);

- **deployment measurements** of calibration, latency, and tier use, followed by a **tier ablation**, a gate-sensitivity analysis, a cache consistency check, and a genuine-LLM evaluation on 1,070 later alerts scored against independent disposition labels (Sec. VI), which show where the composition helps and where it fails.

II. RELATED WORK

Alert triage is the recognized bottleneck of SOC and CSIRT work. Analysts describe most alarms as false positives and distrust tools that do not explain themselves [5], and a recent survey of LLMs for SOC catalogues triage assistants and alert summarizers while noting their cost, hallucination, and evaluation gaps [7]. Our earlier framework, IROF [8], argued for combining XAI, LLMs, and orchestration for prioritization and response; this paper reports how its triage core behaves once deployed and ablated. Composing deciders of increasing cost is established outside security. LLM cascades such as FrugalGPT send a query to cheaper models first and escalate when the answer looks unreliable [9], and learning-to-defer methods decide when a model should hand a case to a human [10]. Both optimize one model family or one model-human hand-off, and neither addresses how an explanation survives the hand-off between heterogeneous tiers, or a setting whose final tier is an accountable analyst.

XAI for network security has progressed from general surveys [11], [12] to a dedicated survey of explainability for traffic classification, prediction, and intrusion detection [13]. Representative systems explain multimodal traffic classifiers to improve their reliability [14], explain both the inputs and the outputs of malicious-traffic detectors [15], make network intrusion detection transparent to operators [16], and explain DDoS identification in IoT networks [17]. Cross-layer work explains intrusion decisions from network data together with application- and protocol-level semantics [18], and the Internet of Explainable Things (IoXT) paradigm argues that explainability must be a system-level property with end-to-end provenance and traceability rather than a per-model add-on [19]. On the tooling side, SHAP grounds feature attribution in Shapley values [3], TreeSHAP computes them exactly for tree ensembles [4] and ships inside XGBoost [20], and model-agnostic methods such as LIME [21] are too slow for per-alert use.

Three gaps remain. First, prior XAI work explains a single model, usually over packet or flow features, whereas CSIRT triage works on third-party observation reports and mixes deterministic, learned, and generative deciders. Second, when such deciders are composed, it is open how explanations and provenance should survive the hand-off between tiers, and whether the composition beats simpler alternatives. Third, we are not aware of deployed triage systems that instrument the transparency and oversight duties the EU AI Act attaches to high-risk uses [6]. The components we use are all established; our contribution is their composition, the principles that govern it, and an ablation that tests it against simpler alternatives.

TABLE I
DEPLOYMENT SNAPSHOT USED FOR THE FIRST MEASUREMENTS.

Shadowserver events (2026-05-21 to 07-09)	24,424,391
Report types/source countries	131/64
Constituent organizations/assets	156/2,645
Derived security alerts	158,332
Alerts with a full per-tier cascade record	3,598

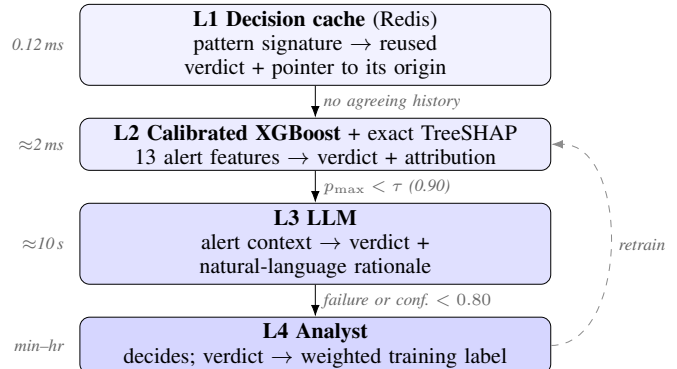


Fig. 1. The four-tier cascade. Each alert returns at the cheapest tier that accepts it; every verdict is written to one decision record (P3); analyst verdicts feed back as training labels (P6). Latencies are measured (Sec. VI).

