technologies

MDPI

Article

Dataset-Calibrated Perturbation Budgets for Adversarial Robustness Evaluation of Machine-Learning Network Intrusion Detection Systems

Mohsen S. Alsaadi [1], Muhammad Imran [2], Noora Fadhel [3], Malik Al-Essa [4], Iulian Sorin Munteanu [5] and Wasim A. Ali [3,*]

[1] Research and Development (R&D) Department, Ebttikar Technology Company Ltd., Riyadh 11573, Saudi Arabia; malsaadi@ebttikar.com
[2] Computer Science Department, University of Bari, 70125 Bari, Italy; muhammad.imran@uniba.it
[3] Department of Electrical and Information Engineering, Polytechnic University of Bari, 70125 Bari, Italy; n.fadhel@studenti.poliba.it
[4] Department of Computer Science, The University of Jordan, Amman 11942, Jordan; m.alessa@ju.edu.jo
[5] Theory of Mechanisms and Robots Department, National University of Science and Technology Politehnica of Bucharest, Splaiul Independentei No. 313, 060042 Bucharest, Romania; iulian.munteanu0306@upb.ro
* Correspondence: wasim.ali@poliba.it

Abstract

Adversarial evaluation of machine-learning-based network intrusion detection systems (NIDSs) requires an explicit bound on attacker perturbations, yet fixed bounds are often reused across datasets without verifying scale comparability. We evaluate two differentiable detectors on CICIDS2017, TON_IoT, and UNSW-NB15 using a constrained, control-surface-aware evasion procedure in which perturbation cost is measured in interquartile-range units. The measured constrained feature-space evasion cost differs by more than two orders of magnitude across benchmarks. On UNSW-NB15, changing only the perturbation bound from a fixed reference value of 50 to a dataset-calibrated P_{95} value changes measured evasion success from 2.6% to 84.0%, with the conclusion remaining stable across P_{90}, P_{95}, and P_{99} calibration. We therefore propose a dataset-specific quantile calibration procedure and a monotonicity-based internal validity test exploiting nested attacker action spaces. Calibration removes the severe fixed-bound monotonicity violations observed on the two most poorly scaled datasets, although the test remains diagnostic rather than certifying. A secondary attribution-based robustness estimator does not generalise consistently across the three benchmarks. These results show that perturbation-budget calibration is an essential component of credible cross-dataset adversarial NIDS evaluation.

Keywords: network intrusion detection; adversarial machine learning; evasion attacks; robustness evaluation; benchmark calibration; feature attribution; explainable artificial intelligence

Check for updates

Academic Editor: Arash Habibi Lashkari

Received: 27 August 2026
Revised: 2 October 2026
Accepted: 3 October 2026
Published: 9 October 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.

1. Introduction

Machine learning-based network intrusion detection systems, hereafter network intrusion detection systems (NIDSs), are now a standard component of the security literature, including hybrid detection designs for IoT and cloud environments [1,2]. Adversarial evaluation of these systems is likewise common. A malicious flow is represented as a feature vector, an optimiser perturbs that vector until the detector labels it benign, and the resulting evasion rate is reported as a measure of the detector's fragility.

Every such evaluation must bound the attacker. Without a bound the exercise is trivial, since a sufficiently large perturbation defeats any classifier. A bound is therefore necessary, but its numerical value is often treated as a fixed implementation choice rather than an empirical quantity to be calibrated.

A common practice in this area is to adopt a single perturbation bound. A bound is chosen once and then reused across datasets and models. This practice is understandable. A shared bound looks like a controlled comparison. If two detectors are attacked under the same budget, then differences in their evasion rates should reflect differences in robustness. In our study, the value $D = 50$ is used only as a fixed reference comparator; it is not derived from prior literature and is not proposed as a universal standard.

This reasoning contains a hidden assumption. It assumes the bound represents a comparable quantity of adversarial effort in every setting it is applied to. Network flow features do not support that assumption. A byte count, an inter-arrival time, or a packet length distribution occupies a numerical range determined by the capture environment, the traffic mix, the sensor placement and the flow extraction tool. Two benchmark datasets can describe the same protocol behaviour on scales that differ by orders of magnitude [3]. A bound expressed as a fixed quantity of feature movement therefore does not encode a fixed quantity of attacker work.

Recent benchmarking work on tabular adversarial robustness has already noticed the symptom. He et al. observe that optimal perturbation budgets vary substantially by dataset and by model, and state explicitly that direct comparison across these contexts would be methodologically unsound [4]. Their remedy is to select the most frequently occurring budget value across all tested settings and adopt it as a standard. That remedy acknowledges the problem but retains a shared numerical value across heterogeneous datasets.

This paper asks a narrower question and answers it with measurements. Does the choice of perturbation bound change what a robustness study concludes about a detector, in the specific setting of NIDS evaluation under realistic attacker constraints? We find that it does, by a wide margin.

Our contributions are as follows, in order of importance. First, we contribute a dataset-specific calibration procedure for the perturbation bound, deriving it from the empirical evasion cost distribution of the dataset under evaluation rather than reusing a fixed constant. Second, we contribute a validity test for constrained adversarial evaluations that requires no ground truth about attack success. Where attacker capability sets are nested, a monotonicity property must hold, and its violation localises a defect in the measurement rather than a property of the detector. Third, we demonstrate empirically that a fixed reference bound can reverse a robustness conclusion. On UNSW-NB15 the identical detector, attacked by the identical algorithm on identical data, evades at 2.6 percent under a fixed reference bound and at 84.0 percent under a calibrated one. We also quantify the underlying cause: measured constrained feature-space evasion cost spans more than two orders of magnitude across the three benchmarks, from a median of one unit on TON_IoT to over two hundred and sixty on UNSW-NB15. A sensitivity analysis across three calibration percentiles establishes that this discrepancy is not an artefact of the particular percentile chosen.

Fourth, and as a secondary result, we report a negative replication. We hypothesised that the fraction of a detector's decision mass placed on features outside the attacker's reach would predict its evasion cost, giving an attack-free robustness estimate derived purely from feature attribution. On CICIDS2017 alone the hypothesis appeared strongly supported. It did not survive extension to two independent datasets. We report the discovery and its falsification together, because a finding supported by a single benchmark can appear robust while failing to replicate on independent data.

The complete evaluation pipeline is summarised in Figure 1. Each of the three benchmark datasets passes independently through preprocessing, detector training and control surface categorisation. The perturbation bound is then calibrated to that dataset's own evasion cost distribution rather than fixed across datasets, and the constrained attack measures cost as an outcome instead of imposing it as a stopping criterion. A monotonicity gate acts as a ground-truth-free validity check on the resulting measurement; a configuration that violates it is treated as a measurement defect and excluded from the pooled analysis rather than passed forward into it.

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 states the threat model. Section 4 presents the control surface categorisation, the constrained attack, the calibration procedure and the validity test. Section 5 describes the experimental setup. Section 6 reports results. Section 7 discusses implications. Section 8 states limitations and Section 9 concludes.

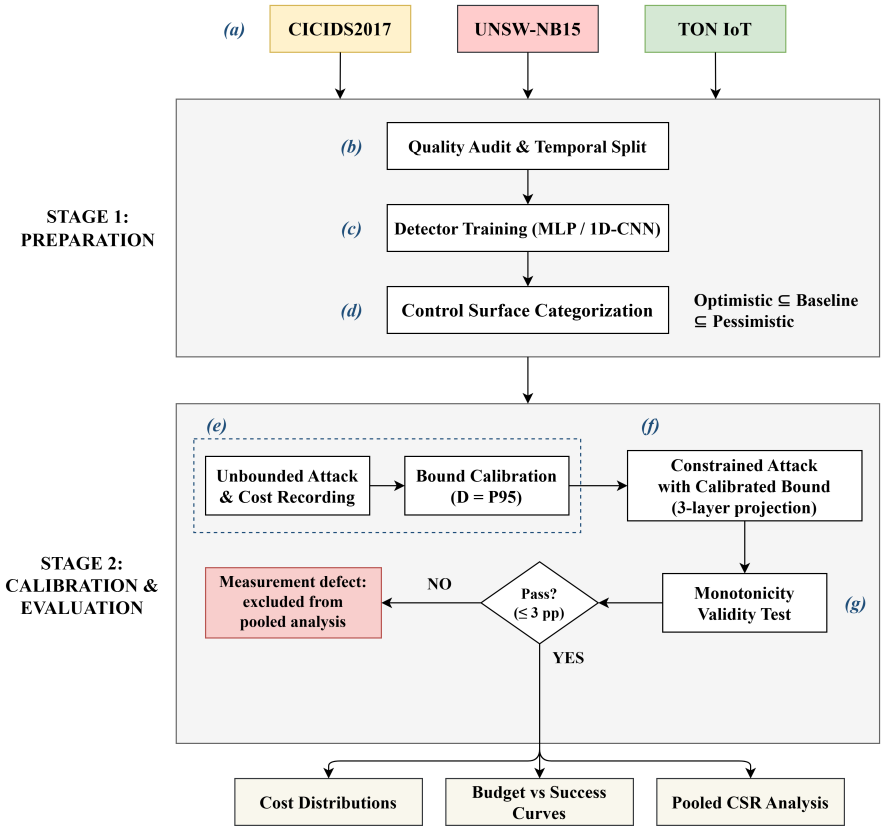


Figure 1. Evaluation pipeline, from raw benchmarks to analyses. Letters (a) to (g) mark the stages referenced in the text.

2. Related Work

Adversarial examples were first characterised in the vision domain [5], and the earliest attacks established the pattern that persists today. A perturbation is computed by following the gradient of a loss with respect to the input, and its magnitude is limited by a norm ball of radius ϵ [6]. Iterative refinements followed, including the basic iterative method [7] and projected gradient descent [8], alongside optimisation-based approaches that minimise perturbation magnitude directly rather than bounding it [9,10]. Saliency-driven variants select which coordinates to perturb rather than moving all of them [11].

In images the bound has an intuitive anchor. Pixel intensities share a common range and a perturbation small in that range is imperceptible to a human observer. Standardised evaluation suites exploit this and fix ϵ across datasets without controversy [12]. The anchor

does not transfer to tabular security data, where no perceptual notion of imperceptibility exists and features do not share a range.

The NIDS community recognised early that unconstrained feature-space perturbation produces vectors that no real traffic could generate, and several lines of work address this. Pierazzi et al. formalise the relationship between the feature-space in which an optimiser works and the problem-space in which an attacker must act [13]. They identify four constraint families that any realisable attack must satisfy, covering available transformations, preserved semantics, robustness to preprocessing and plausibility of the resulting object. They also prove that the existence of a feature-space solution is a necessary condition for a problem-space attack. This result is important for our design, because it licenses feature-space evaluation as a conservative bound on attacker capability rather than as an approximation of unknown fidelity.

