

A Multi-Seed Evaluation of Explanation Faithfulness and Stability under Adversarial Evasion in IoT/IloT Intrusion Detection

S. M. Nihal Ahmed

sm.nihal4@gmail.com

Daffodil International University

Afrim Hossen Khan

Daffodil International University

Sabikun Nahar Sinthia

Daffodil International University

Research Article

Keywords: IoT intrusion detection, IloT security, adversarial evasion, problem-space constraints, explainable artificial intelligence, SHAP, LIME, explanation faithfulness, explanation stability, reproducibility

Posted Date: October 28th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11132050/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

A Multi-Seed Evaluation of Explanation Faithfulness and Stability under Adversarial Evasion in IoT/IIoT Intrusion Detection

S. M. Nihal Ahmed^{1*}, Afrim Hossen Khan¹ and Sabikun Nahar Sinthia¹

^{1*}Department of Software Engineering, Daffodil International University, Birulia, Savar, Dhaka, Bangladesh.

*Corresponding author(s). E-mail(s): sm.nihal4@gmail.com;

Abstract

Explanations are increasingly attached to IoT/IIoT intrusion detectors, but whether they stay faithful and stable under evasion, and whether such findings survive retraining, is rarely tested. We audit SHAP and LIME on de-duplicated ToN-IoT and Edge-IIoTset data, using the training seed as the unit of inference (20 seeds per detector, pre-declared outcomes, Holm control). An MLP, the MLP with FGSM or PGD adversarial training, and XGBoost are attacked under unconstrained, numeric-only and domain-constrained threat models, and robustness is compared at matched false-positive rates. Successful feature-space attacks produced no syntactically valid records; after projection onto valid records, success against the MLP fell from 0.31 to 0.06 (ToN-IoT) but against XGBoost only from 1.00 to 0.60. Domain constraints kept success at small budgets below 0.12, yet at a 1% false-positive budget maximum padding evaded every ToN-IoT MLP, including the adversarially trained ones, and a tree-aware attack evaded XGBoost on 35% of padded attacks where 3,000-query random search found almost none. MLP vulnerability on ToN-IoT was bimodal across seeds, largely a threshold effect, and invisible in clean macro-F1. SHAP was more faithful than LIME in all 52 primary comparisons, for every masking reference, for KernelSHAP and for every LIME configuration; under conditional in-fill masking LIME was near random. SHAP attributions were stable under realisable perturbations, whereas LIME's change under attack was mostly within its own sampling noise. Retaining label-conflicting records or partitioning chronologically within attack families changed clean performance little. Code, seed-level results and logs are openly archived.

Keywords: IoT intrusion detection, IIoT security, adversarial evasion, problem-space constraints, explainable artificial intelligence, SHAP, LIME, explanation faithfulness, explanation stability, reproducibility

1 Introduction

IoT and IIoT networks connect heterogeneous devices and industrial processes, so an intrusion detector has to remain dependable as traffic and attack behaviour vary [1–3]. A clean-test score is only one view of that dependability: it does not show whether the detector resists evasion, or whether the explanation attached to an alert remains useful when the input or the model changes [4, 5].

Three problems make published evidence on these questions hard to interpret. The first is inferential: one trained model on one split cannot show whether a robustness or explanation pattern persists across independently trained models, and treating the records of one test set as replicates understates the variation caused by retraining [6]. The second is realism: gradient attacks in a normalised feature space can change fields an attacker does not control or produce values no network stack would emit, so their success rates are upper bounds of unknown distance from realisable evasion [7, 8]. The third is measurement: deletion-based faithfulness scores depend on what replaces a deleted feature and on which output is tracked, and can be dominated by out-of-distribution artefacts [9, 10].

This paper addresses these problems with one controlled audit. ToN-IoT is the primary benchmark and Edge-IIoTset an external validation benchmark; the two are never pooled. Within each dataset, the same de-duplicated partition, train-only preprocessing, detectors, attack budgets and explanation protocol are applied to 20 independently trained seeds. An MLP with and without adversarial training is compared with gradient-boosted trees, the strongest model family on tabular flow data [11]. Adversarial training serves as a controlled comparison, not as a proposed defence. The contribution is therefore an evaluation methodology and the evidence it produces, not a new detector, explainer or defence.

The closest prior study, by Munilla and Khammas [12], evaluates SHAP and LIME under DeepFool on three IoT datasets with paired tests over test samples. Our design differs in four respects, summarised in Table 1: inference over independently trained models rather than over samples of one model; a paired comparison of feature-space and domain-constrained evasion with a per-record validity audit; faithfulness for the predicted class under three masking references relative to a random-ranking control; and stability separated into input, explainer-call, background and training-seed components.

The contributions are:

1. an evaluation protocol for joint evasion–explanation audits in which the training seed is the unit of inference and primary outcomes and multiplicity families are fixed in advance;

Table 1 Positioning against the closest lines of prior work, characterised by what the cited works describe as their focus (— = not a focus of the cited work). “Seed-level” means that independently trained models, not test records, are the inferential unit.

Line of work	Evasion threat model	Explanation evaluation	Inference
SHAP/LIME under DeepFool [12]	Feature-space DeepFool	Sparsity, completeness and robustness of SHAP and LIME	Paired tests over test samples
Constrained or realistic evasion [8, 13, 14]	Validity, coherence or problem-space constraints	—	—
Quantitative XAI for IoT IDS [15–17]	—	Quantitative explanation metrics	—
This work	Feature-space and domain-constrained; white-box, black-box, tree-aware and transfer attacks; per-record validity audit	Predicted-class AOPC under three references vs. random control; input, call, background and seed stability	Seed-level (20 seeds), pre-declared outcomes, Holm control

2. a paired comparison of feature-space and domain-constrained evasion for neural and tree-based detectors, with a validity audit, projection of invalid records onto the valid input space, and robustness reported at matched false-positive rates and with threshold-free measures;
3. a reference-controlled analysis of deletion-based faithfulness, with sensitivity analyses of the explainer configuration;
4. a decomposition of explanation instability into input, explainer-call, background and training-seed components;
5. an open archive of code, seed-level results and execution logs from which every table and figure is recomputed.

The scope is bounded. The task is binary detection on flow and packet records. The constrained threat model enforces invariants that can be checked on these records; adversarial records are not regenerated as packets or replayed through a network stack. Convolutional, recurrent and attention-based detectors are not evaluated. Section 2 reviews related work, Section 3 specifies the protocol, Section 4 reports the results, Section 5 discusses implications and limitations, and Section 6 concludes. Supplementary material reports the multiclass analysis, the preprocessing replication path, the adversarial-training sweep and resource profiling.

2 Related Work

2.1 Learning-based detection on IoT/IIoT traffic

Surveys of IoT and IIoT intrusion detection report extensive use of supervised, anomaly-based and hybrid detectors, but also that benchmark results depend strongly on the chosen devices, attack families, feature representation and train–test protocol

[1, 2, 18, 19]. Public datasets such as BoT-IoT, Edge-IIoTset, CICIoT2023 and ToN-IoT make single-GPU research feasible, but differ in collection environment, labelling and temporal structure [18]. Deep architectures (convolutional, recurrent, transformer and attention-based) are increasingly proposed, yet surveys repeatedly note limited cross-dataset generalisation and missing out-of-distribution tests [2, 20, 21]. On tabular data generally, gradient-boosted trees remain stronger than deep networks on medium-sized datasets [11, 22], so a finding obtained only for a neural detector may not transfer to the model family most likely to be deployed.

2.2 Evasion attacks and their realism

Surveys of adversarial machine learning for network intrusion detection distinguish evasion from poisoning, white-box from black-box access, and gradient-based from query- and transfer-based attacks [4, 23]. FGSM, JSMA, DeepFool, PGD and substitute-model attacks have all been applied to NIDS, and reported vulnerability varies with the model family and the access assumed [24–27]. Defences include adversarial training, feature squeezing, ensembles and adversarial-example detection [28–32].

The central difficulty for tabular network records is realism. A feature-space perturbation is meaningful only if a realisable object exists that maps to it while respecting the domain’s available transformations and side effects [7]. A realistic NIDS adversary controls only part of the traffic, typically its own direction, volume and timing, and cannot rewrite the victim’s responses or protocol fields [8]. Validity and coherence constraints [13] and constrained-feature-space frameworks with mutability masks and feature relations [14] encode such restrictions. These works motivate reporting both attack success under the constraints and the share of unconstrained adversarial records that violate them.

2.3 Explanations and their evaluation

SHAP [33] and LIME [34] are the most common post hoc explainers in IoT intrusion detection, used to rank traffic features, describe global behaviour and justify individual alerts [15, 16, 35–38]. Both are estimators whose configuration is part of the experiment: SHAP depends on the background data and the estimator, and LIME on its sampling distribution, kernel width and surrogate. Plausible explanations need not be faithful to the model, and faithful explanations need not be useful to analysts [5, 39, 40].

Faithfulness is usually measured by deleting the features an explanation ranks highest and recording the change in output, summarised as an area over the perturbation curve (AOPC) [41] or as comprehensiveness and sufficiency [42]. Replacing a deleted feature by a fixed reference creates inputs outside the data distribution, so part of the output change reflects the model’s response to implausible records [9, 10]; the same issue affects Shapley estimators when features are dependent [43]. Stability has its own pitfalls: LIME’s sampling makes repeated explanations of the same instance differ [44], and both SHAP and LIME can be manipulated by a model that

behaves differently on their perturbation distributions [45]. Exposing explanations is also dual-use, since attributions can reveal decision boundaries to an attacker [46–48].

2.4 Evaluation rigour and variance

Reviews identify inconsistent splits, duplicated or leaked records, unclear preprocessing and incomparable feature spaces as recurring barriers to reproducible IDS evaluation [2, 18, 19], and security-specific guidance lists sampling bias, label inaccuracy and data snooping among the common pitfalls [49]. Variance studies show that seeds for initialisation, data order and data splits can change benchmark conclusions as much as algorithmic choices [6]. Inference over test records of one trained model therefore does not capture between-model variation.

2.5 Gap and research questions

Studies that link explanations to evasion in IoT intrusion detection [12, 48, 50] use feature-space attacks without checking whether the adversarial records are realisable, use a single masking reference whose out-of-distribution effect is not controlled, and draw inferences from the test samples of one trained model. It remains unclear whether explanation changes under attack persist under realisable perturbations, whether they hold for tree ensembles as well as neural networks, how much of an apparent faithfulness difference is an artefact of the evaluation reference, and whether attack-induced instability exceeds the instability already present across explainer calls, background sets and retrained models. We address four research questions:

1. **RQ1:** How does evasion under feature-space and domain-constrained threat models change the predicted-class faithfulness of SHAP and LIME for neural and tree-based detectors, and how sensitive is this to the masking reference and the explainer configuration?
2. **RQ2:** How stable are SHAP and LIME explanations between clean and attacked records, compared with explainer-call, background-set and training-seed variability?
3. **RQ3:** Which differences between explainers, training conditions, model families and threat models are confirmed at the seed level under multiplicity control?
4. **RQ4:** How do detection, evasion resistance and explanation quality trade off at operationally realistic false-positive rates?

3 Methodology

3.1 Study design and analysis plan

The study is a repeated, paired design with five factors: dataset (ToN-IoT, Edge-IIoTset; analysed separately), detector (MLP, MLP with FGSM adversarial training, MLP with PGD adversarial training, XGBoost), threat model (unconstrained, numeric-only, domain-constrained), attack and budget, and explainer (SHAP, LIME). The training seed is the replication unit. Every detector is trained with the same 20 seeds on the same primary partition, and every seed is evaluated on the same fixed

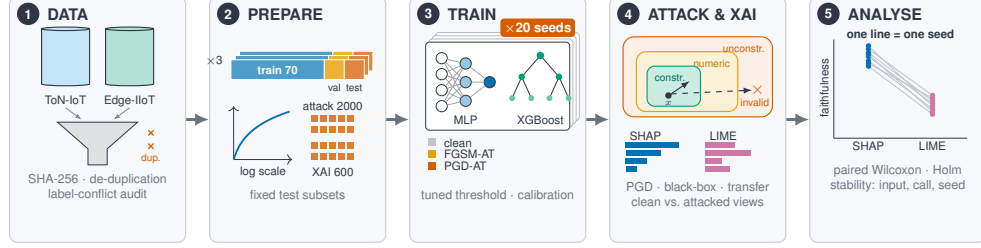


Fig. 1 Experimental pipeline. The two datasets are audited, de-duplicated and partitioned separately; four detectors are trained with 20 seeds; every detector is attacked under unconstrained, numeric-only and domain-constrained threat models; SHAP and LIME are evaluated on the same clean and attacked records; and all inference is carried out at the seed level.

test, attack and explanation records, so all comparisons are paired by seed. Figure 1 summarises the pipeline.