III. SYSTEM AND DATA

CSIRTNova is a multi-tenant SOC/CSIRT platform serving 156 constituencies (national CERTs, sectoral teams, and enterprises). Shadowserver reports and OSINT feeds are parsed into *events* (one observation of one address), matched to a constituency's assets, enriched, risk-scored, and grouped by organization and report category into *security alerts* that list their indicator addresses. Alerts are the unit of triage. Table I summarizes the snapshot on which the cascade was first measured. Internet-wide scans (HTTP, DNS, SSL, SSH, SNMP, DDoS-reflector probing) and sinkhole contacts dominate its 131 report types. For the home constituency (Bangladesh, over 99% of events) the snapshot associates sinkhole contacts with 154,081 distinct hosts in 1,256 autonomous systems, DDoS-amplifier exposure with 336,368 hosts, and open or vulnerable services with 220,658 and 131,373 hosts. Section VI-B adds a second, later population of 28,053 Shadowserver-derived alerts.

IV. THE CASCADE AND ITS DESIGN PRINCIPLES

Each alert passes through four tiers of increasing cost and returns at the first tier that accepts it (Fig. 1).

L1: decision cache. A signature over alert type, severity, indicator count, tags, and ATT&CK tactics keys a Redis list of recent verdicts for that pattern. A verdict is reused only when at least three of the five newest entries agree and the newest three have confidence ≥ 0.85 ; a signature expires seven days after its last write. **L2: calibrated classifier.** An XGBoost model (Sec. V) returns a verdict only if its calibrated top-class probability reaches $\tau = 0.90$, and abstains otherwise. **L3: generative reasoning.** Abstained alerts go to an LLM

that returns a verdict, a confidence, and a rationale; verdicts below 0.80 confidence are not applied. L2 and L3 verdicts are written back to L1. **L4: analyst.** Everything else reaches the analyst queue. In the measured deployment an unreachable L3 fell back to a deterministic rule scorer that was recorded as L3; principle P5 below removes that conflation.

Design principles. The deployment and the ablation of Sec. VI-B lead us to six principles for turning a sequence of tiers into a reusable architecture:

- P1 *Order tiers by marginal cost, and let every tier abstain.* A tier that must always answer hides its failures and removes the escalation path.
- P2 *Gate on calibration and on applicability.* A calibrated probability is trustworthy only on the distribution it was calibrated on, so a gate must also test whether the input resembles the training data, with each threshold set by a sensitivity sweep on labeled data.
- P3 *Keep one decision record across tiers.* Every verdict carries its tier, engine, model version, confidence, class probabilities, input digest and time, so any action traces back to its evidence, in the spirit of IoXT’s end-to-end provenance [19].
- P4 *Type the explanations.* The record holds an *attribution* (L2), a *rationale* flagged as non-faithful (L3), or a *reuse pointer* to the decision a cache hit copied (L1), so no verdict is unexplained.
- P5 *Fail toward the analyst.* An unavailable tier abstains and the record marks the degraded mode; no substitute verdict is issued in its name.
- P6 *Learn only from humans.* Analyst verdicts re-enter training with high weight. Machine verdicts, whether cached, fallback, or model outputs, never do, because self-labels reinforce the model’s own errors (Sec. VII).

Bounded autonomy and the EU AI Act. Confidence maps to an autonomy ladder (observe < 0.4 , investigate $0.4\text{--}0.7$, assisted $0.7\text{--}0.9$, autonomous ≥ 0.9). Two caps override it: an entitlement cap, and a blast-radius cap under which coordinated-response categories (botnet takedown, ransomware, dark-web exposure) always require a human. The decision record (P3, P4) instruments Article 13 transparency. The always-reachable analyst queue, forced routing of high-impact categories, and recorded overrides instrument Article 14 human oversight [6]. We treat both as engineering controls, not as a legal conformity assessment.

V. EXPLAINABILITY: XAI AT L2, GENAI AT L3

L2 is an XGBoost ensemble (600 trees) over 13 alert features: encoded alert type, severity, risk score, correlation confidence, IoC counts (IPs and domains), ATT&CK tactic presence and count, tag count, event count, source-type count, correlation flag, and encoded source. It predicts `true_positive`, `false_positive`, or `needs_review`, and Platt (sigmoid) calibration makes the gate τ a probability rather than a raw score. For an accepted

TABLE II
TREESHAP ATTRIBUTIONS FROM THE DEPLOYED MODEL FOR A HIGH-SEVERITY C2 ALERT (TOP FACTORS BY $|\phi_i|$). BLUE = TOWARD THE VERDICT, RED = AGAINST.