Apruzzese et al. model the realistic powers available to an attacker against a deployed NIDS across five dimensions, covering access to training data, knowledge of the feature set, knowledge of the detection model, oracle access and manipulation depth [14]. Their survey of the literature finds that many published threat models assume access that a real adversary would struggle to obtain. Our threat model follows their framing.

Chernikova and Oprea propose FENCE, a gradient-based framework that preserves mathematical dependencies among engineered features during attack generation [15]. Their treatment of dependency families informs our projection design. Han et al. operate directly in traffic space, mutating packets under an explicit overhead budget expressed in crafted packet count and elapsed time [16,17]. Sheatsley et al. learn domain constraints from data using a constraint learning formulation and show that projecting adversarial examples onto the learned constraint space restores a substantial share of clean accuracy [18].

Teuffenbach et al. introduce attack cost directly into the NIDS setting, weighting perturbation by the difficulty of manipulating each feature and grouping features by how amenable they are to attacker control [19]. Their work is the closest antecedent to our control surface categorisation, and we do not claim the notion of cost-weighted or manipulability-aware attacks as novel. Our contribution concerns what happens to the numerical scale of that cost when the same evaluation is carried across datasets. Recent work by ElShehaby and Matrawy introduces a perturb-ability score for flow-based NIDS features, explicitly quantifying how problem-space constraints limit the attacker's ability to manipulate individual features [20]. That contribution is complementary to our control-surface categorisation and reinforces the need to model attacker reach explicitly. Our question is different: even after manipulability constraints are specified, does the numerical perturbation-cost scale remain comparable when the same evaluation protocol is transferred across datasets? We therefore focus on calibration of the measurement bound rather than on feature-level perturbability as a defense mechanism. Parallel work in the wider tabular setting addresses adjacent constraints, including imperceptibility-weighted perturbation [21], heterogeneity between tabular datasets [22], unified constrained attack and defence formulations [23], and explicit cost and utility awareness in the attacker's objective [24]. Related work on structured data has also used counterfactual generation, in which samples are moved minimally across a decision boundary while respecting the feature domain, as a tool for rebalancing imbalanced classification problems [25]; the same notion of a minimal boundary-crossing displacement underlies the evasion cost measured in this paper. Our setting differs in that the constraints derive from protocol semantics and attacker reach rather than from perceptual or economic plausibility.

Turning to evaluation validity itself, Carlini et al. compile a checklist of sanity tests for adversarial evaluations [26]. Among them is a monotonicity requirement stated on the budget axis. If attack success decreases as the permitted distortion increases, the attack is

likely flawed rather than the model robust. This is the closest existing analogue of the test we adopt. We apply the same underlying principle on a different axis, namely the nesting of attacker action spaces induced by control surface categorisation, which is available in constrained settings but absent from the norm-bounded vision setting for which the original checklist was written. The need for such tests is not hypothetical. He et al. report a case in which attack success declines with increasing budget for one attack and architecture combination on tabular data, and trace it to step size configuration rather than to model robustness [4]. Our results in Section 6 show a structurally similar failure arising from an entirely different cause.

Feature attribution methods, including additive explanation values [27] and local surrogate explanations [28,29], are now standard instruments for interpreting security classifiers, and their use in adversarial NIDS research is an active area rather than an open gap. Two distinct uses should be separated. The first, and by far the better developed, uses attribution to detect adversarial inputs at inference time, on the reasoning that adversarial examples produce attribution patterns that differ measurably from those of clean inputs. Maseno et al. pursue exactly this, combining recurrent detectors with gradient-based attacks and using additive explanation values to identify explanation drift as a detection signal, and reporting that architectures with comparable clean performance can present markedly different robustness profiles [30]. Explainability is also coupled with adversarial robustness assessment in federated intrusion detection at the edge [31]. We make no contribution to this line of work and our secondary result should not be read as one. The second use, which our negative result addresses, is predictive rather than detective. If a detector's decision mass can be located, and if some features are known to lie outside attacker control, then the share of mass placed on those features is a candidate robustness estimator computable without running any attack at all. This is a different question from detecting an attack after it occurs, and it would be attractive because constrained attacks are computationally expensive and their configuration is itself a source of variance, as this paper demonstrates. We test the idea directly and report that it does not hold across datasets.

Finally, the assumption that robustness findings transfer across capture environments has come under scrutiny. González-Ramos et al. report adversarial robustness evaluation of a hybrid convolutional, recurrent and transformer architecture across several benchmarks, and argue for security degradation curves in place of single-point clean performance reporting [32]. Li et al. evaluate robustness under perturbation strength sweeps across intrusion detection benchmarks, comparing architectures under a common protocol [33]. Hakim et al. report outright cross-domain generalisation failure for lightweight industrial intrusion detection models, finding that models performing well within a capture environment degrade sharply outside it [34]. These studies compare detectors under a shared evaluation protocol. Our question is upstream of that comparison: whether the protocol itself remains comparable when carried between datasets. Our results add a specific and previously unquantified mechanism to this picture. Part of what fails to transfer across datasets is not the detector and not the attack, but the evaluation instrument itself.

3. Threat Model

We assume an attacker who has already established a foothold on at least one host inside the monitored network, for example through phishing or other scams that exploit human vulnerability factors [35], and wishes to continue operating without triggering the detector. The attacker fully controls the traffic they themselves emit, including timing, packet sizes, header fields they set, protocol selection and connection counts. The attacker does not control the victim host or its responses, does not control routing or network infrastructure, and does not have access to the feature extraction pipeline. The attacker

must preserve the functional effect of the attack. A variant that is stealthier but no longer accomplishes the original objective is not a successful evasion.

We distinguish the attacker's operational manipulation capabilities from the evaluator's model-knowledge assumptions. Operationally, the attacker controls only the traffic it emits, does not control the victim host, routing, or network infrastructure, and must preserve the malicious functionality of the original activity. For robustness evaluation, however, we grant the optimisation procedure white-box access to the trained detector, including input gradients. This stronger knowledge assumption is deliberate: it isolates the effect of realistic manipulation constraints while subjecting the detector to a conservative stress test. Accordingly, the reported results should not be interpreted as a claim that a real-world attacker necessarily possesses gradient access to the deployed model. This corresponds to the more realistic end of the capability spectrum characterised by Apruzzese et al. [14], while making the stronger evaluator-side knowledge assumption explicit.

Following Pierazzi et al. [13], the existence of a feasible feature-space solution is a necessary condition for a corresponding problem-space attack under the same semantic constraints. The converse does not follow: a feature-space evasion is not, by itself, proof that a packet-level realisation exists at the same cost. We therefore use constrained feature-space evasion as a conservative robustness stress test and avoid interpreting the resulting cost as a direct physical measure of real-world attacker effort.

4. Methodology

4.1. Control Surface Categorisation

Not every input feature is equally available to an attacker. Some are set directly by the packets the attacker emits. Others are produced by the victim's responses, or by protocol mechanics, or by infrastructure the attacker does not control. An evaluation that permits perturbation of the second group measures something no attacker can perform. Stage (d) of Figure 1 situates this step within the wider pipeline.

We therefore partition the feature set of each dataset into three categories. Category 1 contains features the attacker can set essentially at will, at negligible cost to the effectiveness of the attack. A destination port is the canonical example, since relocating a service to a non-standard port costs the attacker almost nothing. Category 2 contains features the attacker can move, but not for free. Volume-tied quantities emitted by the attacker fall here. A flooding attack can reduce its packet count, but doing so weakens the flood. Category 3 contains features determined outside the attacker's reach. Statistics computed exclusively over victim-emitted packets belong here, as do quantities fixed by protocol behaviour. Setting these would require compromising the victim, at which point evasion is moot because the target is already owned.

The categorisation is derived structurally rather than by inspection of feature names. For each dataset we work from the computation performed by the flow extraction tool, determining which packet list each statistic iterates over and what aggregation is applied. The direction of aggregation is the primary signal. Aggregation type matters as well. A minimum or maximum taken over both directions can be set unilaterally by whichever party emits the extreme packet, whereas a sum or mean can be shifted by either party in proportion to its contribution.

A residual set of features remains genuinely ambiguous after this analysis. We do not resolve these by assertion. Instead we construct three categorisation scenarios for each dataset. The baseline scenario reflects our primary structural analysis. The pessimistic scenario reassigns all ambiguous features to attacker control, maximising attacker capability. The optimistic scenario reassigns them to attacker independence, minimising it. Features that remain unresolved under all three are frozen and never perturbed, a conservative

choice that weakens the modelled attacker. The three scenarios' perturbable feature sets are nested by construction, a property we exploit below.

4.2. Constrained Attack and Cost Unit

We use sparse greedy coordinate descent. At each iteration the attacker computes the gradient of the detector's loss with respect to the input, masks it to currently perturbable features, and selects the single feature with the largest cost-aware score. Specifically, for perturbable coordinate j, we compute

$$q_j = \frac{|\partial L/\partial x_j|}{\lambda_j}, \tag{1}$$

where the terms are defined as follows:

- x_j is the current value of the j-th input feature of the flow being attacked, expressed in the raw feature space;
- L is the detector's binary cross-entropy loss computed from its output logit, which the attack optimises in order to drive that logit below zero, that is, towards the benign decision;
- $\partial L/\partial x_j$ is the partial derivative of L with respect to x_j, so that its absolute value measures how strongly a small change in feature j alone would change the loss at the current point;
- λ_j is the category-specific cost multiplier of feature j, with $\lambda_j = 1.0$ for Category 1 and $\lambda_j = 2.5$ for Category 2;
- q_j is the resulting cost-aware selection score of feature j.

Category 3 and unresolved features are excluded from selection. Thus, a Category 2 feature must provide 2.5 times the raw gradient magnitude of a Category 1 feature to obtain the same selection score. Only the feature with the largest q_j is moved. The result is then projected through three constraint layers before the next iteration begins, as shown in stage (f) of Figure 1.

The first layer enforces the control surface mask. Category 3 and unresolved features have their gradients zeroed, and a restoration step verifies after projection that their values are unchanged. The second layer enforces range and type constraints. Non-negativity is imposed on quantities that cannot be negative, empirical bounds are derived from the training split only, integer-valued features are rounded and re-clamped, and sentinel values used by the extractor to indicate absent measurements are preserved rather than perturbed into spurious numeric values. The third layer enforces mathematical dependencies. Features that are exact functions of others are recomputed after perturbation rather than perturbed independently, and internal consistency is restored where an aggregate must agree with its components.