Three primary outcomes, their tests and their multiplicity families were fixed in the analysis code before the main runs: (P1) attack success rate (ASR) at $\epsilon = 0.05$ under each threat model; (P2) predicted-class comprehensiveness relative to a random-ranking control under the marginal reference; and (P3) the Spearman correlation between clean and attacked attributions. All other quantities are secondary. The analyses at matched false-positive rates over all seeds, the threshold-free robustness measures, the projection of invalid adversarial records, the stronger tree attacks, the explainer-configuration analyses and the data variants (label conflicts retained, temporal partitions) were specified after the primary runs and are reported as secondary.

Because all 20 seeds share one partition, the seed-level inference is conditional on that partition: it describes variation over training realisations for a fixed sample of the data, not over samples from the population of networks. Variation over partitions is estimated separately, with two further stratified partitions and with data variants that retain label conflicts or partition the records in time order (Section 3.10).

3.2 Datasets, audit and task formulation

ToN-IoT (network-flow table with Zeek-derived features) is the primary benchmark because it was built to study heterogeneous IoT/IIoT traffic [51, 52]. Edge-IIoTset (the packet-level table that its authors selected for machine learning) is the external validation benchmark because it comes from a separate IoT/IIoT testbed with fourteen attack families [53]. The two datasets are never concatenated. The task is binary: $y_i = 0$ for benign and $y_i = 1$ for malicious traffic. Attack-family labels are used only for stratification and per-family reporting.

For each dataset we record the SHA-256 hash and size of the raw file, the rows read, exact duplicate rows, and duplicates and label conflicts on the model-visible feature vector (Table 4). Records are de-duplicated on the feature vector together with the label *before* partitioning, so that no record can appear in both the training and the test partition. Feature vectors that occur with both labels (label conflicts) are removed in the primary analysis: their label cannot be determined from the model’s

input, so keeping them would place irreducible error in both training and test sets. Because removing them also removes records that are hard to classify, a sensitivity analysis retains them (Section 3.10).

3.3 Feature policy and preprocessing

Table 2 lists the feature policy. Addresses, ports, timestamps, free-text and payload fields, and capture artefacts that identify a stream or a host (for example the UDP stream index, raw TCP acknowledgement numbers, and Modbus transaction and unit identifiers) are removed, because they let a model memorise the testbed rather than learn traffic behaviour [49]. For Edge-IIoTset the removals follow the dataset authors’ recommended preprocessing [53]. Low-cardinality categorical protocol fields are one-hot encoded over the 20 most frequent training levels plus an “other” level. Categorical values that differ only in numeric formatting are merged; in Edge-IIoTset, for example, the values 0 and 0.0 occur as distinct labels of the same field. A protocol field that is missing for a record is encoded as zero, meaning that the field is absent.

All learned preprocessing is fitted on the training partition only. Each non-negative numeric feature is transformed by $\log(1 + x)$ and then min–max scaled to $[0, 1]$ using the training range; validation and test values are clipped to $[0, 1]$. Features that are constant on the training partition are removed. The transformation is monotone and invertible on the training range, which allows adversarial records to be mapped back to raw units for the constraint repair in Section 3.5. For explanations and all explanation metrics, the one-hot columns of a categorical field are aggregated into one *feature group*: a group’s attribution is the sum of its columns’ attributions, and a group is masked as a whole.

One family-stratified 70/15/15 partition per dataset (seed 2026) is the primary partition. Fixed record subsets are drawn from each test partition: 2,000 malicious records stratified by attack family for the attack evaluation, of which the first 300 (again family-stratified, with at least one record per family) are explained together with 300 random benign records. The partitions and subsets are fully determined by the hash-verified raw files and the fixed seeds.

3.4 Detectors, training conditions and operating points

The neural detector is a binary MLP with three fully connected ReLU hidden layers and a sigmoid output:

$$\begin{aligned} h_1 &= \text{ReLU}(W_1 z + b_1), & h_2 &= \text{ReLU}(W_2 h_1 + b_2), \\ h_3 &= \text{ReLU}(W_3 h_2 + b_3), & \hat{p}_\theta(y = 1 \mid z) &= \sigma(w_4 h_3 + b_4). \end{aligned} \quad (1)$$

It uses hidden widths 128–64–32, dropout 0.2, Adam with learning rate 10^{-3} , batch size 2048, at most 30 epochs and early stopping after five epochs without improvement in validation macro-F1. The loss is class-weighted binary cross-entropy,

$$\mathcal{L}(\theta; z, y) = -w_y [y \log \hat{p}_\theta(z) + (1 - y) \log(1 - \hat{p}_\theta(z))], \quad (2)$$

Table 2 Feature policy. Retained features that are not mutable (Table 3) are immutable under the constrained threat model.

	ToN-IoT	Edge-IIoTset
Removed	addresses, ports, DNS query string, SSL subject/issuer, HTTP URI/user agent, anomaly-event name and details	frame time, IP hosts, ARP addresses, ports, TCP payload and options, MQTT message, HTTP URI/query/file data, ICMP timestamp, UDP stream index, raw TCP acknowledgement number, Modbus transaction/unit identifiers, HTTP TLS port
Categorical (one-hot)	transport protocol, service, connection state, DNS flags, SSL version/cipher/flags, HTTP method/version/MIME types, anomaly-notice flag	HTTP method/referer/version, DNS query-name length, MQTT connect-acknowledge flags/protocol name/topic
Numeric	duration, bytes, packets, IP bytes (both directions), missed bytes, DNS codes, HTTP depth/body lengths/status	all remaining numeric packet and protocol fields
Candidate relations	source IP bytes $\geq$ source bytes + $20 \times$ source packets, and likewise for the destination direction (hold in 98.8% and 98.2% of training flows; not enforced)	—
Retained (columns / groups)	88 / 31	53 / 34

with w_y computed from training class frequencies.

Three training conditions share this architecture. *Clean* training minimises $\mathcal{L}$. *FGSM adversarial training* replaces every minibatch by its FGSM perturbation with training budget $\epsilon_t = 0.01$ [54, 55]. *PGD adversarial training* perturbs each minibatch by a five-step PGD ascent with random start and $\epsilon_t = 0.03$, and minimises $\lambda \mathcal{L}(z^{\text{adv}}) + (1 - \lambda) \mathcal{L}(z)$ with $\lambda = 0.5$ [56]. Adversarial training uses the unconstrained perturbation set. The tree-based detector is XGBoost [22] with up to 400 trees of depth 6, learning rate 0.1, row and column subsampling 0.8, the positive-class weight set from the training class ratio and early stopping on validation log-loss. It is trained on the same scaled inputs, which leaves its splits unaffected but places all detectors in the same attack space. Supplementary Table S1 lists the full configuration.

Operating points. Robustness depends on the decision threshold, so three operating points are reported. (i) The *validation-tuned* threshold maximises validation macro-F1. It defines the predicted class used by the explanation metrics and the pre-declared outcome P1, but it is a property of the detector rather than of a deployment: on Edge-IIoTset it implies a false-positive rate of about 14%, which is too high for operation. (ii) *Fixed false-positive budgets*: thresholds are set on the benign validation records to target false-positive rates of 0.1, 0.5, 1, 2 and 5%. We use the 1% budget as the operational reference for robustness, because it compares detectors at equal alert cost. (iii) *Threshold-free measures*: the ROC-AUC between benign test records and attacked

malicious records, and the per-record decrease in the malicious log-odds caused by the attack. Results at the fixed threshold 0.5 are also reported. Calibration is summarised by the Brier score and the 15-bin expected calibration error.

3.5 Threat models

The adversary wants a malicious record that the detector flags to be classified as benign. We evaluate three threat models in the normalised input space $z \in [0, 1]^p$, ordered from least to most restrictive.

Unconstrained. Every input may change within an ℓ_∞ ball inside the unit box, $\mathcal{C}^U(z, \epsilon) = \{z' : \|z' - z\|_\infty \leq \epsilon, 0 \leq z' \leq 1\}$. It allows fractional values in one-hot columns and serves as an upper bound and for comparability with feature-space attacks in the literature.

Numeric-only. All numeric fields may change in both directions within the same ball; the one-hot categorical blocks are frozen.

Domain-constrained. Only the mutable set $\mathcal{M}_d$ may change, and only upwards:

$$\mathcal{C}_d^C(z, \epsilon) = \{z' : 0 \leq z'_j - z_j \leq \epsilon \ \forall j \in \mathcal{M}_d, \ z'_j = z_j \ \forall j \notin \mathcal{M}_d\} \cap [0, 1]^p. \quad (3)$$

Because the scaling is monotone, “upwards” in z means upwards in raw units. After the search, the record is mapped to raw units and *repaired*: integer-valued fields are rounded down, but never below their original value. A candidate protocol relation is enforced, by raising the dependent field, only if it holds in at least 99.5% of clean training records. The ToN-IoT header-overhead relation of Table 2 holds in only 98.8% (source direction) and 98.2% (destination direction) of training flows, so it is not enforced, and constrained records can violate it; the validity audit does not count such violations. No relation is defined for Edge-IIoTset. Success is always evaluated on the repaired record. Because a padding adversary is limited by cost rather than by a small norm, the constrained model is also evaluated at $\epsilon \in \{0.1, 0.25, 0.5, 1\}$, where $\epsilon = 1$ allows each mutable field to be raised up to its training maximum.

Table 3 justifies each mutable feature. A feature is mutable if the sender of the flow or packet can increase it without the victim’s cooperation and without removing the bytes that carry the attack. All other fields are frozen, including fields that the attacker could change but whose change would alter the attack itself (for example the HTTP method or the destination service) and fields set by the victim (responder bytes, response codes). Raising a mutable field to at most its training maximum keeps the padded record inside the range of values seen in training. Table 3 also lists side effects of padding on other fields that the constraints do not model. The constrained model is therefore a set of necessary conditions for realisability, not a sufficient one (Section 5.2).

3.6 Evasion attacks and robustness measures

PGD [56] minimises the malicious logit,

$$z^{(t+1)} = \Pi_C \left[z^{(t)} - \alpha \operatorname{sign} \left(\nabla_z \operatorname{logit} \hat{p}_\theta(z^{(t)}) \right) \right], \quad (4)$$

Table 3 Mutable features of the constrained threat model (increase only). “Mechanism” is how the sending side can increase the field; “side effects” are changes that padding causes in other retained fields, none of which the constraints enforce (Section 3.5).

Feature	Mechanism	Side effects on other features
<i>ToN-IoT (one record per flow; the attacker is assumed to be the originator)</i>		
duration	delay packets or keep the connection open longer	none in the retained feature set, which has no rate features; idle timeouts of the flow monitor not modelled
source bytes	pad the application payload (filler headers or body content)	source IP bytes should rise; HTTP body length if the padding is in the body
source packets	split the payload over more packets or add empty segments	source IP bytes should rise by at least 20 bytes per packet
source IP bytes	rises with payload and packets, or with IP options	should stay at least source bytes plus 20 bytes per packet
HTTP request body length	append unused content to the HTTP request body	source bytes should rise by the same amount
<i>Edge-IIoTset (one record per packet; the attacker is assumed to be the sender)</i>		
TCP segment length	pad the TCP payload of the packet	TCP checksum changes (frozen in the model); sequence numbers of later packets (other records)
HTTP content length	declare and send a longer HTTP body	TCP segment length should rise
MQTT message length	pad the MQTT payload	TCP segment length should rise

with ten steps, $\alpha = 2.5\epsilon/10$ and a random start in $\mathcal{C}$; FGSM [54] is the one-step version. Both require gradients and are applied to the MLPs. A score-based black-box random search in the style of Square Attack [57] is applied to all four detectors. It starts at a random vertex of $\mathcal{C}$, repeatedly resamples a shrinking random subset of mutable coordinates, keeps a proposal if it lowers $\hat{p}(y=1)$, and stops at the first success or after 300 queries; in the constrained model every proposal is repaired before it is scored. PGD records crafted on the clean MLP of a seed are also transferred to the other detectors of the same seed.

