

Article

TRACE: Feature-Space Feasible Action Contracts for Explainable Intrusion Triage

Tran Duc Le , Mohammad Arifuzzaman  and Yida Bao 

Department of Mathematics, Statistics & Computer Science, University of Wisconsin-Stout,
Menomonie, WI 54751, USA; arifuzzamanm@uwstout.edu (M.A.); baoy@uwstout.edu (Y.B.)

* Correspondence: let@uwstout.edu

Abstract

Explainable intrusion detection systems often provide feature attributions without indicating whether a security action should be released, downgraded, or deferred. This paper investigates whether action-governed explanations can provide bounded triage evidence under traffic feature feasibility constraints. We present TRACE, a framework that maps calibrated detector outputs to a finite action ladder, constructs conformal action sets, selects actions via a utility-minimax rule, and releases high-severity actions only when compact support contracts remain stable under feasible perturbations, where feasibility is a property of the processed benchmark features and not of packet-level realizability. Ablations isolate the conformal set and release gate as the primary drivers of system behavior. Across 11 gated dataset–model pairs, TRACE produces non-degenerate action sets with zero full-set collapse and defer/block rates from 0.603 to 1.000. Under held-out sample split tuning, it achieves higher average proxy utility than unconditional release and release rate-matched random release on all 11 pairs. Against the strongest simple selective gate, however, it matches on 6 of 11 pairs and trails on the remaining 5. Robustness sweeps confirm positive all-row utility on all pairs, though pass-only utility becomes fragile in ultra-low-release regimes. Unlike display-only attribution summaries, the TRACE contract records the plausible action set, feasibility checks, stability summaries, and an explicit release rationale. The results support TRACE as a bounded evidentiary framework for action-governed XAI in IDS, rather than claiming superiority over all IDS/XAI methods or general deployment readiness.

Keywords: explainable intrusion detection; network intrusion triage; conformal prediction; actionable XAI; feature-space feasible stability



Academic Editor: Emre Tokgoz

Received: 18 June 2026

Revised: 20 July 2026

Accepted: 24 July 2026

Published: 26 July 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Machine learning has become a central technique in network intrusion detection systems (IDS) research because modern traffic traces are high-volume, heterogeneous, and difficult to monitor with manually maintained rules alone. Yet, the operational difficulty of machine learning-based IDS has been recognized for more than a decade: models are often trained and tested under closed-world benchmark assumptions that do not fully match open, shifting, and adversarial network environments [1,2]. This mismatch is clearest when an IDS output is used in triage. An alert is a class prediction, but it is also a decision point. In that setting, the question extends beyond whether a detector predicts a label such as benign, scan, denial of service, botnet, or exfiltration. A prediction label identifies an attack class, whereas a feasible action is a concrete response step. The relevant

issue is whether the available evidence justifies an action such as isolating a host, dropping a packet, deferring the case, monitoring it, escalating it to an analyst, or initiating containment-oriented handling.

Explainable artificial intelligence (XAI) entered IDS research partly to narrow this gap between opaque predictions and analyst-facing evidence, a motivation reflected in recent XAI-IDS surveys and evaluations [3–5]. Model-agnostic local explanation methods such as Local Interpretable Model-agnostic Explanations (LIME) [6], additive feature-attribution frameworks such as SHapley Additive exPlanations (SHAP) [7], and later global-to-local explanation techniques for tree models [8] have made it practical to associate detector outputs with feature-level rationales, and the resulting XAI-IDS literature is surveyed in Section 2. The usual explanation artifact, however, remains a ranking, attribution vector, saliency summary, rule, or local surrogate. Such artifacts can help explain why a detector produced a score, but they do not by themselves establish that a high-severity triage action is stable, cost-aware, or justified under plausible traffic feature alternatives. In this paper, they remain secondary inspectability artifacts; the main explainability object is the action contract that justifies release, downgrade, deferral, or fail-open behavior.

This distinction leaves three methodological gaps. First, the cited XAI-IDS literature primarily explains detector predictions or feature effects, whereas triage requires reasoning over several plausible actions and their asymmetric consequences [3–5,9]. A false escalation can consume scarce analyst time and trigger unnecessary response activity, whereas an underreaction can delay treatment of a serious incident. Second, explanation stability is frequently assessed in feature space without a sufficiently explicit account of which perturbations are plausible for the traffic representation being used. For tabular flow datasets, unconstrained changes to counts, rates, protocol indicators, or derived statistics can produce explanation-robustness evidence that is mathematically convenient but operationally questionable. Third, explanation depth is rarely governed as a decision problem. In practice, shallow evidence may be acceptable for low-severity monitoring, while containment-oriented handling should require stronger uncertainty, feasibility, and support checks. Taken together, these gaps motivate an action governance view of explainable IDS: explanations should not only describe model behavior but should help decide when an action can be released, downgraded, deferred, or failed open to human investigation.

This paper develops TRACE, a conformal action contract framework for uncertainty-aware intrusion triage under processed feature, feature-space feasibility constraints. The method uses a calibrated detector as a foundation, then maps native IDS labels to a finite action ladder, aggregates detector probabilities into action probabilities, and constructs a split-conformal action set. Conformal prediction provides a principled way to form prediction sets under exchangeability assumptions [10]; in TRACE, the conformal object is moved from class-label reporting to action-level uncertainty reporting. The selected action is chosen by a finite utility-minimax rule over the conformal action set, so the policy accounts for plausible oracle actions instead of committing immediately to a single maximum-posterior label. Because public IDS datasets do not contain validated Security Operations Center (SOC) playbooks, the label-to-action mapping is treated as a benchmark scaffold rather than a production response policy. Throughout the paper, “feature-space feasible” means feasible within the processed feature benchmark representation governed by Ω_D ; it is not a claim that accepted perturbations are packet-level realizable on a live network. Empirically, the ablations plus targeted near-boundary reruns isolate the conformal-action and release gate stages most clearly. The stricter stability, minimax, and deeper support/verifier stages remain implemented, but the current evidence isolates them only partially.

The central technical object in TRACE is an action contract. For an input traffic record x , the method returns

$$K(x) = (a_{\text{conf}}, C_{\alpha}, S_k, \Gamma_D, \rho_{\text{set}}, \rho_{\text{act}}, \rho_{\text{exp}}, \Delta_{\text{cf}}, \text{status}), \quad (1)$$

a finite record pairing a recommended triage action a_{conf} with the evidence for or against releasing it: a conformal action set C_{α} , a compact feature-group support contract S_k whose sufficiency is checked rather than a top- k attribution list, a feature-space feasible perturbation set Γ_D , stability and feasible-contrast summaries, and a status field recording release, downgrade, deferral, or fail-open to investigation. A defer outcome is an explicit explanation result indicating that the evidence stack did not justify release, not a missing explanation; the intended use is analyst inspection and triage support, not autonomous execution. Section 4 develops each field together with the validation-frozen acceptance rules that govern when a high-severity action may be released.

This design shifts robust XAI for intrusion detection toward testing action-level stability: rather than claiming universal stability or certification against deployment-time adversaries, TRACE investigates whether, under a dataset-specific tabular feasibility model, the proposed action remains justified across a calibrated set of plausible actions and feasible local alternatives. Our objective is to evaluate whether this gated action contract layer produces non-degenerate, coverage-aware triage records and exposes, traceably and auditably, when release is supported, when non-release is the safer outcome, and where simple selective gates remain competitive under matched benchmark evaluation.

This paper makes four bounded core contributions.

- It formulates explainable intrusion triage as a feature-space feasible action contract problem, shifting the unit of explanation from feature attribution display to evidence for releasing, downgrading, deferring, or failing open a triage action.
- It couples action-level conformal uncertainty, a utility-minimax selector, and a release gate into a finite contract record, while explicitly treating the conformal set plus release gate mechanism as the most clearly isolated empirical core in the current evidence.
- It introduces a dataset-specific feasibility model Ω_D and perturbation set Γ_D for evaluating action contracts under train-constrained traffic feature alternatives and shows through paired probe diagnostics that generic L_p perturbations can appear stable while still being inadmissible under the dataset feasibility card.
- It establishes a bounded multi-dataset evidence protocol for proxy triage, combining held-out selective gate comparisons, operating region audits, bootstrap intervals, frozen detector probe seed variability checks, and claim readiness gates so that TRACE is evaluated as an auditable evidence layer whose practical reading can be triage-capable, cautious, blocker-like, or a weakly selective value rather than hidden behind one average scalar score.

The empirical study evaluates four public IDS benchmarks under a gate-controlled evidence protocol and matched detector foundations (Section 5). The central scientific question is whether TRACE can match a simple selective gate on a scalar risk objective while exposing the uncertainty, feasibility, and non-release structure behind a proxy triage decision.

The remainder of the paper positions the work (Section 2), defines the action triage problem (Section 3), presents TRACE (Section 4), specifies the empirical protocol (Section 5), and reports the gated results with their interpretation boundaries and validity threats (Sections 6–9).

2. Related Work and Positioning

2.1. Explainable Intrusion Detection

Explainable intrusion detection began as a response to the opacity of machine learning detectors, including unsupervised and representation-based anomaly detectors [11,12], and has developed into a substantial research area with its own survey literature, evaluation criteria, and benchmark practices. General-purpose explanation methods such as LIME and SHAP are now common baselines for explaining black-box or tree-based classifiers, and broader XAI surveys classify them within a larger family of local, global, model-specific, and model-agnostic techniques [6–8,13]. In the IDS setting, recent surveys identify explainability as a recurring requirement for security analysts, Internet of Things (IoT) cyber defense, and model debugging, while also noting that explanation quality depends on properties such as fidelity, stability, compactness, computational cost, and stakeholder fit [3,4].

Empirical XAI-IDS studies illustrate both the usefulness and the boundary of this literature. Explainable IDS models have been developed for specific traffic domains such as DNS-over-HTTPS attack detection [14], and recent evaluations compare LIME and SHAP in IDS classifiers or evaluate black-box explanation frameworks for network intrusion detection [5,9]. Parallel work extends explainable detection to trust-oriented IDS design, IoT and Industry 4.0 deployments, optimized gradient-boosted detectors, and federated settings [15–19].

Recent XAI-IDS studies increasingly pair strong predictive models with post hoc explanation layers, making detector outputs more inspectable and easier to compare across benchmarks [5,9]. Because such attribution artifacts do not by themselves establish whether a security action should be released (as argued in the Introduction), TRACE therefore treats SHAP, LIME, and related explainers as supporting infrastructure rather than as the main contribution: the unit of analysis is the action contract that governs release, downgrade, deferral, or fail-open behavior.

2.2. Actionable XAI and Security Triage

The actionability problem is sharper in security than in many classification settings because different errors have different operational meanings. This cost asymmetry, noted in the Introduction, is related to a long-standing critique of machine learning IDS evaluation: benchmark success can be achieved under closed-world assumptions that differ from operational networks, and public datasets often encode class labels rather than response policies [1,2]. Dataset surveys likewise show that NIDS corpora vary widely in feature construction, attack labeling, capture environment, and realism, which makes direct translation from dataset class to operational action delicate [20].

Encoding this asymmetry in a manually specified cost or utility matrix is a well-established practice, and one with a clear history within intrusion detection. Cost-sensitive learning is built around exactly such an object: a cost matrix whose entries record the penalty of predicting one class when another is true, set by the modeler to encode domain value judgments, after which the optimal decision follows from expected cost rather than from the data alone [21]. The intrusion detection community adopted this approach early, tabulating damage and response costs into an explicit, analyst-specified cost matrix against which the detector is then optimized [22,23]. The same reliance on an analyst-specified matrix appears throughout the wider security-decision literature: alert prioritization ranks alerts through analyst-defined scoring weights [24], and game-theoretic intrusion detection assigns defender and attacker payoff matrices by hand before solving for an equilibrium response policy [25,26]. A predefined matrix is, in this sense, the conventional means of rendering asymmetric error consequences explicit rather than an ad hoc choice. The utility

matrix adopted here, $U(b, a)$, follows the same convention at the action (triage) layer instead of the class layer, and is used as a declared, reproducible proxy scaffold rather than as a validated operational cost model, as detailed in Section 3.

Actionable XAI and algorithmic recourse provide useful language for this issue, but their standard formulation is not identical to IDS triage. Recourse methods ask what feasible changes would alter a model decision for an affected subject, with attention to actionability and constraints [27,28]. In network defense, the decision maker is usually not the traffic source seeking recourse; it is a defender choosing an operational response under uncertainty. The relevant action is therefore not simply a feature change that flips a classifier but a policy choice such as monitor, escalate, or contain. TRACE adopts this decision-centric view while keeping the claim boundary explicit. The label-to-action map and utility matrix are treated as benchmark scaffolds rather than validated SOC playbooks, as detailed in Section 3.

2.3. Robust and Adversarial XAI

Explanation quality is also affected by instability and adversarial manipulation. General XAI work has shown that post hoc explanation methods can be fooled or made misleading even when the underlying model behavior appears acceptable, including attacks against LIME and SHAP explanations [29]. Within IDS, explanation-oriented robustness has been studied more directly; for example, Barnard et al. use explanations as part of a network intrusion detection robustness strategy, while E-XAI evaluates black-box explainers for NIDS using metric families that include stability, robustness, and efficiency [5,30]. These studies motivate evaluating explanations as objects that can fail, rather than as visualizations attached after prediction.

For tabular network flow data, the perturbation model is especially important. Feature vectors often contain counts, rates, protocol indicators, one-hot groups, duration statistics, and derived ratios. Perturbing these columns independently can produce samples that are numerically close under an ℓ_p metric but implausible as traffic records. Recent work on adversarial perturbations of IDS traffic features demonstrates this concern by showing that similarity alone is not enough to ensure realistic perturbations [31]. This concern is shared by the broader adversarial IDS literature. Surveys of adversarial evasion against network intrusion detection note that naive ℓ_p -bounded perturbations routinely violate protocol, counting, or rate semantics, which has motivated constraint-aware attack models [32,33]. Realistic and adaptive perturbation patterns have since been proposed to keep adversarial examples valid as traffic records [34,35], and the same restriction has been formalized as a constrained feature space in general adversarial machine learning [36]. TRACE responds by separating generic feature-space probes from the canonical feasibility model. Its stability checks are defined over Γ_D , a dataset-specific feature-space feasible perturbation set derived from Ω_D . The evaluated objects are explanation drift and action set stability, released-action stability, and support contract stability. Generic ℓ_p perturbations remain useful as a diagnostic comparator, but they are not treated as sufficient evidence for action release.

2.4. Conformal Prediction for Action Uncertainty

Conformal prediction provides a distribution-free framework for constructing prediction sets with marginal coverage under exchangeability assumptions [10]. This property is attractive for IDS research because a single class label or maximum posterior score can hide uncertainty among operationally different outcomes. However, conformal prediction does not by itself specify which security action should be taken when the prediction set contains multiple plausible labels or actions. It also does not remove the usual limits of

benchmark IDS evaluation: under cross-source, temporal, or adversarial shift, coverage must be measured empirically rather than described as an unconditional property [1,2].

TRACE uses conformal prediction as an uncertainty layer rather than as a standalone detector. The conformal set is constructed after aggregating calibrated class probabilities into the action ladder, so $C_\alpha(x)$ represents plausible oracle actions rather than merely plausible dataset labels. The subsequent action is selected by a finite minimax utility rule over $C_\alpha(x)$, and high-severity release still depends on the feature-space feasible contract verifier. This design is related to conformal risk control, which extends conformal prediction from coverage guarantees to decision-focused risk bounds by tuning a parameterized decision rule to control a user-specified loss [37]. TRACE shares the goal of moving from prediction sets to decision sets but differs in using a fixed utility matrix minimax rule rather than online parameter tuning, and in coupling the decision rule to a downstream feasibility and contract verifier that is not part of standard conformal risk control formulations. This distinction is important for positioning: conformal prediction supplies calibrated set-valued uncertainty under its assumptions, while the paper's design contribution is the coupling of action-level conformal uncertainty with utility-aware triage and feasible contract checks.

2.5. Counterfactual Explanations and Feasible Contrast

Counterfactual explanations and recourse methods identify changes that would alter a model decision, often with the goal of making a decision actionable or contestable [13,27,28]. In that literature, restricting candidate counterfactuals to a data-valid region is standard practice: feasibility, plausibility, and causal-link constraints are treated as explicit first-class requirements rather than optional refinements [38,39]. In intrusion detection, the same idea can be useful for diagnosing boundary fragility: if a small feasible change would alter an escalation or containment decision, the action may require additional evidence. At the same time, cybersecurity creates a dual-use concern. Publishing concrete feature edits that evade or downgrade detection can provide operational hints to attackers, and infeasible counterfactuals can mislead defenders about what changes are actually possible in the traffic domain.