Feature	Value	ϕ_i
C2 comms, high sev.	$\rightarrow$ true_positive (0.998)	
severity	high (3)	+1.353
tag_count	2	+0.281
alert_type	c2 (2)	+0.137
ioc_domain_count	1	-0.027
source_type_count	1	-0.021

verdict of class c , XGBoost’s native `pred_contribs` path computes exact TreeSHAP values [4] that satisfy

$$f_c(x) = \phi_0 + \sum_{i=1}^{13} \phi_i(x), \quad (1)$$

where ϕ_0 is the class base value and f_c the class margin. A positive ϕ_i pushed the model toward the verdict and a negative one away from it. The record stores the five features with the largest $|\phi_i|$, with their raw values.

Table II shows attributions from the deployed model: a high-severity C2 alert returns `true_positive` at 0.998 on severity and threat tags. The same model returns `needs_review` at 0.992 on a low-severity scan, again dominated by severity. Across all 28,053 later alerts the mean $|\phi_i|$ is 1.42 for severity, 0.16 for tag count, 0.13 for alert type, and at most 0.02 for every indicator feature. The explanations thus expose, cheaply and consistently, a policy close to a severity threshold, which Sec. VI tests.

When L2 abstains, L3 returns a verdict with a free-text rationale. The two explanations differ in kind: L2’s attribution is faithful to the model over a fixed feature space, while L3’s rationale is flexible but unverified. The record therefore types them (P4), so an analyst always knows which kind of reason they are reading.

VI. EVALUATION

A. Deployment snapshot

The cascade was first measured on the 3,598 alerts of Table I that carry a full per-tier record. L2 (version 20260710) was trained, calibrated, and tested on a labeled subset of them (2,158/720/720 alerts). It reaches 0.9972 test accuracy (log-loss 0.0155), but the labels come from a deterministic alert taxonomy, so this is *agreement with the taxonomy* rather than verified correctness. Table III shows what the figure hides: the whole corpus has only 11 `false_positive` examples, the two in the test split are both missed, and macro-F1 is 0.665. Calibration holds on this distribution: expected calibration error 0.0035 (10 bins), Brier 0.0048, and 715 of 720 predictions in $(0.9, 1]$ with a +0.003 accuracy–confidence gap. L2 costs 1.71 ms per alert on one core (p95 2.11 ms) including TreeSHAP, and an L1 lookup costs 0.12 ms. Of the 3,598 automated verdicts, L1 served 76.7%, L2 14.1%, and L3 9.3%; most of the latter came from the deterministic fallback, so 9.3% is an upper bound on real LLM use.

TABLE III

PER-CLASS RESULTS OF L2 ON THE 720-ALERT TEST SPLIT (TAXONOMY LABELS).

Class	Prec.	Rec.	F1	Support
false_positive	0.000	0.000	0.000	2
needs_review	0.992	1.000	0.996	132
true_positive	0.998	1.000	0.999	586
Macro avg	0.664	0.667	0.665	720
Weighted avg	0.994	0.997	0.996	720

Takeaway. *On its own distribution the cascade is cheap, calibrated, and explained, and its cheap tiers carry 90.7% of the automated load; but its accuracy is agreement with the rules that generated its labels, and the rare false-positive class is not learned.*

B. Tier ablation on later Shadowserver alerts

Setup. To test the composition rather than its parts, we replay the cascade over the 28,053 Shadowserver-derived alerts raised between 2026-07-10 and 2026-09-15 (24,708 sinkhole, 1,221 honeypot, 1,024 vulnerable-service, 697 open/accessible, 221 compromised-website, 182 other). The deployed L2 artifact predates this window, so none of these alerts was in its training data. From them we draw a seeded stratified sample of 1,070 alerts (family $\times$ severity) and label each with the *disposition* it owes, by *hindsight corroboration*. Because the reporter has already defined what was observed, the label does not ask whether the event happened; it asks what this constituency should do about it. An alert is `roll_up` when no indicator is attributable to its constituency (every address belongs to anycast CDN or privacy-relay egress, or falls outside the organization’s country with no domain under its ccTLD): the observation stands, but no constituent owns it. It is `notify` when an attributable indicator is re-observed by Shadowserver on at least two distinct report days within ± 7 days, and `review` otherwise. These labels use only evidence the classifier never sees, namely indicator ASN and geolocation, the organization’s constituency, and later re-observation; they are not analyst labels (Sec. VII). They yield 827 `notify`, 197 `roll_up`, and 46 `review`. We score the classifier’s verdicts against them through the mapping the cascade itself applies when it acts, `true_positive` $\rightarrow$ `escalate/notify`, `false_positive` $\rightarrow$ `telemetry roll-up`, `needs_review` $\rightarrow$ `review`, so the comparison is the one the deployed system makes. L3 is the real local LLM (llama3.2, 3B, 4-bit, on CPU) driven with the production triage prompt. L4 is scored as an oracle, so deferring to it is correct by construction: the informative columns of Table IV are the accuracy of the *automated* verdicts and the analyst load, not end-to-end accuracy.