Because random search may underestimate the vulnerability of tree ensembles, XGBoost is additionally attacked with (i) the same random search with 3,000 queries and no early stopping, and (ii) a tree-aware coordinate search. For one feature at a time, the coordinate search evaluates every split threshold that the trained ensemble uses for that feature and that lies inside the feasible interval, keeps the value that lowers the malicious score most, and cycles over all mutable features for up to three passes. For a single feature this is an exact line search over the ensemble’s decision regions; in the constrained model each candidate is repaired before it is scored. The same coordinate search over a 33-point grid is applied to the MLPs under the constrained model as a gradient-free check of PGD. For each record, the strongest of the attacks run (lowest malicious score) is also reported, and no attack keeps a change that raises the malicious score.

Outcomes. For detector c , seed s , attack a and budget ϵ , the attack success rate is

$$\text{ASR} = \frac{|\{i \in \mathcal{A} : \hat{y}_i^{\text{clean}} = 1, \hat{y}_i^{\text{adv}} = 0\}|}{|\{i \in \mathcal{A} : \hat{y}_i^{\text{clean}} = 1\}|}, \quad (5)$$

where $\mathcal{A}$ is the 2,000-record attack subset. Because the eligible set in the denominator shrinks at strict thresholds, detectors are compared at fixed false-positive budgets by *detection under attack*, the share of all records in $\mathcal{A}$ still flagged after the attack. A per-record validity audit checks one-hot integrity, integrality of changed integer fields, the protocol relations, immutability and direction; a record is *syntactically valid* if it passes the first three checks and *threat-model valid* if it passes all five.

Projection onto the valid space. Unconstrained and numeric-only records fail the syntactic checks largely by construction. To separate this from evasion that survives in syntactically valid records, each such record is also projected onto the nearest syntactically valid record and re-scored. Each one-hot block is set to its nearest valid pattern (the level whose value exceeds 0.5, otherwise the all-zero pattern), each changed integer field is rounded to the nearest integer within its training range, and enforced relations are restored (none is enforced in our data; see Section 3.5). Projected records keep all other changes, so they remain outside the domain-constrained threat model.

3.7 Explanation methods

SHAP [33] approximates additive contributions to the malicious-class probability $f_1(z) = \hat{p}(y=1 \mid z)$, $f_1(z) \approx \phi_0 + \sum_{j=1}^p \phi_j(z)$. For the MLPs we use Deep SHAP, an estimator based on DeepLIFT [58]; for XGBoost we use interventional Tree SHAP on the probability output [59]. Both use the same 100 randomly drawn training records as background. LIME [34] fits a weighted linear surrogate to 1,000 perturbations sampled around the instance, with an exponential kernel of width $0.75\sqrt{p}$ (the library default); continuous features are not discretised, and one-hot columns are treated as categorical. Each LIME call uses a fixed random seed determined by the training seed and the record set. All attributions are signed toward the malicious class and aggregated to feature groups. Explanations are computed on the 600 clean records and on the attacked versions of the 300 malicious records: for the MLPs, PGD records under the constrained model at $\epsilon \in \{0.01, 0.03, 0.05, 1\}$ and under the numeric-only and unconstrained models at $\epsilon = 0.05$; for XGBoost, the corresponding black-box records at $\epsilon = 0.05$ and at maximum padding.

Explainer-configuration analysis. Deep SHAP and Tree SHAP are different estimators, and LIME’s default configuration may disadvantage it. On a subset of 60 benign and 60 malicious clean records (five seeds, all detectors), we therefore also compute KernelSHAP [33] with 50 background records and 2,048 coalition samples, which applies one estimator to both model families, and LIME with 1,000 or 5,000 samples and kernel widths of 0.25, 0.75 and $2\sqrt{p}$, as well as LIME with quartile discretisation. Each configuration is run twice, with different random seeds or background sets, to measure its call-to-call stability.

3.8 Faithfulness metrics

Faithfulness is measured for the class the detector predicts. Let $c_i = \mathbb{1}\{f_1(z_i) \geq \tau\}$, and let $f_c(z) = f_1(z)$ if $c = 1$ and $1 - f_1(z)$ otherwise. Groups are ranked by their attribution *in favour of* c_i (ϕ for $c_i = 1$ and $-\phi$ for $c_i = 0$). Let $\mathcal{G}$ be the set of all G groups and $T_k(z) \subset \mathcal{G}$ the k highest-ranked groups. With a masking operator $m_r(z, S)$ that replaces the groups in S according to reference r , comprehensiveness masks the top-ranked groups and sufficiency masks all *other* groups:

$$\text{Comp}_k^r(z) = f_c(z) - \mathbb{E} f_c(m_r(z, T_k)), \quad (6)$$

$$\text{Suff}_k^r(z) = f_c(z) - \mathbb{E} f_c(m_r(z, \mathcal{G} \setminus T_k)), \quad (7)$$

following DeYoung et al. [42]. Higher comprehensiveness and lower sufficiency indicate a more faithful ranking. Each is summarised as an AOPC [41], the mean over $k \in \{10, 20, 30, 40, 50\}\%$ of the G groups (at least one group and at most $G - 1$). Three references are compared:

1. *median*: numeric groups are set to the background median and categorical groups to the background’s modal level;
2. *marginal*: masked groups are replaced jointly by their values in 16 random background records and the predictions are averaged;
3. *k-NN in-fill*: the 8 training records (from a pool of 5,000) closest to z on the *unmasked* columns provide the masked values, which approximates a draw from the conditional distribution [10, 43].

Every score is also computed for random group orderings (three draws) under the same reference. The faithfulness outcome is $\Delta\text{Comp} = \text{AOPC}^{\text{expl}} - \text{AOPC}^{\text{rand}}$, which removes the part of a score caused by the masking itself. For contrast, we also compute a common variant that scores every record by the malicious output, ranks groups by $|\phi|$ and uses the median reference.

The marginal reference draws from the same training records as the SHAP background. Interventional SHAP estimates the effect of replacing features by draws from this marginal distribution, so a test with the marginal reference asks the question SHAP is built to answer and may favour it. The k -NN reference does not share this construction, and the explainer-configuration analysis tests whether the SHAP–LIME ordering depends on the estimator or on LIME’s settings.

3.9 Stability metrics

For two attribution vectors $a, b \in \mathbb{R}^G$ we compute the top- k Jaccard index on $|a|, |b|$,

$$J_k(a, b) = \frac{|T_k^a \cap T_k^b|}{|T_k^a \cup T_k^b|}, \quad k \in \{3, 5, \lfloor 0.2G + 0.5 \rfloor\}, \quad (8)$$

together with the Spearman rank correlation of $|a|$ and $|b|$, the cosine similarity of the signed vectors, and agreement of the top-ranked group. Small k and rank-based

measures avoid ceiling effects when G is small, and the signed cosine detects sign reversals that set overlap cannot. Four stability axes are separated:

1. *input*: the clean record versus its attacked version, same model and explainer settings;
2. *explainer call*: five LIME calls with different sampling seeds on the same record and model;
3. *background*: SHAP with three independently drawn background sets;
4. *training seed*: the same clean record explained under each pair of the 20 seeds (190 pairs).

Axes 2 and 3 use the first five seeds and 100 malicious records. Axes 2–4 give the level of instability that exists without any attack, against which attack-induced instability is judged.

3.10 Sensitivity analyses

Four data variants are run with five seeds each, all detectors and attacks, and a reduced set of explanation views. (i) Two further stratified partitions (seeds 2027 and 2028) estimate partition-level variation. (ii) *Label conflicts retained*: the primary partition seed is used, but one copy of each conflicting feature vector is kept with each of its labels. (iii) A *global temporal* partition sorts records by capture time and assigns the earliest 70% to training, the next 15% to validation and the latest 15% to testing. (iv) A *within-family temporal* partition applies the same chronological cut within each attack family and within the benign class. Variant (iii) tests extrapolation in time but can leave families absent from training; variant (iv) keeps every family in each partition and isolates drift within families. Edge-IIoTset records carry a frame time; the ToN-IoT train-test table has no time-stamp column, so for ToN-IoT the row order of the released file is used instead, which yields contiguous-block rather than chronological partitions. The logs record how the time stamps were parsed. As Section 4.6 shows, variant (iii) places no benign records in the test partition of either dataset and is therefore not interpreted. The supplementary material reports a multiclass attack-family analysis, the effect of individual preprocessing choices, an adversarial-training sweep and resource profiling.

3.11 Statistical analysis

For each seed, a metric is first averaged over the records of the relevant subset, which gives one value $m^{(s)}$ per seed and cell. A comparison between two conditions uses the seed-paired differences $\Delta^{(s)} = m_A^{(s)} - m_B^{(s)}$, $s = 1, \dots, 20$. For each comparison we report the mean difference, a 95% percentile bootstrap interval over seeds (10,000 resamples), Cohen’s $d_z = \bar{\Delta}/\text{sd}(\Delta)$, the number of seeds with a positive difference, and the two-sided exact Wilcoxon signed-rank p -value. With 20 seeds the smallest attainable exact p -value is $2^{-19} \approx 1.9 \times 10^{-6}$, and a paired t -test with $\alpha = 0.05$ and power 0.8 detects $d_z \geq 0.66$.

Holm’s procedure [60] is applied within pre-declared families. For the explainer comparisons (SHAP minus LIME), one family is formed per dataset, detector and

outcome, and its members are the explanation views. For P2 these are the clean view and the attacked views: seven per MLP (clean; constrained at $\epsilon \in \{0.01, 0.03, 0.05, 1\}$; numeric-only and unconstrained at $\epsilon = 0.05$) and five for XGBoost (clean; constrained at $\epsilon = 0.05$ and 1; numeric-only; unconstrained). Over two datasets, three MLPs and XGBoost this gives eight families with $2 \times (3 \times 7 + 5) = 52$ comparisons. P3 is defined only for attacked views, so the same eight families contain $2 \times (3 \times 6 + 4) = 44$ comparisons. For the training-condition contrasts, one family is formed per dataset, threat model and attack, containing each detector’s contrast with the clean MLP; the threat-model contrasts (constrained versus unconstrained, constrained versus numeric-only, numeric-only versus unconstrained, for every detector) form one family per dataset. Holm-adjusted $p < 0.05$ is called *confirmatory*; all other comparisons are *descriptive*. Secondary analyses are summarised by seed means, standard deviations and bootstrap intervals without multiplicity control. Records within a seed are never treated as independent replicates.

3.12 Reproducibility

All experiments ran in one fixed software environment on Kaggle (NVIDIA Tesla T4 GPU, four virtual CPUs). For every combination of dataset, partition and seed, the predictive, attack and explanation results are stored and also written to the execution log. Every table and figure is computed from the seed-level results by the released analysis code. The code, seed-level results, logs and analysis outputs are publicly archived (see Declarations).

4 Results

Unless stated otherwise, values are means $\pm$ standard deviations over the 20 training seeds on the primary partition, at $\epsilon = 0.05$. “Confirmatory” refers to Holm-adjusted $p < 0.05$ in the pre-declared families (Section 3.11).

4.1 Data audit and clean detection

Table 4 summarises the audit. The ToN-IoT network file contains 211,043 rows, of which 117,373 are exact duplicates on the loaded columns. After removing 110,472 duplicate feature–label vectors and 7,122 rows whose feature vector occurs with both labels, 93,449 records (44% of the file) remain. Edge-IIoTset has fewer duplicates (2,704 removed) but many more label conflicts: 17,982 rows (11%) share their model-visible feature vector with a record of the other class, leaving 137,114 records. No raw record appears in more than one partition. Because of clipping and top-20 level bucketing, 22 transformed ToN-IoT vectors and 1 Edge-IIoTset vector are shared between training and test; these are distinct raw records.