TRACE therefore uses counterfactual reasoning in a restricted way. The method estimates a feasible contrast distance, Δ_{cf} , inside Γ_D as a diagnostic of how close the action is to a decision-changing alternative. The contract reports distance, affected feature groups, and pass/fail status rather than presenting a step-by-step evasion recipe. This use differs from recourse-centered XAI: the goal is not to tell an adversarial source how to change behavior, but to help the defender decide whether the current action has enough support to be released or should instead be downgraded, deferred, or failed open to analyst investigation.

2.6. Synthesis of the Prior-Art Gap

Table 1 summarizes how TRACE responds to the remaining gap at the decision layer after an IDS produces an alert. The contribution of TRACE is deliberately scoped to this boundary, and the empirical claims reported later are bounded to the gated benchmark artifacts rather than phrased as deployment or universal state-of-the-art claims.

Table 1. Positioning of TRACE relative to adjacent research families.

Prior Art Family	Typical Output	Remaining Gap	TRACE Response
XAI-IDS attribution [3–5,9]	Feature attributions or local explanations	Limited action-release evidence	Conformal action contracts
Robustness-oriented XAI benchmarks [29–31]	Stability or perturbation scores	Limited feature-space feasible action stability	Ω_D , Γ_D , and contract stability scores
Actionable or cost-aware decision support [2,20,27,28]	Action motivation, prioritization, or recourse	Limited uncertainty-aware contract	Minimax action policy over C_α
Conformal prediction [10]	Prediction or label uncertainty sets	Limited utility-aware governance	Action-level conformal set plus utility matrix
Counterfactual explanation [13,27,28]	Decision-changing examples	Limited pass/fail contract behavior	Feasible contrast diagnostic

3. Problem Formulation

3.1. Dataset Samples and Label-to-Action Mapping

Consider a benchmark dataset $D = \{(x_i, y_i)\}_{i=1}^n$, where each preprocessed traffic record is represented as $x_i \in \mathbb{R}^d$ and each native class label belongs to a finite label space $y_i \in \mathcal{Y} = \{1, \dots, K\}$. Native IDS labels are dataset-specific: one corpus may distinguish scans, brute-force attempts, denial-of-service traffic, botnet traffic, or benign flows, while another may use a different label hierarchy or feature-construction process. This heterogeneity is a known limitation of public NIDS benchmarks and makes direct translation from dataset class to operational response delicate [2,20]. We therefore separate two objects that are often conflated: the native prediction label y and the benchmark action used for triage evaluation.

The action space is the ordered finite ladder

$$\mathcal{A} = \{A_0, A_1, A_2, A_3\}, \quad \ell(A_j) = j, \quad (2)$$

where $\ell : \mathcal{A} \rightarrow \{0, 1, 2, 3\}$ is the action-level index. The levels denote, respectively, defer or no action (A_0), monitor (A_1), escalate for analyst investigation (A_2), and containment-oriented high-severity handling (A_3). A deterministic mapping

$$\phi_D : \mathcal{Y} \rightarrow \mathcal{A} \quad (3)$$

is frozen separately for each dataset before test evaluation. The subscript D emphasizes that the mapping depends on dataset label semantics and documentation rather than on model outputs. For any labeled evaluation sample, the benchmark oracle action is

$$a_i^* = \phi_D(y_i). \quad (4)$$

This oracle serves as a proxy evaluation target rather than a validated SOC playbook. Public IDS datasets provide labels and traffic features, but they do not provide organization-specific response decisions, analyst workload, containment authorization, or post-incident business impact. Consequently, all action-level evaluations rely on this label-to-action mapping.

The mapping protocol in Table 2 is intentionally conservative: when a native label family plausibly straddles two severity tiers, ϕ_D assigns the lower tier unless the dataset documentation provides a clear reason to use the higher one, and the decision is then recorded in the dataset action map. This ensures that competing policies are evaluated

against the same frozen target rather than against mappings selected after observing test performance, and it prevents the evaluation from manufacturing aggressive action evidence from ambiguous benchmark labels.

Table 2. Benchmark label-to-action construction used to freeze ϕ_D before test evaluation.

Severity Tier	Typical Native Label Families	Evaluation Interpretation	Oracle Action
S_0	Benign, normal, background	No incident evidence in the benchmark label	A_0
S_1	Reconnaissance, scanning, probing, weak anomaly	Low-urgency visibility or watch-list handling	A_1
S_2	Brute force, credential attack, exploitation attempt, unauthorized access attempt	Analyst escalation or ticket creation is warranted	A_2
S_3	distributed denial-of-service (DDoS), botnet, ransomware, worm, backdoor, exfiltration, destructive compromise	Containment-oriented handling is considered in the proxy action space	A_3

An evaluated triage policy is any mapping

$$\pi : \mathbb{R}^d \rightarrow \mathcal{A} \times \mathcal{R}, \quad \pi(x_i) = (\hat{a}_i, r_i), \quad (5)$$

where $\hat{a}_i \in \mathcal{A}$ is the released action and $r_i \in \mathcal{R}$ is an optional evidence record, such as an explanation, uncertainty set, contract summary, or failure status. Detector-only baselines may return a minimal record, whereas TRACE returns the contract record defined later in Equation (1). All finite maximization rules used by a policy are assumed to use a deterministic validation-frozen tie-breaking order, so $\hat{a}_i$ is well defined for every evaluated sample.

3.2. Utility-Aware Triage Objective

Standard classification metrics are necessary for checking the detector foundation, but they do not encode the asymmetric consequences of triage errors. This distinction is central to critiques of closed-world IDS evaluation: a benchmark label error and an operational response error are not the same scientific object [1,2]. We therefore evaluate actions with a proxy utility function

$$U : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}, \quad (6)$$

where $U(b, a)$ denotes the utility of releasing action a when the oracle action is b . Rows in Table 3 are oracle actions and columns are released actions. Exact matches receive utility 1. Underreaction is penalized more strongly as oracle severity increases, while overreaction is penalized because it consumes analyst attention and may trigger unnecessary response work.

For a test set of n samples, the raw proxy triage utility of policy π is

$$\text{TU}_{\text{raw}}(\pi) = \frac{1}{n} \sum_{i=1}^n U(a_i^*, \hat{a}_i). \quad (7)$$

This scalar is not intended to replace per-class detection metrics or a human-factors study. Specifying this matrix by hand follows established cost-sensitive practice, as discussed in Section 2: predefined cost and utility matrices are the conventional way to make

asymmetric error consequences explicit. The proxy utility matrix is a synthetic benchmark scaffold and requires explicit calibration by SOC analysts before real-world operational use. The same proxy status extends to the frozen label-to-action mapping, and because both objects are benchmark scaffolds rather than validated operational targets, the experimental protocol examines how sensitive the reported utility reading is to these choices through a penalty rescaling sweep and an oracle-side action severity rebinning audit, both reported in Section 6. Its role here is to aggregate action quality under a declared and reproducible cost model, so that a method cannot appear attractive merely by maximizing accuracy, always escalating, or always avoiding high-severity release.

Table 3. Frozen proxy triage utility matrix $U(b, a)$. Rows are oracle actions b ; columns are released actions a .

Oracle \ Released	A_0	A_1	A_2	A_3
A_0	1.00	0.30	−0.40	−1.00
A_1	−0.35	1.00	0.20	−0.45
A_2	−0.90	−0.35	1.00	0.35
A_3	−1.20	−0.75	0.20	1.00

To separate different operational failure modes, define the low- and high-urgency action subsets

$$\mathcal{A}_{lo} = \{A_0, A_1\}, \quad \mathcal{A}_{hi} = \{A_2, A_3\}. \quad (8)$$

The false escalation cost rate and missed urgent incident rate are

$$\text{FECR}(\pi) = \Pr\{\hat{a} \in \mathcal{A}_{hi} \mid a^* \in \mathcal{A}_{lo}\}, \quad (9)$$

$$\text{MUIR}(\pi) = \Pr\{\hat{a} \in \mathcal{A}_{lo} \mid a^* \in \mathcal{A}_{hi}\}. \quad (10)$$

The corresponding empirical estimates are computed with the relevant conditional denominator on the test split. If a split contains no samples in the conditioning set, the metric is reported as not applicable for that split rather than forced to zero. Reporting FECR and MUIR alongside TU_{raw} is important because two policies can have similar utility while distributing errors very differently between wasted response effort and missed urgent handling.

Because the evidence record r_i may require different explanation depth, perturbation checks, or contract verification, the problem also includes cost-aware evaluation. Let $L_i \geq 0$ be the normalized inference latency for sample i , let $C_i \geq 0$ be its normalized compute cost, and define

$$\bar{L}(\pi) = \frac{1}{n} \sum_{i=1}^n L_i, \quad \bar{C}(\pi) = \frac{1}{n} \sum_{i=1}^n C_i. \quad (11)$$

For nonnegative validation-frozen penalty weights $\lambda_L, \lambda_C \in \mathbb{R}_{\geq 0}$, the cost-adjusted utility is

$$\text{TU}_{\text{cost}}(\pi) = \text{TU}_{\text{raw}}(\pi) - \lambda_L \bar{L}(\pi) - \lambda_C \bar{C}(\pi). \quad (12)$$

The weights are not interpreted as universal SOC costs. They define a benchmark tradeoff and are subjected to sensitivity analysis in the experimental protocol.

3.3. Research Questions and Testable Hypotheses

The formal problem above turns explainable IDS triage into a comparison among action policies π under a frozen oracle-action map, a declared utility function, and separate diagnostics for escalation, underreaction, uncertainty, feasibility, and cost. The proposed method in Section 4 instantiates one such policy by returning conformal action sets, mini-

max actions, feature-space feasible support contracts, and pass/fail statuses. The empirical study is organized around the research questions in Table 4.

Table 4. Research questions and required evidence.

RQ	Question	Evidence
RQ1	Do conformal action contracts produce non-degenerate, coverage-aware proxy triage records on gated IDS benchmark artifacts?	Oracle-in-set rate, normalized set size, full-set rate, action agreement, release/defer rate
RQ2	Do feature-space feasible probes prevent inadmissible generic perturbation evidence from being treated as action-release support?	Ω_D row- and feature-violation deltas, ρ_{act} , ρ_{set} , release rate deltas
RQ3	Which implemented contract components materially change release behavior under the current proxy operating point?	Ablation deltas for conformal action sets, release gates, stability gates, minimax utility, and verifier depth
RQ4	How does tuned TRACE compare with fixed TRACE, unconditional release, random matched release, and simple selective gates under matched sample split evaluation?	Held-out risk objective, release rate, released action error, matched-comparator deltas
RQ5	What claim boundary is licensed by the executed gates, bootstrap intervals, and claim readiness ledger?	G0–G8 verdicts, bootstrap confidence intervals, C1–C5 evidence status, excluded artifacts

For method comparison, let $\mathcal{B}$ denote the set of baseline policies specified in Section 5, and let π_T denote the TRACE policy. For any baseline $\pi_b \in \mathcal{B}$, define utility and error-rate differences

$$\Delta_b^{TU} = TU_{\text{raw}}(\pi_T) - TU_{\text{raw}}(\pi_b), \quad (13)$$

$$\Delta_b^{\text{MUIR}} = \text{MUIR}(\pi_T) - \text{MUIR}(\pi_b), \quad (14)$$

$$\Delta_b^{\text{FECR}} = \text{FECR}(\pi_T) - \text{FECR}(\pi_b). \quad (15)$$

These differences serve as descriptive estimands for the benchmark. The main hypotheses are: H_1 tests whether $\Delta_b^{TU} > 0$ for fixed and unconditional-release policies and whether tuned TRACE matches or exceeds simple selective gates under the held-out risk objective; H_2 tests whether feature-space feasible contract diagnostics identify action-level feasibility problems not captured by generic L_p probes; H_3 tests whether the implemented support contract diagnostics expose action set, action, and explanation-support instability under feasible probes; H_4 tests whether tuned contract release matches or exceeds fixed-depth or fixed-threshold verification on the benchmark cost-risk frontier; and H_5 quantifies degradation of ACov, CSA, and utility under secondary benchmark stress.

4. Proposed Method: TRACE

4.1. Overview

TRACE is an action governance layer placed on top of a calibrated IDS detector. The detector supplies evidence about native attack labels, but the method releases an operational action only after that evidence has been translated into an action-level uncertainty set, evaluated under the frozen utility model, and checked against a feature-space feasible support contract. This separation is deliberate. It prevents the method from being tied to a single classifier, and it makes the main contribution the contract that justifies or blocks an action rather than another attribution display.

The proposed architecture provides a decision-layer construction that couples calibrated action uncertainty, explicit triage utility, dataset-specific feature-space feasibility, and finite-budget support contract verification into one auditable action record. Each module supplies a distinct form of evidence: conformal prediction controls action set uncer-

tainty under its assumptions; the utility matrix exposes the cost model; Ω_D and Γ_D define which perturbations are treated as feature-space feasible; and the contract verifier decides whether the selected action has enough finite-probe support to be released or should be downgraded, deferred, or failed open.

The full pipeline (Figure 1) has eight stages. First, a detector is trained, calibrated, and evaluated as predictive infrastructure. Second, calibrated label probabilities are aggregated through the frozen dataset-specific map ϕ_D into probabilities over $\mathcal{A}$. Third, a split-conformal action set $C_\alpha(x)$ is constructed to represent plausible oracle actions under the calibration assumptions. Fourth, a finite utility-minimax rule selects a tentative action. Fifth, a train-only traffic-feasibility model Ω_D defines which local alternatives are admissible for the dataset representation. Sixth, a greedy search constructs a compact feature-group support set S_k . Seventh, the verifier estimates action set stability, selected action stability, support stability, feasibility, and feasible contrast distance over $\Gamma_D(x; \epsilon)$. Finally, the system emits a contract record with one of four statuses: release, downgrade, defer, or fail open to analyst investigation. The output is therefore an action label and an auditable record of why the action was or was not released. A defer outcome is not a missing explanation; it is an explicit explanation result indicating that the current evidence stack did not justify action release. The empirical evidence does not isolate these stages equally; Section 6 reports how strongly each stage influences the observed release behavior.

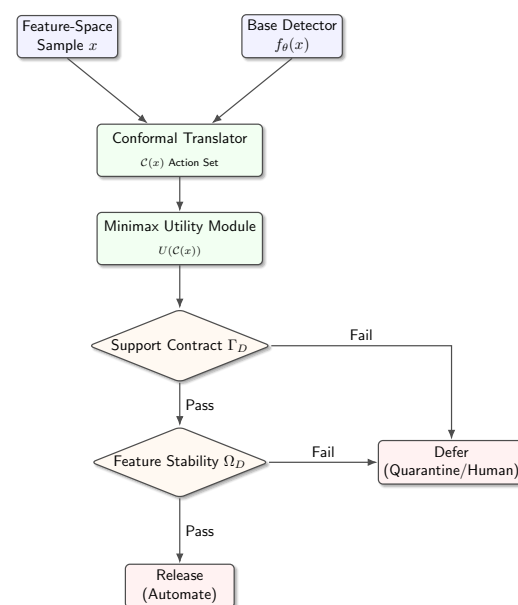


Figure 1. System Architecture of TRACE. The pipeline illustrates the progression from feature-space tabular data inputs through an IDS detector, conformal action set generation, and contract verifier checks before yielding a final decision state (release, downgrade, or defer). The architecture operates entirely at the decision and feature layer, independent of any specific hardware constraints or packet-capture integrations.

4.2. Detector and Action Posterior

Let $f_\theta(y | x)$ be a calibrated detector posterior over native labels. In the experimental protocol, calibration is performed on held-out data and assessed separately from the contract layer; temperature scaling, isotonic calibration, or another validation-selected calibration method may be used provided it is fit without test labels. The detector can be a tree ensemble, neural classifier, or another probabilistic IDS model. This replaceability matters because TRACE is not claiming novelty from the detector architecture. Its contribution begins after the detector has produced a calibrated distribution over native labels.

The first method-specific transformation is action aggregation. The induced action posterior is

$$p_A(a \mid x) = \sum_{y: \phi_D(y)=a} f_\theta(y \mid x). \quad (16)$$

Because ϕ_D is many-to-one, Equation (16) deliberately discards label distinctions that do not change the benchmark action tier. For example, two native labels may differ as attack families while both mapping to analyst escalation. This aggregation is useful for triage because conformal uncertainty and utility are evaluated over actions rather than over every native dataset label. The procedure also exposes ambiguous cases: if detector mass is spread over labels mapped to different action tiers, p_A remains diffuse, $C_\alpha(x)$ tends to widen, and the downstream policy is pushed toward additional evidence or a conservative status.

