

Typed, Calibrated, and Manipulable: Calibration Collapse and Adversarial State Injection in Non-Autoregressive Decision Models for Intrusion Detection and Alert Triage

Anonymous Author(s)

Abstract

A typed decision model answers fixed questions about a serialized record, such as whether a network flow is malicious and which attack class it belongs to, in one non-autoregressive pass with a confidence attached. Its output always conforms to the declared schema, which makes it attractive as an intrusion-detection and alert-triage classifier. Type-safety constrains the form of a decision and does not guarantee its correctness, and we report two failures of correctness. First, the model’s confidence collapses on the rare attack classes operators most need to trust: the pooled calibration error of the attack-class answer remains 0.145, concentrated on the rare classes. Second, the decision can be manipulated while it stays well-typed. On flows the model escalates, appending one public phrase to one attacker-controllable field reaches an attack success rate of 100.0% on the most affected attack class under a realizable perturbation ($n = 118$ flows pooled over 3 seeds, 95% Wilson lower bound 96.8%). The same phrase bank evades 100.0% of 137 escalated flows on the hosted JEV service (lower bound 97.3%). A second, independently trained model family reproduces the evasion at each of its 6 sizes, and its malicious-question confidence is poorly calibrated. Of three inference-time defenses, only input partitioning, which removes the attacker-controllable text, reduces attack success, at a cost in clean utility.

CCS Concepts

• Security and privacy → Intrusion/anomaly detection and malware mitigation; Network security; • Computing methodologies → Neural networks.

Keywords

intrusion detection, alert triage, typed decision models, non-autoregressive models, prompt injection, adversarial examples, confidence calibration, explanation faithfulness

1 Introduction

A security operations center runs on triage. Detection pipelines see far more network flows than analysts can read. An automated stage decides which flows to drop and which to escalate, and analysts read only what it escalates. This stage is increasingly a machine-learning classifier that emits a label and a confidence for each flow, and both the pipeline and the analyst treat that confidence as the probability that the label is correct. A confidently wrong decision either forwards a real attack as benign or spends analyst time on noise. The correctness of these decisions, at the point where the pipeline acts on them, is the asset this paper studies.

Decision models that answer in a single fast pass (System One models) fit this role; we call them typed decision models. Such a model reads a serialized record and answers a fixed set of typed

questions in one non-autoregressive pass: a yes/no question (is this flow malicious?), a choice question (which attack class?), and an ordered severity score. The yes/no answer is a probability, the choice and the severity each carry a confidence, and every answer conforms to the declared schema without generative decoding. The commercial instance, TypeSafe’s JEV [32], is sold on these properties: decisions are cheap, typed so that a pipeline can act on them without parsing free text, and accompanied by a confidence.

The guarantee the vendor sells is type-safety: the output conforms to the declared schema and option set whatever the input. Type-safety constrains the form of an answer and says nothing about whether it is right. A well-typed wrong decision does as much harm as a malformed one and is easier to trust. The vendor’s documentation [32] concedes that text written to steer the model can move the answer, and that the reported confidence is a decision margin rather than the probability that the answer is correct. To our knowledge, no independent study has asked what an adversary who writes part of the input can do to such a model deployed as a detector, or how its calibration behaves under the class imbalance a real detector faces.

We study two objects. We construct attacks and measure calibration on OPEN-JEV, an open replica of JEV that we host: a LoRA adapter and a scalar decision head on a pinned open backbone, with a saved calibration temperature. We then test the attack and the calibration finding on the hosted JEV service at a pinned version. Each serialized flow holds system-derived fields, which a flow meter computes from packet timing and sizes, and attacker-controllable fields, the application-layer strings an endpoint writes. The adversary edits only attacker-controllable text, using a fixed public phrase bank, and cannot touch the schema, the option set, or the model. We report every attack in a realizable regime that keeps the edited flow emittable [25], and in a feature-space upper bound that holds the numeric features fixed. Figure 1 shows the pipeline.

The first failure is calibration. Measured per class on a held-out network split, the replica’s confidence collapses on the rare attack classes an operator most needs to trust and stays better calibrated on the benign majority. The expected calibration error of the choice question is 0.278 under the shipped temperature and 0.145 after refitting a single temperature, and the remaining error sits on the attack classes, where a pooled error hides it (Section 4.2). The deployed model is no better calibrated on the shared yes/no question, and before any manipulation it detects only a minority of attacks, with recall 7.3% at the escalation threshold of 0.5.

The second failure is manipulation. Appending one short phrase from the bank to one attacker-controllable field drives a genuine attack below the escalation threshold, and the output remains schema-valid by construction. Under the realizable regime, the attack success rate (ASR) on escalated flows reaches 100.0% on the most affected attack class ($n = 118$ flows pooled over 3 seeds, 95% Wilson

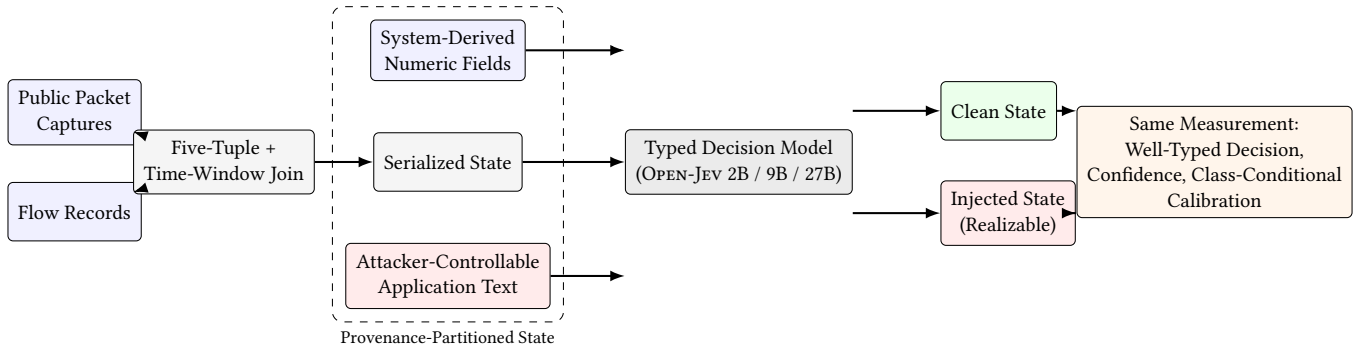


Figure 1: The same flow, read through one serializer, is scored clean and after injection into its attacker-controllable text, and both are measured on the typed output. The state separates system-derived numeric fields from attacker-controllable application text, which is the attack surface and the input-partitioning defense knob.

lower bound 96.8%). A best-of-bank search over the same phrase bank carries over to the hosted service and evades 100.0% of the 137 flows it escalates (lower bound 97.3%). The attack withdraws escalations that a triage queue would otherwise act on. A second, independently trained typed-decision family, Decision 2.0 [33], reproduces the evasion at each of its 6 sizes, and its malicious-question confidence is poorly calibrated (Section 4.6).

To our knowledge, this is the first study to audit the per-class, base-rate-aware calibration of a typed decision model used for intrusion detection and alert triage, and to show that a realizable injection into its attacker-controllable fields flips a well-typed decision. We build the attack on an open replica, test it on the deployed model, and corroborate it on a second family. We report the two findings side by side and do not claim that one causes the other. Prior calibration attacks perturb image pixels without changing labels [21], and prior injection attacks target free-text or tool-calling agents [10] and language-model judges [28]. None addresses a typed decision model serving as a detector. We make four contributions.

- We define a threat model that splits the serialized state into system-derived and attacker-controllable fields and bounds what the adversary can edit (Sections 2 and 3).
- We audit calibration per class and under operational base rates, and show that confidence collapses on rare attack classes and that zero-shot detection degrades on a second intrusion corpus and on real analyst verdicts (Section 4.4).
- We show that editing only application-layer strings flips well-typed decisions and transfers to the hosted service, across replica sizes, and across architectures, and that the built-in explanation lags the flipped decision (Sections 4.3 and 4.7).
- We show that of three inference-time defenses only input partitioning reduces ASR, at a cost in clean utility. (Section 4.8).