Table 5 reports clean-test performance. On ToN-IoT, the three MLPs reach macro-F1 0.969–0.970 and XGBoost 0.995. On Edge-IIoTset, the four detectors reach 0.949–0.953, with ROC-AUC 0.991–0.993. Adversarial training does not reduce clean performance: the paired differences from the clean MLP are at most 0.002 in absolute value and none is significant (Holm $p \geq 0.52$). XGBoost outperforms the MLP on both

datasets (ToN-IoT +0.026, $d_z = 17.8$; Edge-IIoTset +0.003, $d_z = 0.43$; both confirmatory). On Edge-IIoTset, the validation-tuned threshold flags 14.3–14.5% of benign test flows while detecting 99.6–99.9% of attacks; at threshold 0.5, the false-positive rate is 4.4–4.6% and malicious recall 93.4–94.1%. All four detectors misclassify essentially the same benign records (benign recall at threshold 0.5 is 0.954–0.956 for every detector and seed), so we attribute the residual false alarms to benign records that are not separable from attack traffic in the released features. The effect of individual preprocessing choices on these scores is reported in Supplementary Section S2.

Table 4 Dataset audit and primary partition (seed 2026). Exact dup. = rows identical on all loaded columns; dup. removed = repeated (feature vector, label) pairs; conflicts = rows whose feature vector occurs with both labels (removed in the primary analysis). Cols./groups = model columns after one-hot encoding / original fields.

Dataset	Raw rows	Exact dup.	Dup. removed	Conflicts	Retained	Positive rate	Cols./groups	Train / val. / test
ToN-IoT	211,043	117,373	110,472	7,122	93,449	0.747	88 / 31	65,414 / 14,017 / 14,018
Edge-IIoTset	157,800	19,330	2,704	17,982	137,114	0.828	53 / 34	95,979 / 20,567 / 20,568

Table 5 Clean-test performance (20 seeds; means, with SD where shown). Standard deviations are shown for macro-F1 and FPR; all other SDs are ≤ 0.036 . All columns use the validation-tuned threshold except those marked @0.5. Mal. rec. = malicious recall; Ben. prec. = benign precision; FPR = false-positive rate; ECE = expected calibration error (15 bins).

Dataset	Detector	Macro-F1	F1 @0.5	Mal. rec.	Ben. prec.	FPR	FPR @0.5	ROC-AUC	ECE
ToN-IoT	MLP	0.970 $\pm$ 0.001	0.968	0.989	0.967	0.057 $\pm$ 0.019	0.042	0.991	0.026
	MLP+FGSM-AT	0.970 $\pm$ 0.001	0.967	0.994	0.982	0.072 $\pm$ 0.002	0.046	0.992	0.025
	MLP+PGD-AT	0.969 $\pm$ 0.001	0.967	0.993	0.979	0.071 $\pm$ 0.005	0.047	0.991	0.027
	XGBoost	0.995 $\pm$ 0.000	0.996	0.998	0.995	0.009 $\pm$ 0.001	0.006	1.000	0.002
Edge-IIoTset	MLP	0.949 $\pm$ 0.008	0.903	0.996	0.982	0.143 $\pm$ 0.010	0.044	0.992	0.047
	MLP+FGSM-AT	0.951 $\pm$ 0.001	0.903	0.998	0.990	0.145 $\pm$ 0.000	0.044	0.992	0.048
	MLP+PGD-AT	0.950 $\pm$ 0.000	0.901	0.998	0.988	0.145 $\pm$ 0.000	0.044	0.991	0.056
	XGBoost	0.953 $\pm$ 0.000	0.909	0.999	0.996	0.145 $\pm$ 0.000	0.046	0.993	0.041

4.2 Evasion under the three threat models

Figure 2 and Table 6 report attack success at the validation-tuned threshold (outcome P1).

Validity and projection. No successful unconstrained or numeric-only adversarial record passes the validity audit, for any detector on either dataset. Unconstrained records break one-hot integrity; numeric-only records leave byte and packet counts fractional. This failure is partly by construction, so Table 6 also reports success after projecting each record onto the nearest syntactically valid record. For the clean MLP, projection removes most of the success: unconstrained success falls from 0.31 to 0.06 on ToN-IoT and from 0.30 to 0.04 on Edge-IIoTset, and the adversarially trained MLPs stay at or below 0.07. For XGBoost, a substantial part survives: on ToN-IoT,

unconstrained success falls from 1.00 to 0.60 and numeric-only success from 0.73 to 0.65; on Edge-IIoTset, from 0.14 to 0.13. Fractional one-hot and integer values therefore explain most of the feature-space success against the MLPs, but not against the tree ensemble, whose syntactically valid but unconstrained records (which still change immutable fields) remain effective.

Constraints remove most of the attack surface at small budgets. At $\epsilon = 0.05$, attack success on the clean MLP falls from 0.307 ± 0.308 (unconstrained) to 0.095 ± 0.155 (numeric-only) and 0.022 ± 0.032 (constrained) on ToN-IoT, and from 0.305 ± 0.036 to 0.238 ± 0.036 and 0.000 on Edge-IIoTset. All constrained-versus-unconstrained contrasts are confirmatory (d_z between -0.48 and -166 , Holm $p \leq 3.4 \times 10^{-4}$). XGBoost is evaded on 100% of ToN-IoT attacks under the unconstrained model and on $73.1 \pm 3.9\%$ under the numeric-only model, but on only $0.8 \pm 0.6\%$ under the constrained model with random search (see below for stronger attacks). Part of its feature-space vulnerability is an encoding effect: its histogram splits on one-hot columns test against the value 1, so lowering an active indicator from 1 to $1 - \epsilon$ removes the categorical evidence altogether, and small fractional changes to integer-valued fields cross split points that no valid record can reach. PGD records transferred from the clean MLP inherit the same effect (0.870 unconstrained, 0.173 numeric-only, 0.004 constrained on ToN-IoT). The score-based black-box attack is almost as effective as PGD against the clean MLP (0.285 versus 0.307 on ToN-IoT, 0.301 versus 0.305 on Edge-IIoTset), so the MLP results do not depend on gradient access.

Stronger attacks on the tree ensemble. Random search does not measure the constrained robustness of XGBoost reliably. With 3,000 queries instead of 300, its success is unchanged (ToN-IoT: 0.008 at $\epsilon = 0.05$ and 0.023 at maximum padding). The tree-aware coordinate search, in contrast, evades XGBoost on 11.8% of ToN-IoT attacks at $\epsilon = 0.05$ and on 61.3% at maximum padding, against 2.3% for random search. On Edge-IIoTset, all three attacks leave XGBoost essentially unevaded (at most 0.3%). For the MLPs, the gradient-free coordinate search reaches the same success as PGD within 0.05 (for example 0.403 against 0.387 for the clean ToN-IoT MLP at maximum padding), which confirms that PGD is not weakened by gradient masking. Table 6 reports, for each record, the strongest of the attacks run.

Padding. Larger padding budgets evade the MLP (Figure 3). At maximum padding, attack success reaches 0.387 ± 0.339 on ToN-IoT and 0.335 ± 0.097 on Edge-IIoTset. Adversarial training lowers it to 0.07–0.11 (ToN-IoT) and 0.08–0.10 (Edge-IIoTset). The attack changes on average only 1.0–3.4 fields, all of them attacker-controllable.

4.3 Robustness at matched false-positive rates and without a threshold

Validation-tuned thresholds differ between detectors and seeds (on ToN-IoT, from about 0.03 for the adversarially trained MLPs to 0.35 for XGBoost), so attack success at these thresholds mixes robustness with the false-alarm budget each threshold implies. Figure 4 and Table 7 compare all 20 seeds at thresholds matched to the same validation false-positive rate, using the strongest attack per record. At the 1% budget, the test false-positive rate is 1.1–1.6% on ToN-IoT and 0.8–1.3% on Edge-IIoTset, and the detectors find 87–89% of clean attacks (ToN-IoT XGBoost: 99.9%).

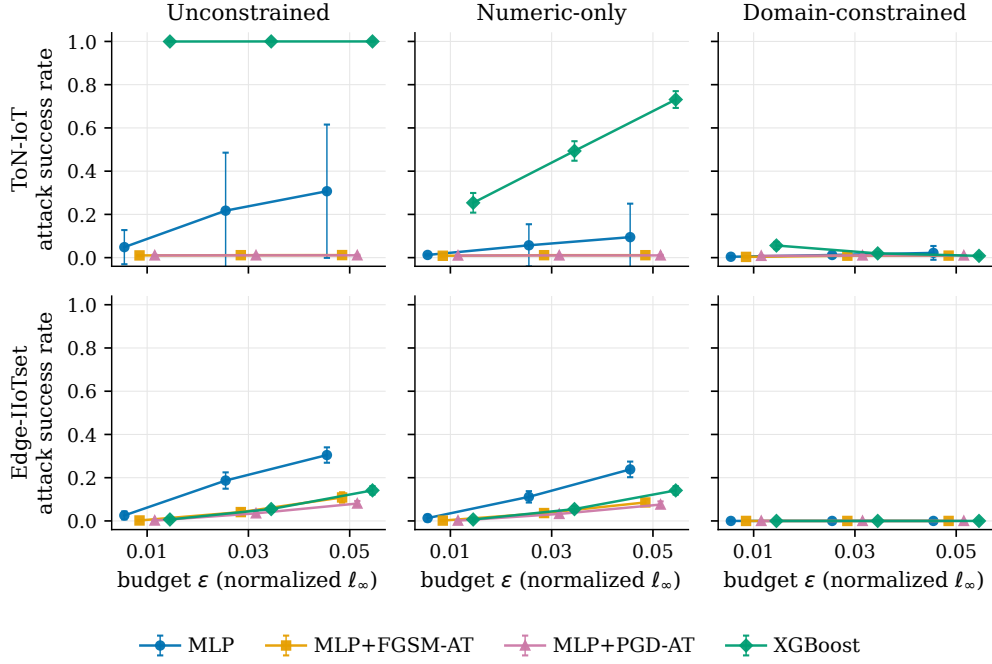


Fig. 2 Attack success rate versus budget under the three threat models at the validation-tuned threshold (PGD for the MLPs, score-based black-box attack for XGBoost). Points are 20-seed means and error bars are seed standard deviations; points are offset horizontally so that overlapping series remain visible.

Table 6 Attack success rate at the validation-tuned threshold (20 seeds, means). Unconstrained and numeric-only ($\epsilon = 0.05$): raw / after projection onto the nearest syntactically valid record. Constrained ($\epsilon = 0.05$ and maximum padding $\epsilon = 1$): main attack / strongest attack per record. The main attack is PGD for the MLPs and 300-query random search for XGBoost; the strongest attack also includes 3,000-query random search and the coordinate search. No successful raw unconstrained or numeric-only record is syntactically valid.

Dataset	Detector	Raw / projected		Main / strongest attack	
		Unconstrained	Numeric-only	Constrained	Constrained, max. padding
ToN-IoT	MLP	0.31 / 0.06	0.09 / 0.07	0.022 / 0.022	0.387 / 0.414
	MLP+FGSM-AT	0.01 / 0.01	0.01 / 0.01	0.009 / 0.009	0.066 / 0.071
	MLP+PGD-AT	0.01 / 0.01	0.01 / 0.01	0.010 / 0.010	0.112 / 0.122
	XGBoost	1.00 / 0.60	0.73 / 0.65	0.008 / 0.118	0.023 / 0.613
Edge-IIoTset	MLP	0.30 / 0.04	0.24 / 0.04	0.000 / 0.000	0.335 / 0.354
	MLP+FGSM-AT	0.11 / 0.06	0.09 / 0.06	0.000 / 0.000	0.098 / 0.143
	MLP+PGD-AT	0.08 / 0.07	0.08 / 0.07	0.000 / 0.000	0.080 / 0.119
	XGBoost	0.14 / 0.13	0.14 / 0.13	0.000 / 0.000	0.002 / 0.003

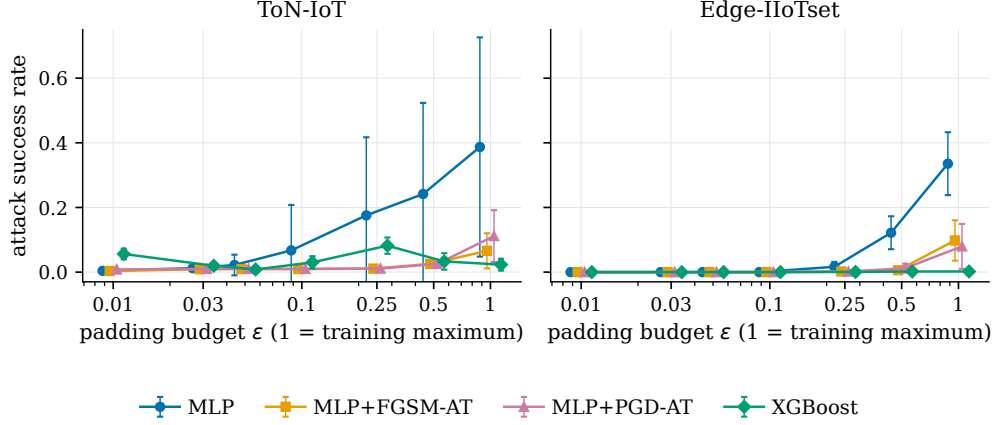


Fig. 3 Attack success under the domain-constrained threat model as a function of the padding budget (log scale; $\epsilon = 1$ allows each mutable field up to its training maximum). Points are 20-seed means and error bars are seed standard deviations.