4.3. Conformal Action Set

TRACE uses split conformal prediction to convert the action posterior into a set-valued uncertainty object. Let $\mathcal{I}_{\text{cal}}$ denote the calibration indices and let $n_{\text{cal}} = |\mathcal{I}_{\text{cal}}|$. On the calibration split, define the nonconformity score

$$s_i = 1 - p_A(a_i^* \mid x_i). \quad (17)$$

For target miscoverage $\alpha \in (0, 1)$, the quantile is frozen as

$$q_\alpha = \text{Quantile}_{[(n_{\text{cal}}+1)(1-\alpha)]/n_{\text{cal}}}(\{s_i : i \in \mathcal{I}_{\text{cal}}\}), \quad (18)$$

with the usual conservative convention if the requested rank exceeds n_{cal} . The conformal action set for a test sample is then

$$C_\alpha(x) = \{a \in \mathcal{A} : 1 - p_A(a \mid x) \leq q_\alpha\}. \quad (19)$$

This set is interpreted as the collection of action tiers that remain plausible after calibration. It is not a confidence interval for a response playbook and it is not a robustness certificate. It is a finite action-level prediction set built from calibrated detector evidence.

The coverage statement follows the standard split-conformal result rather than a new theorem of this paper. If the calibration examples and the test example are exchangeable and a^* is generated under the same action-labeling rule, then

$$\Pr\{a^* \in C_\alpha(X)\} \geq 1 - \alpha, \quad (20)$$

up to the finite-sample quantile convention used in Equation (18) [10]. The empirical action coverage reported in the experiments is

$$\text{ACov} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{a_i^* \in C_\alpha(x_i)\}, \quad \text{ASize} = \frac{1}{n} \sum_{i=1}^n |C_\alpha(x_i)|. \quad (21)$$

Split-conformal prediction assumes exchangeability, an assumption that non-stationary network traffic can violate. Under temporal shift, cross-source transfer, or adversarially selected traffic, Equation (20) is a diagnostic rather than a statistical guarantee. In those settings, ACov and ASize expose calibration degradation, consistent with long-standing cautions that IDS benchmarks can differ from deployment conditions [1,2].

4.4. Utility-Minimax Action Rule

Given $C_\alpha(x)$, TRACE chooses an action by optimizing against the worst plausible oracle action in the conformal set. The selected action is

$$a_{\text{conf}}(x) = \arg \max_{a \in \mathcal{A}} \left[\min_{b \in C_\alpha(x)} U(b, a) - \lambda_L L(a) - \lambda_C C(a) \right]. \quad (22)$$

Here b ranges over plausible oracle actions, a is the candidate released action, and $L(a)$ and $C(a)$ are normalized evidence-depth costs associated with releasing a . All penalty weights, cost normalizations, and tie-breaking rules are validation-frozen before test evaluation.

Equation (22) is the decision-theoretic center of the method. A maximum-posterior action rule commits to $\arg \max_a p_A(a | x)$ even when the next-most-plausible action has very different operational consequences. A threshold rule can similarly collapse uncertainty into a single score. By contrast, the minimax rule treats the conformal set as a finite uncertainty set and selects the action whose worst-case utility is strongest under the declared proxy cost model. When $C_\alpha(x)$ contains only one action, the rule reduces to cost-aware utility maximization for that action. When $C_\alpha(x)$ contains both low- and high-severity actions, the rule naturally exposes the case as action-ambiguous and often sends it to deeper verification or a safer status. This is not reinforcement learning, online control, or a claim of globally optimal SOC response; it is an inspectable finite-action decision rule whose assumptions and utility matrix are explicit.

4.5. Feature-Space Feasible Constraint Model

The contract verifier must decide which local alternatives to treat as plausible processed traffic records. For tabular flow data, independent feature perturbations can produce vectors that are close under an ℓ_p norm while violating protocol indicators, one-hot encodings, count non-negativity, rate relations, or train-observed value ranges. This is a known concern for IDS evaluation because benchmark feature vectors are abstractions of traffic processes rather than free Euclidean points [20,31]. TRACE therefore defines a dataset-specific feasibility region Ω_D from training data and documented feature semantics before evaluating any test sample.

As discussed in Section 2, restricting perturbations to a data-valid region is an established practice in both adversarial machine learning and counterfactual explanation; Ω_D instantiates this principle for processed IDS features.

The region Ω_D contains five classes of constraints. Hard type constraints enforce non-negativity, one-hot validity, and integer-like count fields after inverse preprocessing when applicable. Train-derived bounds use robust quantiles or min–max ranges fitted on the training split only. Semantic invariants encode relations such as duration, packets, bytes, rates, and ratios where the dataset documentation supports them. Immutable groups prevent local probes from altering protocol, direction, service, identity, or label-proxy fields unless a dataset-specific categorical operator is explicitly documented. Finally, residual tolerances model relations that are approximate rather than exact, with tolerances estimated only from training residuals.

A feasibility violation score is written as

$$V_D(x) = \sum_{\ell} \max(0, r_{\ell}(x) - \tau_{\ell}), \quad (23)$$

where r_{ℓ} is the residual for constraint ℓ and τ_{ℓ} is its train-fitted tolerance. A vector is treated as feasible when $V_D(x) \leq \tau_{\text{feas}}$, with τ_{feas} fixed from training residuals or numerical precision. For simple violations, a projection operator Π_{Ω} clips bounded continuous features, repairs one-hot groups, rounds count-like fields, and restores non-negativity. If a

semantic violation cannot be repaired without inventing an unsupported traffic transformation, the probe is rejected rather than silently projected. The complete procedure for training-time construction of Ω_D is outlined in Algorithm 1.

Algorithm 1 Training-time construction of Ω_D

Require: Training feature matrix X_{tr} , feature metadata for dataset D , quantile levels η_{min}, η_{max}
Ensure: Frozen feasibility model $\Omega_D = (\mathcal{G}, \mathcal{T}, \mathcal{B}, \mathcal{I}, \mathcal{R}, \Pi_\Omega)$

- 1: Partition columns into feature groups $\mathcal{G} = \{G_1, \dots, G_m\}$ using metadata, column names, one-hot families, and flow-statistic relations.
- 2: Assign each group a type in $\mathcal{T}$: immutable, bounded-continuous, count-like, ratio-rate, one-hot, derived, or mixed.
- 3: **for** each mutable group G_j **do**
- 4: Fit train-only robust bounds $\mathcal{B}_j = [Q_{\eta_{min}}(G_j), Q_{\eta_{max}}(G_j)]$.
- 5: **end for**
- 6: **for** each documented relation $h_\ell(x) \approx 0$ **do**
- 7: Compute residuals $r_\ell(x) = |h_\ell(x)|$ on X_{tr} and freeze tolerance τ_ℓ .
- 8: **end for**
- 9: Mark immutable groups $\mathcal{I}$ that cannot change under local probes.
- 10: Define V_D by Equation (23) and define Π_Ω for safe clipping, one-hot repair, rounding, and non-negativity repair.
- 11: **return** Ω_D and a dataset Ω_D -card recording constraints, tolerances, immutable groups, and rejected repair types.

The feature-space feasible perturbation set is

$$\Gamma_D(x; \epsilon) = \{x' \in \Omega_D : d_G(x', x) \leq \epsilon, \text{immutable}(x') = \text{immutable}(x)\}. \quad (24)$$

The distance d_G is group-aware: it aggregates changes at the feature-group level so that a one-hot family, a packet count family, or a rate-derived family is not treated as unrelated scalar columns. Three probe families are used inside Γ_D . G-NOISE applies small continuous perturbations to mutable groups to emulate measurement and aggregation variation. G-MASK masks or imputes eligible mutable non-identity groups to test dependence on logging sparsity or missing derived values. G-SEARCH performs constrained grouped coordinate search for action-changing alternatives. Each probe is followed by Π_Ω , scored by V_D , and either accepted into the finite probe set $P(x) \subset \Gamma_D(x; \epsilon)$ or rejected. Generic ℓ_p probes remain in the experimental protocol as a comparator, not as evidence sufficient for action release.

4.6. Sufficient Support Contract

The support contract is the core object that distinguishes TRACE from explanation display IDS. Let $S \subseteq \mathcal{G}$ be a set of feature groups and let $x'[S] = x[S]$ mean that the groups in S are fixed to their observed values while all other mutable groups may vary within $\Gamma_D(x; \epsilon)$. Ideally, a sufficient support set would solve

$$S_k \in \arg \min_{S \subseteq \mathcal{G}} |S| \quad (25)$$

$$\text{s.t. } |S| \leq k_{\max}, \quad (26)$$

$$\max_{x' \in \Gamma_D(x; \epsilon) : x'[S] = x[S]} d_A(a_{\text{conf}}(x'), a_{\text{conf}}(x)) \leq \delta_A, \quad (27)$$

where

$$d_A(a, b) = \frac{|\ell(a) - \ell(b)|}{3}. \quad (28)$$

Equation (27) states the intended sufficiency criterion: after fixing the support groups, feasible variation in the remaining mutable groups should not move the selected action by more than the accepted action distance tolerance. The exact search is combinatorial, so the implemented method uses the validation-frozen greedy approximation in Algorithm 2. We therefore call S_k a compact sufficient support set relative to the specified greedy search, probe budget, and $k_{\max}$, not a globally minimal mathematical certificate.

Algorithm 2 Greedy support contract search and verification

Require: Sample x , action $a_{\text{conf}}(x)$, action set $C_{\alpha}(x)$, feasibility model Ω_D , groups $\mathcal{G}$, shallow budget B_1 , deep budget B_2 , maximum size $k_{\max}$
Ensure: Contract fields $(S_k, \rho_{\text{set}}, \rho_{\text{act}}, \rho_{\text{exp}}, \Delta_{\text{cf}}, \text{status})$

- 1: Generate shallow feasible probe set $P_1(x) \subset \Gamma_D(x; \epsilon)$ using budget B_1 .
- 2: Initialize $S \leftarrow \emptyset$ and compute baseline contract violation $J(S; P_1)$.
- 3: **while** acceptance thresholds fail and $|S| < k_{\max}$ **do**
- 4: **for** each candidate group $G_j \in \mathcal{G} \setminus S$ **do**
- 5: Estimate violation decrease $\Delta J_j = J(S; P_1) - J(S \cup \{G_j\}; P_1)$.
- 6: **end for**
- 7: Add the group with the largest validation-defined decrease: $S \leftarrow S \cup \{\arg \max_j \Delta J_j\}$.
- 8: **end while**
- 9: **if** shallow thresholds pass and $a_{\text{conf}}(x) \neq A_3$ **then**
- 10: **return** $(S, \rho_{\text{set}}, \rho_{\text{act}}, \rho_{\text{exp}}, \Delta_{\text{cf}}, \text{RELEASE})$.
- 11: **end if**
- 12: Generate deep probe set $P_2(x) \subset \Gamma_D(x; \epsilon)$ and run constrained feasible contrast search.
- 13: Recompute stability scores on $P_1(x) \cup P_2(x)$ and estimate Δ_{cf} .
- 14: **if** deep acceptance thresholds pass **then**
- 15: **return** $(S, \rho_{\text{set}}, \rho_{\text{act}}, \rho_{\text{exp}}, \Delta_{\text{cf}}, \text{RELEASE})$.
- 16: **else if** a lower-severity action has higher feasible contract utility **then**
- 17: **return** $(S, \rho_{\text{set}}, \rho_{\text{act}}, \rho_{\text{exp}}, \Delta_{\text{cf}}, \text{DOWNGRADE})$.
- 18: **else**
- 19: **return** $(S, \rho_{\text{set}}, \rho_{\text{act}}, \rho_{\text{exp}}, \Delta_{\text{cf}}, \text{DEFER/FAIL-OPEN})$.
- 20: **end if**

For the greedy search, let $P_S(x)$ denote the finite feasible probe set generated while holding the groups in S fixed. Algorithm 2 ranks candidate groups by a scalar contract violation functional

$$J(S; P) = [\tau_{\text{act}} - \hat{\rho}_{\text{act}}(S; P_S)]_+ + [\tau_{\text{set}} - \hat{\rho}_{\text{set}}(S; P_S)]_+ + [\tau_{\text{exp}} - \hat{\rho}_{\text{exp}}(S; P_S)]_+ + \mathbf{1}\{a_{\text{conf}}(x) = A_3\} [\tau_{\text{cf}} - \hat{\Delta}_{\text{cf}}(S; P_S)]_+, \quad (29)$$

where $[z]_+ = \max(z, 0)$. The hatted quantities are provisional versions of the stability and contrast scores defined below, computed under the candidate support set S . Equation (29) is used only to order candidate groups inside the finite greedy search; it is not an additional training loss, and it does not create an optimality claim beyond the searched probe budget.

The contract is assessed by stability scores computed over a finite feasible probe set $P(x)$. Selected action stability is

$$\rho_{\text{act}}(x; P) = 1 - \max_{x' \in P(x)} d_A(a_{\text{conf}}(x'), a_{\text{conf}}(x)). \quad (30)$$

Action set stability uses the Hausdorff distance between action levels in the two conformal sets. For nonempty action sets $C, C' \subseteq \mathcal{A}$, define

$$d_H(C, C') = \max \left\{ \max_{a \in C} \min_{b \in C'} |\ell(a) - \ell(b)|, \max_{b \in C'} \min_{a \in C} |\ell(a) - \ell(b)| \right\}. \quad (31)$$

The normalized set stability score is then

$$\rho_{\text{set}}(x; P) = 1 - \frac{1}{3} \max_{x' \in P(x)} d_H(C_\alpha(x'), C_\alpha(x)). \quad (32)$$

Support stability compares the discovered support set and its normalized group weights under feasible probes:

$$\rho_{\text{exp}}(x; P) = \frac{1}{|P(x)|} \sum_{x' \in P(x)} \left[\frac{1}{2} \text{Jac}(S_k(x), S_k(x')) + \frac{1}{2} \text{Cos}(w_S(x), w_S(x')) \right]. \quad (33)$$

Here, w_S denotes the normalized nonnegative group weight vector used by the support ranking routine. Empty-support conventions are fixed on validation data: two empty supports are treated as identical and a one-sided empty comparison is treated as unstable. The feasible contrast distance is reported as the best distance found by the constrained search. Let $P_{\text{cf}}^\Delta(x) = \{x' \in P_{\text{cf}}(x) : a_{\text{conf}}(x') \neq a_{\text{conf}}(x)\}$, where $P_{\text{cf}}(x) \subset \Gamma_D(x; \epsilon)$ is the finite contrast search set. Then,

$$\Delta_{\text{cf}}(x) = \begin{cases} \min_{x' \in P_{\text{cf}}^\Delta(x)} d_G(x', x), & P_{\text{cf}}^\Delta(x) \neq \emptyset, \\ +\infty, & P_{\text{cf}}^\Delta(x) = \emptyset. \end{cases} \quad (34)$$

The $+\infty$ case is not interpreted as proof of global invariance; it is reported as right-censoring at the largest searched distance. The paper reports contrast direction and affected feature groups rather than exact attack-enabling edits.

The contract acceptance rule is

$$\rho_{\text{act}}(x; P) \geq \tau_{\text{act}}, \quad \rho_{\text{set}}(x; P) \geq \tau_{\text{set}}, \quad \rho_{\text{exp}}(x; P) \geq \tau_{\text{exp}}, \quad (35)$$

$$V_D(x') \leq \tau_{\text{feas}} \quad \forall x' \in P(x), \quad \Delta_{\text{cf}}(x) \geq \tau_{\text{cf}} \quad \text{if } a_{\text{conf}}(x) = A_3. \quad (36)$$

For containment-oriented handling, TRACE also requires $A_3 \in C_\alpha(x)$ and $a_{\text{conf}}(x) = A_3$. If any condition fails, the method does not fabricate a confident explanation. It returns the failure reason, downgrades when a lower-severity action has stronger contract utility, defers when the evidence is inconclusive, or fails open to analyst investigation when the budget is exhausted for a high-severity unresolved case.