A validator checks any candidate sample against all three layers and returns a list of violations. Every reported adversarial sample passes this validator. A vector that violates the constraint set is a bug, not an adversarial example.

We chose sparse selection over dense perturbation deliberately. Dense projected gradient descent moves every unfrozen feature at every step and charges cost for all of that movement, including movement with negligible effect on the decision. When the perturbable feature set grows, dense descent wastes proportionally more of its budget, which can make a larger action space appear less effective than a smaller one. Sparse selection charges cost only for movement the attacker chose to make.

Perturbation must be measured in some unit. The natural first choice, a multiple of each feature's standard deviation, fails on this data. Flow features are heavy-tailed, with skewness reaching three orders of magnitude on some columns, and a standard

deviation computed over such a distribution is dominated by a small number of extreme observations. A step of a fixed fraction of a standard deviation therefore means very different things on different features of the same dataset. We use the interquartile range instead. It is robust to the tail, and it is also the scale used internally by the robust scaler applied before detection, which means it is the scale the detector itself experiences. Cost is accumulated as the sum, across all modified features, of the displacement in interquartile range units, weighted by the category-specific cost multiplier. We refer to the resulting quantity as measured constrained feature-space evasion cost rather than true attacker cost.

4.3. Cost Measurement and Bound Calibration

We do not impose a budget as a stopping criterion. The attack runs until it evades, until every perturbable feature is exhausted, or until a generous iteration cap that we verify does not bind in practice. Cost is recorded at the moment of success. Budget versus success curves are derived retrospectively by thresholding the recorded costs, which produces the same curves a budget-capped search would produce without the distortion that a hard shared budget introduces.

A displacement bound D is nonetheless enforced during the search, limiting the total movement of a sample from its original position. D is not treated as an evaluation operating point. It functions solely as a plausibility constraint, in the sense identified by Pierazzi et al. [13] as one of the four requirements for a realisable adversarial object. A flow displaced arbitrarily far from any realistic operating point is conspicuously anomalous regardless of what the classifier reports. The distinction from an evaluation budget is that D bounds what counts as a valid adversarial object, whereas an evaluation budget selects the operating point at which robustness is reported. We report robustness across the full range of budgets, not at D.

The central methodological point of this paper concerns how D is chosen. We calibrate it to each dataset separately, corresponding to stage (e) of Figure 1. For each dataset we first draw 200 malicious samples that are correctly classified as malicious by the clean detector from the dataset's test partition and run the attack with D effectively unbounded on this calibration draw. We record the resulting evasion-cost distribution and set D to a fixed percentile of that distribution. The final robustness matrices use a separate deterministic draw from the same test partition. The calibration and final-evaluation draws overlap by 0, 4, and 2 samples for CICIDS2017, TON_IoT, and UNSW-NB15, respectively; these overlaps are consistent with incidental overlap between independent draws from the same parent pool. Because no dedicated calibration partition was used, however, we do not describe this procedure as statistically independent calibration. This is a limitation of the present study and motivates the use of a held-out calibration split in future work. In future applications of the procedure, we strongly recommend a fully independent calibration partition. Under the temporal protocol used here, this can be obtained by dividing each dataset chronologically into three disjoint segments, used respectively for training, calibration and final evaluation, with the duplicate-signature grouping of Section 5 applied across all three so that no exact record crosses a boundary. The calibration bound is then estimated only on the middle segment and applied unchanged to the final segment. Where the data are too limited for a three-way split, a cross-fitting scheme provides an alternative: the evaluation pool is divided into K disjoint folds, and the bound applied to each fold is estimated from the evasion costs recorded on the remaining $K - 1$ folds, so that no sample contributes to the bound under which it is itself evaluated. The scalar D is estimated once per dataset from the MLP baseline calibration pass and is then applied to the dataset's final evaluation matrix.

Formally, let c_i denote the recorded evasion cost of sample i under an unbounded search on the calibration draw for dataset $\mathcal{D}$, over the set $E_{\mathcal{D}}$ of calibration samples that ever evade. Then

$$D_{\mathcal{D}} = P_q\big(\{c_i : i \in E_{\mathcal{D}}\}\big), \tag{2}$$

and the retrospective success rate at budget b is

$$S_{\mathcal{D}}(b) = \frac{1}{|N|}\,\big|\{i \in N : c_i \le b\}\big|, \tag{3}$$

where N is the set of malicious evaluation samples that are correctly classified as malicious by the clean detector before adversarial optimisation, and c_i is the recorded evasion cost for that same evaluation population under an effectively unbounded search. Samples already misclassified as benign at baseline are excluded from the evasion denominator and are reported only through clean detection metrics.

For the fixed reference comparison, $D = 50$ has no literature-derived provenance. It originated as one value in an earlier displacement sweep over $\{10, 20, 30, 50, 75, 100, 150, 200, 300, 500\}$ during an abandoned surrogate-transfer investigation. Every value in that sweep produced zero transfer, so $D = 50$ was subsequently retained as a fixed, dataset-agnostic reference comparator rather than as an empirically optimal setting. We use it only to expose the consequences of transferring an absolute bound without calibration; it is not proposed as a universal standard.

We fix $q = 95$ as the primary calibration rule, specified in advance of any analysis and applied uniformly across datasets. Section 6 reports a sensitivity analysis at $q = 90$ and $q = 99$ to establish whether the paper's conclusions depend on this choice. We do not claim that the ninety-fifth percentile is optimal in any sense. It is a pre-specified rule, and the sensitivity analysis exists so that its influence is visible rather than assumed away.

The calibration in (2) is descriptive rather than optimised, and the distinction matters. D is set from the empirical cost distribution before any validity check is run, and it is never adjusted in response to a monotonicity outcome, an evasion rate, or any other downstream result. Selecting D to maximise monotonicity compliance would be circular, since the calibration would then be tuned on the criterion it is meant to be tested against. We do not do this, and the percentile sweep is reported as observed rather than as a search for a preferred value.

4.4. Monotonicity as a Validity Test and the Reliance Measure

The three categorisation scenarios induce nested perturbable feature sets,

$$F_{\text{opt}} \subseteq F_{\text{base}} \subseteq F_{\text{pess}}, \tag{4}$$

where F denotes the set of features the attacker may modify. An attacker granted F_{pess} can always replicate any strategy available under F_{opt} by simply declining to use the additional features. It follows that for an attack that searches its action space adequately,

$$S_{\text{opt}} \;\le\; S_{\text{base}} \;\le\; S_{\text{pess}}, \tag{5}$$

where S denotes the asymptotic evasion success rate, that is, the fraction of samples that ever evade under an unbounded budget.

Equation (5) requires no ground truth. It is a statement about internal consistency, checkable on any dataset without labelled evasion outcomes and without a reference implementation to compare against. A violation indicates that the measurement apparatus is failing to find solutions that provably exist, which points to a defect in the attack,

the bound, or the cost accounting rather than to a robust detector. Stage (g) of Figure 1 shows how the test gates the analyses.

We stress the asymmetry. A violation is informative; passing is not proof. The test is a diagnostic check on the measurement, not a certificate of correctness, and Section 6 shows that the validity status of a borderline configuration can itself shift with the calibration percentile. Because our attack is a non-exhaustive heuristic, exact monotonicity is not guaranteed. We adopt a tolerance of three percentage points throughout, stated explicitly rather than buried in configuration, and interpret violations beyond that tolerance as measurement defects. Configurations that violate the tolerance are excluded from the pooled correlation analysis rather than reported alongside compliant ones.

The relationship to the budget axis monotonicity check of Carlini et al. [26] is worth stating plainly. Both rest on the same principle, that a strictly stronger attacker cannot do worse. The axis differs. Their formulation varies the magnitude of a single norm bound, which is the natural free parameter in the vision setting. Ours varies the composition of the attacker's action space, which is the natural free parameter once domain constraints are introduced and is unavailable in the unconstrained setting.

The final quantity we require is a candidate robustness estimator derived from attribution alone. We define control surface reliance, or CSR, as the share of a detector's decision mass placed on features outside attacker control:

$$\mathrm{CSR} = \frac{\sum_{j \in C_3} w_j}{\sum_{j \in \mathcal{F}} w_j}, \tag{6}$$

where w_j is the attribution weight of feature j, C_3 is the set of Category 3 features and $\mathcal{F}$ is the full feature set. Weights are normalised within each model so that detectors whose attributions are computed by different methods remain comparable. For neural detectors we use the mean absolute gradient of the output logit with respect to each input feature, averaged over the test set. Unresolved features are excluded from the numerator, consistent with the conservative treatment applied elsewhere. We additionally define a variant that credits partial protection from Category 2 features,

$$\mathrm{CSR}_{\mathrm{B}} = \frac{\sum_{j \in C_3} w_j + \frac{1}{2} \sum_{j \in C_2} w_j}{\sum_{j \in \mathcal{F}} w_j}, \tag{7}$$

and an inverse quantity, exposed mass, given by the share of attribution on Categories 1 and 2 combined. The hypothesis under test is that CSR predicts evasion cost. If it holds, robustness could be estimated from attribution alone, without running an attack. This is an explainability-driven claim, and it is the reason we treat feature attribution as an instrument in this work rather than as a presentational device.

5. Experimental Setup

5.1. Datasets

We use three benchmarks that, between them, span enterprise, synthetic and Internet of Things capture environments, and three distinct flow extraction tools, as shown in stage (a) of Figure 1. Public intrusion detection benchmarks are known to differ substantially in capture setup, attack coverage and feature definitions [3], and this heterogeneity is precisely the condition under which a shared absolute perturbation bound becomes questionable.

CICIDS2017 [36] provides five days of enterprise traffic with flow features extracted by CICFlowMeter. The release we use contains no timestamp column, so we validated the assumption that row order is capture chronological before relying on it. For each attack class we compared the width of the positional window containing the class against a null

distribution generated by randomly shuffling row order one hundred times. Observed window widths fell at or below the fifth percentile of the null for every attack class and file pair, and the order of attack blocks matched the documented attack schedule in every checkable file.

UNSW-NB15 [37] provides synthetically generated traffic with features derived from Argus and Bro. The distributed files carry no header row, and column names must be applied from the accompanying feature description file. The capture comprises two sessions with a pronounced difference in attack rate between them, which we account for in splitting.

TON_IoT [38] provides heterogeneous traffic from an Internet of Things testbed with network features extracted by Zeek. We use the network layer only. Row order in this release proved to be grouped by class rather than chronologically, which we established by the same positional analysis applied to CICIDS2017, and we split accordingly.