On ToN-IoT, the operating point reverses the apparent benefit of adversarial training. At their tuned thresholds the adversarially trained MLPs still detect 87–92% of maximally padded attacks, but these thresholds imply a false-positive rate of about 7%. At the 1% budget, maximum padding leaves every MLP detecting at most 3% of attacks. XGBoost detects 65% at this budget, but only 15% at 0.5% and 3% at 0.1%, so its advantage also depends on the alert budget. Against the unconstrained attack, adversarial training roughly doubles detection at 1% (0.41 against 0.23), while XGBoost detects nothing. On Edge-IIoTset, detection under attack is nearly flat across budgets. Adversarial training keeps its benefit against padding at matched false-positive rates (0.73–0.75 against 0.54 for the clean MLP at 1%), and XGBoost detects 88%.

Threshold-free robustness and seed dependence. On ToN-IoT, the clean MLP’s attack success at the tuned threshold is bimodal across seeds: seven of the 20 seeds are evaded on 64–72% of attacks under the unconstrained model and the other 13 on 4–9%, although clean macro-F1 does not separate the two groups (0.971 versus 0.969). The vulnerable seeds settled on higher tuned thresholds (mean 0.43 against 0.037), so their median malicious record lies closer to the threshold (4.4 against 6.6 in log-odds). Two observations show that the threshold explains much, but not all, of the bimodality. First, at a common 1% false-positive budget, unconstrained attack success no longer separates the groups (0.70 against 0.76). Second, the threshold-free measures still differ: the attack lowers the malicious log-odds by 6.6 on average for the vulnerable seeds against 3.1 for the others, and the ROC-AUC between benign and attacked malicious records is 0.81 against 0.93. The vulnerable training realisations therefore respond more strongly to the attack, and their higher thresholds turn this into a large difference in attack success at the tuned operating point. Edge-IIoTset shows no such bimodality (unconstrained success 0.24–0.41 over seeds).

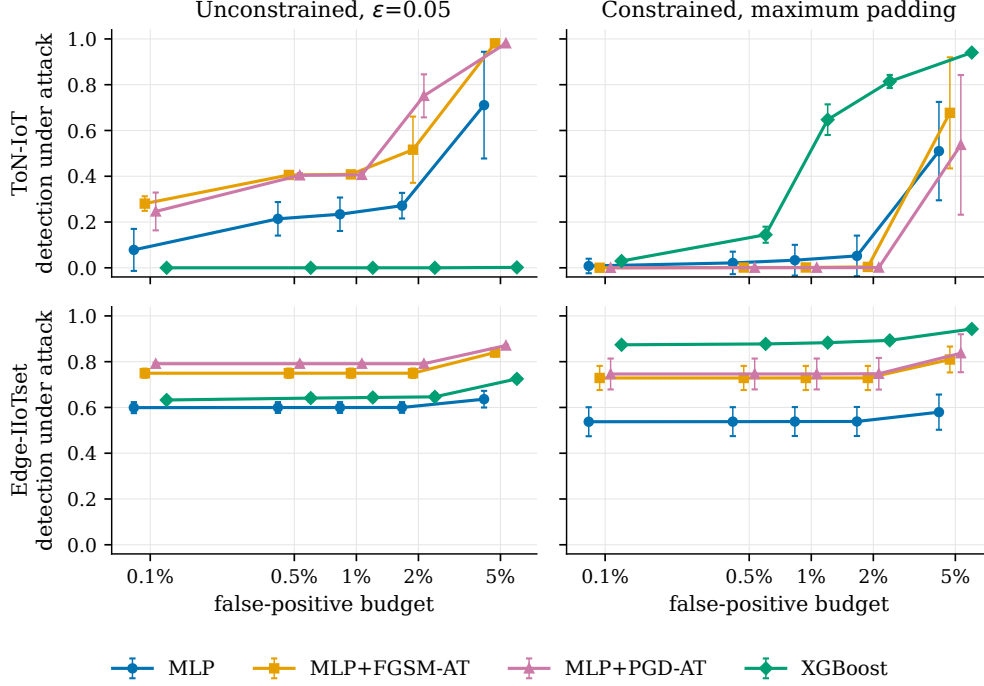


Fig. 4 Detection under attack (share of all records of the 2,000-record attack subset still flagged after the attack) at thresholds matched to a validation false-positive rate (20 seeds; mean $\pm$ SD; strongest attack per record). Left: unconstrained, $\epsilon = 0.05$; right: domain-constrained maximum padding.

4.4 Explanation faithfulness

Table 9 and Figure 5 report predicted-class comprehensiveness relative to the random-ranking control (ΔComp) and the corresponding sufficiency difference (ΔSuff ; more negative is better) on clean records.

Evaluation target. When every record is scored by the malicious output and groups are ranked by $|\phi|$ (with the median reference), 39–76% of records receive a negative comprehensiveness score, depending on dataset, detector and explainer. Under the predicted-class definition with the marginal reference, the share is 0–16% (at most 1.6% on ToN-IoT). Negative scores of the first kind arise because deleting features that argue *against* the malicious class raises the malicious score; they do not show that the explainers are unfaithful.

SHAP versus LIME. SHAP identifies decision-relevant fields better than LIME for every detector, dataset, view and reference. For P2 (marginal reference), all 52 SHAP–LIME comparisons are confirmatory in favour of SHAP: 20 of 20 seeds positive, d_z from 1.8 to 58, Holm $p \leq 1.3 \times 10^{-5}$. The gap is small for the ToN-IoT MLPs (ΔComp 0.37–0.41 against 0.31–0.32) and large elsewhere. On Edge-IIoTset, LIME’s ranking is barely better than random for every detector (ΔComp 0.03–0.06 against 0.28–0.34 for SHAP), and for XGBoost on attacked Edge-IIoTset records it is worse

Table 7 Detection under attack at matched false-positive budgets, and threshold-free ROC-AUC (20 seeds, means; strongest attack per record). Clean = malicious recall without attack at the 1% budget. Unc. = unconstrained, $\epsilon = 0.05$ (raw / projected); pad. = constrained maximum padding at the 1% budget and at the tuned threshold. AUC = ROC-AUC between benign test records and clean / maximally padded malicious records.

Dataset	Detector	Clean	Unc., 1% FPR		Pad.		AUC clean / pad.
			raw	proj.	1% FPR	tuned	
ToN-IoT	MLP	0.88	0.23	0.50	0.03	0.58	0.991 / 0.856
	MLP+FGSM-AT	0.88	0.41	0.46	0.00	0.92	0.991 / 0.917
	MLP+PGD-AT	0.88	0.41	0.49	0.00	0.87	0.991 / 0.899
	XGBoost	1.00	0.00	0.51	0.65	0.39	1.000 / 0.960
Edge-IIoTset	MLP	0.88	0.60	0.87	0.54	0.64	0.991 / 0.813
	MLP+FGSM-AT	0.88	0.75	0.82	0.73	0.85	0.991 / 0.948
	MLP+PGD-AT	0.87	0.79	0.82	0.75	0.88	0.991 / 0.956
	XGBoost	0.89	0.64	0.78	0.88	1.00	0.993 / 0.991

than random (-0.01 to -0.04 ; the only 4 of 104 explainer-versus-random comparisons with a negative sign).

Reference choice. The reference changes the magnitude of ΔComp and, for LIME, its interpretation. Under k -NN in-fill, LIME’s advantage over random almost vanishes on every detector ($\Delta\text{Comp} \leq 0.007$), while SHAP retains 0.08–0.28. For the ToN-IoT MLPs, the median reference makes LIME look nearly as faithful as SHAP (0.33 versus 0.36); under k -NN in-fill the ratio is more than tenfold. A single-reference evaluation can therefore misrank explainers.

Explainer configuration. Table 8 reports the configuration analysis (five seeds, 120 clean records). KernelSHAP, a single estimator for both model families, is as faithful as the model-specific estimators under the marginal reference (ΔComp 0.33–0.39 against 0.27–0.40) and comparably faithful under k -NN in-fill (0.14–0.25 against 0.06–0.26). LIME’s sample size and kernel width change its ΔComp by at most 0.05 under any reference; 5,000 samples mainly make it more repeatable (call-to-call ρ of 0.30–0.84 with 1,000 samples and 0.52–0.95 with 5,000). Quartile discretisation is LIME’s best configuration under the marginal reference (0.13–0.34), but under k -NN in-fill no LIME configuration exceeds 0.09. In all 8 combinations of dataset and detector and under all three references, no LIME configuration reaches the faithfulness of either SHAP estimator. The SHAP–LIME ordering therefore does not depend on LIME’s settings, on the SHAP estimator, or on the shared construction of the marginal reference and the SHAP background. KernelSHAP is, however, much slower (60–75 ms per record for the MLPs and 220–440 ms for XGBoost, against 6–19 ms for Deep and Tree SHAP).

Under attack. SHAP explanations of attacked records remain faithful to the model’s behaviour on those records (ΔComp 0.21–0.53 across all attacked views), whereas LIME’s faithfulness falls under the unconstrained attack on ToN-IoT (0.05–0.10).

Table 8 Explainer-configuration analysis (ΔComp on 60 benign and 60 malicious clean records, 5 seeds, means). SHAP = Deep SHAP (MLP) or Tree SHAP (XGBoost) as in the main analysis; LIME default = 1,000 samples, kernel width $0.75\sqrt{p}$, no discretisation; LIME best = the configuration with the highest ΔComp under k -NN in-fill (last column; disc. = 1,000 samples with quartile discretisation; 5k = 5,000 samples, kernel width $2\sqrt{p}$).

Dataset	Detector	SHAP		KernelSHAP		LIME default		LIME best	
		marg.	k NN	marg.	k NN	marg.	k NN	marg.	k NN
ToN-IoT	MLP	0.402	0.089	0.380	0.143	0.314	0.001	0.339	0.056
	MLP+FGSM-AT	0.378	0.064	0.381	0.146	0.307	0.001	0.324	0.014
	MLP+PGD-AT	0.365	0.113	0.371	0.142	0.308	0.004	0.331	0.023
	XGBoost	0.384	0.243	0.388	0.222	0.074	0.003	0.077	0.006
Edge-IIoTset	MLP	0.292	0.174	0.331	0.232	0.021	-0.007	0.132	0.016
	MLP+FGSM-AT	0.267	0.217	0.332	0.238	0.037	-0.010	0.143	0.015
	MLP+PGD-AT	0.285	0.250	0.341	0.234	0.005	-0.010	0.161	0.016
	XGBoost	0.331	0.261	0.351	0.246	0.028	-0.005	0.225	0.083

Table 9 Faithfulness on clean records (20-seed means). ΔC = comprehensiveness AOPC minus the random-ranking control (higher is better) under the median, marginal and k -NN in-fill references; ΔS = sufficiency AOPC minus control under the marginal reference (more negative is better). Seed standard deviations are ≤ 0.045 in every cell.