4.7. Anytime Contract Scheduler

The contract verifier is scheduled as an anytime process (see Table 5) because not every sample needs the same explanation depth. Low-severity cases with narrow conformal sets can be handled with a cheap diagnostic record. Ambiguous or high-severity cases require stronger support checks before release. Let $w(x) = 1$ indicate that $C_\alpha(x)$ contains both low- and high-severity actions, and let $h(x) = 1$ indicate that the tentative action is A_3 . A conformal set is called narrow when $|C_\alpha(x)| \leq \tau_{\text{size}}$ and $w(x) = 0$, where τ_{size} is validation-frozen. The scheduler enters deeper states when $w(x) = 1$, $h(x) = 1$, shallow stability scores fail, feasible contrast distance is too small, or queue policy allows deeper verification for high-severity cases.

All scheduler thresholds are validation-frozen. The scheduler is therefore a cost-control mechanism, not an adaptive procedure tuned on the test set. Its behavior is still part of the evidence record, but the frozen reported artifacts capture final contract outcomes (selected action, final status, and stability summaries) rather than a complete Z0–Z3 stopping distribution audit. We therefore interpret the scheduler through its observable release, defer, and downgrade consequences on the final contract rows.

Table 5. Anytime contract verification states.

State	Name	Entry Condition	Exit Condition
Z0	Conformal-only	Always compute p_A , C_α , and a_{conf} .	Stop only for low-severity, narrow-set cases that do not require a contract.
Z1	Shallow contract	Enter when $a_{\text{conf}} \in \{A_1, A_2\}$, $ C_\alpha $ is large, or low/high ambiguity exists.	Release A_0 – A_2 if shallow contract thresholds pass; otherwise enter Z2.
Z2	Deep contract	Required for tentative A_3 , failed shallow checks, unstable conformal sets, or small feasible contrast distance.	Release only if deep stability, feasibility, and contrast thresholds pass.
Z3	Unresolved	Enter when budget is exhausted or deep verification fails.	Downgrade, defer, or fail open with logged failure reason.

4.8. Inference Procedure

Algorithm 3 (and its tabular breakdown in Table 6) summarizes the inference-time policy. The key design point is that every branch returns a record. A passed contract contains the action and support evidence; a failed contract contains the failed predicate and the fallback status. This prevents the evaluation from counting only successful explanations while hiding unresolved cases.

Algorithm 3 TRACE inference for one sample

Require: Sample x , calibrated detector f_θ , frozen ϕ_D , q_α , utility matrix U , feasibility model Ω_D , scheduler budgets

Ensure: Contract record $K(x)$

- 1: Compute $f_\theta(y | x)$ and aggregate $p_A(a | x)$ using Equation (16).
- 2: Construct $C_\alpha(x)$ using Equation (19).
- 3: Select $a_{\text{conf}}(x)$ by Equation (22) with frozen tie-breaking.
- 4: **if** $a_{\text{conf}}(x) \in \{A_0, A_1\}$ and $C_\alpha(x)$ is narrow **then**
- 5: **return** $K(x)$ with RELEASE or DIAGNOSTIC status at Z0.
- 6: **end if**
- 7: Generate feasible probes from $\Gamma_D(x; \epsilon)$ and search for S_k using Algorithm 2.
- 8: Estimate ρ_{set} , ρ_{act} , ρ_{exp} , Δ_{cf} , cost, and feasibility diagnostics.
- 9: **if** contract acceptance rule in Equations (35) and (36) passes **then**
- 10: **return** $K(x) = (a_{\text{conf}}, C_\alpha, S_k, \Gamma_D, \rho_{\text{set}}, \rho_{\text{act}}, \rho_{\text{exp}}, \Delta_{\text{cf}}, \text{RELEASE})$.
- 11: **else if** fallback utility favors a lower-severity action with a valid or cheaper contract **then**
- 12: **return** $K(x)$ with DOWNGRADE status and logged failing predicate.
- 13: **else**
- 14: **return** $K(x)$ with DEFER or FAIL-OPEN status.
- 15: **end if**

Table 6. TRACE-IDS inference procedure for a sample x .

Step	Operation
1	Compute $f_\theta(y x)$ and aggregate it into $p_A(a x)$.
2	Construct $C_\alpha(x)$ using the calibration quantile q_α .
3	Select $a_{\text{conf}}(x)$ by the utility-minimax rule in Equation (22).
4	Generate feasible probes from $\Gamma_D(x; \epsilon)$.
5	Search for a compact support contract S_k .
6	Estimate ρ_{set} , ρ_{act} , ρ_{exp} , and Δ_{cf} .
7	Return $K(x)$ with a release, downgrade, defer, or fail-open status.

4.9. Computational Complexity

This subsection characterizes the per-sample time and space complexity of the inference pipeline. The dominant operations are the greedy support set search, the multi-round feasible-probe generation, and the multi-index stability detection; their costs, together with those of the conformal and utility-minimax base steps, are summarized in Table 7.

Let p be the number of processed features, m the number of feature groups in $\mathcal{G}$, $|\mathcal{A}| = 4$ the number of actions, B_1 and B_2 the shallow and deep feasible-probe budgets, and $k_{\max}$ the maximum support set size. These quantities are validation-frozen rather than data-dependent: the support index cardinality is fixed at $k_{\max} = 10$ across every reported pair, the probe budget is roughly $B_1 \approx 16$ probes per round, and the action ladder has four levels. Let c_f denote the cost of a single detector forward pass together with the action-posterior aggregation; for the tree ensemble detectors used here (XGBoost, random forest, CatBoost), this is the cost of a bounded-depth ensemble evaluation, which is itself well suited to edge hardware.

Table 7. Per-sample time and space complexity of the TRACE inference pipeline.

Pipeline Component	Time (per Sample)	Space (per Sample)
Conformal set + utility-minimax (base, Z_0)	$O(\mathcal{A} ^2)$ with frozen q_α	$O(\mathcal{A})$
Multi-round feasible probe generation	$O((B_1 + B_2)(c_f + p))$	$O((B_1 + B_2)p)$
Greedy support set search	$O(k_{\max} m B_1 c_f)$	$O(k_{\max} + B_1 p)$
Multi-index stability detection	$O((B_1 + B_2) \mathcal{A} ^2)$	$O(\mathcal{A})$

The greedy support contract search dominates the per-sample cost. Because $k_{\max}$, m , B_1 , and B_2 are frozen constants, this cost reduces to a bounded constant multiple of one detector forward pass c_f and is independent of the number of training or calibration samples seen at inference time. The conformal quantile q_α , the feasibility model Ω_D , and the utility matrix are computed offline; at inference, conformal set construction amounts to $|\mathcal{A}|$ threshold comparisons against the frozen q_α , and the utility-minimax selection ranges over the four-element action set. The corresponding offline costs are one-time: fitting Ω_D (Algorithm 1) requires $O(n_{\text{tr}} p)$ work for the bounds and relation residuals, and split-conformal calibration requires $O(n_{\text{cal}} \log n_{\text{cal}})$ to extract the quantile in addition to $O(n_{\text{cal}} c_f)$ for the conformity scores.

The anytime contract scheduler (Table 5) lowers the amortized cost below this worst case. Low-severity cases with narrow conformal sets return from the Z_0 state without any probe generation or support search, so the multi-probe machinery is invoked only for ambiguous or high-severity alerts. This staged structure gives concrete meaning to the normalized latency and compute quantities L_i and C_i defined in the problem formulation (Equation (11)).

The cost profile bears directly on deployment at resource-limited edge IoT terminals. Per-sample inference is polynomial, bounded by validation-frozen budgets, and driven by a small constant number of tree ensemble forward passes that do not grow with traffic volume. The base detector is a tree ensemble of a kind already common on edge devices. At the same time, the present experiments were not profiled on edge or IoT hardware, so the results do not establish real-time line-rate inline per-packet processing on such platforms. TRACE is better read as a per-alert triage governance layer that emits an auditable evidence record for analyst inspection than as an inline packet classifier; this reading is consistent with the intended use set out in the method overview (Section 4), where an alert record supports inspection and triage rather than autonomous execution. The probe budgets and scheduler thresholds are exactly the levers an edge deployment would adjust to meet a

target latency budget, and characterizing on-device latency under that budget is left to future work.

5. Experimental Protocol

5.1. Research Design and Gate Discipline

The empirical study is a gated, claim-bounded evaluation of TRACE rather than a conventional leaderboard experiment. This design choice follows a practical concern in IDS research: public benchmarks can support apparently strong numerical results while leaving leakage, fragile threat assumptions, feature-space infeasibility, and operational overclaiming unresolved [1,2,20]. We therefore require each reported claim to pass a corresponding evidence gate before it is stated as a result. The executed evidence package contains frozen preprocessing artifacts, detector models, test predictions, TRACE contract records, selective gate tuning outputs, paired probe-mode comparisons, ablation summaries, bootstrap intervals, frozen detector probe seed variability outputs, and a claim readiness ledger. We report only the subset of claims permitted by those artifacts.

The gate sequence (Table 8) covers eight reporting checks spanning data readiness (G0–G1), detector sanity (G2), mechanism non-degeneracy (G3), ablation and robustness (G4–G5), and result, claim readiness, and wording integrity (G6–G8). In the final gate ledger, G3 produced 12 PASS and 3 FAIL_REPAIR records; after excluding the smoke test dataset and the failed TON_IoT/XGBoost mechanism run, 11 research dataset–model pairs are eligible for reported TRACE analysis.

Table 8. Executed evidence gates and claim permissions.

Gate	Evidence Checked	Claim Permission
G0–G1	Dataset manifests, splits, train-only preprocessing, feature cards	Data-readiness and leakage-control claims with dataset-specific split caveats
G2	Trained detector artifacts, test predictions, prediction diversity, metric sanity	Detector-foundation claims; no standalone security-action claim
G3	TRACE contract records, action set sharpness, release/defer behavior, noncollapse checks	Proposed method mechanism claims for 11 G3-passing research pairs
G4–G5	Ablation deltas, threshold sensitivity, feature-space feasible versus generic probes	Component and robustness-diagnostic claims, bounded to feature-space contracts
G6–G8	Result table integrity, fairness audit, claim readiness ledger	Final wording limited to proxy triage evidence, not deployment or universal state-of-the-art superiority

5.2. Datasets, Exclusions, and Split Boundaries

The evaluation uses four public IDS benchmarks as research datasets (detailed in Table 9). CICIoT2023 provides large-scale IoT attack traffic [40]. Edge-IIoTset adds an edge/industrial Internet of Things (IIoT) setting with centralized and federated-learning motivation [41]. CSE-CIC-IDS2018 provides a conventional network IDS bridge benchmark [42]. TON_IoT contributes IoT/IIoT telemetry with categorical and numeric network fields [43]. SMOKE_SYNTHETIC is retained only to exercise the pipeline and is excluded from all research claims. The dataset choice is not claimed as a contribution; the contribution is the action contract formulation and the disciplined evidence stack applied to these datasets.

The processed inventory ranges from roughly 200k rows (TON_IoT) to over 6 million (CICIoT2023), with feature dimensions spanning 16 numeric plus 24 categorical features (TON_IoT, after dropping direct source and destination identifiers) up to 78 numeric features (CSE-CIC-IDS2018, after dropping timestamps and file stems). The disjoint-versus-

reused split structure that gates each dataset’s claim boundary is detailed in Table 9; these boundaries support bounded processed split wording only and do not establish temporal, live-network, or deployment-valid OOD evaluation. Across all valid datasets, the executed contract tables materialize strong tree ensemble detector foundations (XGBoost, RF-GPU, and CatBoost), and a reproduced DeepXAI-IDS run is surfaced as a predictive reference in Table 10, with its action contract outputs discussed separately rather than merged into the main tree ensemble contract tables.

Table 9. Dataset roles and claim boundaries.

Dataset	Role	Claim Boundary
CICIoT2023	Primary IoT benchmark	Processed train/validation/test groups are disjoint; strongest dataset for bounded cross-group processed split wording and for detector, mechanism, and selective gate claims
Edge-IIoTset	Primary IIoT/edge benchmark	Used for mechanism and selectivity claims, but the processed representation exposes only one usable group, so shift wording remains benchmark-bounded
CSE-CIC-IDS2018	Secondary transfer/stress benchmark	Used for cross-benchmark stress; processed splits reuse the same groups across train/validation/test, so group-held-out wording is not claimed
TON_IoT	Secondary IoT/IIoT telemetry benchmark	Processed train/validation/test groups are disjoint, supporting bounded split structure wording; the XGBoost mechanism record is still excluded because G3 failed repair
SMOKE_SYNTHETIC	Engineering smoke test	Excluded from the reported results and claims

Table 10. Protocol-aligned DeepXAI-IDS predictive reference. The reproduced deep baseline uses the same split, preprocessing, action map, and proxy utility scaffold as the main evaluation, but it is surfaced only as predictive context rather than as a fully matched action contract comparator. Smoke test rows are excluded.

Dataset	Val. Acc.	TU _{raw}	FECR	MUIR
CICIoT2023	0.9898	0.9837	0.2043	0.0049
CSE-CIC-IDS2018	0.9602	0.9470	0.0001	0.0000
Edge-IIoTset	0.8869	0.9107	0.0422	0.0062
TON_IoT	0.7433	0.8928	0.0118	0.0042

5.3. Leakage Controls and Threat-Model Boundary

The implementation follows split-before-fit preprocessing. Raw features are canonicalized, identity-like metadata are removed from model inputs or retained only for grouping, split identifiers are frozen, and imputation, encoding, and scaling are fit only inside the training partition. Validation and test matrices are transformed by frozen training preprocessors. SHAP and LIME reference pools are sampled from the training partition when explanation display records are generated [6–8]. Detector predictions, TRACE contracts, ablations, and selective gate comparisons are computed from frozen artifacts, not from refitted test-time pipelines.

The robustness scope is feature-space- and dataset-constrained. The contract verifier tests local alternatives generated inside Γ_D and projected to the dataset feasibility card Ω_D . It does not generate packet-level adversarial traffic, prove protocol-valid exploitability, or certify live-network robustness. Feature-space feasibility is therefore a property of the processed benchmark representation, not synonymous with packet-level realizability. Generic probes are retained only as a diagnostic comparator, because perturbing IDS features without feasibility constraints can create vectors that are numerically

close but semantically implausible [31]. This boundary is also a safety decision: the paper reports aggregate action stability and feasibility diagnostics rather than operational edit recipes. For exact reproducibility of the predictive foundations, all model configurations, including the DeepXAI-IDS network architecture, embedding dimensions, learning rates, and XGBoost hyperparameters, are frozen and documented in the supplementary repository artifacts.

5.4. Models, Baselines, and Fair Comparison

The executed detector suite contains XGBoost, random forest (RF-GPU in the reported tables), and CatBoost artifact variants, representing strong tabular tree ensemble foundations [44–46]. To broaden the predictive context, we also reproduce a representative deep representation learning XAI-IDS architecture (DeepXAI-IDS) under the same split, preprocessing, action map, and metric budget [47,48]; Table 10 reports it as predictive context using the frozen summary metrics already materialized in the artifact package. The reported action contract tables focus on the tree ensemble artifacts for which the full gated contract suite is available, so the deep baseline serves as predictive and positional context rather than a fully matched contract-table participant.

The main decision baselines are: an all-action release policy with no selective gate; fixed TRACE status; confidence-only tuned gates; chosen-confidence tuned gates; stability-only tuned gates; sharpness-plus-confidence tuned gates; and release rate-matched random gates. The gate tuning fairness audit reports PASS for all eight compared methods: they use the same evaluation half, frozen G3 contract artifacts, action error/risk objective metrics, release threshold grid where applicable, and proxy triage action ladder. This audit is essential because a selective system can appear strong merely by releasing fewer cases. We therefore compare TRACE under matched release grids and report both action error and risk objective on held-out evaluation halves.

5.5. Metrics, Statistical Analysis, and Claim Rules

Metrics are organized by claim tier. Detector metrics include macro-F1, balanced accuracy, and log loss. Conformal-action diagnostics include q_{α} , normalized action set size, full-set rate, oracle-in-set rate, action agreement, and mean action distance. Contract release diagnostics include release rate, defer/block rate, ρ_{act} , ρ_{set} , and robustness pass rate. Selective gate metrics include release rate, action error on released cases, risk objective, and deltas versus fixed TRACE, all-action release, random matched release, and the best simple selective gate. Robustness diagnostics compare feature-space feasible and generic probes through changes in release rate, action-change rate, set-change rate, and Ω_D violation rate. Bootstrap confidence intervals use 500 resamples at the contract-row level for seven contract metrics [49]. The specific evidence claims (C1–C5) mapped by this protocol are summarized in Table 11.