2 Problem Setting and Threat Model

2.1 Deployment scenario

A typed decision model sits in a detection pipeline as a high-volume classifier. It reads a serialized record of network telemetry, the

state, and answers three fixed questions: a yes/no question (is the flow malicious), a choice question (which attack class), and a score question (how severe for triage). The choice and score answers carry a confidence. An automated stage escalates or drops each flow using the malicious probability and a severity threshold. An analyst reads the severity and confidence when triaging escalated alerts. Both consumers treat the confidence as if it were the probability that the answer is correct.

2.2 Assets and adversary goal

The asset is the correctness of the decision at the point where the pipeline acts on it. The adversary controls a source of traffic that a detector observes, such as a scanning host, a brute-force client, or a web-application attack tool. The adversary wants one of two failures, each of which leaves the typed output schema-valid. Evasion drives the malicious probability of a genuine attack below the escalation threshold, or moves the class choice to benign, and the automated stage does not escalate the flow. Confidence suppression keeps the class choice correct but drives the confidence below the level at which an analyst would escalate. The vendor documents confidence suppression. We define it for completeness and measure evasion only.

2.3 Adversary knowledge and capability

The state holds two kinds of fields. *System-derived* fields are the numeric flow statistics a flow meter computes from packet timing and sizes. *Attacker-controllable* fields are the application-layer strings an endpoint writes directly: the HTTP method, host, request target, user-agent, and referrer; the FTP user, command, and argument; the SSH client version; the TLS server name; and the DNS query. In real traffic the adversary sets these strings at will.

The adversary edits text only inside the attacker-controllable fields. The schema, the option set, the answer types, and the flow meter are outside its control. Protocol rules, timing, and cross-field dependencies constrain real traffic, and the numeric features are computed from the same packets that carry the string. A text edit can change numeric features too. We report every attack under two regimes. In the *realizable regime*, each text edit is propagated to the numeric features it changes, so that the edited record keeps

a problem-space preimage in the sense of Pierazzi et al. [25]: a flow an endpoint could emit. The *feature-space bound* edits the text and holds the numeric features fixed. It is an upper bound that leaves numeric side effects unmodeled. The headline claims use the realizable regime.

2.4 Access settings

In the full-access setting the adversary reads the decision-head scores of the replica we host. This setting bounds what the phrase bank can achieve. No attack uses gradients. In the transfer setting the adversary searches the same bank against another target, sees only the typed outputs and confidences, and has a bounded query budget. This setting measures how much of the full-access effect survives.

2.5 Out of scope and measurement posture

The adversary cannot change the model weights, the adapter, the decision head, the calibration temperature, the schema, or the option set. It cannot inject packets into another host’s flow or change fields the endpoint does not write. Attacks that modify the training data, the serialization code, or the pipeline configuration are out of scope.

We construct every attack, and measure calibration, on open weights we host. These frozen replicas carry the headline results and make them reproducible. We also query the hosted service, only with benign functional requests that, in our reading, its terms of service permit (Section 8). We submit our own records, including records carrying crafted attacker-controllable text, and observe the typed outputs. These queries test, at a single point in time, whether the attack transfers to the deployed model and whether that model is calibrated on operational data. The interaction is a bounded best-of-bank search over the flows the service scores above the escalation threshold, at normal query rates. Section 8 states how the hosted outputs are retained and used.

3 Method

3.1 Targets

A typed decision model answers typed questions about a block of state in one non-autoregressive pass. It returns a choice, a yes/no probability, or an ordered score, and the choice and score carry a confidence. It generates no text, and its output is always schema-valid. The vendor documents that steering text can move the answer and that the confidence is a margin (Section 1).

The reproducible basis is OPEN-JEV, an open replica of this interface: a LoRA adapter and a trained scalar decision head on a pinned open backbone, with a saved calibration temperature. We study the 2B, 9B, and 27B replicas. The 9B replica is the primary full-access target; the 2B and 27B replicas are transfer targets. The weights are frozen and released (Appendix A). The released adapter, head, and temperature were trained on a generic non-network decision mixture and contain no network-intrusion data, which we confirmed by inspecting the released training manifest.

The hosted target is TypeSafe’s JEV [32], version 1.13, queried as a black box under the posture of Section 2. We pin the version and record the served revision on every call. Every recorded call resolved to the revision dated 17 September 2026.

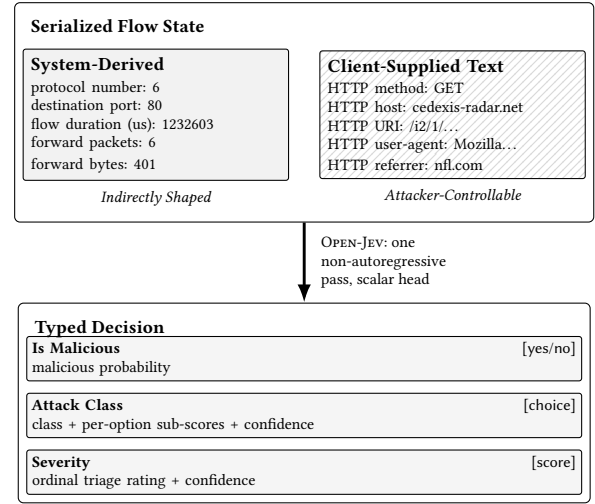


Figure 2: Input and output of the typed decision model. Hatching marks the attacker-controllable block; the choice and score answers add a confidence.

Two further open models are cross-architecture transfer targets (Section 4.3): Laya, a ModernBERT encoder [6], and SemIf, a training-free logit reader over a stock open backbone with no trained decision head. Both are weak zero-shot detectors. Transfer to them shows that manipulability requires neither the decoder-with-head architecture nor a trained decision head; it does not separate the effect of decision training from that of the backbone. A second family, the Decision 2.0 ladder [33] of 6 sizes from 0.6B to 27B, shares the decoder-with-head interface but comes from a separate training pipeline (Section 4.6).

3.2 From flow to typed decision

Each labeled flow is serialized to state as two blocks (Figure 2). The first holds system-derived numeric features from the corrected flow records. The second holds the application-layer text recovered for the flow, under an explicit “client-supplied” heading, with the provenance of each field (Section 2) preserved. The serializer records the exact character span of every attacker-controllable field. That span is the surface the attack perturbs and the surface input partitioning removes. Both regimes read the same flow through one serializer, which keeps the comparison controlled.

The questions are fixed. The yes/no question asks whether the flow is malicious. The choice question assigns an attack class from a fixed option set; the same form is applied to analyst verdicts on the alert-triage corpus (Section 4.1). The score question rates triage severity on ordered levels. The option order is fixed and recorded so that the order-sensitivity control can permute it.

3.3 Attacks

Every attack appends a short clause from a fixed public phrase bank to one attacker-controllable field. The bank covers the three failure modes the vendor documents: a clause that argues for the benign class, an injected instruction that reframes the classification, and a fabricated context field that claims prior approval. The bank holds

no model-specific text. It is used unchanged against every target except the adversarially hardened replica. That replica trains on 6 of the 9 bank phrases and is tested on the held-out phrases (Section 4.8). Search is a bounded best-of-bank: for each flow, the target scores every pairing of a phrase with an attacker-controllable field once, and the strongest pairing is kept. The score is the decision-head output on the replica and the returned malicious probability on the hosted service. No gradient is used. In the realizable regime, the appended bytes are propagated to the dependent numeric features: the forward byte totals, the forward mean packet length, the average and forward segment packet sizes, and the flow byte rate.

3.4 Calibration design

The original temperature was fit on non-network data. The refit multiplies the original log-probabilities by one scalar a , which gives an effective temperature of T/a for shipped temperature T . It is fit per question type on a held-out network calibration split and evaluated on a disjoint test split; for the choice question, $a = 0.650$. The design follows the temperature-scaling and classwise-calibration literature [11, 16]. The stratified evaluation sample is balanced for attack-surface work and does not match the deployed base rate. We report each operational quantity per class, as an aggregate reweighted by each flow’s inverse sampling weight, and on a separate random sample that preserves the base rate. ECE is reported per class because a pooled figure hides a collapse on the rare attack classes. Section 4.2 defines the calibration metrics; Appendix C gives the reliability diagram.