Dataset	Detector	SHAP				LIME			
		ΔC med.	ΔC marg.	ΔC k NN	ΔS marg.	ΔC med.	ΔC marg.	ΔC k NN	ΔS marg.
ToN-IoT	MLP	0.361	0.409	0.104	-0.317	0.332	0.319	0.004	-0.236
	MLP+FGSM-AT	0.344	0.377	0.084	-0.312	0.315	0.310	0.005	-0.269
	MLP+PGD-AT	0.340	0.370	0.123	-0.316	0.319	0.315	0.007	-0.274
	XGBoost	0.383	0.389	0.264	-0.359	0.052	0.066	0.002	-0.058
Edge-IIoTset	MLP	0.383	0.310	0.191	-0.231	0.149	0.029	0.003	-0.051
	MLP+FGSM-AT	0.368	0.281	0.223	-0.238	0.174	0.063	0.005	-0.083
	MLP+PGD-AT	0.336	0.295	0.252	-0.264	0.128	0.027	0.003	-0.073
	XGBoost	0.277	0.338	0.278	-0.326	0.107	0.032	0.007	-0.033

4.5 Explanation stability

Table 10 and Figure 6 compare four sources of attribution change, measured by the Spearman correlation of absolute group attributions.

Variability without attack. LIME rankings already vary substantially between calls on the same record and model ($\rho = 0.51$ – 0.82) and between retrained models (0.50 – 0.80). SHAP rankings are far more reproducible: $\rho = 0.89$ – 0.94 between background draws and 0.92 – 0.99 between seeds. The top-3 sets agree across seeds with Jaccard 0.64 – 0.91 for SHAP and 0.31 – 0.54 for LIME.

Change under realisable perturbations. Under the constrained attacks with $\epsilon \leq 0.05$, LIME’s clean-to-attacked agreement is indistinguishable from its own call-to-call agreement (for example 0.83 against 0.82 for the ToN-IoT MLP, and 0.51 against 0.51 for Edge-IIoTset XGBoost), so LIME’s apparent instability there is sampling noise; the same holds under the numeric-only attack. At maximum padding, LIME’s agreement falls below its call-to-call level by 0.07 – 0.09 on ToN-IoT and by at most

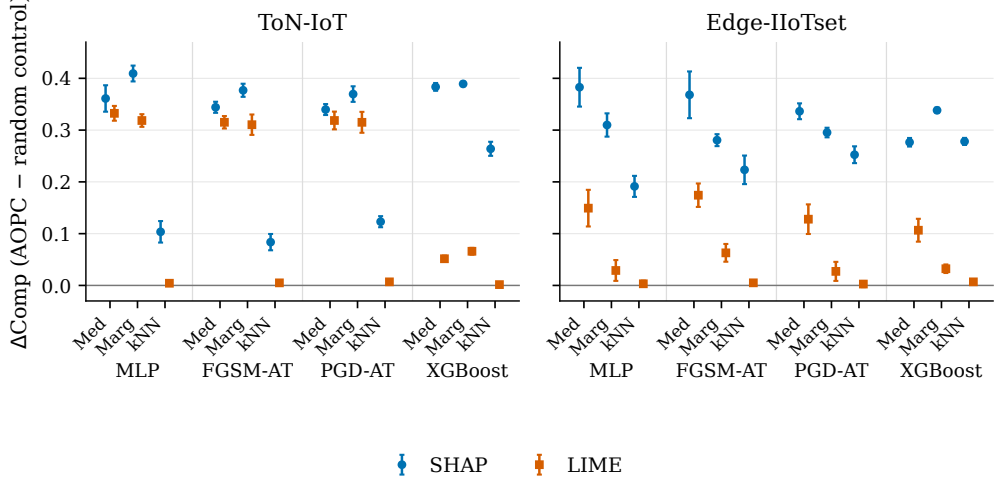


Fig. 5 Comprehensiveness AOPC relative to the random-ranking control for SHAP and LIME under the median (Med), marginal (Marg) and k -NN in-fill (kNN) references, on clean records (20 seeds, mean $\pm$ SD). FGSM-AT and PGD-AT denote the MLP with FGSM and PGD adversarial training.

0.03 on Edge-IIoTset, so only there does part of its change reflect the perturbation. SHAP is essentially unchanged by the small constrained perturbations ($\rho \geq 0.98$ at $\epsilon = 0.05$, where few fields change) and stays at 0.84–0.99 under maximum padding. Within the constrained threat model, SHAP is therefore more stable than LIME in every comparison.

Change under invalid perturbations. Under the unconstrained attack, SHAP rankings of the clean MLP change more ($\rho = 0.27$ on ToN-IoT, 0.56 on Edge-IIoTset) than LIME rankings (0.41, 0.72), and the same holds under the numeric-only attack on Edge-IIoTset (0.73 against 0.78). These are the only three of the 44 P3 comparisons in which LIME is confirmatorily more stable. Because these attacked records are not valid inputs (Section 4.2), we do not draw conclusions about explanation quality from them: they show only that SHAP responds more strongly than LIME to inputs outside the data domain. The signed cosine reveals a further effect that rank measures miss: on ToN-IoT, black-box evasion of XGBoost reverses the sign of the attribution vector (signed cosine -0.24 and -0.28 under the numeric-only and unconstrained attacks) while the Spearman correlation of absolute values stays at 0.94 and 0.66.

4.6 Seed-level inference and sensitivity analyses

With 20 seeds, all 52 P2 comparisons and all 44 P3 comparisons are confirmatory; the P3 comparisons favour SHAP in 41 cases and LIME in the three invalid-input cases above. All adversarial-training and threat-model contrasts in Table 6 are confirmatory except where values are identical by construction (zero constrained attack success on Edge-IIoTset), four ToN-IoT constrained-versus-numeric contrasts for the adversarially trained MLPs (Holm $p = 0.50$; both attacks near zero) and the numeric-versus-unconstrained contrast for XGBoost on Edge-IIoTset ($p = 0.06$). None of the

Table 10 Stability of absolute group attributions (Spearman ρ). Call/bg. = repeated LIME calls with different sampling seeds, or SHAP with resampled background sets (5 seeds, 100 records); seed = same clean record under different training seeds (190 seed pairs); pad. = clean versus maximally padded record (constrained); unc. = clean versus unconstrained adversarial record ($\epsilon = 0.05$, invalid inputs).

Dataset	Detector	SHAP				LIME			
		call/bg.	seed	pad.	unc.	call/bg.	seed	pad.	unc.
ToN-IoT	MLP	0.94	0.96	0.84	0.27	0.82	0.80	0.75	0.41
	MLP+FGSM-AT	0.93	0.97	0.87	0.75	0.79	0.76	0.70	0.53
	MLP+PGD-AT	0.93	0.97	0.90	0.77	0.79	0.76	0.69	0.57
	XGBoost	0.92	0.99	0.97	0.66	0.71	0.71	0.62	0.60
Edge-IIoTset	MLP	0.93	0.99	0.97	0.56	0.75	0.73	0.77	0.72
	MLP+FGSM-AT	0.90	0.95	0.95	0.62	0.60	0.57	0.58	0.57
	MLP+PGD-AT	0.89	0.92	0.95	0.63	0.59	0.56	0.56	0.54
	XGBoost	0.93	0.96	0.99	0.95	0.51	0.50	0.50	0.51

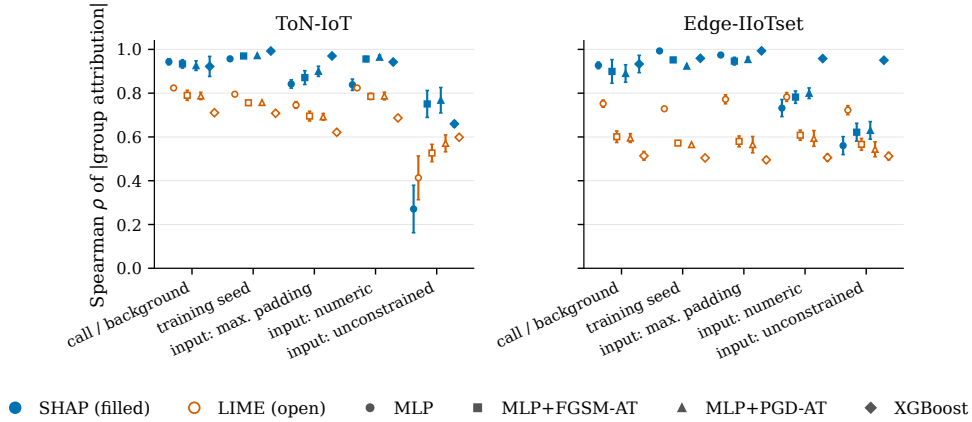


Fig. 6 Sources of attribution change. Spearman correlation of absolute group attributions between repeated explainer calls (LIME) or background draws (SHAP), between training seeds, and between clean and attacked records under maximum constrained padding, the numeric-only attack and the unconstrained attack ($\epsilon = 0.05$). Points are seed means; error bars are seed standard deviations.

clean-performance differences between adversarial and clean training is significant (95% intervals within ± 0.006 macro-F1).

Other stratified partitions. Mean macro-F1 differs between partitions by at most 0.006, the SHAP-over-LIME ordering of ΔComp holds on every partition, dataset and detector, and the attack-success patterns of Table 6 repeat. For ToN-IoT MLP unconstrained attack success, the between-partition variance component is zero and the between-seed variance is 0.077: seed 29 is vulnerable on the primary partition (0.68) but not on the others (0.10, 0.07), whereas seed 41 is vulnerable only on the third (0.72). The bimodality is thus a property of individual training realisations, not of a seed value or a partition.

Label conflicts retained. The conflicting rows are highly duplicated: the 7,122 conflicting ToN-IoT rows and 17,982 conflicting Edge-IIoTset rows reduce to 26 and 558 additional unique records once each feature vector is kept with each of its labels. Retaining them therefore changes the data little, and clean performance changes by at most 0.003 in macro-F1 and 0.013 in false-positive rate (Table 11). Removing the conflicts does not inflate the reported performance.

Temporal partitions. Neither dataset supports a clean global chronological split. The ToN-IoT train-test table has no time-stamp column, and its rows are grouped by attack type, so ordering by row places only attack records, three of whose families never occur in training, in the test partition. The Edge-IIoTset frame time is parseable for 90% of rows but records only the year and time of day, so its ordering is chronological within a day at best; ordering by it also places no benign records in the test partition. Both global variants therefore cannot measure false positives and are not interpreted. The within-family variants are informative. On Edge-IIoTset, training on the earlier and testing on the later traffic of each family lowers macro-F1 by about 0.01 (0.940 and 0.942 against 0.951 and 0.953) and raises the false-positive rate from 14.5% to 17.8% at the tuned threshold, while attack success and the SHAP-LIME ordering are unchanged. On ToN-IoT, where only row order is available, a contiguous-block split within each family leaves clean performance unchanged. Attack success on the MLP varies more between data variants than clean performance does (for example 0.41 and 0.14 under the unconstrained attack on ToN-IoT with conflicts retained and in row order, against 0.30 on the primary partition with the same five seeds), consistent with the seed and realisation dependence described above, and the SHAP-LIME ordering of ΔComp holds in every variant.

Table 11 Sensitivity to data preparation (5 seeds, means; validation-tuned threshold). ASR: clean MLP under unconstrained PGD ($\epsilon = 0.05$) and maximum padding, XGBoost under 300-query random search at maximum padding. ΔComp : marginal reference, clean records, SHAP / LIME, averaged over detectors. ^aToN-IoT has no time stamps; contiguous blocks in row order within each family. ^bChronological within each family by frame time (time of day).

Dataset	Partition	Macro-F1		FPR		ASR			ΔComp
		MLP	XGB	MLP	XGB	MLP unc.	MLP pad.	XGB pad.	
ToN-IoT	primary (stratified)	0.969	0.995	0.057	0.009	0.30	0.42	0.017	0.38 / 0.25
	conflicts retained	0.972	0.996	0.044	0.006	0.41	0.64	0.174	0.39 / 0.25
	row order, per family ^a	0.969	0.995	0.053	0.006	0.14	0.19	0.090	0.37 / 0.28
Edge-IIoTset	primary (stratified)	0.951	0.953	0.145	0.144	0.29	0.31	0.004	0.30 / 0.03
	conflicts retained	0.948	0.950	0.155	0.153	0.27	0.27	0.001	0.30 / 0.03
	temporal, per family ^b	0.940	0.942	0.178	0.178	0.30	0.28	0.004	0.29 / -0.02

4.7 Answers to the research questions

RQ1 (faithfulness under evasion).