As Table 11 records, C1–C4 are supported with scope, and C5 is supported only in the bounded sense that TRACE improves over fixed TRACE and over all-action policies on average for the finite-comparator rows, while remaining tied with or weaker than the best simple selective gates for several pairs. These results position TRACE as a benchmark evidence contract framework for action-governed triage; operational correctness, autonomous containment, certified robustness, and universal IDS/XAI superiority lie beyond what the present artifacts establish.

Table 11. Evidence claims (C1–C5) and their corresponding status.

Claim	Description
C1	TRACE provides coverage-aware proxy triage evidence.
C2	TRACE provides paired feature-space feasible-versus-generic probe diagnostics.
C3	TRACE provides contract-gated containment behavior.
C4	TRACE provides action-level misguidance measurements on frozen benchmark artifacts.
C5	TRACE improves over fixed baselines on average.

6. Results and Analysis

The analyses below build on one another, each stage narrowing the question for the next. The evaluation boundary first restricts attention to the non-degenerate artifact pairs; on that set, the detection foundation verifies that the action contracts rest on strong, heterogeneous detector substrates. The TRACE mechanism tables then establish, on those same pairs, whether the contracts produce sharp action sets and selective release. That evidence frames the comparisons that follow: how the release behavior stands against simpler selective gates and whether it survives wider operating regions and alternative utility encodings; where the method genuinely diverges from those gates, particularly near decision boundaries; and what role the feature-space feasibility constraint plays in keeping the stability evidence valid. The ablation traces this behavior back to its components, and the contract taxonomy with its bootstrap confidence intervals closes the section by assessing the aggregate stability of the record.

6.1. Evaluation Boundary for Interpreting the Results

The empirical results are interpreted conservatively to ensure that the evaluation relies only on models and datasets passing relevant deterministic diversity and sharpness checks. Non-passing artifacts are treated as boundary evidence rather than hidden failures.

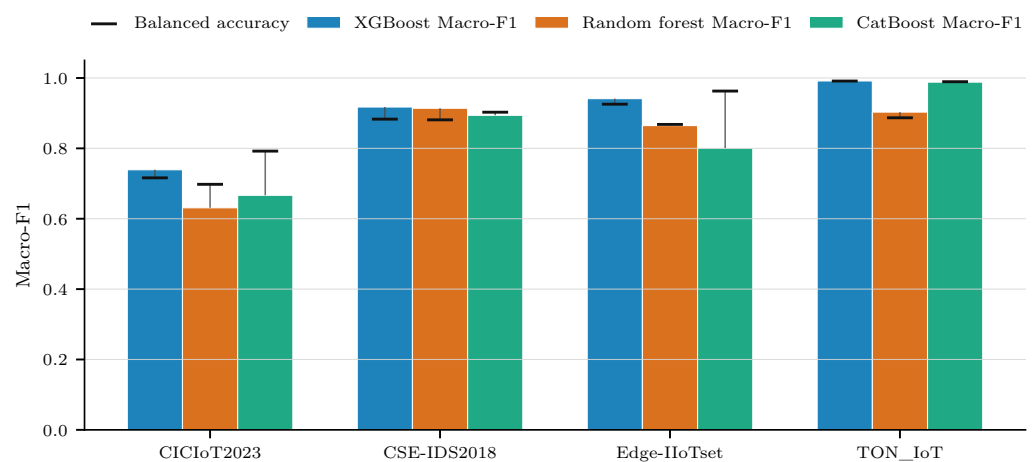
For TRACE mechanism analysis we use four research datasets: CICIoT2023, CSE-CIC-IDS2018, Edge-IIoTset, and TON_IoT. The smoke test artifacts are excluded. We also exclude the TON_IoT/XGBoost pair from the mechanism tables because it fails the conformal-sharpness criterion. These exclusions are themselves part of the result. TRACE is evaluated only where the contract mechanism is demonstrably non-degenerate, so the quantitative interpretation below is bounded to the 11 passing research pairs. We therefore treat the selective gate comparisons as held-out benchmark evidence rather than universal superiority claims.

6.2. Detection Foundation

The analysis begins at the detector layer, because the action contracts are evaluated on top of these substrates. Table 12 and Figure 2 report the frozen test detector foundation for the executed tree ensemble suite. XGBoost gives the highest macro-F1 on all four research datasets, with 0.739 on CICIoT2023, 0.918 on CSE-CIC-IDS2018, 0.941 on Edge-IIoTset, and 0.992 on TON_IoT. CatBoost gives the highest balanced accuracy on CICIoT2023, CSE-CIC-IDS2018, and Edge-IIoTset. This pattern matters for TRACE because the action contract is evaluated on top of strong but not identical detector substrates. The downstream action layer is therefore not being tested on a trivial classifier, and model-dependent contract behavior remains visible rather than hidden by selecting one detector after observing test outcomes. To keep the deep comparator from disappearing into provenance notes, Table 10 separately surfaces the reproduced DeepXAI-IDS run as a protocol-aligned predictive reference under the same split, preprocessing, action map, and proxy utility scaffold, but without promoting it to a matched action contract comparator.

Table 12. Detector foundation on frozen test partitions. Smoke test results are excluded; model names abbreviate the frozen B11 artifacts.

Dataset	Model	Macro-F1	Bal. Acc.	Log Loss
CICIoT2023	CatBoost	0.667	0.792	0.031
CICIoT2023	RF-GPU	0.631	0.698	0.118
CICIoT2023	XGBoost	0.739	0.716	0.019
CSE-CIC-IDS2018	CatBoost	0.894	0.903	0.180
CSE-CIC-IDS2018	RF-GPU	0.914	0.881	0.252
CSE-CIC-IDS2018	XGBoost	0.918	0.883	0.111
Edge-IIoTset	CatBoost	0.801	0.963	0.140
Edge-IIoTset	RF-GPU	0.865	0.868	0.394
Edge-IIoTset	XGBoost	0.941	0.926	0.074
TON_IoT	CatBoost	0.988	0.989	0.075
TON_IoT	RF-GPU	0.903	0.887	0.313
TON_IoT	XGBoost	0.992	0.991	0.040

**Figure 2.** Macro-F1 of the executed detector suite on the four research datasets. The figure visualizes detector substrate quality only; it is not a TRACE superiority claim.

6.3. TRACE Mechanism Quality

On these detector substrates, the first TRACE-specific question is whether the contract layer produces valid, non-degenerate action evidence. The final G3-passing contracts are strictly non-degenerate. Table 13 shows that every evaluated pair has a full-set rate of 0.000, meaning no reported research pair contract collapsed to $C_\alpha(x) = \mathcal{A}$ for all actions. The conformal thresholds vary sharply by model: CICIoT2023/CatBoost has $q_\alpha = 0.001248$, while CICIoT2023/XGBoost has $q_\alpha = 0.000004$; TON_IoT/CatBoost and RF-GPU require much larger thresholds, 0.971084 and 0.924729, respectively. This variation shows that the contract layer is sensitive to detector calibration and dataset geometry, so mechanism evidence cannot be inferred from detector accuracy alone.

Most action sets have mean size one, so the contracts are sharp at the selected operating point. The exceptions are Edge-IIoTset/RF-GPU, with mean set size 1.046, and the two reported TON_IoT pairs, with mean set sizes 1.308 and 1.515. Oracle-in-set rates range from 0.769 to 0.996. TRACE does not return a single confident action everywhere. It can represent ambiguity, but the accepted contracts remain much sharper than the full action space.

The release pattern is selective rather than uniformly permissive. Release rates span from 0.000 on Edge-IIoTset/CatBoost to 0.3975 on CSE-CIC-IDS2018/XGBoost; defer-or-block rates span from 0.6025 to 1.000. The main mechanism result is not high release

volume. Instead, the contract layer produces sharp action sets and meaningful non-release behavior under frozen G3-passing artifacts. For proxy action governance, this behavior fits the intended safety boundary: non-release is an inspectable outcome when conformal, utility, support, or stability evidence is insufficient.

Table 13. TRACE mechanism quality on G3-passing research dataset–model pairs. Model names abbreviate the frozen B11 artifacts. TON_IoT/XGBoost and smoke test records are excluded by gate policy.

Dataset	Model	q_α	ASize	Full-Set	Oracle-in-Set	Release	Defer/Block
CICIoT2023	CatBoost	0.001248	1.000	0.000	0.849	0.350	0.650
CICIoT2023	RF-GPU	0.017759	1.000	0.000	0.769	0.338	0.663
CICIoT2023	XGBoost	0.000004	1.000	0.000	0.810	0.338	0.663
CSE-CIC-IDS2018	CatBoost	0.396629	1.000	0.000	0.906	0.204	0.796
CSE-CIC-IDS2018	RF-GPU	0.191514	1.000	0.000	0.879	0.321	0.679
CSE-CIC-IDS2018	XGBoost	0.123400	1.000	0.000	0.886	0.398	0.603
Edge-IIoTset	CatBoost	0.320848	1.000	0.000	0.963	0.000	1.000
Edge-IIoTset	RF-GPU	0.603341	1.046	0.000	0.876	0.003	0.998
Edge-IIoTset	XGBoost	0.147651	1.000	0.000	0.938	0.025	0.975
TON_IoT	CatBoost	0.971084	1.308	0.000	0.996	0.006	0.994
TON_IoT	RF-GPU	0.924729	1.515	0.000	0.996	0.021	0.979

A stricter practical reading comes from the frozen operating region audit, not from the strict release rate column alone. Of the 11 G3-passing pairs, four are triage-capable with scope, one is moderate throughput with scope, two are cautious low-throughput regimes, three are frontier-recoverable strict blockers, and one remains weak selective value under the current benchmark scaffold. Accordingly, 10 of the 11 pairs are marked supported with scope in the frontier audit, although the strict final gate can behave much more conservatively. In particular, Edge-IIoTset/CatBoost, Edge-IIoTset/RF-GPU, and TON_IoT/CatBoost release less than 1% of cases under the final mechanism gate even though their best frontier thresholds reach release rates between 0.4463 and 0.5838. A dedicated threshold-plateau audit shows that this blocker-like behavior is stable within the current frozen suite rather than the result of one arbitrary high threshold: all 11 pairs are unchanged from $\rho \geq 0.67$ through 0.95, all 11 gain release somewhere below 0.67, and the best frozen coverage-quality point occurs at $\rho = 0.20$ for every pair. The result should therefore be read at two levels: the frozen final gate is sometimes blocker-like by design, while the frontier tables still show where the same artifact family can support a bounded triage interpretation.

To make the contract record concrete without exceeding frozen artifact provenance, consider a released row from the CICIoT2023/CatBoost contract table. For evaluation sample 50772, the oracle and selected action are both A_2 (escalate). The stored action posterior assigns $p_A(A_2 | x) = 0.6023$ and $p_A(A_0 | x) = 0.3170$, the conformal quantile is $q_\alpha = 0.001248$, and the conformal action set is the singleton $C_\alpha(x) = \{A_2\}$. The stored stability values are $\rho_{\text{act}} = \rho_{\text{set}} = 1.00$, and the row carries a support index budget of 10 entries. This is enough to show what the released record exposes: the selected action, the plausible action set, the stored stability summaries, and the fact that the final status passed the contract gate. We intentionally avoid attaching additional semantic support group labels, probe count statistics, or feasible contrast claims here because those details are not directly traceable from the frozen contract parquet.

6.4. Selective Gate Tuning and Baseline Comparison

With the mechanism shown to be non-degenerate, the next question is whether tuned TRACE adds measurable value relative to simpler selective release rules. Table 14 reports the held-out sample split comparison between tuned TRACE and simple selective gates. For six of the 11 pairs, TRACE matches the best simple gate objective on the held-out evaluation half. Two additional pairs improve over fixed TRACE but still trail the best simple gate, and the remaining three pairs stay weaker than the best simple gate even though they still outperform unconditional and random matched release. Averaged across the 11 pairs, tuned TRACE ties with the best simple gate on 6 pairs, trails on 5, and leads on 0; the mean held-out delta is therefore slightly negative under the risk objective sign convention ($\Delta = -0.0056$). By contrast, it improves over fixed TRACE by 0.1490 on average over the 10 pairs with finite fixed comparator deltas, over the all-action policy by 0.1475 on average across all 11 pairs, and over release rate-matched random release by 0.1459 on average across all 11 pairs. The appropriate conclusion is narrower than a blanket superiority claim: TRACE adds bounded measurable value over fixed, unconditional, and random matched-release policies, while simple selective gates remain competitive and must be reported as serious baselines. The comparison is pairwise and conditional on the frozen split and release grid, not a claim that the mean delta alone settles the comparison class.

The selective gate baseline summary gives the same message from a different angle. Six pairs demonstrate parity with tuned confidence or sharpness gates, while five are weaker than the best simple gate. Those weaker cases are not hidden by the average: CSE-CIC-IDS2018/CatBoost, Edge-IIoTset/RF-GPU, Edge-IIoTset/XGBoost, TON_IoT/CatBoost, and TON_IoT/RF-GPU all favor the best simple gate on the held-out split. This clarifies the operational boundary of the method: TRACE functions as an action contract framework that exposes release conditions, feasibility violations, action set coverage, and non-release decisions, rather than a universal replacement for every selective threshold.

Table 14. Held-out sample split gate tuning. Positive deltas indicate lower risk objective for tuned TRACE than the comparator (*n/a* indicates the fixed comparator violated feasibility constraints, resulting in a non-finite objective). The random comparator is release rate-matched random release. Model names abbreviate the frozen B11 artifacts.

Dataset	Model	Δ Best Simple	Δ Fixed	Δ All-Action	Δ Random
CICIOT2023	CatBoost	0.000	0.000	0.161	0.156
CICIOT2023	RF-GPU	0.000	0.000	0.226	0.225
CICIOT2023	XGBoost	0.000	0.000	0.200	0.201
CSE-CIC-IDS2018	CatBoost	−0.007	0.049	0.089	0.088
CSE-CIC-IDS2018	RF-GPU	0.000	0.239	0.125	0.120
CSE-CIC-IDS2018	XGBoost	0.000	0.231	0.111	0.104
Edge-IIoTset	CatBoost	0.000	n/a	0.047	0.046
Edge-IIoTset	RF-GPU	−0.010	−0.010	0.172	0.173
Edge-IIoTset	XGBoost	−0.016	−0.016	0.059	0.058
TON_IoT	CatBoost	−0.004	−0.017	0.132	0.132
TON_IoT	RF-GPU	−0.025	1.014	0.299	0.300

In addition to simple selective gates, we reproduced a DeepXAI-IDS-style baseline as a separate predictive reference. These runs confirm that TRACE is not positioned solely against weak detectors. However, the deep architecture is not treated as a fully matched action contract table participant, because the reported contract tables materialize the frozen

gated tree ensemble suite. The results indicate that predictive strength alone does not answer the proxy triage release question. The core evidence demonstrates when release is supported and when the evidence stack instead returns non-release.

6.5. Frontier Practicality and Utility-Scaffold Sensitivity

The gate tuning evidence concerns one frozen operating point, so the analysis then asks whether the same artifacts support wider operating regions and whether the reading survives alternative utility encodings. The strict final mechanism gate is intentionally conservative, so benchmark-scoped triage usefulness should be read from the selective frontier together with the frozen-gate release table, not from a single release rate column. Under the frontier audit, 10 of the 11 G3-passing dataset–model pairs are classified as *supported with scope*, while CSE-CIC-IDS2018/XGBoost is flagged as *weak selective value*. The best frontier threshold by the coverage-quality criterion is $\rho \geq 0.20$ for all 11 pairs, and the corresponding best-release rates span 0.418 to 0.824. This does not replace the stricter final gate, whose purpose is to report what survives the frozen release discipline. It indicates that the same contract artifacts expose a wider operating region on most pairs, so the low-throughput strict gate behavior should be interpreted as one conservative point on a frontier rather than as the only regime available in the artifacts.

Table 15 makes that practical split explicit. Four pairs are triage-capable with scope, one is moderate throughput with scope, two are cautious low-throughput regimes, three are frontier-recoverable strict blockers, and one remains weak selective value. The blocker category deserves attention because those rows release at most 0.6% of cases under the strict final gate, yet their best frozen frontier points still reach release rates between 44.6% and 58.4%. The strict gate acts as a conservative evidence filter, while the frontier shows where the artifact family supports bounded triage-style use.

Even under that wider reading, the practical boundary remains real. The weak-selective-value row and the cautious low-throughput categories show that some dataset–model pairs still lack an attractive throughput-versus-risk tradeoff under the current scaffold. The results therefore indicate bounded practical usefulness on most pairs, not universal triage readiness.