3.5 Explanation faithfulness

The choice question splits a decision into per-option sub-scores, which the vendor presents as its built-in interpretable readout. We test whether this readout stays consistent with the malicious decision it accompanies, using the faithfulness criteria of Warnecke et al. [34]. On flows that the attack moves from escalated to cleared, we compare the direction and size of the change in the benign sub-score with the change in the malicious probability. We report the fraction whose choice readout still names an attack class after the yes/no question has cleared the flow. We also report how often the two typed heads that read the shared decision score disagree on the hard label, tested clean against attacked (Section 4.2).

3.6 Defenses

We evaluate three inference-time defenses against the realizable attack. Input partitioning removes the attacker-controllable text block from the state. Option-order permutation reorders the choice options per request. Threshold lowering moves the escalation threshold from 0.5 to 0.3 to escalate more flows. Each defense is scored by its reduction in ASR against its cost in clean utility (Section 4.2).

We also run two training-time probes on the 9B replica. Fine-tuning trains the replica on network labels. Adversarial hardening builds an intrusion detector on the frozen backbone: a new low-rank adapter and scalar head, trained on clean and attacked flows, with public BoolQ questions and synthetic records as a forgetting control. Both train on the Monday through Wednesday traffic in CIC-IDS2017. Seen-family results use evaluation flows from the

training days. Held-out results use the Thursday and Friday traffic. Fine-tuning uses 3 seeds and hardening uses 3 (Section 4.8).

4 Evaluation

We answer four questions. RQ1: can an adversary who controls only the attacker-controllable fields flip a well-typed decision under a realizable perturbation, and does the attack transfer? RQ2: how calibrated is the model under class imbalance, against a refit baseline? RQ3: when the injection flips the malicious decision, does the built-in explanation move with it? RQ4: which defenses reduce attack success, and at what cost in clean utility?

4.1 Setup

The primary corpus is the corrected CICIDS2017 flow records [7, 17], which repair known labeling and flow-construction errors of the original release. A passive analyzer recovers application-layer text from the public captures. Each text record joins a flow by five-tuple when the connection’s active window contains the flow start. On a miss the text is null, and the match status is recorded per flow. The stratified sample places completed attacks in the positive class. Payload-free “attempted” flows form a separate stratum. Completed web attack and completed infiltration are small classes and are reported as underpowered. The attack applies only to flows that carry an attacker-controllable field. No infiltration or Heartbleed flow carries one, and ASR is undefined for both.

GUIDE supplies real triage verdicts [8]. An analyst grades each incident once as a confirmed true positive, a benign positive, or a false positive. Each incident holds several correlated evidence records serialized from alert metadata, with no attacker-controllable text block. The triage question is a three-way choice over these verdicts. We wrote one fixed description per option and release them. The analyst grade is excluded from the serialized state.

The decision model is the OPEN-JEV replica at 2B, 9B, and 27B parameters on pinned backbones. The two tabular baselines are a gradient-boosted tree and a random forest on the numeric feature vector, each trained under a temporal and a stratified split. A tree ensemble is the appropriate baseline for tabular flow data [3]. Two models serve as cross-architecture transfer targets. Laya is an independently released ModernBERT encoder [6]. SemIf is a training-free logit reader over a stock open backbone with no trained decision head. Stochastic results are repeated over 3 evaluation-sample seeds unless stated otherwise. All seeds, revisions, and configurations are released.

4.2 Metrics

Escalation and attack success rate. The detector escalates a flow when its malicious probability is at least 0.5, the escalation threshold. An analyst sees only escalated flows. We call a malicious flow *clean-flagged* when the model escalates it before any attack. Attack success rate (ASR) is the fraction of clean-flagged flows that fall below the threshold after the attack. The attack is the best-of-bank search of Section 3. A flow the model never escalates reaches no analyst and has no decision to flip. ASR measures how many true alerts an attacker can withdraw from the queue. Lower is better for the defender. We report the mean and standard deviation over

seeds and a Wilson lower bound, which gives a conservative floor when few flows are escalated.

Recall at a fixed false-positive rate. A security operations center can absorb only a fixed volume of false alarms. We report recall, the fraction of malicious flows escalated, at fixed false-positive rates of 1% and 0.1%. The threshold that meets each budget is set on benign test flows. It states how many attacks are caught at an alarm load the operator can absorb. Higher is better.

Calibration error. A confidence value is calibrated when, among decisions made with confidence p , a fraction p are correct. Analysts and automated stages read confidence as this probability. A miscalibrated detector leads them to trust wrong answers or discard right ones. Expected calibration error (ECE) takes the confidence of the predicted answer (top-label confidence), groups predictions into 15 equal-width confidence bins, and averages the gap between mean confidence and accuracy, weighted by bin size. Maximum calibration error (MCE) is the largest such gap. Both range from zero to one, and lower is better. An ECE of e means stated confidence and observed accuracy differ by about $100e$ percentage points on average. A pooled ECE is dominated by the benign majority. It can look acceptable while every rare attack class is badly miscalibrated. We report ECE per class, under the shipped temperature and under the refit of Section 3.

Base-rate projection. The attack experiments use a sample balanced across attack classes. Deployed traffic is overwhelmingly benign, and a rate read off a balanced sample misstates the false-alarm load. Each sampled flow carries an inverse sampling weight. Reweighting by it projects a rate to the class frequencies of the full corpus. We confirm each projected figure on a separate sample that preserves the corpus base rate.

Clean utility. Clean utility is detection accuracy on one fixed set of unattacked flows. It is reported as a fraction of flows, and a defense’s cost is the drop in that fraction. It is the cost side of a defense. A defense that lowers ASR and also lowers clean utility trades evasion for missed detections. Higher is better.

Significance. When two conditions score the same flows, such as clean against attacked, we use a paired McNemar test. It counts only the flows whose decision differs between the two conditions and asks whether the changes favor one direction. Paired bootstrap resampling of flows or incidents gives confidence intervals for other differences. A result is called significant only when its test rejects at the stated level.

Cost of operation. Throughput is the number of flows scored per second. Latency is the time to score a single flow. Peak memory is the largest accelerator memory footprint during scoring. Throughput decides whether a model keeps up with a link. Latency decides whether it can sit inline. The cost column gives the cost model for scoring one million flows. Open models are self-hosted, and their cost is the accelerator time implied by throughput. The hosted service is metered per call at a provider-set rate, and we report no fixed figure for it. Table 3 reports them where they can be measured.

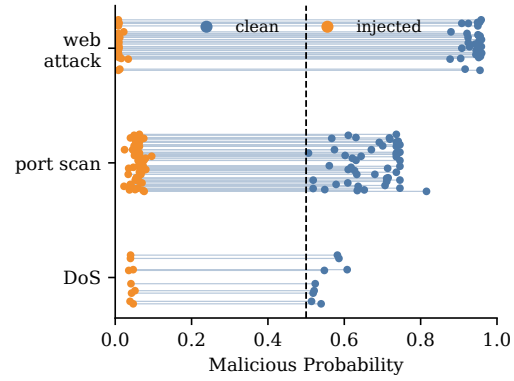


Figure 3: Malicious probability of each flow the 9B replica flags on the clean input, before and after one phrase-bank clause is appended to a single attacker-controllable field. The vertical line is the escalation threshold of 0.5.

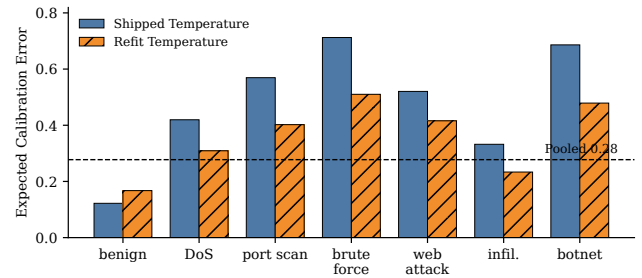


Figure 4: Per-class expected calibration error of the 9B choice question under the shipped and the refit temperature. The dashed line is the pooled error.