SHAP explanations remain faithful to the model’s behaviour on attacked records under all three threat models and for both model families (ΔComp 0.21–0.53), and SHAP is more faithful than LIME in all 52 primary comparisons. The ordering holds for every masking reference, for KernelSHAP as a common estimator and for every LIME configuration tested. The masking reference strongly affects the magnitudes: under conditional in-fill, LIME is barely better than random.

RQ2 (stability).

Under realisable (constrained) perturbations, SHAP rankings barely change ($\rho \geq 0.84$ even at maximum padding), whereas LIME’s change under attack is mostly no larger than its own call-to-call variability and exceeds it clearly only at maximum padding on ToN-IoT. Changes under invalid feature-space perturbations are larger for SHAP but are not used to judge explanation quality. Signed metrics expose sign reversals that rank metrics miss.

RQ3 (seed-level effects).

With 20 seeds, all primary explainer comparisons and almost all training-condition and threat-model contrasts are confirmatory, conditional on the primary partition. The training seed is the dominant source of variation in ToN-IoT MLP robustness. Its bimodality is largely, though not entirely, a threshold effect, and it is invisible in clean metrics.

RQ4 (trade-offs at realistic operating points).

At a 1% false-positive budget, no detector is robust to maximum padding on ToN-IoT: the MLPs detect at most 3% of padded attacks, with or without adversarial training, and XGBoost detects 65% once a tree-aware attack is used. On Edge-IIoTset, adversarial training keeps its benefit at matched false-positive rates and XGBoost remains robust. Adversarial training costs no clean accuracy. Unconstrained feature-space attack success overstates realisable risk, mostly through invalid values for the MLPs, but projection shows that the tree ensemble’s feature-space vulnerability is not purely an encoding artefact.

5 Discussion

5.1 Operational implications

The operating point is as consequential as the model. On Edge-IIoTset, the validation-tuned threshold gives a test false-positive rate of 14.3–14.5% (Table 5), that is, about 1,430–1,450 alerts per 10,000 benign flows, while detecting 99.6–99.9% of attacks. At the 1% validation budget, the test false-positive rate is 0.8–1.3% (Section 4.3), that is, about 80–130 alerts per 10,000 benign flows, and 87–89% of attacks are still detected. Alert volumes of the first kind would overwhelm an analyst team [61]. Robustness

should therefore be reported at fixed false-positive budgets: at the detectors’ own tuned thresholds, robustness and alert volume are confounded, and on ToN-IoT the ranking of detectors changes with the operating point. No detector is robust in every setting. XGBoost has the highest clean accuracy and resists padding on Edge-IIoTset, but on ToN-IoT a tree-aware attack evades it on 35% of padded attacks at the 1% budget, and a 300-query random search would have missed this almost entirely. Robustness claims for tree ensembles therefore need attacks that exploit the tree structure. Input validation at the flow-parsing stage can reject the syntactically invalid records that dominate feature-space attack success against the MLPs, but it would not stop the projected, syntactically valid records that still evade XGBoost. SHAP is preferable to LIME on every criterion measured here, including cost (Supplementary Section S5).

5.2 What the threat models do and do not establish

The constrained threat model encodes conditions that every realisable evasion must satisfy on a flow or packet record: which fields the sender controls, the direction in which they can change, and integer granularity. It does not establish that a modified record can be produced by a real attack tool while preserving its malicious function, because adversarial records were neither regenerated as packets nor replayed through the original feature extractors (Zeek for ToN-IoT, tshark for Edge-IIoTset) [7, 8]. Several side effects of padding are also not modelled (Table 3): coupled length fields that should rise together (including the header-overhead relation between IP bytes, payload bytes and packets, which holds in only 98.8% of ToN-IoT training flows and was therefore not enforced), checksums, sequence numbers of later packets, and timeouts of the flow monitor. The ToN-IoT feature set contains no rate features, so a longer duration changes no other retained field, but a feature set with rates would couple duration to them. Constrained attack success is therefore an estimate of evasion that satisfies the stated necessary conditions, not a measured end-to-end evasion rate. It can err in both directions: unmodelled side effects can make some constrained records unrealisable, and an attacker who can change fields we froze (for example by switching protocol or splitting a flow) has options we did not evaluate. The projection results show how much of the feature-space attack success depends on syntactically invalid values; they do not turn projected records into realisable ones, since projected records still change immutable fields.

5.3 Evaluating explanations

Deletion-based faithfulness has two failure modes that our results separate. The first is the target: scoring every record by the malicious output and ranking by absolute attribution produces negative comprehensiveness for 39–76% of records, which evaluating the predicted class with evidence-signed rankings reduces to 0–16%. The second is the reference: random rankings achieved comprehensiveness of 0.12–0.17 under the marginal reference, a large part of the raw explainer scores, and conditional in-fill removes most of LIME’s advantage over random. We recommend reporting predicted-class AOPC relative to a random control under at least one distribution-preserving reference, and checking that explainer rankings agree across references before drawing conclusions.

Two caveats limit what these scores show. First, the marginal reference and interventional SHAP share the same construction, which may favour SHAP; the k -NN reference does not, and the explainer-configuration analysis (Section 4.4) shows that the ordering also holds for KernelSHAP and for every LIME configuration tested. The k -NN in-fill only approximates the conditional distribution locally, with eight neighbours from a pool of 5,000 records, so dependencies that this pool does not represent remain unmodelled. Second, comprehensiveness and sufficiency measure the model’s response to masking. They do not establish that a feature causes the attack to be detected in the data-generating sense, and they say nothing about whether analysts find the explanations useful; neither causal relevance nor human interpretability was evaluated.

5.4 Explanations as an attack surface

All attacks in this study target the detector’s decision; none targets the explainer. An adaptive adversary could instead search for records that keep the decision but change the explanation, or use exposed attributions to find the fields with the most leverage [45, 46]. Our results bound only part of this risk. Under the constrained model, SHAP attributions changed little even when the decision changed, so an analyst would see the same leading fields for a padded and an unpadded record, while LIME attributions changed by about as much as LIME’s own sampling noise, and somewhat more only at maximum padding on ToN-IoT. Whether exposing SHAP attributions to an attacker accelerates evasion, and whether explanation-targeted attacks succeed under the same constraints, are open questions that require attacks designed for that purpose.

5.5 Limitations

Datasets and generalisation. Two public testbed datasets are used, one flow-based and one packet-based, both recorded in controlled environments. They do not represent the heterogeneity of operational IoT/IIoT deployments, and no live or cross-environment traffic was evaluated. Temporal generalisation could be tested only partly: ToN-IoT’s train–test table has no time stamps, Edge-IIoTset’s frame time records only the time of day, and neither supports a global chronological split with benign test traffic. The within-family partitions (Section 4.6) test drift within families; evolving attack behaviour across deployments and cross-environment transfer, which the different record formats of the two datasets prevent, are not tested. *Architectures.* The audit covers an MLP and gradient-boosted trees. Convolutional, recurrent, transformer and attention-based detectors, which operate on packet sequences or other representations, are not evaluated, and our conclusions about model families should not be extended to them without testing. *Explanation subset.* Explanations are computed for 300 malicious and 300 benign records per seed, stratified by family with at least one record per family. Rare families are therefore represented by few records (the rarest families have 1–4 explained malicious records per seed: ransomware and MITM in ToN-IoT, MITM, fingerprinting and DDoS-UDP in Edge-IIoTset), and per-family explanation results would be imprecise; the seed-level aggregates are the basis of inference. *Deployment.*

Runtime and memory were measured on a cloud machine with recorded data; feasibility on a specific embedded device and under continuous industrial traffic was not measured. *Scope of the contribution.* The paper contributes an evaluation methodology and evidence obtained with it; it does not propose a new detector, explainer or defence. *Statistical conclusion validity.* Inference is conditional on the primary partition, and comparisons smaller than the minimum detectable effect may remain non-significant; such results are reported as descriptive rather than as evidence of no difference.

6 Conclusion

This paper presents a seed-level audit of explanation quality under evasion for IoT/IIoT intrusion detection, covering two de-duplicated datasets, 20 training seeds, neural and tree-based detectors, three threat models with a validity audit, faithfulness under three masking references against a random control, and four separately measured sources of explanation instability. Feature-space attack success largely reflects syntactically invalid records for neural detectors, but not for the tree ensemble, part of whose vulnerability survives projection onto valid records. Under domain constraints, small budgets rarely evade any detector, but large padding budgets evade every MLP on ToN-IoT at a 1% false-positive budget, with or without adversarial training, and a tree-aware attack evades XGBoost on 35% of padded ToN-IoT attacks where random search found almost none. SHAP was more faithful than LIME in all 52 primary comparisons, a result that held for every masking reference, for KernelSHAP and for every LIME configuration, and SHAP attributions were stable under realisable perturbations while LIME’s change under attack was mostly within its own sampling noise.

Two findings concern evaluation practice. Neural-detector robustness on ToN-IoT varied bimodally across training seeds and was invisible in clean metrics, so single-seed or few-seed robustness claims are unreliable; and comparisons of detectors at their own tuned thresholds confound robustness with alert volume. Future work should regenerate and replay constrained adversarial flows through the original feature extractors to test end-to-end realisability, evaluate sequence and attention-based detectors and further datasets, and study attacks that target the explanations themselves.

Supplementary information. The online Supplementary Information reports the full experimental configuration, the effect of individual preprocessing choices, an adversarial-training sweep, a multiclass attack-family analysis and resource profiling.

Acknowledgements. The authors used Claude (Anthropic), an AI assistant, for language editing and as a coding assistant. The authors carried out all experiments, verified all code, results and text, and are solely responsible for the content of this article.

Declarations

Ethical approval

Not applicable. This study used only publicly available, de-identified benchmark network-traffic datasets (ToN-IoT and Edge-IIoTset). It involved no human participants, human data, human tissue or animals.

Consent to participate

Not applicable. No human participants were involved in this study.

Consent to publish

Not applicable. The manuscript contains no data, images or other information relating to an identifiable individual. All authors have read and approved the final manuscript and consent to its publication.

Competing interests

The authors declare no competing interests.

Funding

No funding was received for this research.

Data availability

The raw ToN-IoT and Edge-IIoTset datasets are publicly available from their original providers [51–53]. They are not redistributed by the authors, but the SHA-256 hash of each raw file used is published so that copies can be verified. All derived data are openly available at [doi:10.5281/zenodo.23682203](https://doi.org/10.5281/zenodo.23682203) under a CC BY 4.0 licence. These comprise the duplicate and label-conflict audit, the partition summaries, the seed-level predictive, attack, explanation-faithfulness, stability and runtime tables for all runs, the results of the robustness, explainer-configuration and data-variant analyses, the secondary analyses, the complete execution logs and the analysis outputs (tests, tables and figures). The partitions and the attack and explanation subsets are not stored as files; they are regenerated exactly by the archived code from the hash-verified raw files and the fixed partition seeds.

Code availability

The complete executable pipeline (Python package, Kaggle notebook, environment specification, smoke test and analysis scripts that regenerate every table and figure) is openly available at <https://github.com/nihal4/iot-xai-evasion-audit> and archived at [doi:10.5281/zenodo.23682203](https://doi.org/10.5281/zenodo.23682203) under the MIT licence.

Author contributions

S. M. Nihal Ahmed conceived and designed the study, led the methodological formulation, implemented the analysis, conducted the experiments, interpreted the results, and drafted the manuscript. Afrim Hossen Khan contributed to the methodological discussions, interpretation of the findings, and critical revision of the manuscript. Sabikun Nahar Sinthia contributed to the literature review, manuscript review, and editorial revision. All authors reviewed and approved the final manuscript.