We also tested how dependent the proxy utility reading is on the specific frozen off-diagonal penalties in Table 3. A 5×5 sensitivity sweep rescales underreaction and overreaction penalties over $\{0.25, 0.5, 1, 2, 4\}$ while preserving the action ordering. Across the four research datasets, the minimum observed TU_{raw} over all tested model–matrix combinations remains positive: 0.577 for CICIoT2023, 0.709 for CSE-CIC-IDS2018, 0.847 for Edge-IIoTset, and 0.652 for TON_IoT. The smoke dataset intentionally fails this stability gate. This sweep does not validate the benchmark utility matrix as a SOC cost model, but it does show that the bounded positive-utility interpretation of the frozen contract suite is not created by one single arbitrary penalty scale.

To address action-map sensitivity, we performed an oracle-side severity-rebinning audit on the 11 G3-passing pairs while holding the TRACE predictions fixed. The first alternate map relaxes severity by demoting A_2 to A_1 and most A_3 labels to A_2 , while the second tightens severity by upgrading A_1 to A_2 and A_2 to A_3 . Under these two alternate rebinnings, the mean pairwise all-row utility remains positive on all 11 pairs. The frozen mean pairwise all-row utility is 0.8397; it decreases to 0.7025 under the relaxed downshift and to 0.5402 under the strict upshift, with mean oracle shift rates of 0.2009 and 0.3847, respectively. In ultra-low-release regimes, pass-only row utility becomes conservative: TON_IoT/RF-GPU becomes slightly negative on pass row utility under both alternate rebinnings, and Edge-IIoTset/RF-GPU drops to -0.3500 under the strict upshift while releasing only 0.25% of rows. This confirms that the positive all-row utility is robust to the

specific action severity encoding, although it remains an evaluation-side audit as detectors are not retrained under alternate action targets.

Table 15. Operating region summary over the 11 G3-passing dataset–model pairs. Counts and release ranges come from the frozen operating region audit and characterize practical throughput across the evaluation suite.

Operating Region Label	Pairs	Strict Release Rate	Characterization
Triage-capable with scope	4	32.1–35.0%	Shows bounded triage utility under both the strict gate and the frontier.
moderate throughput with scope	1	20.4%	Provides bounded triage with non-trivial release and substantial deferral.
Cautious low-throughput	2	2.1–2.5%	Operates as explicit low-throughput triage with strong deferral.
Frontier-recoverable strict blocker	3	0.0–0.6%	The strict final gate limits release, but optimal frontier points reach 44.6–58.4% release.
Weak selective value	1	39.8%	Shows non-trivial release without a strong throughput-versus-risk advantage.

6.6. Disagreement and Hard Slice Governance Profile

To see where that selective behavior actually changes decisions, the analysis turns to the disagreement profile and isolates the near-boundary hard slices. TRACE adds a bounded governance signal by deferring cases that simple gates would otherwise release, particularly in lower-stability regions (Figure 3). On the pooled full set, TRACE defers while a tuned simple gate releases 22.2% of cases, whereas the reverse pattern occurs for 11.1% of cases; both methods release 28.1% of cases and both defer 38.6%. The artifacts do not materialize action severity disagreement because simple gates reuse the same underlying action. However, when evaluating on hard slices that isolate a more contentious near-boundary subset, pooled disagreement rises from 0.335 on the full set to 0.622 on the hard slice. Over the same transition, TRACE release falls from 0.387 to 0.353, while the simple-gate release share remains high (0.501 to 0.497), and the TRACE hard slice risk objective increases to 0.0277 (Figure 4). This makes the governance contrast clearer than on the full frozen set, reinforcing that TRACE’s value lies in cautious containment near decision boundaries rather than in unconditional risk dominance.

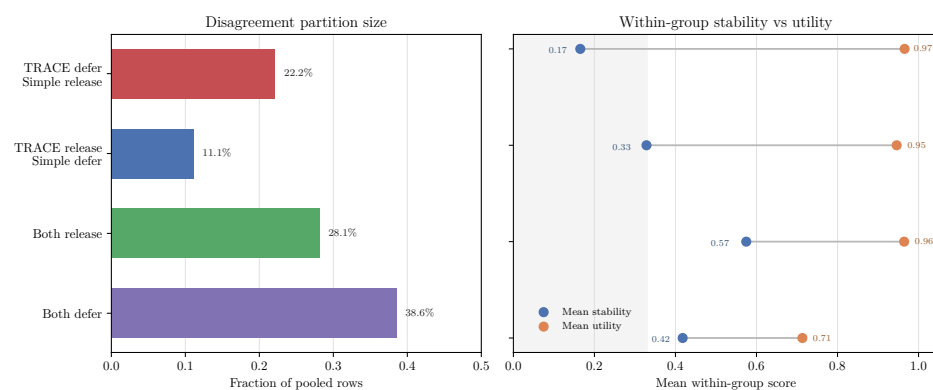


Figure 3. Disagreement profile between TRACE and tuned simple selective gates. The figure shows regions where TRACE defers cases that simple gates would release.

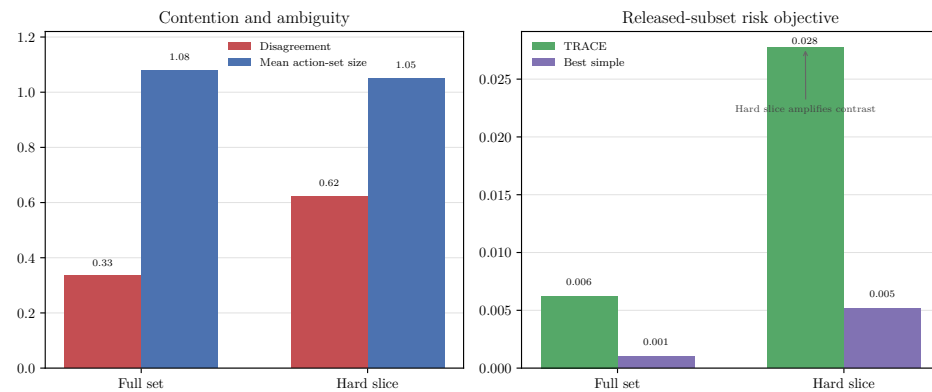


Figure 4. Hard slice profile isolating near-boundary subsets. The hard slice demonstrates substantially higher TRACE-versus-simple disagreement, a lower TRACE release share, and a more contentious decision region than the pooled full set.

6.7. Feature-Space Feasible Versus Generic Probes

The governance evidence rests on feature-space feasible probes, so the next diagnostic checks whether the feasibility restriction materially affects the stability evidence. The paired probe-mode comparison evaluates the same 11 G3-passing pairs under feature-space feasible and generic perturbation modes. Mean changes from generic minus feature-space feasible probes are small for the observed stability metrics: $\Delta\rho_{\text{act}} = 0.0006$, $\Delta\rho_{\text{set}} = 0.0033$, and $\Delta\text{release rate} = 0.0048$. While generic probes exhibit comparable aggregate stability, the feasibility diagnostics reveal significant violations: generic probes have an Ω_D row violation rate delta of 1.000 on every pair, and mean feature violation rate deltas range from 0.442 to 0.947. Thus, generic probes achieve stability scores through inadmissible rows. The feature-space feasible protocol is necessary to prevent invalid counterfactual evidence from incorrectly licensing action release.

6.8. Ablation and Threshold Sensitivity

With the diagnostic thread established, the ablation isolates which contract components actually drive the observed release behavior. Table 16 reports the per-component effects on release behavior, and Figure 5 visualizes the corresponding release rate deltas. The clearest signal comes from the explanation-coupling family that removes the conformal action set (V1) or the release gate (V3), approximating a post hoc action display without full contract discipline. That family yields mean absolute release rate deltas of 0.818 on the full suite and 0.974 on the hard slice, so its verdict is strongly supported. The remaining families are weaker but are now more precisely localized by a boundary ablation audit over all 8800 frozen contract rows plus direct near-boundary reruns across seeds and probe budgets. The strict stability and shallow verifier variants (V4 and V6) do not move the current release frontier at all: both change zero released rows, zero chosen actions, and zero expected-utility entries, while 2,096 raw status strings simply collapse from `downgrade_or_defer` to `defer` on already non-released rows. This dormancy has a measurable operating region explanation rather than being a vague null result: 7198 rows lie below stability 0.67, 1602 lie at or above 0.90, and the current suite contains zero rows in the intermediate bands [0.67, 0.90) where relaxed strict stability or verifier-depth rules could have shifted release. The targeted reruns do not overturn that reading: across the 10 frontier/disagreement pairs selected for V4/V6 stress, the mean V4 and V6 release-flip rates remain 0.000. The minimax variant (V5) likewise leaves release rates unchanged, but its effect is not null. It changes the chosen action on 332 of 454 multi-action rows (73.1%), with 319 of those changes occurring on rows that TRACE already keeps non-released and only 13 on released rows; moreover, only 4 changed rows fall inside the existing revision hard slice. The dominant transition

is from escalate to contain on 191 rows. That localized reading is reinforced rather than weakened by the reruns: on the ambiguity-targeted TON_IoT/RF-GPU batch, all 12 selected rows retain an V5 action flip rate of 1.000 across the executed seed and probe budget grid, while V4/V6 remain at 0.000. The correct reading is therefore a localized ambiguity slice action-selection effect rather than a broad release-shift effect. The support feasible search family remains only partially supported because its main empirical signal still comes from the paired generic-versus-feasible probe diagnostics, and the current suite does not yet populate richer near-boundary cases where deeper verifier/search logic would visibly move the release frontier.

Table 16. Per-component effects of the contract-layer ablation on release behavior and action selection.

Component	Release Rate Effect	Action Selection/Operating Region Effect
Conformal action set (V1)	Mean absolute Δ 0.818 (full suite), 0.974 (hard slice)	Largest shift; post hoc display regime
Release gate (V3)	Large Δ , same family as V1	Post hoc display regime
Strict stability gate (V4)	0.000 (no released row moves)	No rows in the intermediate stability band [0.67, 0.90)
Minimax utility (V5)	0.000 (release unchanged)	Shifts the chosen action on 73.1% of multi-action rows
Verifier depth (V6)	0.000 (no released row moves)	No rows near the release boundary at the current frontier

The threshold-plateau audit sharpens the same boundary. Across all 11 G3-passing pairs, release rate, released-set action agreement, and coverage-quality index are identical from $\rho \geq 0.67$ through 0.95. The transition region is therefore not spread across the upper thresholds: no pair changes anywhere inside that high-threshold band, whereas all 11 pairs change between 0.50 and 0.67, and all 11 recover additional release somewhere below 0.67. The best frozen coverage-quality point also occurs at $\rho = 0.20$ for every pair, with strict-to-frontier release gain ranging from 0.2700 to 0.5838. The blocker-like strict gate behavior is a stable operating region property of the current frozen suite rather than noise from a single arbitrary threshold choice. At the same time, the audit does not validate $\rho = 0.67$ as a universal or deployment-optimal threshold. It indicates that richer support contracts, deeper feasible search, and near-boundary records are required to improve the final frontier.

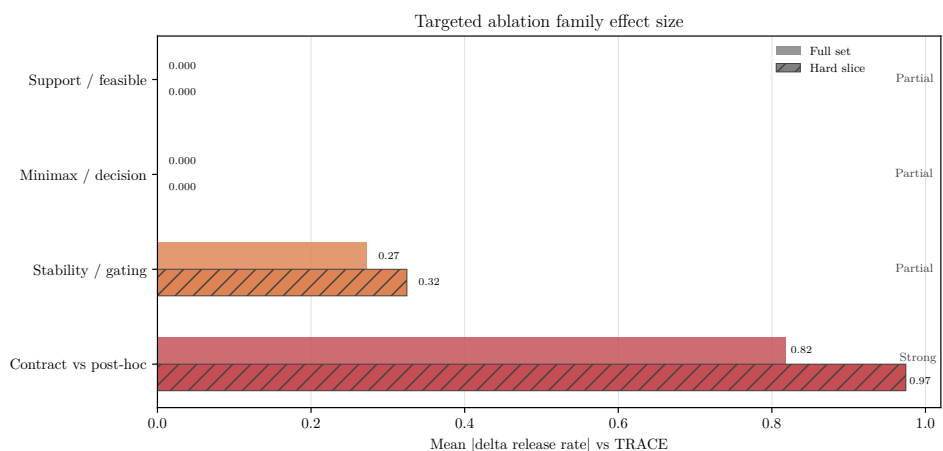


Figure 5. Ablation family effects showing the impact of removing different components of the TRACE contract layer on the release rate. Large release rate deltas (V1, V3) indicate components that materially change release behavior. Exact zero bars indicate zero observed release rate delta at the current operating point; they do not imply zero utility or feasibility effect.

6.9. Contract Taxonomy and Confidence Intervals

The final stage closes the thread by aggregating the contract record across dataset roles and testing its statistical stability through bootstrap confidence intervals. The contract taxonomy is summarized in Table 17 and Figure 6, presenting pooled workload evidence. The six primary artifacts have mean pass rate 0.175, mean defer rate 0.653, mean conformal set size 1.008, and mean oracle-in-set rate 0.867. The five secondary-transfer artifacts have mean pass rate 0.190, mean defer rate 0.492, mean conformal set size 1.164, and mean oracle-in-set rate 0.933. In both groups, the mean stored support index size remains fixed at 10.0 in the reported artifacts. The summary quantifies contract burden, selectivity, and uncertainty sharpness.

Table 17. Aggregate support contract audit by dataset role. The support index size remains fixed at 10 in the frozen reported artifacts, quantifying contract burden and selectivity.

Dataset Role	Pairs	Mean Pass Rate	Mean Defer Rate	Mean Set Size	Mean Support Size	Mean Oracle-in-Set
Primary	6	0.175	0.653	1.008	10.0	0.867
Secondary transfer	5	0.190	0.492	1.164	10.0	0.933

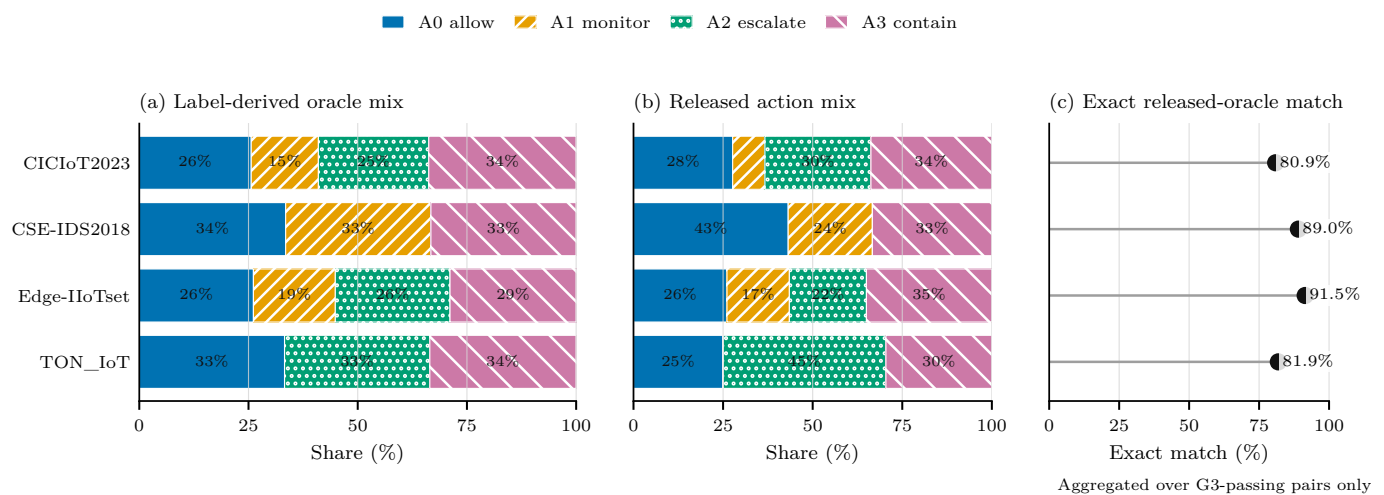


Figure 6. Dataset-level contract action distributions aggregated over the 11 evaluated dataset-model pairs. The panels compare the label-derived oracle action mix, the actions released under TRACE, and the exact released-oracle agreement rate.