5.2. Preprocessing, Detectors and Metrics

Each dataset receives an independent quality audit rather than a common inherited cleaning recipe, corresponding to stage (b) of Figure 1. On CICIDS2017 the audit surfaced infinite values, zero variance columns, exactly duplicated columns and a duplicate row rate of 10.9 percent overall, reaching 63.9 percent within the port scan class. Automated attack tools emit near-identical flows, so this is a structural property of the data rather than a collection error.

Duplication of this magnitude interacts badly with splitting. Under a per-class chronological split, 13.3 percent of test rows were byte-for-byte duplicates of training rows, and the affected detectors were partly being evaluated on records they had memorised verbatim. We therefore group rows by exact feature and label signature before splitting, assign every row sharing a signature to whichever partition its signature lands in, and verify zero signature overlap across partitions afterwards. Duplicate rows are retained in the dataset. Only their placement is controlled.

Table 1 reports the effect of correcting this leakage on measured clean performance for CICIDS2017. The tree ensembles absorb the larger correction, which is consistent with their capacity to memorise exact records.

Table 1. Effect of duplicate leakage correction on clean performance, CICIDS2017 binary task.

Model	Leaked Split F_1 (Macro)	Group-Aware Split F_1 (Macro)
Random Forest	0.991	0.963
XGBoost	0.9971	0.9965

We report this not as a finding about one dataset, but as evidence for the paper's premise. Evaluation validity has to be established before robustness conclusions are drawn from it, and duplication of this kind is invisible to every metric a study would normally inspect.

We use temporal splitting rather than random splitting throughout. The distinction is not cosmetic. Table 2 reports the gap on CICIDS2017.

Table 2. Random versus temporal splitting, CICIDS2017, random forest, binary task.

Split Protocol	F_1 (Macro)	Gap vs. Random
Random	0.998	–
Per-class chronological	0.963	−0.035
Whole-day holdout	0.619	−0.379

The whole-day holdout figure in Table 2 deserves comment. In CICIDS2017 each capture day contains a distinct attack scenario, so holding out an entire day means every attack class in the test set is unseen during training. That protocol measures zero-day generalisation rather than in-distribution detection, and a macro F_1 of 0.619 under it is a reasonable result rather than a poor one. It is not, however, a usable baseline for adversarial evaluation. Measuring the cost of hiding an attack from a detector that already misses roughly forty percent of attacks confounds evasion with baseline failure. We therefore use the per-class chronological split as the primary protocol and retain the day holdout result as a separate generalisation observation.

We evaluate two natively differentiable architectures for each dataset, a multilayer perceptron and a one-dimensional convolutional network, on the binary detection task, as shown in stage (c) of Figure 1. Both families are common choices in comparative studies of machine and deep learning attack detectors [2,39]. Hyperparameters are modest and documented, and no large search was performed, since the object of study is the evaluation protocol rather than peak detection performance.

We restrict the adversarial analysis to natively differentiable architectures. Attacking tree ensembles by gradient requires a differentiable surrogate, and in our experiments adversarial examples generated against distilled neural surrogates transferred poorly to the target ensembles under strict domain constraints. The mechanism appears to be structural. A forest requires agreement across many trees to change its vote, whereas a dense surrogate can be driven across its boundary by movement in a small number of coordinates, and a sparse attacker discovers and exploits that discrepancy. We report constrained transfer attacks against tree ensembles as an open problem and note the restriction as a limitation in Section 8.

Table 3 reports clean detection performance for the detectors used in the adversarial analysis. The TON_IoT convolutional detector is weak, and we flag it as marginal wherever its results appear. A detector that barely separates the classes is a poor attack target, and conclusions drawn from attacking it carry correspondingly less weight.

Table 3. Clean detection performance of attack targets.

Dataset	Model	ROC-AUC	F1	PR-AUC	Note
CICIDS2017	MLP	0.925801	0.871754	0.743730	
	1D-CNN	0.905811	0.894986	0.807362	
TON_IoT	MLP	0.917471	0.874093	0.973797	
	1D-CNN	0.657564	0.716464	0.855225	marginal detector
UNSW-NB15	MLP	0.999417	0.991110	0.995507	
	1D-CNN	0.999077	0.990188	0.994204	

F1 and PR-AUC are reported using the weighted binary-classification convention used for the derived metrics.

To make the adversarial evaluation reproducible without relying on private configuration files, Table 4 gives the exact detector and training configuration used across all three datasets. Table 5 gives the corresponding attack configuration. The classifier hyperparameters are identical across datasets; the input dimensionality is dataset-specific and is fixed by the feature columns stored with each trained checkpoint. Exact software-version metadata were not retained in the archived experiment artifacts and are therefore not reported here rather than inferred retrospectively.

Table 4. Detector and training configuration.

Parameter	MLP	1D-CNN
Hidden/convolutional layers	3: 128–64–32	2: 16–32 filters
Activation	ReLU	ReLU
Kernel size/stride/padding	–	3/1/1
Pooling	–	AdaptiveMaxPool1d(1)
Dropout	0.2 after each hidden layer	0.2 in classification head
Dense output	Linear(32,1)	Linear(32,1)
Output activation	None (raw logit)	None (raw logit)
Loss	BCEWithLogitsLoss	BCEWithLogitsLoss
Optimiser	Adam	Adam
Learning rate	10^{-3}	10^{-3}
Batch size	1024	1024
Epochs	15	15
Early stopping	None	None
Seed	42	42

Table 5. Constrained attack configuration.

Parameter	Configuration
Attack algorithm	Sparse greedy coordinate descent; one feature updated per iteration
Loss and target	Binary cross-entropy with logits; target logit is driven below zero (benign)
Feature selection	$q_j = \lvert\partial L/\partial x_j\rvert/\lambda_j$; $\lambda_1 = 1.0$, $\lambda_2 = 2.5$; Category 3/unresolved features excluded
Step size	$(\mathrm{IQR}_j/\lambda_j) \times 0.2$, with a floor of 1 raw unit for integer-valued features
Maximum iterations	30,000 safety iterations
Stopping	Evasion, given-up condition, iteration cap, or 900 s per-configuration wall-clock guard
Success	Detector output logit < 0
Attack population	Malicious samples correctly classified as malicious before attack; target 500 per dataset/model, with the exact per-configuration count retained in the archived summaries
Projection	Control-surface mask; empirical training-split range/type constraints; integer rounding; sentinel preservation; functional-dependency recomputation where supported
Feature bounds	Training-split empirical minima and maxima
Feature space	Raw feature space; the trained RobustScaler is applied internally by the model wrapper
Cost	Sum of IQR-normalised absolute displacement weighted by category multiplier
Restarts	None in the final MLP/CNN matrices
Random seed	42

Before any attack results were produced, the constraint engine was stress tested by applying random corruption of eight magnitudes to sampled inputs and projecting the corrupted vectors back through all three layers. Across four thousand projections the validator reported zero violations. Two implementation defects were found and corrected during this process, both invisible on clean data and both involving interactions between the frozen feature restoration step and the dependency recomputation step. We report the validation figure rather than the defects, but note that clean data testing alone would not have surfaced either. Across the thirty-six attack configurations reported in Section 6, the validator likewise reported zero violations.

For each dataset, model and scenario, we report the asymptotic evasion success rate, the median evasion cost among samples that ever evade, and the quartiles of the cost distribution. We report the full budget versus success curve rather than a single operating point. We deliberately avoid reporting evasion success at a single matched budget as a headline metric. Cutting two cost distributions of different shape at one arbitrary point measures the difference in their tails, not the difference in evasion difficulty, and the choice of cut point is exactly the free parameter this paper is about.

6. Results

Table 6 reports the measured constrained feature-space evasion cost distribution obtained under unbounded search, together with the calibrated bounds derived from it by (2) at three percentiles.

Table 6. Measured constrained evasion cost and calibrated bounds by dataset (IQR units).

Dataset	Median Cost	P_{90}	P_{95} (Primary)	P_{99}
TON_IoT	1.00	6.00	27.70	169.00
CICIDS2017	5.50	767.96	914.68	1048.94
UNSW-NB15	264.50	397.00	399.50	408.70

The spread in median measured evasion cost between TON_IoT and UNSW-NB15 in Table 6 exceeds two orders of magnitude. This is measured in a unit that is already scale-robust and that is already the unit the detector's own preprocessing operates in. The variation is therefore not an artefact of unnormalised features. It reflects a genuine difference in how far a flow must move, relative to the spread of the data around it, before the detector's decision changes.

The shape of the upper tail also differs sharply between datasets, and Table 6 makes this visible in the spacing of the three percentiles. On TON_IoT the bound grows by a factor of twenty-eight between the ninetieth and the ninety-ninth percentile, indicating a heavy tail. On UNSW-NB15 the same range spans three percent, indicating a nearly flat one. CICIDS2017 sits between the two. A calibration rule expressed as a percentile therefore has very different practical consequences depending on the dataset it is applied to, which is itself an argument against fixing the bound in absolute units.

Figure 2 shows these distributions directly, restricted to samples that evade under the baseline scenario, with a logarithmic vertical axis. Beyond the difference in medians, the distributions differ in shape. CICIDS2017 exhibits a heavy tail in which a minority of samples require displacement orders of magnitude above the median, whereas UNSW-NB15 is comparatively concentrated. Two datasets can therefore differ not only in typical cost but in the shape of the cost distribution, which is why we report curves rather than points and why a single matched budget is an unreliable basis for cross-dataset comparison.

Table 7 isolates the effect of the bound. The detector is identical, the attack algorithm is identical, the data are identical, and the categorisation is identical. Only D changes.

Table 7. Effect of bound calibration on measured evasion, UNSW-NB15, MLP, baseline scenario.

Bound	Value (IQR Units)	Evasion Success
Fixed, inherited	50.0	2.6%
Calibrated, P_{90}	397.0	80.0%
Calibrated, P_{95} (primary)	399.5	84.0%
Calibrated, P_{99}	408.7	84.8%

A study reporting the first row of Table 7 would describe this detector as highly resistant to constrained evasion. A study reporting any of the remaining three would describe it as substantially vulnerable. All four use the same detector and the same attack. The difference arises because a bound of 50 sits far below the median evasion cost on this dataset, so the search is terminated before it can reach the decision boundary for the great majority of samples.