References

- [1] Rahman, M.M., Shakil, S.A., Mustakim, M.R.: A survey on intrusion detection system in IoT networks. *Cyber Security and Applications* (2025) <https://doi.org/10.1016/j.csa.2024.100082>
- [2] Rehman, H.M.R.U., Liaquat, S., Gul, M.J., Jhandir, M.Z., Gavilanes, D., Vergara, M.M., Ashraf, I.: A systematic literature study of machine learning techniques based intrusion detection: datasets, models, challenges, and future directions. *Journal of Big Data* (2025) <https://doi.org/10.1186/s40537-025-01323-2>
- [3] Singhal, S., Kumar, K.A.: AI-driven cybersecurity for industrial Internet of Things: architectures, challenges, datasets, and future research directions. *Frontiers in Big Data* **9**, 1938279 (2026) <https://doi.org/10.3389/fdata.2026.1938279>
- [4] He, K., Kim, D.D., Asghar, M.R.: Adversarial machine learning for network intrusion detection systems: A comprehensive survey. *IEEE Communications Surveys & Tutorials* (2023) <https://doi.org/10.1109/COMST.2022.3233793>
- [5] Rjoub, G., Bentahar, J., Wahab, O.A., Mizouni, R., Song, A., Cohen, R., Otrok, H., Mourad, A.: A survey on explainable artificial intelligence for cybersecurity. *IEEE Transactions on Network and Service Management* (2023) <https://doi.org/10.1109/TNSM.2023.3282740>
- [6] Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., Sepahvand, N.M., Raff, E., Madan, K., Voleti, V., Kahou, S.E., Michalski, V., Arbel, T., Pal, C., Varoquaux, G., Vincent, P.: Accounting for variance in machine learning benchmarks. In: *Proceedings of Machine Learning and Systems*, vol. 3, pp. 747–769 (2021)
- [7] Pierazzi, F., Pendlebury, F., Cortellazzi, J., Cavallaro, L.: Intriguing properties of adversarial ML attacks in the problem space. In: *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1332–1349 (2020). <https://doi.org/10.1109/SP40000.2020.00073>
- [8] Apruzzese, G., Andreolini, M., Ferretti, L., Marchetti, M., Colajanni, M.: Modeling realistic adversarial attacks against network intrusion detection systems. *Digital Threats: Research and Practice* **3**(3), 1–19 (2022) <https://doi.org/10.1016/j.dtrp.2022.100001>

- [9] Hooker, S., Erhan, D., Kindermans, P.-J., Kim, B.: A benchmark for interpretability methods in deep neural networks. In: *Advances in Neural Information Processing Systems* 32 (2019)
- [10] Hase, P., Xie, H., Bansal, M.: The out-of-distribution problem in explainability and search methods for feature importance explanations. In: *Advances in Neural Information Processing Systems* 34 (2021)
- [11] Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? In: *Advances in Neural Information Processing Systems* 35, *Datasets and Benchmarks Track* (2022)
- [12] Munilla, J., Khammas, R.M.: Evaluation of explainable artificial intelligence in IoT intrusion detection systems under DeepFool adversarial conditions. *Sensors* **26**(10), 2924 (2026) <https://doi.org/10.3390/s26102924>
- [13] Vitorino, J., Praca, I., Maia, E.: Towards adversarial realism and robust learning for IoT intrusion detection and classification. *Annals of Telecommunications* (2023) <https://doi.org/10.1007/s12243-023-00953-y>
- [14] Simonetto, T., Dyrmishi, S., Ghamizi, S., Cordy, M., Traon, Y.L.: A unified framework for adversarial attack and defense in constrained feature space. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*, pp. 1313–1319 (2022). <https://doi.org/10.24963/ijcai.2022/183>
- [15] Kalakoti, R., Bahsi, H., Nomm, S.: Improving IoT security with explainable AI: Quantitative evaluation of explainability for IoT botnet detection. *IEEE Internet of Things Journal* (2024) <https://doi.org/10.1109/JIOT.2024.3360626>
- [16] Gaspar, D., Silva, P., Silva, C.: Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron. *IEEE Access* (2024) <https://doi.org/10.1109/ACCESS.2024.3368377>
- [17] Gummadi, A.N., Napier, J.C., Abdallah, M.: XAI-IoT: An explainable AI framework for enhancing anomaly detection in IoT systems. *IEEE Access* (2024) <https://doi.org/10.1109/ACCESS.2024.3402446>
- [18] Kaur, B., Dadkhah, S., Shoeleh, F., Neto, E.C.P., Xiong, P., Iqbal, S., Lamon-tagne, P., Ray, S., Ghorbani, A.A.: Internet of Things (IoT) security dataset evolution: Challenges and future directions. *Internet of Things* (2023) <https://doi.org/10.1016/j.iot.2023.100780>
- [19] Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., Han, H.: A systematic

- literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security* (2022) <https://doi.org/10.1016/j.cose.2022.102675>
- [20] Neto, E.C.P., Iqbal, S., Buffett, S., Sultana, M., Taylor, A.: Deep learning for intrusion detection in emerging technologies: a comprehensive survey and new perspectives. *Artificial Intelligence Review* (2025) <https://doi.org/10.1007/s10462-025-11346-z>
 - [21] Sahani, N., Zhu, R., Cho, J.-H., Liu, C.-C.: Machine learning-based intrusion detection for smart grid computing: A survey. *ACM Transactions on Cyber-Physical Systems* (2023) <https://doi.org/10.1145/3578366>
 - [22] 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*, pp. 785–794 (2016). <https://doi.org/10.1145/2939672.2939785>
 - [23] Alotaibi, A., Rassam, M.A.: Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense. *Future Internet* (2023) <https://doi.org/10.3390/fi15020062>
 - [24] Zhang, C., Costa-Perez, X., Patras, P.: Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms. *IEEE/ACM Transactions on Networking* (2022) <https://doi.org/10.1109/TNET.2021.3137084>
 - [25] Alshahrani, E., Alghazzawi, D., Alotaibi, R., Rabie, O.: Adversarial attacks against supervised machine learning based network intrusion detection systems. *PLOS ONE* (2022) <https://doi.org/10.1371/journal.pone.0275971>
 - [26] Sharma, S., Chen, Z.: A systematic study of adversarial attacks against network intrusion detection systems. *Electronics* (2024) <https://doi.org/10.3390/electronics13245030>
 - [27] Aloraini, F., Javed, A., Rana, O.: Adversarial attacks on intrusion detection systems in in-vehicle networks of connected and autonomous vehicles. *Sensors* (2024) <https://doi.org/10.3390/s24123848>
 - [28] Alkadi, S., Al-Ahmadi, S., Ismail, M.M.B.: RobEns: Robust ensemble adversarial machine learning framework for securing IoT traffic. *Sensors* (2024) <https://doi.org/10.3390/s24082626>
 - [29] Barik, K., Misra, S.: A comprehensive defense approach of deep learning-based NIDS against adversarial attacks. *Multimedia Tools and Applications* (2025) <https://doi.org/10.1007/s11042-025-21008-5>
 - [30] Heydari, V., Nyarko, K.: Enhancing adversarial robustness in network intrusion

- detection: A novel adversarially trained neural network approach. *Electronics* (2025) <https://doi.org/10.3390/electronics14163249>
- [31] Danesh, W., Sapireddy, S.R., Rahman, M.: Understanding and detecting adversarial examples in IoT networks: A white-box analysis with autoencoders. *Electronics* (2025) <https://doi.org/10.3390/electronics14153015>
 - [32] Ziras, G., Farao, A., Zarras, A., Xenakis, C.: From vulnerability to resilience: Adversarial training and real-time detection for AI security. *Array* (2025) <https://doi.org/10.1016/j.array.2025.100546>
 - [33] Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems* 30 (2017)
 - [34] Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you? Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144 (2016). <https://doi.org/10.1145/2939672.2939778>
 - [35] Keshk, M., Koroniotis, N., Pham, N., Moustafa, N., Turnbull, B., Zomaya, A.Y.: An explainable deep learning-enabled intrusion detection framework in IoT networks. *Information Sciences* (2023) <https://doi.org/10.1016/j.ins.2023.119000>
 - [36] Shtayat, M.M., Hasan, M.K., Sulaiman, R., Islam, S., Khan, A.U.R.: An explainable ensemble deep learning approach for intrusion detection in industrial Internet of Things. *IEEE Access* (2023) <https://doi.org/10.1109/ACCESS.2023.3323573>
 - [37] Sharma, B., Sharma, L., Lal, C., Roy, S.: Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach. *Expert Systems with Applications* (2024) <https://doi.org/10.1016/j.eswa.2023.121751>
 - [38] Kaur, N., Gupta, L.: Enhancing IoT security in 6G environment with transparent AI: Leveraging XGBoost, SHAP and LIME. In: *2024 IEEE 10th International Conference on Network Softwarization (NetSoft)* (2024). <https://doi.org/10.1109/NetSoft60951.2024.10588922>
 - [39] Moss, J., Gordon, J., Duclos, W., Liu, Y., Wang, Q., Wang, J.: Explainable AI in IoT: A survey of challenges, advancements, and pathways to trustworthy automation. *Electronics* (2025) <https://doi.org/10.3390/electronics14234622>
 - [40] Mohale, V.Z., Obagbuwa, I.C.: A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity. *Frontiers in Artificial Intelligence* (2025) <https://doi.org/10.3389/frai.2025.1526221>
 - [41] Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R.: Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions*

- on Neural Networks and Learning Systems **28**(11), 2660–2673 (2017) <https://doi.org/10.1109/TNNLS.2016.2599820>
- [42] DeYoung, J., Jain, S., Rajani, N.F., Lehman, E., Xiong, C., Socher, R., Wallace, B.C.: ERASER: A benchmark to evaluate rationalized NLP models. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4443–4458 (2020). <https://doi.org/10.18653/v1/2020.acl-main.408>
 - [43] Aas, K., Jullum, M., Løland, A.: Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence* **298**, 103502 (2021) <https://doi.org/10.1016/j.artint.2021.103502>
 - [44] Visani, G., Bagli, E., Chesani, F., Poluzzi, A., Capuzzo, D.: Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models. *Journal of the Operational Research Society* **73**(1), 91–101 (2022) <https://doi.org/10.1080/01605682.2020.1865846>
 - [45] Slack, D., Hilgard, S., Jia, E., Singh, S., Lakkaraju, H.: Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 180–186 (2020). <https://doi.org/10.1145/3375627.3375830>
 - [46] Pawlicki, M., Pawlicka, A., Kozik, R., Choras, M.: Explainability versus security: The unintended consequences of xAI in cybersecurity. In: Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems (2024). <https://doi.org/10.1145/3665451.3665527>
 - [47] Masud, M.T., Keshk, M., Moustafa, N., Linkov, I., Emge, D.K.: Explainable artificial intelligence for resilient security applications in the Internet of Things. *IEEE Open Journal of the Communications Society* (2025) <https://doi.org/10.1109/OJCOMS.2024.3413790>
 - [48] Ojo, S., Krichen, M., Alamro, M., Mihoub, A.: TXAI-ADV: Trustworthy XAI for defending AI models against adversarial attacks in realistic CIoT. *Electronics* (2024) <https://doi.org/10.3390/electronics13091769>
 - [49] Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K.: Dos and don’ts of machine learning in computer security. In: 31st USENIX Security Symposium (USENIX Security 22), pp. 3971–3988 (2022)
 - [50] Sharma, D.P., Xue, L., Sun, X., Lin, X., Xiong, P.: Enhancing adversarial robustness of IoT intrusion detection via SHAP-based attribution fingerprinting. In: 2025 IEEE 24th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) (2025). <https://doi.org/10.1109/Trustcom66490.2025.00125>

- [51] 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* **8**, 165130–165150 (2020) <https://doi.org/10.1109/ACCESS.2020.3022862>
- [52] Booiij, T.M., Chiscop, I., Meeuwissen, E., Moustafa, N., Hartog, F.T.H.: ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet of Things Journal* **9**(1), 485–496 (2022) <https://doi.org/10.1109/JIOT.2021.3085194>
- [53] 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* **10**, 40281–40306 (2022) <https://doi.org/10.1109/ACCESS.2022.3165809>
- [54] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: *International Conference on Learning Representations* (2015)
- [55] Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. In: *International Conference on Learning Representations* (2020)
- [56] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: *International Conference on Learning Representations* (2018)
- [57] Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: A query-efficient black-box adversarial attack via random search. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science*, vol. 12368, pp. 484–501. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58592-1_29
- [58] Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: *Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 70, pp. 3145–3153 (2017)
- [59] 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. *Nature Machine Intelligence* **2**(1), 56–67 (2020) <https://doi.org/10.1038/s42256-019-0138-9>
- [60] Holm, S.: A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* **6**(2), 65–70 (1979)
- [61] Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: *2010 IEEE Symposium on Security and Privacy*, pp. 305–316 (2010). <https://doi.org/10.1109/SP.2010.25>

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryInformation.pdf](#)