Bootstrap intervals (visualized in Figure 7) are available for 77 metric rows, covering seven contract metrics per pair with 500 resamples. For example, CICIoT2023/CatBoost has release rate 0.350 with 95% confidence interval [0.318, 0.383], and action agreement 0.849 with 95% confidence interval [0.824, 0.871]. At the summary level, the median 95% interval width is 0.0525 for release rate and 0.0444 for action agreement. These intervals expose point-estimate uncertainty under contract-row resampling, distinguishing measured selectivity from sampling noise. To move beyond row bootstrap alone, we also reran the frozen contract layer on 32 sampled rows per G3-passing pair across three probe seeds with 16 feasible probes per rerun, yielding 96 repeated evaluations per pair. In that medium-strength audit, the selected action stayed unchanged on every sampled row across all 11 pairs, while row-level status-change rates ranged from 0.000 to 0.146 depending on the pair. Action choices remain stable on the sampled rows under probe seed variation, but release/defer status can still shift modestly near the contract boundary. This strengthens the uncertainty evaluation beyond simple row resampling.

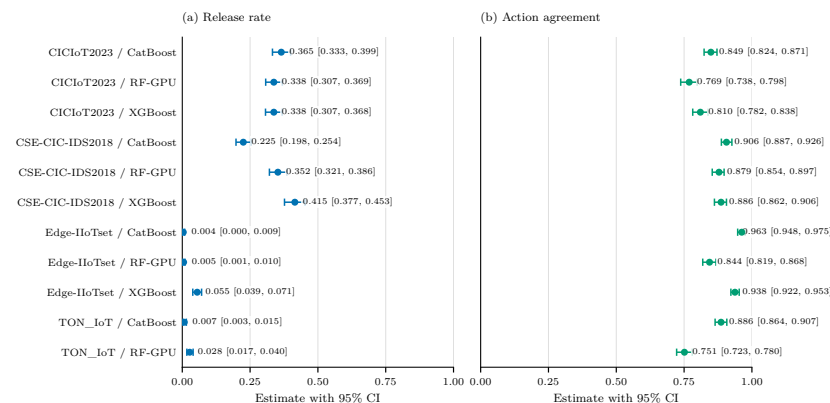


Figure 7. Bootstrap 95% confidence intervals for release rate and action agreement across the 11 evaluated dataset–model pairs, using 500 resamples per pair.

The evaluation framework supports five scoped outcomes. First, it supports coverage-aware proxy triage on gated contracts. Second, it reports a paired feature-space feasible-versus-generic probe comparison. Third, it records cases where contract gates defer or downgrade fragile proxy containment decisions under the specified action schema. Fourth, it measures action-level instability and release/defer sensitivity. Finally, it supports bounded baseline comparisons: while matched-coverage and sample split gate tuning are credible, TRACE’s selective value is not uniformly better than the best simple selective gate.

7. Discussion

The main empirical message is that action-governed XAI differs from explanation display: as motivated in the Introduction, attribution summaries do not by themselves decide whether a security action should be released. TRACE shifts the evaluated object from an attribution vector to an action contract: a finite record containing the plausible action set, minimax action, support contract, feasible probe model, stability scores, and release status. That shift is the paper’s conceptual contribution.

The main positive result is the validity of the mechanism. The final contracts avoid full-set collapse, produce sharp conformal action sets, and release only a selective subset of cases. These outcomes ground the method in observed behavior rather than theoretical design alone. However, tuned TRACE does not uniformly outperform the best simple selective gate; across 11 pairs, it matches the best simple gate on six held-out rows and is slightly lower on average by the mean held-out risk objective delta (−0.0056). The findings demonstrate TRACE’s utility as an action contract framework rather than a universal classifier. The contract record exposes uncertainty, admissibility, and non-release on the subset where a simpler gate would still release.

This boundary matters because a simple confidence gate may be enough when the only goal is to release low-error predictions at a fixed coverage level. TRACE becomes useful when the reader also needs to know why an action is released, whether it remains stable under admissible local alternatives, whether generic perturbation evidence is invalid under Ω_D , and whether the system records non-release as a first-class outcome. The contribution therefore is larger than a scalar score: it defines an evidence protocol for deciding when an explanation-supported action should stay blocked. If two methods achieve similar selective risk, TRACE still changes the observable object by recording which actions remained plausible, the observed stability and feasibility summaries, and whether the case ended in release or non-release. The negative comparisons against the best simple gate therefore do not invalidate the method; they show that TRACE has to justify its added complexity through auditable contract information, not only through the mean risk objective number.

Accordingly, a released contract should be interpreted as an auditable benchmark-time triage recommendation with explicit uncertainty and feasibility checks, not as proof that a human analyst would endorse the same action in live SOC operations.

7.1. Beyond Auditability: The Utility, Selectivity, and Interpretability Trade-Off

On the scalar utility axis, the selective gate comparison does not separate TRACE from the best simple confidence threshold: at matched release grids it matches on six of eleven pairs and trails on five (mean held-out risk objective delta -0.0056). On that axis alone, a confidence threshold remains a strong baseline, and the case for TRACE rests on auditability. The contribution beyond auditability is better read as a trade-off across three axes than as a win on any single one. A simple selective gate optimizes one axis, scalar selective risk at a threshold, and collapses every decision to release or defer. TRACE operates on all three at once: it resolves the action through a utility-minimax rule over the conformal set instead of a binary threshold, selects where to defer using boundary-aware and feasibility-constrained stability, and emits a contract record that makes the decision interpretable. Table 18 summarizes this trade-off.

At matched scalar risk the value of the additional axes is clearest. On the six parity pairs, TRACE and the best simple gate achieve similar risk objectives, but TRACE also records which action was plausible, whether it survived admissible local alternatives, and whether non-release was the safer outcome, none of which a threshold gate exposes. On the five pairs where TRACE trails, the scalar deficit is the cost of carrying that multi-axis structure: the disagreement and hard slice profiles (Section 6) show that TRACE defers cases a simple gate would still release, concentrating its caution near decision boundaries where the cost of an unsafe release is highest. This is a deliberate selectivity profile rather than noise, since hard slice disagreement rises to 0.622 against 0.335 on the full set while the TRACE release share falls and the simple-gate share does not.

For use cases that need action-level resolution, boundary-aware selectivity, and decision interpretability at the same time, a single-axis threshold is structurally insufficient, and the contribution of TRACE is to provide all three within one auditable record. The scalar-risk parity or small deficit of TRACE on the utility axis is the cost of carrying that multi-axis structure. This is a benchmark-scoped trade-off argument rather than a claim of operational superiority over all selective gate designs.

Table 18. The utility, selectivity, and interpretability trade-off between a simple selective gate and TRACE. The TRACE cells are factual readings of the reported artifacts, not superiority claims.

Axis	Simple Selective Gate	TRACE	Evidence
Scalar utility	Competitive; wins or ties at the threshold	Matches best gate on 6/11 pairs, trails on 5/11 ($\Delta = -0.0056$)	Table 14
Selectivity	Binary release/defer at one threshold	Boundary-aware; defers 22.2% of gate-released cases (hard slice disagreement 0.622)	Figures 3 and 4
Action resolution	None (release/defer only)	Four-action utility-minimax over the conformal set	Sections 3 and 4
Decision interpretability	Threshold crossing	Contract record (plausible actions, minimax action, support, feasibility, status)	Equation (1)
Feasibility-awareness	None	Generic probes inadmissible under Ω_D (row violation delta 1.000)	Section 6

7.2. Sensitivity to the Utility Matrix and Action Map

Both the utility matrix $U(b, a)$ and the label-to-action map ϕ_D are benchmark scaffolds rather than validated operational objects (Section 3), so a natural question is how much the reported reading depends on the specific choices frozen in Tables 2 and 3. Two evaluation-side audits in Section 6 bound this dependence. A 5×5 penalty rescaling sweep varies the underreaction and overreaction penalties over the set $\{0.25, 0.5, 1, 2, 4\}$ while preserving the action ordering, and the minimum raw proxy utility stays positive on all four research datasets. An oracle-side action severity rebinning audit then relaxes and tightens the severity tiers with predictions held fixed, and the mean pairwise all-row utility stays positive on all eleven pairs, decreasing from the frozen 0.84 to 0.70 under the relaxed downshift and to 0.54 under the strict upshift. The bounded positive-utility reading is therefore not an artifact of one penalty scale or one severity encoding.

The scope of this finding is set by the audit design. Both audits hold the TRACE predictions fixed, so they characterize the sensitivity of the evaluation reading rather than of a detector retrained under alternate targets. The audits reveal marginal fragility: under the strict severity upshift one pair drops to a negative pass row utility while releasing only a small fraction of cases.

The two audits vary the penalty scale and the severity encoding but hold the matrix asymmetry, the action ladder granularity, and the label-to-action mapping fixed. A more asymmetric matrix would shift the operating point toward deferral; characterizing that regime and the effect of alternate label-to-action mappings requires analyst-elicited utilities and retraining under each alternate target, which is left to future work.

The result pattern also clarifies what a comparison against prior IDS/XAI systems would require. Existing XAI-IDS methods often report predictive metrics, explanation fidelity, or explanation stability under their own feature and split assumptions. A comparative evaluation against TRACE requires alignment at the action contract layer: the same label-to-action map, utility matrix, split-conformal calibration, feature-space feasible perturbation card, and release/defer rule. Without that alignment, a detector with higher macro-F1 and a TRACE contract with lower release risk are answering different questions.

The ablation findings clarify the impact of individual components. The conformal action set and release gate materially affect release behavior. In the evaluated operating region, V4 and V6 remain dormant at the release frontier, as no rows occupy the intermediate stability bands where those stricter rules could change release. By contrast, V5 shows a localized ambiguity slice action effect concentrated on multi-action rows that are mostly non-released. Accordingly, the conformal action set and release gate are the primary drivers of release behavior, whereas the strict stability, minimax, and verifier-depth components leave the release frontier unchanged at the current operating point; Table 16 reports these per-component effects. Future work should strengthen support set construction, run deeper feasible search, and evaluate intermediate-band records.

From a deployment perspective, the most important design implication is that deferral is not failure. In a security setting, the absence of sufficient contract support should be represented explicitly because unsafe release can be worse than non-release. The low-release regime is only part of the picture: the operating region audit classifies four pairs as triage-capable with scope, one as moderate throughput with scope, two as cautious low-throughput, three as frontier-recoverable strict blockers, and one as weak selective value. This classification separates conservative final gate behavior from the broader frozen frontier evidence. Even so, several dataset-model pairs behave more like blocker filters than triage assistants. At the extreme, Edge-IIoTset/CatBoost releases no cases at all under the final mechanism gate, and Edge-IIoTset/RF-GPU releases only 0.25% of cases.

When deferral rates approach unity, the contract framework provides auditability but little actionable decision throughput.

The new pooled support contract audit sharpens a different boundary. Across the 11 final pairs, the mean conformal set size is only 1.08, indicating that most stored action sets are already close to singleton decisions, while the stored support index cardinality is fixed at 10 entries for every pair in the reported artifacts. The empirical variance therefore appears mainly in the release/defer/downgrade status mix rather than in adaptive support-size shrinkage. This is still useful because it documents how often the contract layer blocks or downgrades actions under the frozen evidence stack, so we do not claim rich support contract size adaptation. Future work should characterize the operating region where TRACE transitions from a blocking evidence filter to a useful triage assistant and should expose more varied support contract burdens under deeper feasible search.

The proposed TRACE system provides a reproducible action contract layer for IDS explanations that avoids degenerate certainty, exposes infeasible counterfactual evidence, and enforces feature-space feasibility constraints. The results demonstrate the utility of an action contract account of IDS explanations, though they do not demonstrate universal superiority over all IDS/XAI methods or adaptive attacker robustness. Validating deployment readiness would require packet-level feasibility validation, independently aligned benchmarking, adaptive-adversary testing, and human/SOC validation of the action ladder and utility model.

8. Threats to Validity

8.1. Threat Model Realism

The threat model is a benchmark-time decision-support model. The defender has a trained IDS, frozen preprocessing, a label-to-action map, and a finite set of candidate triage actions. The adversarial or uncertain environment is represented through feature-space alternatives inside Γ_D and through generic probes used only for comparison. Under a Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE)-style reading, the paper touches tampering-like perturbations of feature records, spoofing-like ambiguity in benchmark labels, information quality risks in explanations, and denial-of-service-like analyst overload from over-release. It does not model live lateral movement, packet crafting, command-and-control interaction, adversary dwell time, SOC queue dynamics, or human response behavior. Under a kill chain reading, the evaluation sits after detection and before response selection; it is not an end-to-end intrusion campaign simulator.

This boundary strictly limits security claims. A released TRACE action means that a benchmark record passed the specified conformal, utility, feasibility, and contract gates under the processed representation. It does not mean that the action would be safe in a production SOC, legally appropriate, or effective against an adaptive attacker. This study formulates the problem as a proxy triage task rather than an autonomous containment system.

The assumed attacker knowledge is also limited. A critical limitation of this work is the absence of a full adaptive attacker evaluation. The preliminary tabular FGSM-style diagnostic in the artifact package does not grant an adversary white-box access to the detector, the conformal calibration set, the utility matrix, or the release thresholds, and it does not uniformly reduce attack success across datasets. The experiments also do not simulate an attacker who optimizes packet traces to pass Ω_D while changing the released action. This leaves a gap between our tabular-feature-feasible representation and true packet-level payload realizability. Because TRACE relies on explanations to govern triage actions, it also inherits the risk of adversarial explanations (e.g., scaffolding attacks where an adversary manipulates the input such that the classifier's output remains malicious,

but the explanation points to benign features). An adaptive attacker might specifically target the explanation layer to force a triage downgrade or deferral, bypassing the action governance entirely even if the underlying detector remains accurate. The evaluated uncertainty therefore reflects local ambiguity and representation-level fragility rather than a full adversarial machine learning game. Consequently, robustness claims are strictly limited to tabular-feature stability, and any deployment would require testing against adaptive, evasion-optimized adversaries targeting both the predictive and explanatory layers.

8.2. Construct, Internal, and External Validity

Construct validity is limited by the label-to-action map. Public IDS labels are not SOC playbooks, and the oracle action a^* is a benchmark scaffold derived from attack-label semantics. We mitigate this by freezing ϕ_D , using conservative action tiers, reporting action agreement separately from detector accuracy, and avoiding claims about analyst trust or operational correctness. A bounded oracle-side severity-rebinning audit further shows that the all-row proxy utility reading stays positive under two alternate rebinnings of the frozen test labels, although pass-only utility can become fragile in ultra-low-release regimes. Because that audit holds predictions fixed, it strengthens construct validity only modestly and does not replace retraining under alternate action targets. The utility matrix is likewise a proxy cost model, not an empirically elicited SOC utility function.

Internal validity is threatened by leakage, threshold tuning, and post hoc comparator selection. The pipeline mitigates leakage through split-before-fit preprocessing, train-only reference objects, frozen test matrices, and gate checks over detector materialization and contract records. The selective gate comparison mitigates post hoc optimism by tuning on one half of the contract sample and evaluating on the other half under the same release grid and risk objective. The fairness audit further requires each executed comparator to use the same sample eligibility, action map, metric definitions, and release budget. Nonetheless, the sample split is still within benchmark artifacts, and the paper does not treat it as a replacement for external validation. The current robustness claims are also bounded by contract-row bootstrapping plus a frozen detector probe seed audit; they still do not incorporate detector retraining, cross-split, or hierarchical uncertainty propagation to account for variance in model initialization, data partitioning, or full pipeline refitting.

External validity is limited by public benchmark age, feature abstractions, and processed split structure. The formal split audit clarifies this boundary: within the frozen processed metadata, CICIoT2023 and TON_IoT use group disjoint train/validation/test splits, CSE-CIC-IDS2018 reuses groups across splits, and Edge-IIoTset exposes only one usable group. This indicates processed split realism for the first two datasets, benchmark-only validity for CSE-CIC-IDS2018, and weak shift characteristics for Edge-IIoTset. None of these audit outcomes establishes temporal, live-network, or deployment-valid OOD evidence. The results provide multi-benchmark evidence for an action contract protocol, not as deployment evidence across arbitrary networks.

8.3. Conclusions, Security, and Dual-Use Validity

Conclusion validity is strengthened by bootstrap intervals, frozen detector probe seed variability, paired probe comparisons, ablations, and a claim readiness ledger, but several comparisons remain bounded. The reproduced DeepXAI-IDS reference addresses the need for a strong modern detector context, but the reported tables do not materialize fully matched action contract rows for that architecture. It is therefore a predictive baseline. TRACE is an evidence contract framework layered on top of strong detector foundations. It also does not prove that feature-space feasible probes are universally better than generic probes. The evidence shows that generic probes violate Ω_D on all rows in the paired

comparison, while aggregate action stability deltas are small in this run. The new probe seed audit strengthens that statement by showing no sampled action flips across the executed repeated evaluations, but it still does not close retraining or external-shift uncertainty. This matters for alert interpretability because an explanation that appears stable only under infeasible perturbations can create false analyst confidence rather than trustworthy triage evidence.