Results. Every configuration that contains L2 returns exactly the same verdicts, because the deployed gate never fires: the calibrated top-class probability is at least 0.9918 on all 28,053 alerts, so L2 accepts every alert and L3 and

TABLE IV

TIER ABLATION ON 1,070 ALERTS WITH DISPOSITION LABELS ($\tau = 0.90$, L3 ACCEPTANCE 0.80). TIER COLUMNS AND LOAD ARE PERCENTAGES OF THE SAMPLE; “AUTO” IS ACCURACY OVER AUTOMATED VERDICTS ONLY, “LOAD” THE SHARE LEFT TO AN ANALYST, “MS” THE MEAN MACHINE LATENCY PER ALERT. SEC. VI-C SWEEPS τ .

Configuration	L1	L2	L3	L4	Auto	Load	ms
Human only	—	—	—	100	—	100	—
LLM $\rightarrow$ human	—	—	0.4	99.6	1.000	99.6	7290
XGBoost only	—	100	—	—	0.508	27.1	2.3
XGBoost $\rightarrow$ human	—	100	—	—	0.508	27.1	2.3
Cache $\rightarrow$ XGB $\rightarrow$ human	53.4	46.6	—	—	0.508	27.1	1.2
XGB $\rightarrow$ LLM $\rightarrow$ human	—	100	—	—	0.508	27.1	2.3
Full cascade	53.4	46.6	—	—	0.508	27.1	1.2

L4 are unreachable. Against those labels the verdicts are 0.508 accurate (95% CI 0.477–0.536, bootstrap) with macro-F1 0.227, and the model never predicts `false_positive`, the verdict the cascade maps to a roll-up, so all 197 alerts that belong to no constituent are escalated as if someone had to act on them. Accuracy is very uneven across families: 0.777 on sinkhole and 0.700 on compromised-website alerts, but 0.253 on honeypot, 0.007 on open/accessible, and 0.000 on the other family. The sample is stratified so that the rare families are measurable at all, which over-weights them: sinkhole is 88.1% of the population but 32.7% of the sample. Re-weighted to the deployed mix the same verdicts score 0.726; we headline the stratified figure because it is the one that exposes where the policy fails. The cache halves the mean cost (2.3 ms to 1.2 ms) and changes no verdict; the LLM tier changes nothing at all, because nothing reaches it.

Takeaway. *The cheap tiers deliver the cost economy they were built for, halving per-alert latency with no change in verdict; outside the training distribution that is the whole gain, since the gate admits everything and the escalation paths go unexercised, so a gate needs an applicability test as well (P2).*

C. Gate sensitivity

The gate is effectively binary on this population. Any $\tau \leq 0.99$ reproduces the deployed behaviour exactly. At $\tau = 0.995$ the 27.1% of alerts the model calls `needs_review` fall through; L3 accepts almost none of them, so they reach analysts, automated accuracy rises to 0.696 and end-to-end accuracy to 0.779, at a mean 1.76 s per alert. At $\tau \geq 0.999$ everything falls through to the analyst queue. Varying the L3 acceptance threshold between 0.5 and 0.9 changes nothing, because the local model almost never reports confidence ≥ 0.8 ; those figures reflect the deployed local tier, and Sec. VI-E shows what a hosted model does to them.

Takeaway. *A gate on calibrated confidence alone has no useful operating point here: the model is uniformly confident, including where it is wrong, so the gate must also test whether the alert resembles the training distribution (P2).*

D. Is cache reuse correct?

Replaying L1 over the full 28,053-alert stream, 76.2% of alerts are served from cache (74% of sinkhole, 95% of

honeypot, 99% of open/accessible alerts), and every hit reproduces the verdict L2 would have given, so the cache is self-consistent by construction. Correctness is another matter: on the labelled sample, cache-served decisions are 0.465 accurate while decisions L2 takes directly are 0.773. The signature is coarse enough to mix outcomes; of the 74 signatures with at least two sampled alerts, 22 cover more than one corroboration label, and the majority label accounts for only 86.3% of alerts in such groups.