4.3 RQ1: State injection flips well-typed decisions

Finding. A short clause appended to one attacker-controllable field pulls escalated attacks below the threshold. On port scan flows the realizable attack reaches an ASR of 100.0% (± 0.0 pp), and every typed output stays schema-valid. Pooled over the 118 distinct flows escalated in at least one seed, the Wilson lower bound is 96.8%. The feature-space upper bound is 100.0%. Figure 3 shows the shift, and Table 4 in Appendix B gives per-class rates.

The attack transfers without per-model tuning. On a single-seed subsample, the same phrase bank reaches an ASR of 100.0% on the web attack flows each of the 2B, 9B, and 27B replicas escalates (68, 31, and 31 flows). It also reaches 100.0% on Laya (84 flows) and 100.0% on SemIf (33 flows). Both read the same serialized text. The manipulability requires neither a decoder with a scalar head nor a trained decision head. Transfer to the hosted service is reported in Section 4.5.

Table 1: Calibration of the 9B choice question: per-class ECE under the shipped and a refit temperature. †: too few flows to estimate. Pooled and base-rate rows aggregate all classes. Lower block: top-label ECE of the tree ensembles under stratified and temporal splits; seeds are stratified/temporal; multi-seed entries are means.

Class	<i>n</i>	ECE shipped	ECE refit
benign	3000	0.122	0.167
dos	600	0.420	0.309
portscan	600	0.569	0.402
bruteforce	362	0.712	0.510
web attack	52	0.520	0.416
infiltration	18	0.332	0.233
botnet	360	0.686	0.479
heartbleed†	6	n/a	n/a
pooled	4998	0.278	0.145
base-rate proj.		0.185	
<i>Tabular baselines, top-label ECE</i>			
Model	seeds	stratified	temporal
XGBoost	5/1	0.001	0.445
Random forest	5/1	0.008	0.421

4.4 RQ2: Calibration collapses on the rare classes

Finding. Refitting the temperature lowers pooled ECE, but the residual error sits on the rare attack classes an operator most needs to trust. On the choice question, pooled ECE falls from 0.278 under the shipped temperature to 0.145 after the refit. The refit lowers attack-class error and raises benign error (Figure 4; Table 1). The base-rate-projected aggregate is lower than the shipped pooled figure because benign flows dominate it. A miscalibrated attack flow still costs the analyst the same, whatever its share of traffic.

Tabular baselines. The tree ensemble calibrates well under a stratified split and fails under a temporal split (Table 1). The temporal split is a novel-class split: the Thursday and Friday attack classes never appear in training. The destination port is a known spurious feature in these datasets [14]. Dropping it leaves held-out recall at 38.9%, against 37.3% with it. The port does not explain the recall gap between the two splits.

Cross-dataset. Zero-shot detection of the 9B replica collapses on TON-IoT [18], a corpus it was never adapted to. Malicious recall at the escalation threshold falls from 27.6% on CICIDS2017 to 5.7%, and ECE on the malicious question worsens from 0.174 to 0.481. Each rate is measured on that corpus’s own mixture. This is a single zero-shot estimate on one further corpus, without an interval. It agrees with the expectation that predictive uncertainty degrades under distribution shift [22].

Real verdicts. On alert metadata, the replica’s triage choice does not track the analyst. We score 931 GUIDE incidents from a verdict-stratified subsample, averaging option probabilities over each incident’s evidence records. Incident-level accuracy is 18.0% (95% interval 15.7% to 20.7%), below the 33.3% uniform-chance rate. The decision collapses onto one verdict. The replica returns the confirmed true positive grade for 81.1% of incidents, and its MCE is

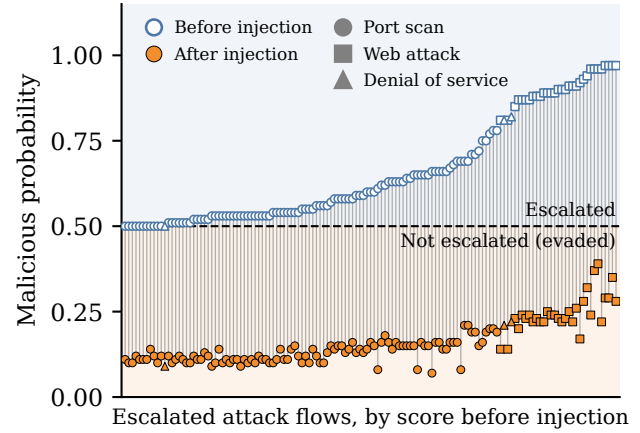


Figure 5: Malicious probability assigned by the hosted Jev service to the 137 attack flows it escalates without manipulation, before and after injection. Flows run left to right in order of their score before injection. For each flow, a hollow marker gives the score before injection and a filled marker on the same vertical line gives the score after injection; marker shape gives the attack class. A filled marker below the dashed escalation threshold (0.5) is a flow the service no longer escalates.

0.627. This result rests on one checkpoint, one subsample, and one authored option set and order. It may be sensitive to the option wording.

4.5 Validation on the hosted service

Finding. The attack built on the replica transfers to the deployed model, which is no better calibrated. We queried the hosted Jev service at version 1.13 (revision dated 17 September 2026) through benign functional requests, under the posture of Section 2. These figures are point-in-time measurements of that revision. On a base-rate-preserving sample the service escalates few attacks before any manipulation: at the escalation threshold of 0.5, recall is 7.3% and the false-positive rate is 0.13% (Figure 6). On the malicious question its ECE is 0.215, against 0.174 on the replica. Of the 137 flows the service escalates, the best-of-bank search of Section 3 reaches an ASR of 100.0% (Wilson lower bound 97.3%). Figure 5 shows the escalation boundary.

4.6 A second typed-decision family

Finding. An independently trained typed-decision family reproduces the evasion at each of its 6 sizes, with malicious-question ECE from 0.16 to 0.32. Decision 2.0 [33] is an open family on a Qwen backbone with a trained decision head. We score all sizes, spanning 0.6B to 27B, zero-shot on the same evaluation slice at pinned revisions. The realizable injection reaches an ASR of 100.0% on every size. Recall at a false-positive rate of 1% stays between 1.5% and 7.5% across sizes (Table 3). We neither trained nor tuned this family. Its agreement with the replica and the hosted service

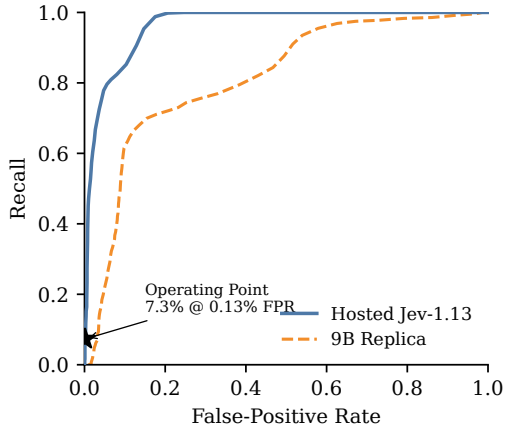


Figure 6: Recall against false-positive rate as the malicious-probability threshold sweeps, on a base-rate-preserving sample, for the hosted service and the 9B replica. The star marks the operating point at the escalation threshold of 0.5.

is consistent with these failures belonging to the typed-decision interface.

4.7 RQ3: The built-in explanation lags the flipped decision

Finding. When the injection clears a flow, the built-in explanation moves the same way but by less. The two typed questions read one shared decision score: the yes/no malicious question and the choice among attack classes. The vendor presents the choice decomposition as its rationale. Agreement between the two is a minimum requirement for that rationale to be faithful. We follow the prediction-interpretation gap of Zhang et al. [36] and the faithfulness criteria of Warnecke et al. [34], within the evaluation of explanations under attack [19, 20].