8.4. Limitations: Tabular Boundaries

Our empirical evaluation and the phrase “feature-space feasible” are bounded to the tabular features provided by the benchmark datasets. We do not generate raw network traffic (PCAP) to verify that a mathematical perturbation in the dataset maps perfectly to a realizable malicious TCP/IP packet that bypasses hardware firewalls or IDS engines. As such, the feasibility constraints model feature-level realism but do not constitute a full network-level evasion proof. Readers should interpret these action contracts as decision-layer triage tools, not deployment-ready network filters.

Security validity is limited by the feature-space representation. A tabular vector may encode traffic rates, counts, flags, or categorical protocol values, but feasibility in that vector is not equivalent to packet-level realizability. Dual-use risk is also real: contrast search and near-boundary diagnostics could be misused if released as step-by-step evasion guidance. The reproducibility package exposes aggregate metrics, code for benchmark verification, and feasibility card construction while avoiding operational attack recipes beyond what is necessary for scientific audit. The paper also avoids publishing per-record evasion instructions, exploit chains, or deployment-specific containment playbooks.

The benchmark scope also limits evaluation of fairness, inclusion, and organizational equity. Public IDS datasets do not carry the human, demographic, workflow, or institutional metadata required to evaluate demographic fairness, inclusive usability across diverse analyst roles and expertise levels, or equitable transfer across organizations, sectors, and operating regions. This study makes no claims regarding fairness, accessibility, or cross-organization equity, and any such evaluation would require deployment-context studies, analyst-population sampling, and policy-level validation that lie outside the present evidence.

9. Conclusions

TRACE treats explainable intrusion triage as a contract problem: the detector proposes, the conformal action set bounds plausible alternatives, a utility-minimax rule picks the action, and feasibility-checked support contracts decide whether release is earned. The conformal set and release gate stages are the components demonstrated most clearly by the ablation studies. The final artifact provides an auditable release, defer, downgrade, or fail-open record rather than a standalone attribution display.

The final experiments cover the 11 dataset–model pairs that successfully passed robustness and calibration checks. Across those pairs, TRACE produces non-degenerate action sets with zero full-set collapse, release rates from 0.000 to 0.398, and defer/block rates from 0.603 to 1.000. Sample split gate tuning gives TRACE higher average proxy utility than all-action and release rate-matched random release on all 11 pairs, and higher than fixed TRACE on the 10 pairs with finite fixed comparator deltas. Against the best simple selective gate, however, TRACE only matches on 6 of 11 pairs and trails on the other 5. That bounded utility reading survives the 5×5 penalty rescaling sweep and two alternate oracle-side action severity rebinnings with predictions held fixed, although pass-only utility becomes fragile in ultra-low-release regimes, and the rebinning audit does not stand in for retraining under different action maps.

The operating region audit complicates those headline numbers without contradicting them. It marks 10 of 11 pairs as supported with scope, but only four pairs read as triage-capable. Three strict gate points still behave more like blocker-style filters than triage assistants. Probe comparisons stay consistent: generic probes are inadmissible under Ω_D on every paired row even when their aggregate stability deltas look small. The frozen detector probe seed audit over 32 sampled rows per pair and three probe seeds per row observed zero sampled action flips across all 11 pairs, with row-level status-change rates between 0.000 and 0.146. That sharpens the uncertainty story beyond row bootstrap alone, but it does not close retraining or cross-split variance. Component-wise, the ablations plus targeted near-boundary reruns confirm the conformal set plus release gate core. Across the remaining variants, V4 and V6 stay frontier-invariant on both the frozen suite and the targeted rerun batches, while V5 shows a localized ambiguity slice action effect rather than a broad release shift.

TRACE provides an evidence framework for action-governed XAI in IDS, demonstrating improved auditability without claiming autonomous security response or uniform superiority over all IDS/XAI methods. Where a simple selective gate matches the release risk, the remaining value of TRACE is the auditable record itself: which actions remained plausible, which evidence was admissible, and why a case did or did not release. The reproduced deep baseline maintains strong predictive context, although it only supports a benchmark-time evaluation for the artifact suite. Future work should close this gap with materialized matched deep action contract tables, packet-level feasibility checks, richer intermediate-band support contract cases, detector-retraining, analyst-facing evaluations, and ongoing checks for model and threat drift under operational data.

Author Contributions: Conceptualization, T.D.L., M.A. and Y.B.; methodology, T.D.L.; software, T.D.L. and Y.B.; validation, T.D.L., M.A. and Y.B.; formal analysis, T.D.L.; investigation, T.D.L. and Y.B.; data curation, T.D.L. and Y.B.; writing—original draft preparation, T.D.L.; writing—review and editing, T.D.L., M.A. and Y.B.; visualization, T.D.L.; supervision, M.A.; project administration, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code and supporting data for this study are available in the project repository at <https://github.com/tranducle/TRACE> (accessed on 23 July 2026). The benchmark datasets (CICIoT2023, Edge-IIoTset, CSE-CIC-IDS2018, and TON_IoT) are publicly available from their respective original sources.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sommer, R.; Paxson, V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010; pp. 305–316. <https://doi.org/10.1109/SP.2010.25>.
2. Apruzzese, G.; Laskov, P.; Schneider, J. SoK: Pragmatic Assessment of Machine Learning for Network Intrusion Detection. In Proceedings of the 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P), Delft, The Netherlands, 3–7 July 2023; pp. 592–614. <https://doi.org/10.1109/EuroSP57164.2023.00042>.
3. Neupane, S.; Ables, J.; Anderson, W.; Mittal, S.; Rahimi, S.; Banicescu, I.; Seale, M. Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities. *IEEE Access* **2022**, *10*, 112392–112415. <https://doi.org/10.1109/ACCESS.2022.3216617>.

4. Moustafa, N.; Koroniotis, N.; Keshk, M.; Zomaya, A.Y.; Tari, Z. Explainable Intrusion Detection for Cyber Defences in the Internet of Things: Opportunities and Solutions. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1775–1807. <https://doi.org/10.1109/COMST.2023.3280465>.
5. Arreche, O.; Guntur, T.R.; Roberts, J.W.; Abdallah, M. E-XAI: Evaluating Black-Box Explainable AI Frameworks for Network Intrusion Detection. *IEEE Access* **2024**, *12*, 23954–23988. <https://doi.org/10.1109/ACCESS.2024.3365140>.
6. Ribeiro, M.T.; Singh, S.; Guestrin, C. Why Should I Trust You?: Explaining the Predictions of Any Classifier. In Proceedings of the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144. <https://doi.org/10.1145/2939672.2939778>.
7. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
8. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From Local Explanations to Global Understanding with Explainable AI for Trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. <https://doi.org/10.1038/s42256-019-0138-9>.
9. Gaspar, D.; Silva, P.; Silva, C. Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron. *IEEE Access* **2024**, *12*, 30164–30175. <https://doi.org/10.1109/ACCESS.2024.3368377>.
10. Angelopoulos, A.N.; Bates, S. Conformal Prediction: A Gentle Introduction. *Found. Trends Mach. Learn.* **2023**, *16*, 494–591. <https://doi.org/10.1561/2200000101>.
11. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An ensemble of autoencoders for online network intrusion detection. *arXiv* **2018**, arXiv:1802.09089.
12. Goodge, A.; Hooi, B.; Ng, S.K.; Ng, W.S. LUNAR: Unifying Local Outlier Detection Methods via Graph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI: Menlo Park, CA, USA, 2022; Volume 36, pp. 6737–6745. <https://doi.org/10.1609/aaai.v36i6.20629>.
13. Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; Pedreschi, D. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* **2018**, *51*, 1–42. <https://doi.org/10.1145/3236009>.
14. Zebin, T.; Rezvy, S.; Luo, Y. An Explainable AI-Based Intrusion Detection System for DNS Over HTTPS (DoH) Attacks. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2339–2349. <https://doi.org/10.1109/TIFS.2022.3183390>.
15. Houda, Z.A.E.; Brik, B.; Khoukhi, L. Why Should I Trust Your IDS?: An Explainable Deep Learning Framework for Intrusion Detection Systems in Internet of Things Networks. *IEEE Open J. Commun. Soc.* **2022**, *3*, 1164–1176. <https://doi.org/10.1109/OJCOMS.2022.3188750>.
16. Sharma, B.; Sharma, L.; Lal, C.; Roy, S. Explainable Artificial Intelligence for Intrusion Detection in IoT Networks: A Deep Learning Based Approach. *Expert Syst. Appl.* **2024**, *238*, 121751. <https://doi.org/10.1016/j.eswa.2023.121751>.
17. Sivamohan, S.; Sridhar, S.S. An Optimized Model for Network Intrusion Detection Systems in Industry 4.0 Using XAI Based Bi-LSTM Framework. *Neural Comput. Appl.* **2023**, *35*, 11459–11475. <https://doi.org/10.1007/s00521-023-08319-0>.
18. Hosain, Y.; Çakmak, M. XAI-XGBoost: An Innovative Explainable Intrusion Detection Approach for Securing Internet of Medical Things Systems. *Sci. Rep.* **2025**, *15*, 22278. <https://doi.org/10.1038/s41598-025-07790-0>.
19. Kalakoti, R.; Nömm, S.; Bahsi, H. Federated Learning of Explainable AI (FedXAI) for Deep Learning-Based Intrusion Detection in IoT Networks. *Comput. Netw.* **2025**, *270*, 111479. <https://doi.org/10.1016/j.comnet.2025.111479>.
20. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A Survey of Network-Based Intrusion Detection Data Sets. *Comput. Secur.* **2019**, *86*, 147–167. <https://doi.org/10.1016/j.cose.2019.06.005>.
21. Elkan, C. The Foundations of Cost-Sensitive Learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*; IJCAI: Vienna, Austria, 2001.
22. Lee, W.; Fan, W.; Miller, M.L.; Stolfo, S.J.; Zadok, E. Toward Cost-Sensitive Modeling for Intrusion Detection and Response. *J. Comput. Secur.* **2002**, *10*. <https://doi.org/10.3233/JCS-2002-101-202>.
23. Fan, W.; Lee, W.; Stolfo, S.J.; Miller, M. A Multiple Model Cost-Sensitive Approach for Intrusion Detection. In *Proceedings of the European Conference on Machine Learning (ECML)*; Springer: Berlin/Heidelberg, Germany, 2000. https://doi.org/10.1007/3-540-45164-1_15.
24. Alsubhi, K.; Al-Shaer, E.; Boutaba, R. Alert Prioritization in Intrusion Detection Systems. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*; IEEE: Piscataway, NJ, USA, 2008. <https://doi.org/10.1109/NOMS.2008.4575114>.
25. Alpcan, T.; Basar, T. A Game Theoretic Approach to Decision and Analysis in Network Intrusion Detection. In *Proceedings of the 42nd IEEE Conference on Decision and Control (CDC)*; IEEE: Piscataway, NJ, USA, 2003. <https://doi.org/10.1109/CDC.2003.1273013>.
26. Manshaei, M.H.; Zhu, Q.; Alpcan, T.; Basar, T.; Hubaux, J.P. Game Theory Meets Network Security and Privacy. *ACM Comput. Surv.* **2013**, *45*, 1–39. <https://doi.org/10.1145/2480741.2480742>.

27. Ustun, B.; Spangher, A.; Liu, Y. Actionable Recourse in Linear Classification. In Proceedings of the Conference on Fairness, Accountability, and Transparency, Atlanta, GA, USA, 29–31 January 2019; pp. 10–19. <https://doi.org/10.1145/3287560.3287566>.
28. Karimi, A.H.; Scholkopf, B.; Valera, I. Algorithmic Recourse. In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual, 3–10 March 2021; pp. 353–362. <https://doi.org/10.1145/3442188.3445899>.
29. Slack, D.; Hilgard, S.; Jia, E.; Singh, S.; Lakkaraju, H. Fooling LIME and SHAP. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, 7–9 February 2020; pp. 180–186. <https://doi.org/10.1145/3375627.3375830>.
30. Barnard, P.; Marchetti, N.; DaSilva, L.A. Robust Network Intrusion Detection Through Explainable Artificial Intelligence (XAI). *IEEE Netw. Lett.* **2022**, *4*, 167–171. <https://doi.org/10.1109/LNET.2022.3186589>.
31. Catillo, M.; Pecchia, A.; Villano, U. Similarity Is Not Enough: Issues with Adversarial Perturbations of Traffic Features against Intrusion Detection Systems. In Proceedings of the 12th International Conference on Information Systems Security and Privacy, Marbella, Spain, 4–6 March 2026; pp. 325–332. <https://doi.org/10.5220/0014605900004061>.
32. Pawlicki, M.; Choraś, M.; Kozik, R. Defending Network Intrusion Detection Systems Against Adversarial Evasion Attacks. *Future Gener. Comput. Syst.* **2020**, *110*, 148–154. <https://doi.org/10.1016/j.future.2020.04.013>.
33. Vitorino, J.; Praça, I.; Maia, E. SoK: Realistic Adversarial Attacks and Defenses for Intelligent Network Intrusion Detection. *Comput. Secur.* **2023**, *134*, 103433. <https://doi.org/10.1016/j.cose.2023.103433>.
34. Vitorino, J.; Oliveira, N.; Praça, I. Adaptive Perturbation Patterns: Realistic Adversarial Learning for Robust Intrusion Detection. *Future Internet* **2022**, *14*, 108. <https://doi.org/10.3390/fi14040108>.
35. Sheatsley, R.; Papernot, N.; Weisman, M.; Verma, G.; McDaniel, P. Adversarial Examples for Network Intrusion Detection Systems. *J. Comput. Secur.* **2022**, *30*, 727–752. <https://doi.org/10.3233/JCS-210094>.
36. Simonetto, T.; Dyrnishi, S.; Ghamizi, S.; Cordy, M.; Traon, Y.L. A Unified Framework for Adversarial Attack and Defense in Constrained Feature Space. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI); IJCAI: Vienna, Austria, 2022. <https://doi.org/10.24963/ijcai.2022/183>.
37. Angelopoulos, A.; Bates, S.; Fisch, A.; Lei, L.; Schuster, T. Conformal Risk Control. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024; Kim, B., Yue, Y., Chaudhuri, S., Fragkiadaki, K., Khan, M., Sun, Y., Eds.; Curran Associates, Inc.: New York, NY, USA, 2024; Volume 2024, pp. 55198–55218.
38. Mahajan, D.; Tan, C.; Sharma, A. Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers. *arXiv* **2019**, arXiv:1912.03277.
39. Barocas, S.; Selbst, A.D.; Raghavan, M. The Hidden Assumptions Behind Counterfactual Explanations and Principal Reasons. In Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*), Barcelona, Spain, 27–30 January 2020. <https://doi.org/10.1145/3351095.3372830>.
40. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIOT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors* **2023**, *23*, 5941. <https://doi.org/10.3390/s23135941>.
41. Ferrag, M.A.; Friha, O.; Hamouda, D.; Maglaras, L.; Janicke, H. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access* **2022**, *10*, 40281–40306. <https://doi.org/10.1109/ACCESS.2022.3165809>.
42. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, Funchal, Madeira, Portugal, 22–24 January 2018; pp. 108–116. <https://doi.org/10.5220/0006639801080116>.
43. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 165130–165150. <https://doi.org/10.1109/ACCESS.2020.3022862>.
44. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. <https://doi.org/10.1145/2939672.2939785>.
45. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased Boosting with Categorical Features. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, Canada, 2–8 December 2018; Curran Associates, Inc.: New York, NY, USA, 2018; Volume 31, pp. 6638–6648.
46. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the Advances in Neural Information Processing Systems; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
47. Arik, S.O.; Pfister, T. TabNet: Attentive Interpretable Tabular Learning. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 6679–6687. <https://doi.org/10.1609/aaai.v35i8.16826>.

48. Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; Babenko, A. Revisiting Deep Learning Models for Tabular Data. In *Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021*; Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W., Eds.; MIT Press: Cambridge, MA, USA, 2021; Volume 34, pp. 18932–18943.
49. Efron, B. Bootstrap Methods: Another Look at the Jackknife. *Ann. Stat.* **1979**, *7*, 1–26. <https://doi.org/10.1214/aos/1176344552>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.