Takeaway. *Cache reuse is as good as the tier it copies: self-consistent and very cheap, but bounded by L2’s correctness, and a signature blind to attribution will reuse one verdict across alerts owing different dispositions.*

E. The generative tier, actually run

An L3 tier is only as good as the model behind it, so we ran all 1,070 sampled alerts through the three models that completed it, on the identical production prompt: the local llama3.2 (3B, 4-bit, CPU) the deployment falls back to, and two hosted models, Claude Haiku 4.5 and Claude Sonnet 5 (Table V). Every response parsed. The differences are large, but not in the direction a pure capability argument predicts.

The local model is effectively single-valued: it answers `false_positive` on 1,066 of 1,070 alerts, so it reaches the right disposition on the roll-up cases only by rolling up everything, and it reports confidence ≥ 0.8 on 0.4% of cases, so under the production acceptance rule it abstains almost always. The hosted models are far more discriminating and agree with L2 on 81–88% of alerts, yet neither beats the classifier: 0.477 (Haiku) and 0.383 (Sonnet) against 0.508 for L2 alone. Neither ever answers `false_positive`, so like L2 they escalate all 197 alerts that no constituent owns. The reason is structural rather than a matter of model size: the disposition labels turn on indicator ASN, constituency membership, and re-observation across days, and none of that reaches the prompt, just as none of it reaches the 13-feature vector. A generative tier cannot recover evidence its input never carried.

Where the models differ sharply is in the quality of the self-reported confidence that the acceptance gate consumes. At the production bar of 0.80 and $\tau = 0.999$, Haiku accepts 41.9% of the stream and is right on 0.699 of what it decides, cutting analyst load to 34.6%; Sonnet accepts 4.5% but is right on 0.875. The cheap model buys throughput, the stronger one buys selectivity. A fourth engine, Claude Opus 5, could not complete the sample: its run was interrupted after 128 alerts. Scored on exactly those alerts (Table V, lower panel) it reaches 0.289, level with Sonnet 5, below Haiku 4.5 at 0.367 and the classifier’s 0.453, at 5.65 s and \$10.15 per thousand alerts — the ordering is unchanged. Fluency also brings unsupported specificity: 50 (Haiku) and 40 (Sonnet) rationales discuss domains for alerts that carry none, against 5 for the local model. Finally, the live path shows why measuring this tier mattered: of the 28,053 alerts, 20,614 (73.5%) carry an LLM-failure fallback verdict caused by hosted-provider

TABLE V
L3 ENGINES ON THE SAME 1,070 ALERTS WITH THE SAME PRODUCTION PROMPT. “ACC” IS AGAINST THE DISPOSITION LABELS (L2 ALONE SCORES 0.508); “CONF ≥ 0.8 ” IS THE SHARE THE PRODUCTION RULE WOULD AUTO-APPLY; COST IS PER 1,000 ALERTS AT LIST PRICES. THE LOWER PANEL SCORES EVERY ENGINE ON THE SAME 128 ALERTS AN INTERRUPTED CLAUDE OPUS 5 RUN COMPLETED; THAT SUBSET IS SKEWED TOWARD VULNERABLE-SERVICE ALERTS, SO ITS ABSOLUTE NUMBERS ARE NOT COMPARABLE WITH THE UPPER PANEL, ONLY WITH EACH OTHER.

L3 engine	Acc	mF1	agree L2	conf ≥ 0.8	s/alert	\$/1k
llama3.2 3B (local)	0.188	0.107	0.4	0.4	7.29	0
Claude Haiku 4.5	0.477	0.253	88.1	61.9	2.23	1.19
Claude Sonnet 5	0.383	0.210	81.0	4.5	2.48	2.67
<i>matched subset, n=128 (L2 scores 0.453 on it)</i>						
llama3.2 3B (local)	0.117	0.074	1.6	1.6	7.36	0
Claude Haiku 4.5	0.367	0.184	91.4	43.8	2.85	1.18
Claude Sonnet 5	0.289	0.154	83.6	4.7	2.56	2.64
Claude Opus 5	0.289	0.155	82.8	25.8	5.65	10.15

authentication and credit errors, 5,246 a classifier verdict, and only 81 a genuine LLM verdict.