Over 3 seeds, on the 73 to 78 flows per seed that flip from escalated to cleared, the benign sub-score rises by less than the malicious probability falls in 91.3% (± 3.8 pp) of cases. The choice question still names an attack class in 29.5% (± 4.4 pp) of them. Only port scan and web attack have at least twenty flipped flows. The lag holds in both. The hard choice label differs: for web attack it follows the decision to benign, and for port scan it keeps an attack class on 61.0% of flipped flows on the base seed.

The hard-label disagreement rate between the two questions does not shift under attack: a paired McNemar test finds no significant change in 3 of 3 seeds. The attack shows in the direction and size of the sub-scores.

4.8 RQ4: Defenses and the utility trade-off

Finding. Of three inference-time defenses on the 9B replica, only input partitioning lowers ASR, from 100.0% to 50.9%, and it costs clean utility. Lowering the escalation threshold and permuting the option order leave ASR unchanged (Table 2; Figure 7). Each defense is a single run. The realizable injection acts through two channels: the client-supplied strings and the byte-count features

Table 2: Defenses against the realizable injection attack on the 9B replica: residual attack success, clean utility, and the utility cost relative to no defense. Residual attack success is over the flows each defended replica flags before manipulation. Clean utility is detection accuracy on one fixed set of flows.

Defense	ASR (%)	clean util.	util. cost
none	100.0	0.550	0.000
input partitioning	50.9	0.463	0.088
threshold lowering	100.0	0.568	-0.017
option-order permutation	100.0	0.550	0.000

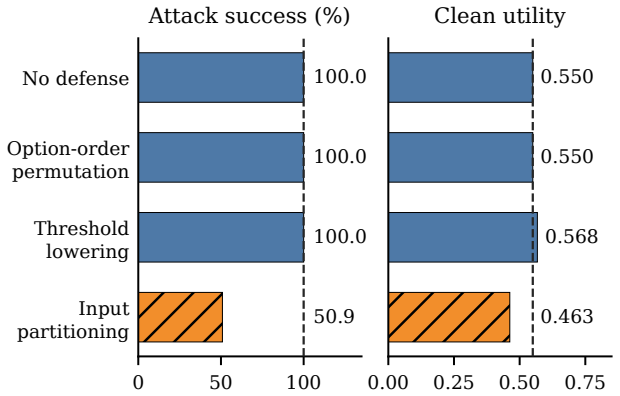


Figure 7: Defenses on the 9B replica against the realizable injection attack. Left: attack success (%) over flows each defended replica flags before manipulation. Right: clean utility, detection accuracy on the same clean flows for every defense. Dashed lines mark no-defense values. Hatched orange bars mark defenses that lower attack success.

the appended bytes change. Partitioning removes the strings but keeps the numeric block; 50.9% of the flagged flows are still evaded. Removing them also lowers clean utility by 0.088. A threshold of 0.3 flags more flows before manipulation and still loses every one of them. The malicious question is a yes/no question with no option order to permute (Section 3), and permutation leaves ASR unchanged.

Fine-tuning. Domain adaptation improves detection without closing the attack. We fine-tune the 9B replica on the Monday through Wednesday flows and test on the Thursday and Friday population across 3 fine-tuning seeds. Mean recall, ECE, and false-positive rate rise, the last by a large and variable amount (Figure 8). A paired McNemar test on evasion finds a significant ASR reduction in only 1 of the 3 seeds, and that seed has the highest false-positive rate.

Hardening. Training covers the Monday through Wednesday traffic. Its attack families, brute force and denial of service, are seen in training. The Thursday and Friday families are held out. They include Friday distributed denial of service (DDoS), a separate capture from Wednesday denial of service. Training against the injection closes the evasions on held-out web attack and botnet. It also closes

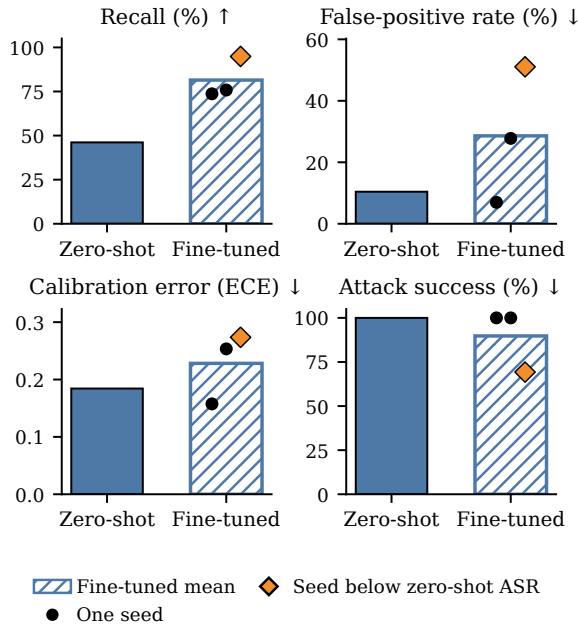


Figure 8: Zero-shot (solid) and fine-tuned (hatched, seed mean) 9B replica; dots: single seeds. Orange diamond: seed whose attack success (ASR) falls below zero-shot. Recall, false-positive rate (escalation threshold), and calibration error (ECE) use clean flows. ASR counts flows flagged before the realizable injection attack. Title arrows show the better direction.

them on seen brute force and denial of service. Held-out port scan and DDoS stay open. We build the hardened detector of Section 3 over 3 seeds. Its injected training flows include benign flows carrying the clause with a benign label, and no hosted-service output enters training. ASR falls to 0.0% (± 0.0 pp) on held-out web attack and to 0.0% (± 0.0 pp) on held-out botnet. Seen brute force falls to 0.0% (± 0.0 pp) and seen denial of service to 0.0% (± 0.0 pp). They are measured on evaluation flows from the training days. These seen results measure resistance on families present in training, not generalization. Held-out port scan retains 51.1% (± 22.0 pp). On held-out Friday DDoS, only 2 of 3 seeds flag any flow, and the injection evades 100.0% (± 0.0 pp) of the 56 flagged flows. Clean recall is low on the two classes the attack still evades: 15.0% (± 3.6 pp) on port scan and 2.1% (± 3.7 pp) on held-out DDoS. Pooled held-out recall at a false-positive rate of 1%, projected to deployment base rates, is 12.4% (± 3.5 pp).

5 Related Work

Calibration and attacks on confidence. Temperature scaling and its multiclass successors calibrate neural classifiers and define the reliability-diagram and ECE machinery we use [11, 16], and predictive uncertainty degrades under dataset shift [22]. Obadinma et al. [21] show that an adversary can miscalibrate an image classifier’s confidence through pixel perturbations without changing

its predicted label. Our target is a typed, non-autoregressive decision model serving as an intrusion-detection and alert-triage classifier. We measure clean calibration with class-conditional, base-rate-aware metrics that a pooled ECE conceals, and our edit is a problem-space-realizable change to attacker-controllable fields that flips a well-typed decision.

Adversarial machine learning and the problem space. Adversarial perturbation of neural classifiers [9, 31] and its decade in security [4] frame the attack. Pierazzi et al. [25] formalize the constraints that make a feature-space perturbation realizable as a real object, the standard our realizable regime meets. Apruzzese et al. [2] argue that practical attackers rarely compute gradients, which motivates our black-box transfer setting and bounded query budget.

Typed, non-autoregressive decision models. Models of this class answer typed schema questions in one non-autoregressive pass and report a probability rather than free text. Rafe and Das [26] use a System One model to convert police crash narratives into probabilistic crash variables. Open members of the class include a model that recovers decisions from a hidden-state readout [5], an encoder that scores option markers [6], and an independently trained decoder-with-head ladder [33]. The closest concurrent study is that of Sun and Xu [30], who show on generic workflow decisions that a constrained decision head follows the surface name of an answer option rather than the rubric bound to it. We study a different surface: realizable edits to attacker-controllable payload, with the serialized state partitioned by provenance, in intrusion detection and alert triage, together with per-class calibration. This work is concurrent with and complementary to ours, and we claim no priority over it.