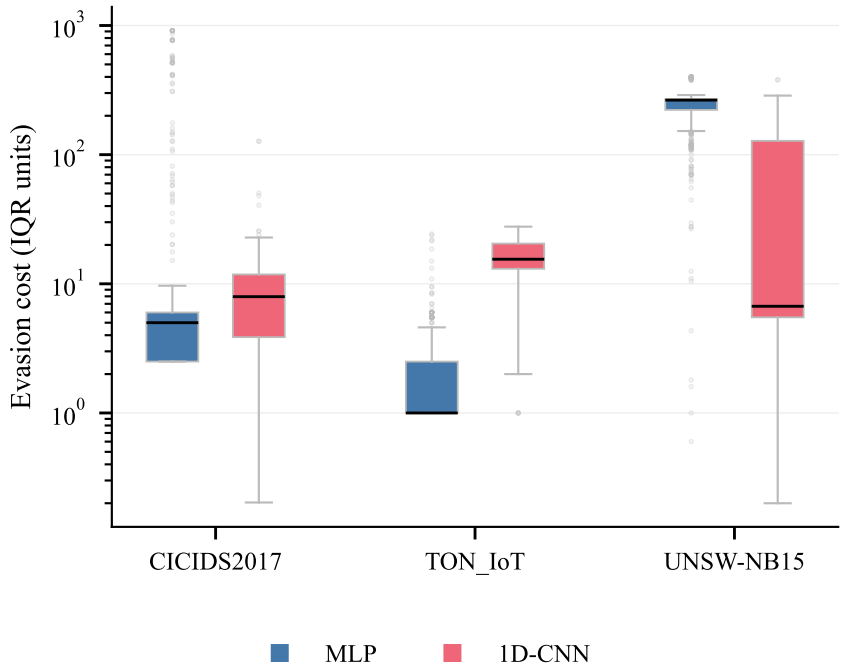


Figure 2. Evasion cost distributions by dataset and detector.

The three calibrated rows are tightly clustered, spanning 4.8 percentage points, while the gap to the fixed reference bound ranges from 30.8× to 32.6×. The discrepancy is therefore a property of transferring a fixed reference bound, not of the particular percentile used to replace it. Median evasion cost among evaders is identical at 264.5 units under all three calibrated bounds, indicating that the same set of genuinely evadable samples is measured in each case.

Figure 3 shows why single point reporting is fragile in general. Each panel plots the baseline scenario for both detectors on a logarithmic budget axis, with a solid vertical line marking that dataset's calibrated bound and a dashed line marking the inherited bound of 50 IQR units. The three panels span different ranges by roughly two orders of magnitude, which is itself the clearest available depiction of the incomparability the calibration argument concerns. On the UNSW-NB15 panel the dashed line falls far to the left of the region where the curves rise, which is the mechanism behind Table 7 shown graphically.

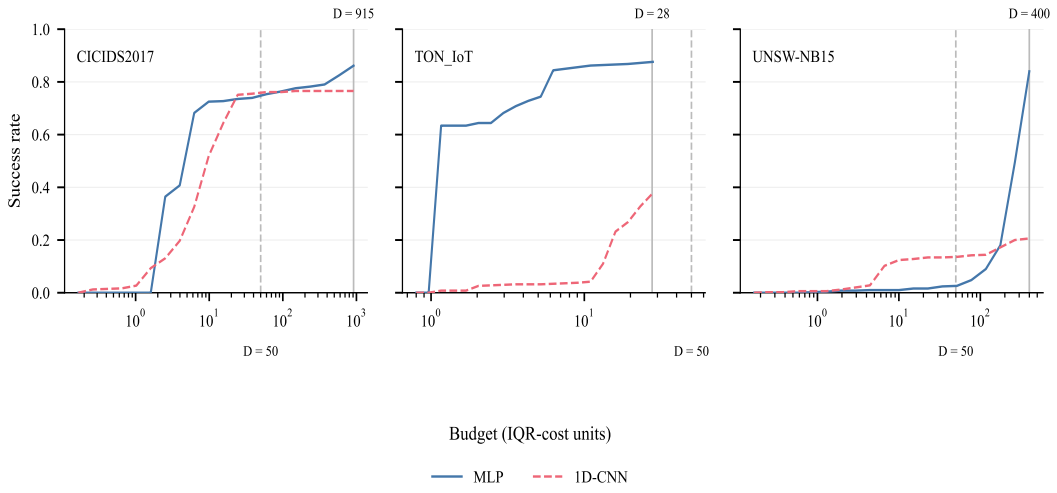


Figure 3. Budget versus success curves under calibrated bounds. Vertical labels are rounded to the nearest integer for display; exact calibrated values are reported in Table 6.

Reading the panels of Figure 3 together makes the transfer problem concrete. On TON_IoT both detectors have largely saturated by ten cost units. On UNSW-NB15 the curves are still near zero at that budget and only rise steeply beyond one hundred units. A bound calibrated on the first dataset applied to the second measures almost nothing.

Table 8 reports the validity test of Section 4.4 under both bound regimes, for all six dataset and detector combinations.

Table 8. Monotonicity validity test, maximum violation in percentage points, three-point tolerance.

Dataset	Model	Fixed $D = 50$	Calibrated P_{95}
CICIDS2017	MLP	0.41	**5.50**
	1D-CNN	0.00	0.00
TON_IoT	MLP	**3.60**	0.60
	1D-CNN	0.00	0.00
UNSW-NB15	MLP	**14.00**	0.60
	1D-CNN	0.00	0.80

Bold entries exceed the three-point tolerance.

Three features of Table 8 matter. First, the two violations under the fixed bound occur precisely on the two datasets whose cost scale that bound misrepresents. On TON_IoT a bound of 50 sits far above the entire cost distribution, whose median is 1.0, and on UNSW-NB15, it sits far below it, whose median is 264.5. On CICIDS2017, whose median of 5.5 places a bound of 50 in a workable range, no violation appears. The failures of the fixed bound are therefore not random across datasets; they track exactly the mis-scaling that motivates calibration. Calibration removes both violations, the severe one on UNSW-NB15 falling from fourteen percentage points to 0.60.

Second, calibration does not produce a clean sweep. CICIDS2017 with the multilayer perceptron passes under the fixed bound and fails under the calibrated one, at 5.50 percentage points. We report this rather than smooth it over. The percentile sweep below shows that this violation shrinks monotonically as the calibration percentile rises, from 7.54 at the ninetieth percentile to 2.85 at the ninety-ninth, and therefore behaves as a graded property of the operating bound rather than as an erratic one. Applying our exclusion rule at the primary percentile removes this configuration and its three scenarios from the pooled correlation analysis, which is why that analysis rests on fifteen points rather than eighteen.

Third, the test does not require knowing whether the reported evasion rates are correct. It only requires knowing that the scenarios are nested, which is a property of how the scenarios were constructed. This is what makes it usable as a routine check, and also what limits it. A configuration that passes has not been shown to be correct, only to be free of this particular inconsistency.

Figure 4 presents the test graphically for both bound regimes. Scenarios appear on the horizontal axis in order of increasing attacker capability, so a valid measurement produces non-decreasing lines.

Two parameters in the preceding analysis were chosen by us: the percentile at which the bound is calibrated, and the tolerance applied to the monotonicity test. We report a sweep over both, so that their influence is visible. The full attack matrix was rerun at the ninetieth and ninety-ninth percentiles, using the same sample draw, seed and attack configuration as the primary runs, giving thirty-six configurations in total across the three percentiles. Table 9 reports the resulting monotonicity violations, and Figure 5 presents the same result alongside the headline comparison of Table 7.

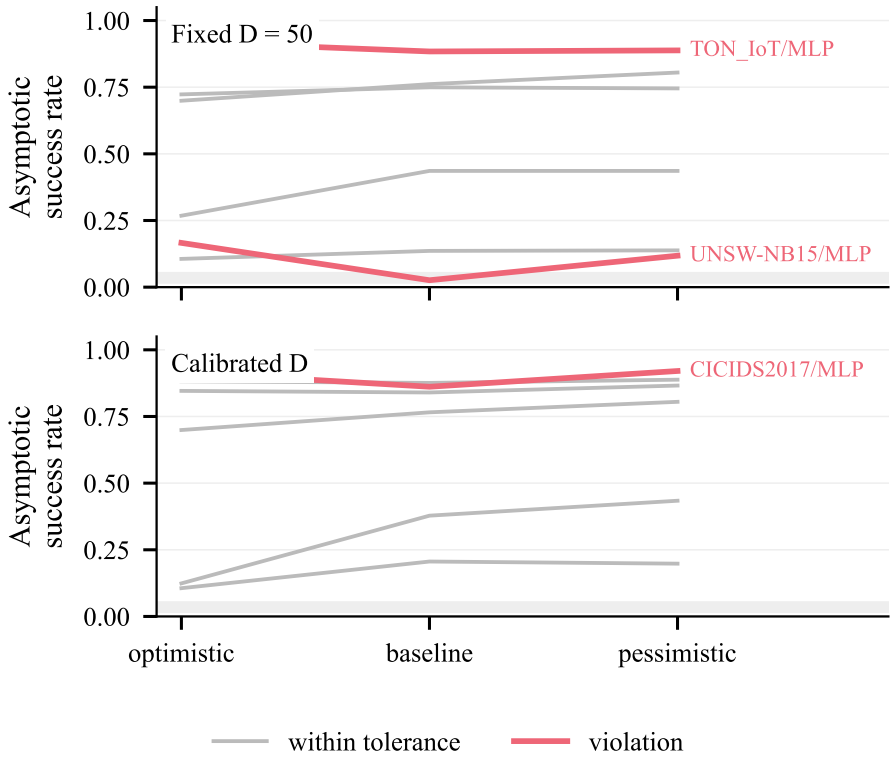


Figure 4. Monotonicity test under fixed and calibrated bounds.

Table 9. Monotonicity violation (pp) across calibration percentiles.

Dataset	Model	P_{90}	P_{95}	P_{99}
CICIDS2017	MLP	**7.54**	**5.50**	2.85
	1D-CNN	0.00	0.00	0.00
TON_IoT	MLP	0.40	0.60	1.60
	1D-CNN	0.00	0.00	0.40
UNSW-NB15	MLP	**4.60**	0.60	0.00
	1D-CNN	0.80	0.80	0.80

Bold entries exceed the three-point tolerance.