Takeaway. *The worth of a generative tier is set by grounding and by the quality of its confidence, not by raw capability: no model beat a severity threshold on evidence none of them were given, the cheap model bought throughput while the strong one bought selectivity, and both asserted details the alert never contained.*

VII. DISCUSSION, LIMITATIONS, AND FUTURE DIRECTIONS

Labels. The disposition labels are independent of the classifier’s inputs and use hindsight the cascade lacks, but they encode our own attribution rules. Requiring re-observation on three days instead of two moves 39 alerts from `notify` to `review` and leaves the ablation unchanged (accuracy 0.507, macro-F1 0.258); dropping the shared-egress rule moves 21 alerts to `notify` and gives 0.526. They remain a proxy: the platform holds seven analyst-verified labels in total, none of them in our sample. An analyst-verified benchmark is the most valuable missing artifact here, and the disagreement set between tiers is where to build it.

Attribution as a data-quality probe. TreeSHAP showed early that the indicator features contributed almost nothing, which traced to a data-integration defect: IoC and ATT&CK enrichment ran on intelligence events but never reached the alert feature vector. After we wired it in, a controlled re-train showed that adding the model’s own verdicts as labels regressed accuracy without giving the indicator features any weight. The hazard then recurred at deployment scale: a later scheduled retrain drew 287k machine-generated triage records, most of them LLM-failure fallbacks, and reported 0.9999 accuracy against seven human labels. That is precisely what P6 forbids, and it explains the behaviour measured in Sec. VI-B: a model trained on taxonomy and machine labels reproduces the severity rule that generated them.

Detection labels for a disposition decision. The classifier predicts a detection vocabulary inherited from its bootstrapped

taxonomy, while the cascade delegates a disposition. For confirmed third-party reports those differ, and the gap is where the errors sit: no class means “real, but nobody here owns it”. The pipeline also discards the evidence that would settle it. Alerts are grouped by report *category*, so the precisely defined report type never reaches the alert: baseline `population_*` report types, which the schema marks as research telemetry rather than threat indicators, fall into the same categories that raise alerts and are indistinguishable at the triage surface from genuine threat reports. Carrying the report type onto the alert would give the features and the prompt the distinction they now lack.

Scope. We instrument Articles 13 and 14 as engineering controls, not as a legal conformity assessment. We report no controlled study of analyst time or error against a no-automation baseline, the deployment is a single national platform, and L3 was a 3B local model; a larger or retrieval-grounded model may behave differently.

Future directions. Five follow directly. (i) Applicability-aware gating (P2): conformal or out-of-distribution scores over report type and indicator mix, so L2 abstains on shifted alerts instead of accepting them at 0.99 confidence. (ii) Promoting the corroboration evidence to features and to the prompt: shared-egress ASN, constituency match, and re-observation counts are cheap, already available, and are what separated the dispositions here. (iii) An analyst-verified benchmark with inter-annotator agreement, seeded from inter-tier disagreement. (iv) Portable provenance across tiers and federated CSIRTs, in the IoXT sense [19]. (v) Grounding L3 in retrieved evidence with schema-validated output, measured against analysts rather than the classifier.

VIII. CONCLUSION

We described, deployed, and ablated a triage subsystem that composes a decision cache, a calibrated XGBoost classifier with exact TreeSHAP attributions, an LLM, and a human analyst into one confidence-gated cascade over Shadowserver-derived security alerts. In production it does what it was built to do: verdicts are calibrated and explained at ≈ 1.7 ms with no extra dependency, the two cheap tiers carry 90.7% of the automated load, and every automated decision leaves a record an analyst can read and contest. Extending the evaluation to 1,070 later alerts, judged against independent disposition labels, maps the envelope in which that design holds: the 0.90 gate admits every alert, so the cascade resolves at its classifier, cache reuse inherits that accuracy, and no generative model we tried improves on it, because the evidence these dispositions turn on reaches neither the features nor the prompt. The explanation layer made the envelope visible cheaply: TreeSHAP showed a policy close to a severity threshold, and the per-tier record showed a fallback logged as an LLM verdict. That is the contribution we would carry forward: a cost-tiered cascade is as good as its gates and its labels. Tiers must be able to abstain, gates must test applicability as well as confidence, the evidence that decides a disposition must reach the tier deciding it, and only humans should close the training loop.