Injection into structured and judging models. Indirect prompt injection plants instructions in content a model later reads [10], and optimization-based injection can force the verdict of a language-model judge through one controlled field [28]. These attacks target free-text or tool-calling agents and judges. We target a schema-constrained decision interface, where the serialized state the model treats as data becomes the injection surface. Readouts that recover a decision from allowed-answer logits are sensitive to option order [24, 37], which motivates our order-sensitivity control and the option-order permutation defense.

Explanations for security classifiers. Multi-criteria evaluation of security explanations [19, 34], fidelity-oriented explainers [12, 13], and explanation frameworks for deep-learning intrusion detection [35] frame the faithfulness question we put to the built-in decomposition. Kalakoti et al. [15] study explanations in the closest deployment setting, alert triage; we add an attacker who controls the input.

Methodology and datasets. We follow the pitfall checklist of Arp et al. [3] and the experiment-design practices of Rossow et al. [27], apply a temporal split to the tabular baselines as in TESSER-ACT [23], and frame false-positive cost as in Sommer and Paxson [29] and Alahmadi et al. [1]. We use the corrected CICIDS2017 flow records of Engelen et al. [7] and Liu et al. [17]. Trustee exposes a destination-port shortcut in published intrusion models [14]; our port-dropped control is consistent with this shortcut not being

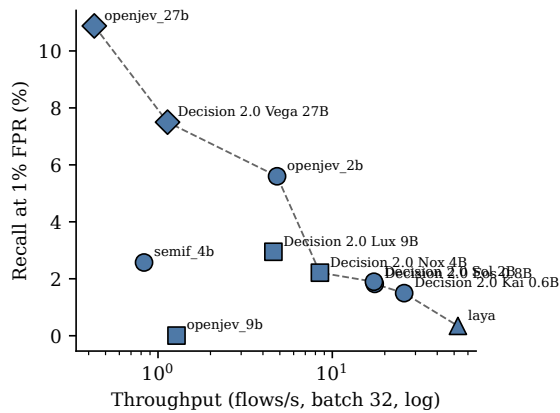


Figure 9: Throughput (flows per second, log scale) against recall at a fixed 1% false-positive rate for the panel. Marker shape encodes size tier; the dashed line marks the Pareto frontier over the models with both measurements.

the primary driver of the gap between our stratified and temporal tabular baselines, which is a novel-class gap.

6 Discussion and Limitations

Implications for deployers. A deployer of zero-shot models cannot buy out of these failures with scale. Table 3 arranges the panel by size tier, and Figure 9 places throughput against recall at a false-positive rate of 1%. Every zero-shot model that exposes a typed injection surface is evaded on nearly all of its flagged flows, poorly calibrated on the malicious question, and weak at a fixed false-positive rate, and no size tier removes these failures. At 1%, the 9B replica has 0.0% recall and the hosted service 11.9%. The same pattern holds in a second, independently trained family (Section 4.6). Three practices follow. A deployer should treat client-supplied text as attacker input and keep it out of the scored state where clean utility allows (Section 4.8). It should refit the temperature on its own network and read calibration per class, not pooled (Section 4.4). It should not read the choice decomposition as a faithful rationale for the escalation decision (Section 4.7). The tabular incumbents are supervised on the training days; their recall is not comparable to the zero-shot decision models.

The replica and the hosted service. Attacks are built and calibration is measured on OPEN-JEV. Its weights are fixed (Section 3, Appendix A). The hosted service is queried to test whether the attack transfers and whether the deployed model is calibrated on operational data (Section 4.5). The replica, not the hosted service, is the reproducible basis.

Dataset scope. The core evaluation uses one flow-dataset lineage. The corrected CICIDS2017 records remove known labeling and flow-construction errors, but one traffic generator is not the open Internet. Completed web attack and completed infiltration are small classes and are reported as underpowered. Each attack class except denial of service occurs on a single capture day. The day split thus

tests novel classes rather than a distribution shift. The destination-port shortcut [14] does not explain the tabular recall gap between the two splits (Section 4.4). The cross-dataset and GUIDE results each rest on one further corpus.

Realizability. The realizable regime propagates each text perturbation to its first-order dependent numeric features. It does not re-derive packet-level statistics from a re-synthesized capture, and second-order effects on inter-arrival timing and packet counts are not modeled. The feature-space regime bounds the effect from above and is labeled as such wherever it appears.

Base rates. Projected rates assume the class frequencies of the corrected corpus. A network with a different attack mix would see a different false-alarm load at the same threshold.

Adaptive adversaries. The defenses are measured against the attack family defined here. An adversary who adapts to a specific defense is stronger than the one modeled, and no defense result should be read as robustness to that adversary.

7 Conclusion

Type-safety guarantees the form of a typed decision model’s answer and says nothing about its correctness. Used for intrusion detection and alert triage, the model’s confidence collapses on the rare attack classes that a pooled error hides, and an adversary who writes only the application-layer strings of their own flow can flip a well-typed decision on the replica, on the hosted service, and on a second model family. A deployer should keep attacker-controllable text out of the decision state and accept the clean-utility cost (Section 4.8), recalibrate per class, and judge detection by recall at a fixed false-positive rate.

8 Ethics Considerations

Attack construction and calibration measurement run on OPEN-JEV, an open replica whose frozen weights we host (Appendix A). We also query the hosted JEV service with benign functional requests: we submit our own network records, including records carrying crafted attacker-controllable text, for classification and observe the typed outputs. In our reading of the vendor’s master customer agreement (<https://typesafe.ai/legal/mca>, last updated 23 September 2026), these queries are permitted functional use rather than security testing. The agreement scopes its security- and vulnerability-testing restriction to the circumvention of access restrictions and protection mechanisms, and states no benchmarking or publication restriction. This is our interpretation of the terms, not an adjudicated fact. We send no abnormal or denial-of-service load and circumvent no access restriction or protection mechanism.

We honor the obligations that attach to that use. We submitted 45,628 flow records to the hosted service at revision jev-1.13-20260917 and kept the returned outputs as a local auxiliary label set. No reported model is trained on these labels, and they are not released. The agreement assigns these outputs to the customer and bars only distillation, imitation training, and building a competing product. The refit temperature, the fine-tuning probe, and the hardening adapter and head are the only components fit on network labels, and none uses hosted output. Requests stay within documented usage limits at normal rates, and we claim no endorsement

Table 3: Size-tiered panel. Flows/s at batch 32; latency in ms at batch 1. Recall projected to deployment base rates at 1% and 0.1% false-positive rates. ECE on the malicious question over 15 bins. Realizable ASR on flagged flows. Keys: ^s no injection surface; ⁿ hosted, internals undisclosed; ^h CPU; ^t top-label multiclass ECE; ^u supervised; ^d two GPUs, VRAM per card.