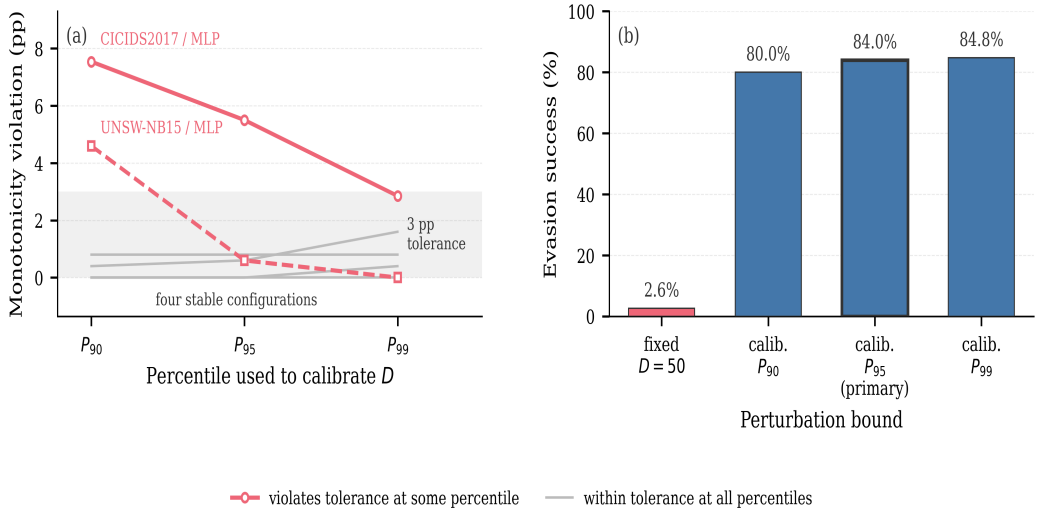


Figure 5. Sensitivity of the validity test and the headline comparison to the calibration percentile.

Two distinct conclusions follow, and separating them matters. The principal finding is insensitive to the percentile. As panel (b) of Figure 5 shows, the fixed bound produces an answer qualitatively different from every calibrated bound, at every percentile tested. Whether the bound is set at the ninetieth, ninety-fifth or ninety-ninth percentile of the dataset's own cost distribution, the measured evasion rate for the same detector rises from 2.6 percent to between 80.0 and 84.8 percent. The paper's central claim does not depend on the calibration rule we happened to specify.

The validity finding is more sensitive. Panel (a) shows that four of the six dataset and detector combinations remain within tolerance at all three percentiles, while two do not. CICIDS2017 with the multilayer perceptron exceeds the tolerance at the ninetieth and ninety-fifth percentiles and falls within it at the ninety-ninth. UNSW-NB15 with the multilayer perceptron exceeds it at the ninetieth only. In both cases the violation shrinks monotonically as the bound grows; it does not oscillate or change direction. We draw the conservative reading from this. Whether a borderline configuration is judged valid can itself depend on the operating bound, which reinforces rather than weakens the case for treating monotonicity as a diagnostic check rather than as a certificate. We hold the ninety-fifth percentile as the primary rule throughout, as pre-specified, and we do not adopt the ninety-ninth percentile despite its producing the cleanest validity outcome. Selecting the percentile that resolves a violation would be precisely the circularity that Section 4.3 rules out. Table 10 reports the second sweep, over the tolerance applied to the monotonicity test.

Table 10. Configurations passing the monotonicity test at three tolerances.

Bound Regime	1 pp	3 pp	5 pp
Fixed $D = 50$	4 / 6	4 / 6	5 / 6
Calibrated P_{95}	5 / 6	5 / 6	5 / 6

Under the calibrated bound the outcome is identical at all three tolerances, for every configuration. Under the fixed bound it is not: TON_IoT with the multilayer perceptron, whose violation is 3.60 percentage points, flips from failing to passing between the three and five point tolerances. The three-point choice was therefore not a knife edge for the regime the paper actually adopts, though it partially was for the regime the paper argues against.

Turning from the bound to what the attack actually does, Table 11 reports the most frequently modified features in successful constrained evasions on CICIDS2017, annotated with their control surface category.

Table 11. Most frequently modified features in successful constrained evasions, CICIDS2017.

Model	Feature	Cat.	Mod. Rate
MLP	Fwd Packet Length Std	2	100.0%
	Flow IAT Mean	2	100.0%
	Flow IAT Max	2	100.0%
	Avg Fwd Segment Size	2	100.0%
	Destination Port	1	99.7%
	Packet Length Std	2	99.7%
1D-CNN	Avg Fwd Segment Size	2	100.0%
	Average Packet Size	2	95.7%
	Subflow Fwd Bytes	2	91.4%
	Max Packet Length	2	87.3%
	Fwd Header Length	1	83.9%
	Init_Win_bytes_forward	1	79.6%

Two observations follow from Table 11. Both detectors draw overwhelmingly on Category 2 features, that is, quantities the attacker emits but cannot alter without cost. No Category 3 feature appears, which is expected since those are frozen by construction and serves as a further check that the mask is being enforced. The two detectors differ in emphasis. The dense network leans on inter-arrival timing in addition to packet length statistics, whereas the convolutional network leans on byte volume, rate and header level fields. We are able to produce this analysis for CICIDS2017 only. The corresponding adversarial feature vectors were not retained for the other two datasets, and reconstructing them would require rerunning the attack. We state this rather than extrapolate, and note it as a limitation.

We now report the negative result. Before presenting the correlation, we establish that the predictor varies. Figure 6 shows the distribution of normalised decision mass across control surface categories for every dataset and detector combination under the baseline scenario. Control surface reliance, defined in (6), is the Category 3 share of each bar. It differs substantially across configurations, ranging from near zero on UNSW-NB15 to roughly one quarter on CICIDS2017. This matters because an absent correlation would be uninformative if the predictor were near constant.

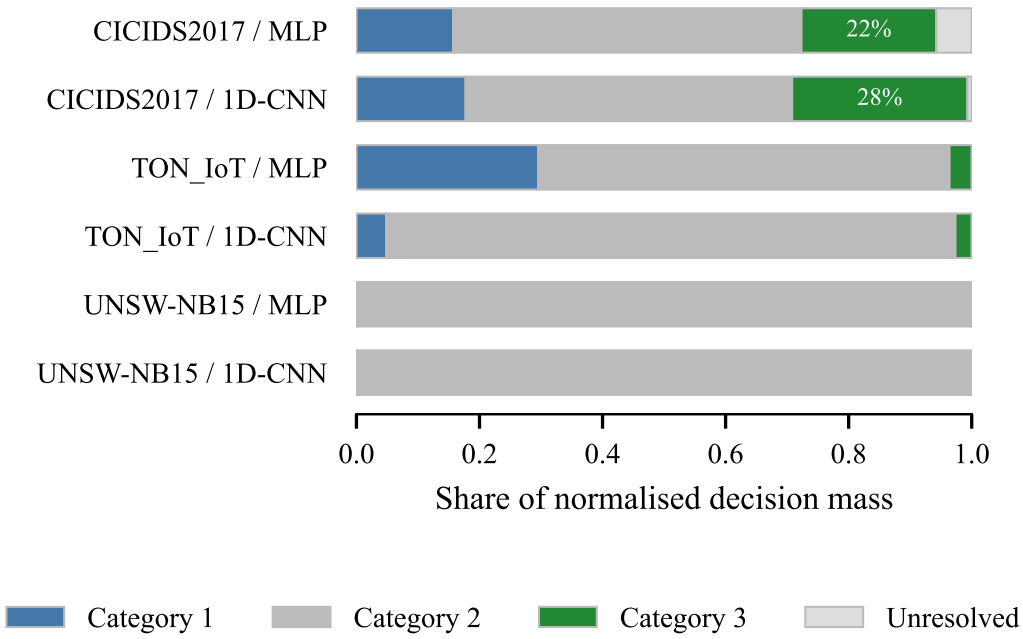


Figure 6. Decision mass by control surface category.

On CICIDS2017 considered alone, the hypothesis appeared well supported. Every predictor and outcome pair we examined carried the sign the hypothesis predicts. At that point, a relationship between decision mass placement and adversarial robustness could have been reported on the basis of a single benchmark. Extending the analysis to TON_IoT and UNSW-NB15 removed that support. Table 12 reports the pooled correlations, computed over the fifteen configurations that satisfy the validity test at the primary percentile, with bootstrap confidence intervals and permutation-based p values. Because the three categorisation scenarios within a dataset/model pair are structurally nested rather than independent experimental replicates, the pooled correlation analysis is treated as exploratory. We therefore emphasise effect direction, uncertainty intervals, and cross-dataset sensitivity rather than interpreting the nominal p values as confirmatory inference.

Table 12. Correlation between attribution-derived predictors and evasion outcomes, pooled across three datasets.

Predictor	Outcome	n	Spearman ρ	Asymptotic p	Bootstrap 95% CI	CI Contains 0	Permutation p
CSR (Cat.3 mass)	Median evasion cost	15	−0.248	0.373	[−0.735, 0.350]	Yes	0.383
CSR (Cat.3 mass)	Asymptotic success rate	15	−0.002	0.995	[−0.600, 0.651]	Yes	0.998
CSR_B (Cat.3 + $\frac{1}{2}$Cat.2)	Median evasion cost	15	+0.005	0.985	[−0.544, 0.587]	Yes	0.988
CSR_B (Cat.3 + $\frac{1}{2}$Cat.2)	Asymptotic success rate	15	−0.257	0.355	[−0.775, 0.384]	Yes	0.355
Exposed mass (Cat.1 + Cat.2)	Median evasion cost	15	+0.253	0.363	[−0.342, 0.740]	Yes	0.372
Exposed mass (Cat.1 + Cat.2)	Asymptotic success rate	15	+0.004	0.990	[−0.644, 0.596]	Yes	0.992

The pooled analysis has a nested dependence structure: the fifteen observations form five dataset/model clusters of three scenarios each, with the three scenario rows sharing the same trained detector and attack procedure. As a sensitivity check, excluding CICIDS2017, TON_IoT, and UNSW-NB15 in turn gives $\rho = -0.301$, -0.211, and -0.155, respectively, so the sign remains negative and no single dataset reverses the direction. A dataset-level block bootstrap gives a 95% interval of $[-0.3106, 0.0000]$, while a dataset/model-level block bootstrap gives $[-0.6246, -0.0225]$. Because only three datasets and five dataset/model clusters are available, these block-bootstrap intervals have limited resolution and are treated as sensitivity checks rather than confirmatory inference. The primary pooled analysis and its conclusion are unchanged.

Every bootstrap interval in Table 12 contains zero. The permutation p values agree closely with the asymptotic ones, which indicates that the small sample approximation was not itself the source of the null result. The widest interval spans from a moderately strong negative correlation to a moderate positive one, which is what an uninformative point estimate looks like when its uncertainty is made explicit. The sign test provides an independent view. Across all predictor and outcome comparisons, twelve of twenty-four matched the hypothesised direction. That proportion is exactly fifty percent, giving an exact binomial p value of 1.000 against a null of fifty percent, with a ninety-five percent confidence interval on the proportion of $[29.1\%, 70.9\%]$.