REFERENCES

- [1] The Shadowserver Foundation, “Free daily network reports,” <https://www.shadowserver.org/what-we-do/network-reporting/>, 2026.
- [2] The Shadowserver Foundation, “Report schema: report types,” https://github.com/The-Shadowserver-Foundation/report_schema/blob/main/types.md, 2026.
- [3] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.
- [4] S. M. Lundberg *et al.*, “From local explanations to global understanding with explainable AI for trees,” *Nature Machine Intelligence*, vol. 2, pp. 56–67, 2020.
- [5] B. A. Alahmadi, L. Axon, and I. Martinovic, “99% false positives: A qualitative study of SOC analysts’ perspectives on security alarms,” in *Proc. 31st USENIX Security Symp.*, 2022, pp. 2783–2800.
- [6] European Parliament and Council, “Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act),” *Official Journal of the European Union*, 2024.
- [7] A. Habibzadeh, F. Feyzi, and R. E. Atani, “Large language models for security operations centers: A comprehensive survey,” *arXiv:2509.10858*, 2025.
- [8] M. M. Alam, M. R. Z. Anik, and A. D. Jurcut, “IROF: An AI-driven framework for automated threat detection, prioritization & incident response optimization with explainability,” *OIC-CERT J. Cyber Security*, vol. 6, no. 1, pp. 43–49, Aug. 2025.
- [9] L. Chen, M. Zaharia, and J. Zou, “FrugalGPT: How to use large language models while reducing cost and improving performance,” *arXiv:2305.05176*, 2023.
- [10] D. Madras, T. Pitassi, and R. Zemel, “Predict responsibly: Improving fairness and accuracy by learning to defer,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018.
- [11] N. Capuano, G. Fenza, V. Loia, and C. Stanzione, “Explainable artificial intelligence in cybersecurity: A survey,” *IEEE Access*, vol. 10, pp. 93575–93600, 2022.
- [12] A. Nadeem, D. Vos, C. Cao, L. Pajola, S. Dieck, R. Baumgartner, and S. Verwer, “SoK: Explainable machine learning for computer security applications,” in *Proc. IEEE EuroS&P*, 2023, pp. 221–240.
- [13] A. Nascita, G. Aceto, D. Ciunzio, A. Montieri, V. Persico, and A. Pescapé, “A survey on explainable artificial intelligence for Internet traffic classification and prediction, and intrusion detection,” *IEEE Commun. Surveys Tuts.*, vol. 27, no. 5, pp. 3165–3198, 2025.
- [14] A. Nascita, A. Montieri, G. Aceto, D. Ciunzio, V. Persico, and A. Pescapé, “Improving performance, reliability, and feasibility in multimodal multitask traffic classification with XAI,” *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1267–1289, 2023.
- [15] W. Lin, C. Xia, T. Wang, Y. Zhao, L. Xi, and S. Zhang, “Input and output matter: Malicious traffic detection with explainability,” *IEEE Network*, vol. 39, no. 2, pp. 259–267, 2025.
- [16] M. T. Islam, M. K. Syfullah, M. G. Rashed, and D. Das, “Bridging the gap: Advancing the transparency and trustworthiness of network intrusion detection with explainable AI,” *Int. J. Mach. Learn. Cybern.*, vol. 15, no. 11, pp. 5337–5360, 2024.
- [17] C. S. Kalutharage, X. Liu, C. Chrysoulas, N. Pitropakis, and P. Papadopoulos, “Explainable AI-based DDOS attack identification method for IoT networks,” *Computers*, vol. 12, no. 2, art. 32, 2023.
- [18] M. Georgiades and F. Hussain, “An explainable AI approach for interpretable cross-layer intrusion detection in Internet of Medical Things,” *Electronics*, vol. 14, no. 16, art. 3218, 2025.
- [19] M. Georgiades, “IoXT: The Internet of Explainable Things. Why explainability in IoT requires a new system-level paradigm and protocol design,” *arXiv:2608.08556*, 2026.
- [20] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. ACM SIGKDD*, 2016, pp. 785–794.
- [21] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why should I trust you?’: Explaining the predictions of any classifier,” in *Proc. ACM SIGKDD*, 2016, pp. 1135–1144.