Model	Params	VRAM	Flows/s	Lat. p50/p99	Cost/1M	Rec@1%	Rec@0.1%	ECE _{mal}	ASR	Interpretability
<i>Encoder (~400M)</i>										
Laya (ModernBERT enc.)	421M	3.5	52.5	36/37	self-host	0.3	0.1	0.162	100.0	[MASK]-option scores
<i>Small decoder (2-4B)</i>										
OPEN-JEV 2B (zero-shot)	2B	14.6	4.8	122/170	self-host	5.6	0.2	0.255	100.0	scalar head + Choice
SemLf-4B (training-free)	4B	40.7	0.8	324/447	self-host	2.6	0.2	0.252	100.0	logit reader (Yes/No)
AR decoder-LLM (baseline)	2B	3.8	1.1	902/946	self-host	n/a ^s	n/a ^s	n/a ^s	n/a ^s	free-text (no typed head)
<i>Mid decoder (9B)</i>										
OPEN-JEV 9B (zero-shot)	9B	34.8	1.3	583/792	self-host	0.0	0.0	0.174	100.0	scalar head + Choice
<i>Large decoder (27B)</i>										
OPEN-JEV 27B (zero-shot)	27B	41.9 ^d	0.4	1806/2473	self-host	10.9	4.6	0.250	100.0	scalar head + Choice
<i>Open Decision 2.0 ladder (independently trained typed-decision family)</i>										
Decision 2.0 Kai 0.6B	0.6B	1.5	25.8	39/44	self-host	1.5	0.0	0.164	100.0	decision head + options
Decision 2.0 Eos 0.8B	0.8B	2.0	17.5	58/65	self-host	1.8	0.2	0.319	100.0	decision head + options
Decision 2.0 Sol 2B	2B	4.6	17.3	59/65	self-host	1.9	0.9	0.318	100.0	decision head + options
Decision 2.0 Nox 4B	4B	9.4	8.5	112/131	self-host	2.2	0.0	0.212	100.0	decision head + options
Decision 2.0 Lux 9B	9B	17.0	4.6	202/242	self-host	2.9	0.2	0.208	100.0	decision head + options
Decision 2.0 Vega 27B	27B	32.0 ^d	1.1	838/1004	self-host	7.5	5.0	0.212	100.0	decision head + options
<i>Tabular incumbent (CPU, numeric-only)</i>										
XGBoost	400 trees	n/a ^h	43349.4	0/1	self-host	37.3 ^u	29.9 ^u	0.445 ^{t,u}	n/a ^s	feature importances
Random forest	400 trees	n/a ^h	693.6	46/55	self-host	36.2 ^u	34.9 ^u	0.421 ^{t,u}	n/a ^s	feature importances
<i>Black-box reference</i>										
Closed Jev-1.13 (hosted)	n/a ⁿ	n/a ⁿ	n/a ⁿ	n/a ⁿ	metered	11.9	3.4	0.215	100.0	typed API, no internals

by the vendor. The submitted payloads and returned outputs are retained privately (Appendix A).

The paper discloses no previously unknown exploitable flaw in the deployed model. The vendor’s own limitations page [32] already documents, at a high level, that text written into the model’s input can move a typed decision. We quantify that effect under operational conditions, on the replica and on the deployed model, and measure defenses. Any finding we communicate to the vendor follows coordinated disclosure.

The study involves no human subjects and no personal data. All experiments use public research datasets of network flows and security-operations alert metadata, together with inputs we craft from them. The state submitted to any model excludes raw network addresses, and no traffic is collected from real users.

The released artifact contains the phrase bank and the measurement and defense code. Releasing it lowers the bar to reproduce this manipulation. We judge that cost outweighed by the defensive value of a reproducible measurement and of the defense evaluation it enables, which a deployer needs to assess and mitigate the risk on their own systems.

References

- [1] Bushra A. Alahmadi, Louise Axon, and Ivan Martinovic. 2022. 99% False Positives: A Qualitative Study of SOC Analysts’ Perspectives on Security Alarms. In *31st USENIX Security Symposium (USENIX Security 22)*. 2783–2800.
- [2] Giovanni Apruzzese, Hyrum S. Anderson, Savino Dambra, David Freeman, Fabio Pierazzi, and Kevin A. Roundy. 2023. “Real Attackers Don’t Compute Gradients”: Bridging the Gap Between Adversarial ML Research and Practice. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. 339–364.
- [3] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2022. Dos and Don’ts of Machine Learning in Computer Security. In *31st USENIX Security Symposium (USENIX Security 22)*. 3971–3988.
- [4] Battista Biggio and Fabio Roli. 2018. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. *Pattern Recognition* 84 (2018), 317–331. doi:10.1016/j.patcog.2018.07.023
- [5] Zehua Cheng, Wei Dai, and Jiahao Sun. 2026. this-that-model-1.0: A Typed Decision Model that Decides in 30 ms, for a Millionth of a Cent. arXiv:2609.23886.
- [6] Convai Innovations. 2026. Laya: A Non-Autoregressive Typed Decision Model. Model artifact, huggingface.co/convaiinnovations/laya; code github.com/NandhaKishorM/laya. 421M bidirectional ModernBERT-large encoder with an option-marker scoring head; Apache-2.0.
- [7] Gints Engelen, Vera Rimmer, and Wouter Joosen. 2021. Troubleshooting an Intrusion Detection Dataset: the CICIDS2017 Case Study. In *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE, 7–12. doi:10.1109/SPW53761.2021.00009
- [8] Scott Freitas, Jovan Kalajdjieski, Amir Gharib, and Robert McCann. 2024. AI-Driven Guided Response for Security Operation Centers with Microsoft Copilot for Security. *arXiv preprint arXiv:2407.09017* (2024). doi:10.48550/arXiv.2407.09017
- [9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. *International Conference on Learning Representations (ICLR)* (2015). arXiv:1412.6572.
- [10] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not What You’ve Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec)*. 79–90.
- [11] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. 1321–1330.
- [12] Wenbo Guo, Dongliang Mu, Jun Xu, Purui Su, Gang Wang, and Xinyu Xing. 2018. LEMNA: Explaining Deep Learning based Security Applications. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 364–379. doi:10.1145/3243734.3243792
- [13] Dongqi Han, Zhiliang Wang, Wenqi Chen, Ying Zhong, Su Wang, Han Zhang, Jiahai Yang, Xingang Shi, and Xia Yin. 2021. DeepAID: Interpreting and Improving Deep Learning-based Anomaly Detection in Security Applications. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 3197–3217. doi:10.1145/3460120.3484589
- [14] Arthur S. Jacobs, Roman Beltukov, Walter Willinger, Ronaldo A. Ferreira, Arpit Gupta, and Lisandro Z. Granville. 2022. AI/ML for Network Security: The Emperor has no Clothes. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 1537–1551. doi:10.1145/3548606.3560609

- [15] Rajesh Kalakoti, Risto Vaarandi, Hayretin Bahsi, and Sven Nömm. 2025. Evaluating Explainable AI for Deep Learning-Based Network Intrusion Detection System Alert Classification. In *Proceedings of the 11th International Conference on Information Systems Security and Privacy (ICISSP)*. arXiv:2506.07882.
- [16] Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. 2019. Beyond Temperature Scaling: Obtaining Well-calibrated Multi-class Probabilities with Dirichlet Calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32.
- [17] Lisa Liu, Gints Engelen, Timothy Lynar, Daryl Essam, and Wouter Joosen. 2022. Error Prevalence in NIDS Datasets: A Case Study on CIC-IDS-2017 and CSE-CIC-IDS-2018. In *2022 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 254–262.
- [18] Nour Moustafa. 2021. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON IoT datasets. *Sustainable Cities and Society* 72 (2021), 102994. doi:10.1016/j.scs.2021.102994
- [19] Azqa Nadeem, Daniel Vos, Clinton Cao, Luca Pajola, Simon Dieck, Robert Baumgartner, and Sicco Verwer. 2023. SoK: Explainable Machine Learning for Computer Security Applications. In *2023 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 221–240.
- [20] Maximilian Noppel and Christian Wressnegger. 2024. SoK: Explainable Machine Learning in Adversarial Environments. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE. doi:10.1109/SP54263.2024.00021
- [21] Stephen Obadinma, Xiaodan Zhu, and Hongyu Guo. 2024. Calibration Attacks: A Comprehensive Study of Adversarial Attacks on Model Confidence. *Transactions on Machine Learning Research (TMLR)* (2024). arXiv:2401.02718.
- [22] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32.
- [23] Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney, Johannes Kinder, and Lorenzo Cavallaro. 2019. TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time. In *28th USENIX Security Symposium (USENIX Security 19)*. 729–746.
- [24] Pouya Pezeshkpour and Estevam Hruschka. 2023. Large Language Models Sensitivity to the Order of Options in Multiple-Choice Questions. arXiv:2308.11483.
- [25] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. 2020. Intriguing Properties of Adversarial ML Attacks in the Problem Space. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1332–1349.
- [26] Amir Rafe and Subasish Das. 2026. Calibrated Decisions at Scale: Converting Police Crash Narratives into Probabilistic Crash Variables with a System One Model (Jev). arXiv:2609.24052.
- [27] Christian Rossow, Christian J. Dietrich, Chris Grier, Christian Kreibich, Vern Paxson, Norbert Pohlmann, Herbert Bos, and Maarten van Steen. 2012. Prudent Practices for Designing Malware Experiments: Status Quo and Outlook. In *2012 IEEE Symposium on Security and Privacy*. IEEE, 65–79.
- [28] Jiawen Shi, Zenghui Yuan, Yinyao Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. 2024. Optimization-based Prompt Injection Attack to LLM-as-a-Judge. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*.
- [29] Robin Sommer and Vern Paxson. 2010. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In *2010 IEEE Symposium on Security and Privacy*. IEEE, 305–316.
- [30] Yu Sun and Junhao Xu. 2026. Type-Safe Is Not Error-Free: A Constrained Decision Head Follows the Option Name, Not the Rubric Bound to It. arXiv:2609.26758.
- [31] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing Properties of Neural Networks. *International Conference on Learning Representations (ICLR)* (2014). arXiv:1312.6199.
- [32] TypeSafe AI. 2026. Introducing System One Models and Jev. Vendor documentation, docs.typesafe.ai. vendor-self-reported.
- [33] vLLM Semantic Router Team. 2026. Decision 2.0: Towards Open Foundation Decision Models. Model collection, huggingface.co/vllm-sr. Open typed-decision family on Qwen backbones with a trained decision head, sizes Kai 0.6B, Eos 0.8B, Sol 2B, Nox 4B, Lux 9B, Vega 27B; Apache-2.0.
- [34] Alexander Warnecke, Daniel Arp, Christian Wressnegger, and Konrad Rieck. 2020. Evaluating Explanation Methods for Deep Learning in Security. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 158–174. doi:10.1109/EuroSP48549.2020.00018
- [35] Feng Wei, Hongda Li, Ziming Zhao, and Hongxin Hu. 2023. xNIDS: Explaining Deep Learning-based Network Intrusion Detection Systems for Active Intrusion Responses. In *32nd USENIX Security Symposium (USENIX Security 23)*. 4337–4354.
- [36] Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. 2020. Interpretable Deep Learning under Fire. In *29th USENIX Security Symposium (USENIX Security 20)*. 1659–1676.
- [37] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large Language Models Are Not Robust Multiple Choice Selectors. In *The Twelfth International Conference on Learning Representations (ICLR)*. arXiv:2309.03882.