Figure 7 shows the pooled relationship for the primary predictor and outcome pair, with one panel per dataset on a shared logarithmic vertical axis. Within each panel, marker shape denotes the categorisation scenario and marker colour and fill denote the detector. In panel (b), the MLP baseline and pessimistic configurations have identical values and their markers therefore coincide. We fit no trend line, since fitting one would imply a relationship the data does not support.

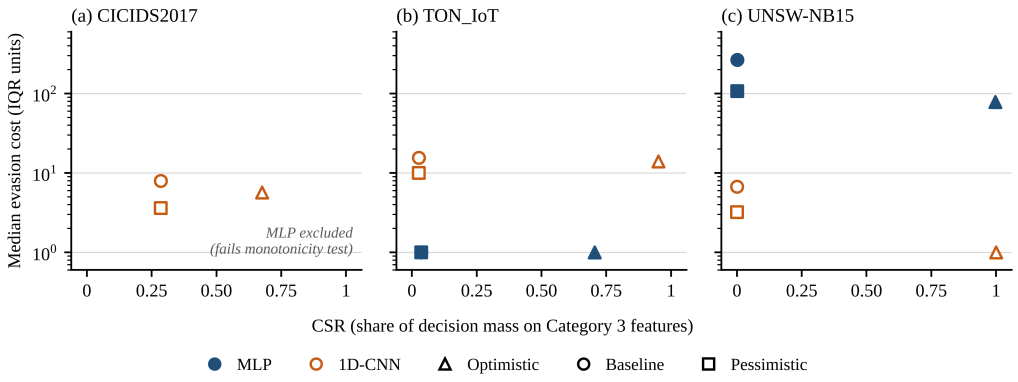


Figure 7. Control surface reliance against median evasion cost, by dataset.

We state the conclusion in the form the evidence supports. We find no evidence of a consistent relationship between attribution-derived control surface reliance and evasion

cost across the three datasets and two architectures evaluated. We do not claim that no such relationship exists. Fifteen points cannot support that stronger claim, and the width of the bootstrap intervals in Table 12 makes the limitation explicit rather than hidden.

One further observation arose during detector selection and is worth recording. We evaluated a reconstruction-based anomaly detector alongside the discriminative models. On CICIDS2017 it performed close to chance, reaching a ROC-AUC of 0.542, and a bounded retuning sweep across bottleneck width, depth, threshold percentile and training length improved this only to 0.591. We excluded it from adversarial targeting on the grounds that measuring evasion cost against a detector that barely separates the classes is uninformative, the same reasoning applied to the marginal detector flagged in Table 3.

Per-class analysis of reconstruction error suggested a structural explanation. Attacks producing an extreme statistic within a single flow separated strongly, with reconstruction error ratios against benign traffic of 153.6 for slowloris, 35.8 for slow HTTP test and 20.8 for infiltration. Attacks whose anomaly appears only across flows were invisible, with ratios of 0.0023 for port scan, effectively zero for web attacks and 0.035 for bot traffic. An individual port scan flow is statistically indistinguishable from an ordinary failed connection. The signal lives in the rate, not in the record.

We recorded a prediction from this before evaluating the other datasets. If the explanation is general rather than specific to CICIDS2017, then a per-flow reconstruction detector should perform better on datasets whose attack mix contains more within-flow anomalous attacks. The subsequent results were mixed. On UNSW-NB15 the detector achieved a ROC-AUC of 0.910, consistent with an attack mix containing exploits, shellcode and backdoors that produce within-flow irregularities. On TON_IoT it achieved 0.321, which is not merely weak but below chance, and which the prediction does not account for. We report this as a secondary observation rather than a contribution. It illustrates the same theme as the main results, namely that properties which appear to be characteristics of a detection approach can turn out to be characteristics of the benchmark it was measured on.

7. Discussion

Our results support two claims, and they are not equally strong. Stating them separately is more useful than merging them. The first is robust. A fixed reference perturbation bound can drastically distort measured evasion, and this conclusion holds across every calibration percentile we tested. It rests on a more-than-thirty-fold discrepancy in measured evasion rate that persists at the ninetieth, ninety-fifth and ninety-ninth percentiles alike, and on a cost spread across benchmarks of more than two orders of magnitude measured in a scale-robust unit. The second is more contingent. Calibration generally improves the internal consistency of the measurement, and it removes both of the severe violations produced by the fixed bound. It does not guarantee consistency everywhere, and for a borderline configuration the validity verdict can itself shift with the operating bound. This is a limitation of the test and also reinforces the paper's central point: the bound influences not only what is measured but whether the measurement can be trusted.

The sequence of results in this study followed a specific shape. A relationship was observed on a single widely used benchmark, appeared robust under every check applied within that benchmark, and then failed when the same procedure was applied to independent data. We report the sequence rather than only its endpoint, because the intermediate result is the point at which many studies would stop. There is a structural reason to expect this shape in adversarial NIDS research. Benchmark datasets in this field differ from one another far more than benchmark datasets in vision do. They are collected on different networks, with different tools, over different durations, with different attack tooling and different labelling procedures, and in some domains, such as vehicular networks, the train-

ing data are themselves generated by network simulation rather than captured from live traffic [17]. Figure 2 makes one dimension of that heterogeneity visible directly, and Table 6 makes a second visible in the shape of the upper tail. A relationship that holds on one benchmark is not obviously a property of intrusion detection. It may be a property of that capture environment. The general form of this observation is the paper's central claim. Part of what fails to transfer across datasets is not the detector and not the attack, but the evaluation instrument itself.

Three practices follow for evaluation design. Perturbation bounds should be derived from the dataset under evaluation rather than reused as fixed absolute values across datasets. The procedure in Section 4.3 is inexpensive. It requires one unbounded pass to obtain the cost distribution, after which the bound follows from (2). The cost of not doing it is illustrated by Table 7 and Figure 3. Because the shape of the cost tail is itself dataset dependent, as Table 6 shows, no percentile can be recommended universally; what can be recommended is that the rule be pre-specified and its influence reported. Cost should also be measured rather than imposed wherever the evaluation design permits. A hard shared budget interacts with the size of the attacker's action space in ways that distort comparison between configurations, and retrospective thresholding of recorded costs produces the same curves without that interaction. Finally, nested attacker capability sets should be checked for monotonicity before results are reported. The check costs nothing beyond the categorisation that a constrained evaluation already requires; it needs no ground truth, and in our experiments it flagged a defect that clean performance metrics and constraint validation both passed over. It should be read as a diagnostic rather than as a certificate, for the reasons set out above.

The calibration procedure itself does not depend on the detector being differentiable, which matters for practitioners whose deployed detectors are tree ensembles such as random forest or XGBoost. Equation (2) requires only a per-sample record of the cost at which evasion first occurs under an effectively unbounded search, and any attack that produces such a record can supply it. For non-differentiable models, the gradient step in Equation (1) can be replaced by a gradient-free search, for example a query-based decision attack that moves along the decision boundary using only predicted labels, a zeroth-order method that estimates the score in Equation (1) by finite differences of the model output, or an evolutionary search over the perturbable features. The essential requirement is that the replacement attack operate inside the same control-surface mask and projection layers, and that it record cost in the same category-weighted IQR units, so that the resulting distribution is comparable in kind to the one reported here. The bound is then set at the pre-specified percentile exactly as before, and the monotonicity test of Section 4.4 applies unchanged because it depends only on the nesting of the scenarios. Two cautions apply. Gradient-free attacks are generally weaker searchers, so the recorded costs are upper bounds on the true minimum evasion cost and the calibrated bound should be reported together with the attack that produced it; and because calibration is tied to the attack, a bound derived with one attack family should not be reused with another without recalibration. Validating this extension empirically is part of the future work identified in Section 8.

The negative result reported above is, in substance, a result about explainability. CSR is computed entirely from feature attribution, and Figure 6 is a direct visualisation of that attribution partitioned by attacker reach. The hypothesis it encodes is that attribution reveals where a model is vulnerable, in a way that predicts what an attacker will actually be able to do. We are careful about how this relates to existing work. Attribution-based adversarial detection in NIDS is an established and active line of research, with explanation drift used as a signal for identifying adversarial inputs at inference time [30,31]. Our

result neither extends nor contradicts that line, because it addresses a different question. Detecting that an input is adversarial after the fact is not the same as predicting, in advance and without running an attack, how much effort evasion will require. What our result does suggest is that the predictive use of attribution for robustness estimation deserves more scepticism than its intuitive appeal invites. Attribution measures where a model places weight, and that placement can itself shift with how the same data are represented to the model [29]. Evasion cost depends additionally on the geometry of the decision boundary in the directions the attacker can actually move, on the joint distribution of the features involved and on how far the sample sits from the boundary to begin with. Weight placement is one input to that, and on our evidence it is not a sufficient one.

8. Limitations

Several limitations should be considered when interpreting the findings. First, the study evaluates two differentiable detector architectures and a single constrained attack algorithm. While the consistency of the results across three independent datasets and two architectures supports the robustness of the main finding, broader validation with tree-based detectors and structurally different attack strategies remains an important direction for future work.

Second, the statistical analysis is based on fifteen dataset–model–scenario combinations, and the resulting uncertainty is therefore non-negligible. We address this by reporting confidence intervals and dataset-level sensitivity analyses rather than relying solely on point estimates. Larger experimental designs would provide tighter uncertainty bounds.

Third, the feature-space constraints capture the dependencies explicitly represented by the underlying feature extractor, but cannot guarantee semantic validity for relationships that are not encoded in the extractor. Consequently, the reported values should be interpreted as constrained feature-space evasion costs rather than direct measurements of physical attacker effort. In addition, the feature-level attack-surface analysis was available for only one dataset because the corresponding adversarial feature vectors were not retained for the other datasets.

Finally, the calibration bound was estimated from 200 baseline-correct malicious samples drawn from the same parent test partition as the final evaluation. Although the calibration and evaluation samples were separately drawn and their observed overlap was small, this does not constitute a fully independent calibration/test split. A held-out calibration partition or cross-fitting would provide a cleaner design. A genuine cross-dataset P_{95} transfer analysis would likewise require rerunning the constrained attack with a foreign dataset's calibrated bound imposed during optimisation and is therefore left to future work. During feature categorisation, one documented CICFlowMeter relationship for average packet size was also found to be inconsistent with a substantial portion of the released CICIDS2017 values; the feature was therefore excluded from the dependency specification rather than enforcing an unverifiable constraint.

9. Conclusions

Bounding attacker perturbations is unavoidable in adversarial evaluation of network intrusion detectors, but our results show that an absolute feature-space bound should not be assumed to have comparable meaning across heterogeneous datasets without empirical calibration. Across CICIDS2017, TON_IoT, and UNSW-NB15, the measured constrained feature-space evasion cost differs by more than two orders of magnitude under the same attack procedure and the same IQR-normalised cost definition. On UNSW-NB15, changing only the operating bound from the fixed reference value of 50 to the dataset-calibrated

P_{95} value of 399.5 changes measured evasion success from 2.6% to 84.0%, while the same qualitative conclusion persists at P_{90} and P_{99}.

Dataset-specific quantile calibration and action-space monotonicity provide complementary safeguards for robustness evaluation: calibration aligns the numerical bound with the benchmark being evaluated, whereas monotonicity tests the internal consistency of nested attacker capabilities without requiring attack-success ground truth. Neither procedure certifies packet-level realisability, and the resulting costs should be interpreted within the constrained feature-space threat model. The present calibration protocol uses a dedicated sample draw from the test partition rather than a separate held-out calibration partition; although observed sample overlap with the final evaluation draw was low and consistent with incidental overlap, future work should use independent calibration/test splits.

Our secondary attribution analysis does not provide evidence for a stable cross-dataset relationship between control-surface reliance and evasion cost. This negative result is intentionally retained because it illustrates the same broader point: patterns that appear robust on one benchmark may reflect the capture environment rather than a transferable property of the detector. Future work should evaluate independent calibration/test splits, structurally different constrained attacks, tree-based detector families, packet- or traffic-space realisations of the proposed calibration framework, and the cross-dataset transfer of calibrated bounds under foreign search-time constraints.

Author Contributions: Conceptualisation, M.S.A.; Software, M.S.A.; Investigation, M.S.A. and M.A.-E.; Data Curation, M.S.A. and N.F.; Writing—Original Draft Preparation, M.S.A.; Methodology, W.A.A. and I.S.M.; Validation, W.A.A. and I.S.M.; Formal Analysis, W.A.A. and I.S.M.; Resources, W.A.A. and M.I.; Writing—Review and Editing, W.A.A. and I.S.M.; Visualisation, I.S.M.; Project Administration, W.A.A.; Supervision, W.A.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 control surface categorisations for all three datasets, the calibrated bounds, the per-sample evasion cost records, the constraint engine, and the analysis scripts are archived in a private repository. An anonymised reviewer-access mechanism will be provided during peer review where the submission system permits it; the repository will be made publicly available upon publication.

Conflicts of Interest: Author Mohsen S. Alsaadi was employed by Ebttikar Technology Company Ltd. in the Research and Development Department. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The authors declare no conflicts of interest.

References

1. Lhioui, C. Intelligent hybrid intrusion detection systems for IoT-Cloud environments. *J. Intell. Syst. Appl. Data Sci.* **2025**, *3*, 91–103. [CrossRef]
2. Al-Essa, M.; Ali, W.A.; Roccotelli, M.; Fanti, M.P. A Practical Enhancement for Intrusion Detection Systems with Validation-Gated Mixture-of-Experts (VG-MoE). In Proceedings of the 2026 IEEE Conference on Artificial Intelligence (CAI), Granada, Spain, 8–10 May 2026; pp. 7–12.
3. Ali, W.A.; Sandhya, P.; Roccotelli, M.; Fanti, M.P. A comparative study of current dataset used to evaluate intrusion detection system. *Int. J. Eng. Appl. (IREA)* **2022**, *10*, 336–344. [CrossRef]
4. He, Z.; Ouyang, C.; Wen, L.; Liu, C.; Moreira, C. TabAttackBench: A benchmark for adversarial attacks on tabular data. *Expert Syst. Appl.* **2026**, *301*, 130491. [CrossRef]

5. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
6. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
7. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. In Proceedings of the International Conference on Learning Representations (ICLR) Workshop, Toulon, France, 24–26 April 2017.
8. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
9. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (S&P), San Jose, CA, USA, 22–26 May 2017; pp. 39–57.
10. Moosavi-Dezfooli, S.-M.; Fawzi, A.; Frossard, P. DeepFool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.
11. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. In Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P), Saarbruecken, Germany, 21–24 March 2016; pp. 372–387.
12. Croce, F.; Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Proceedings of the International Conference on Machine Learning (ICML), Virtual, 13–18 July 2020; pp. 2206–2216.
13. Pierazzi, F.; Pendlebury, F.; Cortellazzi, J.; Cavallaro, L. Intriguing properties of adversarial ML attacks in the problem-space. In Proceedings of the IEEE Symposium on Security and Privacy (S&P), San Francisco, CA, USA, 18–21 May 2020; pp. 1332–1349.
14. Apruzzese, G.; Andreolini, M.; Ferretti, L.; Marchetti, M.; Colajanni, M. Modeling realistic adversarial attacks against network intrusion detection systems. *Digit. Threat. Res. Pract.* **2022**, *3*, 1–19. [CrossRef]
15. Chernikova, A.; Oprea, A. FENCE: Feasible evasion attacks on neural networks in constrained environments. *ACM Trans. Priv. Secur.* **2022**, *25*, 1–34. [CrossRef]
16. Han, D.; Wang, Z.; Zhong, Y.; Chen, W.; Yang, J.; Lu, S.; Shi, X.; Yin, X. Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2632–2647. [CrossRef]
17. Ali, W.A.; Alsaadi, M.S.; Roccotelli, M.; Mangini, A.M.; Fanti, M.P. Real-time Sybil attack detection in vehicular networks using simulation-based machine learning. In Proceedings of the 2025 12th International Conference on Wireless Networks and Mobile Communications (WINCOM), Riyadh, Saudi Arabia, 25–27 November 2025; pp. 1–6.
18. Sheatsley, R.; Papernot, N.; Weisman, M.; Verma, G.; McDaniel, P. On the robustness of domain constraints. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), Virtual, 15–19 November 2021; pp. 495–515.
19. Teuffenbach, M.; Piatkowska, E.; Smith, P. Subverting network intrusion detection: Crafting adversarial examples accounting for domain-specific constraints. In Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction (CD-MAKE), Dublin, Ireland, 25–28 August 2020; pp. 301–320.
20. ElShehaby, M.; Matrawy, A. A novel perturb-ability score to mitigate evasion adversarial attacks on flow-based ML-NIDS. *J. Inf. Secur. Appl.* **2026**, *99*, 104409. [CrossRef]
21. Ballet, V.; Renard, X.; Aigrain, J.; Laugel, T.; Frossard, P.; Detyniecki, M. Imperceptible adversarial attacks on tabular data. *arXiv* **2019**, arXiv:1911.03274.
22. Mathov, Y.; Levy, E.; Katzir, Z.; Shabtai, A.; Elovici, Y. Not all datasets are born equal: On heterogeneous tabular data and adversarial examples. *Knowl. Based Syst.* **2022**, *242*, 108377. [CrossRef]
23. Simonetto, T.; Dyrmishi, S.; Ghamizi, S.; Cordy, M.; Le Traon, Y. A unified framework for adversarial attack and defense in constrained feature-space. In Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI), Vienna, Austria, 23–29 July 2022; pp. 1313–1319.
24. Kireev, K.; Kulynych, B.; Troncoso, C. Adversarial robustness for tabular data through cost and utility awareness. *arXiv* **2022**, arXiv:2208.13058.
25. Al-Essa, M.; Ali, W.A.; Alsharo, M.; Atoum, M.S.; Imran, M. COBRA: Counterfactual oversampling framework for imbalanced structured data classification. *Expert Syst. Appl.* **2025**, *304*, 130764. [CrossRef]
26. Carlini, N.; Athalye, A.; Papernot, N.; Brendel, W.; Rauber, J.; Tsipras, D.; Goodfellow, I.; Madry, A.; Kurakin, A. On evaluating adversarial robustness. *arXiv* **2019**, arXiv:1902.06705.
27. Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 4765–4774.
28. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.

29. Malik, A.E.; Alsharo, M.; Alnsour, Y.; Ali, W.A.; Almomani, O. Transforming data representation: A comparative analysis of tabular and image-based approaches with XAI. *J. Comput. Cogn. Eng.* **2026**, *5*, 333–340. [CrossRef]
30. Maseno, E.M.; Sun, Y.; Wang, Z. Detecting adversarial evasion in deep learning intrusion detection systems using explainable AI. *Int. J. Inf. Sec.* **2026**, *25*, 125. [CrossRef]
31. Chowdhury, A.P.; Nur, F.N.; Islam, A.H.M.S.; Alam, K.; Karim, A.; Shah, M.A. FLEX-IDS: A secure and explainable federated intrusion detection framework for Edge-of-Things environments under adversarial conditions. *Comput. Electr. Eng.* **2026**, *129*, 110827. [CrossRef]
32. González-Ramos, J.A.; Abril, E.J.; Fernández, P.; Prieto, J.; Chamoso, P. Adversarial robustness evaluation of hybrid CNN-LSTM-transformer NIDS on evolving threats. *J. Inf. Secur. Appl.* **2026**, *100*, 104467. [CrossRef]
33. Li, M.; Qiao, Y.; Lee, B. Adversarial robustness evaluation for multi-view deep learning cybersecurity anomaly detection. *Future Internet* **2025**, *17*, 459. [CrossRef]
34. Hakim, M.A.; Uddin, M.S.; Anis, T.I. Cross-domain generalization failure in lightweight intrusion detection models for IIoT networks. *arXiv* **2026**, arXiv:2607.00553. [CrossRef]
35. Darem, A.; Alhashmi, A.; Sheatah, H.K.; Mohamed, I.B.; Jabnoun, C.; Allan, F.M.; Dhibi, A.; Elmourssi, D.M. Decoding the deception: A comprehensive analysis of cyber scam vulnerability factors. *J. Intell. Syst. Appl. Data Sci.* **2024**, *2*, 29–41.
36. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP), Funchal, Portugal, 22–24 January 2018; pp. 108–116.
37. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems. In Proceedings of the Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6.
38. Moustafa, N. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustain. Cities Soc.* **2021**, *72*, 102994. [CrossRef]
39. Elshaik, H.; Edam, S. Analyzing ML/DL techniques for SDN-based DDoS attack detection: A comparative study. *J. Intell. Syst. Appl. Data Sci.* **2023**, *1*.

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.