A Reproducibility and Configuration

All attack construction and calibration measurement use fixed open weights: the OPEN-JEV 2B, 9B, and 27B replicas on pinned backbone revisions, with the released adapter, decision head, and calibration temperature. The corrected CICIDS2017 records are joined to application-layer text that a passive analyzer recovers from the public packet captures, matched by five-tuple and time window. The artifact releases the sampling seeds, backbone revisions, library versions, and per-flow join status. The submitted payloads and the hosted-service outputs are retained privately and are available on request. The served revision is pinned as described in Section 3. The 27B models run model-parallel across two GPUs.

B Per-class attack success

Table 4: Evasion attack success per class on the 9B replica (base evaluation seed): the fraction of clean-flagged attacks pushed below the escalation threshold. Brackets are 95% Wilson intervals on the realizable rate. Well-typed is whether every attacked output stayed schema-valid.

Class	n	flagged	ASR real	ASR feat	typed
dos	100	9	1.00 [0.70,1.00]	1.00	yes
portscan	100	49	1.00 [0.93,1.00]	1.00	yes
bruteforce	100	0	n/a	n/a	yes
web attack	100	31	1.00 [0.89,1.00]	1.00	yes
botnet	100	0	n/a	n/a	yes

C Choice-question reliability

Figure 10 shows the reliability diagram behind the calibration collapse. Benign confidence tracks the diagonal; on attack traffic the choice question stays far below it across most of the confidence range.

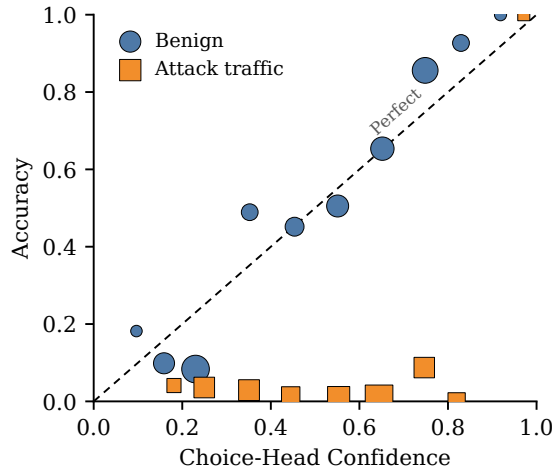


Figure 10: Reliability of the 9B choice question on the held-out split, benign against pooled attack traffic. Axes are confidence and accuracy over 15 bins; marker area encodes the number of flows in each bin; the diagonal is perfect calibration.

Table 5: Secondary measurements. ASR is realizable attack success on flagged flows; n is the number of flagged flows.

<i>Fine-tuning probe, 9B replica (fine-tuned / zero-shot; fine-tuned entries are mean $\pm$ std over seeds)</i>		
Accuracy	76.6% ($\pm$ 5.7 pp) / 67.2%	
Malicious recall	81.5% ($\pm$ 11.7 pp) / 46.1%	
False-positive rate	28.6% ($\pm$ 22.1 pp) / 10.4%	
False-positive rate, seed range	7.0% to 51.1%	
ECE	0.228 ($\pm$ 0.062) / 0.184	
Realizable ASR	89.8% ($\pm$ 17.7 pp) / 100.0%	
Realizable ASR, lowest seed	69.3%	
<i>Adversarially hardened 9B replica (3 seeds)</i>		
Recall at 1% FPR	12.4% ($\pm$ 3.5 pp)	
Recall at 0.1% FPR	4.4% ($\pm$ 1.2 pp)	
Operating thresholds	0.99985 to 0.99995	
ASR, botnet	0.0% ($\pm$ 0.0 pp) (n = 300)	
ASR, brute force	0.0% ($\pm$ 0.0 pp) (n = 300)	
ASR, DoS family, seen and held out	18.7% ($\pm$ 16.2 pp) (n = 272)	
ASR, port scan	51.1% ($\pm$ 22.0 pp) (n = 129)	
ASR, web attack	0.0% ($\pm$ 0.0 pp) (n = 300)	
ASR, DoS on a training day	0.0% ($\pm$ 0.0 pp)	
ASR, held-out DDoS day	100.0% ($\pm$ 0.0 pp) (n = 56, 2 seeds)	
Recall, botnet	100.0% ($\pm$ 0.0 pp)	
Recall, held-out DDoS	2.1% ($\pm$ 3.7 pp)	
Recall, infiltration	100.0% ($\pm$ 0.0 pp)	
Recall, port scan	15.0% ($\pm$ 3.6 pp)	
Recall, web attack	98.1% ($\pm$ 1.7 pp)	
Training phrases of the bank	6 of 9	
<i>Zero-shot 9B replica, malicious question</i>		
CICIDS2017 recall / FPR / accuracy	27.6% / 8.1% / 66.2%	
TON-IoT flows	11200	
TON-IoT recall / FPR / accuracy	5.7% / 20.2% / 32.2%	
TON-IoT ECE	0.481	
TON-IoT attack rate; all-benign accuracy	64.3%; 35.7%	
Choice-question ECE, shipped, base-rate projected	0.185	
<i>Transfer, Wilson lower bounds on realizable ASR</i>		
Replica sizes, web attack, lowest Wilson lower bound across sizes	89.0%	
Laya	95.6%	
Semlf	89.6%	
<i>Hosted Jev, clean</i>		
Malicious accuracy; base-rate projected	62.2%; 77.4%	
Choice accuracy, attack classes	70.3%	
<i>Decision 2.0 ladder, range over sizes</i>		
ECE, malicious question	0.16 to 0.32	
Recall at 1% FPR	1.5% to 7.5%	
<i>Head disagreement on the hard label, 9B replica</i>		
Clean; attacked	6.2% ($\pm$ 0.4 pp); 5.6% ($\pm$ 0.6 pp)	

D Secondary results

Table 5 reports the secondary measurements.