Robust Detection of Malicious Network Traffic Under Adversarial Attacks

Mohammed Amine Merzoug
Computer Science Department
University of Batna 2
Batna 05078, Algeria
amine.merzoug@univ-batna2.dz

Selma Rihani
Computer Science Department
University of Batna 2
Batna 05078, Algeria
selma.rihani@univ-batna2.dz

Ahmed Mostefaoui
DISC Dept, FEMTO-ST Institute
University of Marie et Louis Pasteur
Belfort 90000, France
ahmed.mostefaoui@univ-fcomte.fr

***Abstract*—Adversarial machine learning has become a foundational component of modern network intrusion detection systems (NIDS). Yet, existing adversarial vulnerability assessments heavily rely on unrealistic white-box assumptions, unconstrained adversarial features, or homogeneous data distributions, which fail to reflect production networking realities. To bridge these gaps, this paper introduces a realistic, zero-knowledge black-box evaluation framework for NIDS. Our methodology models an adversary operating in a cross-dataset paradigm, using a local surrogate multi-layer perceptron (MLP) optimized on the CICIDS2017 dataset to launch transfer-based evasion attempts against four target classifiers trained independently on a disjoint dataset (NF-UQ-NIDS-v2). To preserve network traffic realism, we integrate the Tabular Constraint Guaranteed Evasion (TCGE) algorithm to ensure all generated adversarial examples strictly comply with network protocols. In particular, we systematically evaluate four representative classifiers (Logistic Regression, Random Forest, XGBoost, and MLP) across a designed suite of five scenarios, spanning baseline performance (no defense and no adversarial attack), vulnerability to realistic adversarial attacks (no defense against attacks), adversarial training defense (defense against seen attacks), cross-attack generalization (defense against unseen attacks), and robustness–accuracy trade-off (defense against no attack).**

***Index Terms*—Adversarial Machine Learning; Network Intrusion Detection; Realistic Adversarial Attacks; Adversarial Training; Tabular Feature Constraints; Machine Learning Robustness**

I. Introduction

In modern cybersecurity, AI acts both as a threat and as a defense mechanism. While it can be exploited by attackers to develop more sophisticated and adaptive cyberattacks, it also serves as a powerful tool for enhancing prevention, threat detection, and response capabilities. For instance, nowadays, AI has been widely adopted in network security systems such as firewalls and intrusion detection/prevention systems (IDS/IPS) to help detect malicious stealth traffic and protect network infrastructures [1], [2].

Despite these advances, AI-based network intrusion detection systems (NIDS) remain vulnerable and face increasing difficulty in coping with the growing sophistication and volume of modern cyberthreats [3]. In particular, AI-based NIDS are vulnerable to adversarial attacks [4], [5], which introduce small, carefully crafted perturbations to input traffic to cause the model to misclassify malicious flows as benign, while remaining imperceptible to detection systems.

Adversarial attacks can occur at different stages of the model lifecycle, including training time (e.g., data poisoning, which injects malicious training data to corrupt the learning phase) and inference time (e.g., evasion attacks, where malicious inputs are manipulated to bypass a deployed model's detection boundaries) [5]. Among these, evasion attacks are a critical threat in operational NIDS deployment, as they allow malicious traffic to bypass detection at inference time without requiring any modification to the deployed model [5]. Further details on adversarial attacks and adversarial defense mechanisms are provided in the Background section (II).

A. Research Motivation

Among existing defense mechanisms, adversarial training has emerged as a promising preventive approach for improving robustness by exposing models to adversarial examples during the optimization phase. However, despite considerable research in this area [3], [6]–[8], critical limitations remain that obscure the true operational efficiency of these defenses:

- **Isolated Assessments:** Existing studies frequently focus on an isolated AI model architecture [8], a singular attack vector [7], [9], or restrict their evaluation to highly symmetric, unrealistic threat vectors [3].
- **Unrealistic Adversarial Capabilities:** A significant portion of existing literature relies on relaxed, unconstrained adversarial assumptions [3], [6], [8]. They assume attackers possess complete structural knowledge of the model (white-box configurations) or enjoy unlimited oracle interactions without enforcing physical protocol constraints, disregarding whether such manipulation can be feasibly deployed within real-world network streams without terminating the session.
- **Homogeneous Data:** Prior evaluations heavily assume that both the attacker and the victim NIDS operate on identical underlying data distributions and matching feature spaces [3], [8]. This assumption rarely holds in practice, as real-world adversaries must collect, parse, and mutate traffic independent of the victim's proprietary training set.

As a result of these gaps, the reported vulnerability of AI-based NIDS in current literature may be vastly overstated or practically unfeasible, leading to misleading conclusions

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

regarding their actual robustness under production conditions. Consequently, this research is motivated by a fundamental question: *To what extent can AI-based NIDS be trusted if they are evaluated under artificial benchmarks rather than realistic operational constraints?*

Rather than merely comparing a loose set of models, this study establishes a rigorous cross-dataset, constrained black-box framework to systematically investigate the adversarial impact on heterogeneous machine learning and deep learning architectures. Our research design is explicitly structured to address the following four key Research Questions (RQs):

- **RQ1:** How do undefended, heterogeneous ML and DL-based NIDS behave when subjected to realistic adversarial attacks (constrained, hard-label black-box adversarial attacks) that strictly comply with real-world network protocols?
- **RQ2:** How effective is adversarial training in enhancing the empirical robustness of network classifiers against attacks seen during the training phase?
- **RQ3:** Does defensive robustness built against a specific seen attack vector successfully generalize to counter structurally distinct or stronger unseen adversarial strategies?
- **RQ4:** Does the integration of adversarial training introduce a detrimental trade-off that compromises the nominal classification accuracy of standard non-adversarial traffic?

B. Methodology and Contributions

This paper proposes a unified and realistic evaluation framework for assessing the robustness of AI-based NIDS under adversarial conditions. We consider a fully realistic black-box threat model in which the attacker has no access to the victim model, its architecture, parameters, or training data, and interacts with the victim system only through oracle queries that return binary classification responses (benign or malicious), with no access to model confidence scores or internal states.

We evaluate four representative AI classifiers: Logistic Regression, Random Forest, XGBoost, and Multi-Layer Perceptron, trained on the NF-UQ-NIDS-v2 dataset [10] against an attacker operating on a different dataset (CICIDS2017 [11]). The empirical assessment is organized into five distinct experimental configurations:

- **Scenario 1 – Baseline Performance:** Models are trained and evaluated on standard network traffic containing benign and malicious samples, entirely free of adversarial perturbations. This unattacked and undefended configuration establishes the nominal classification performance baseline under standard operating conditions.
- **Scenario 2 – Vulnerability To Realistic Adversarial Attacks:** Models are optimized on standard network streams containing benign and malicious samples, but evaluated against adversarial test streams. This setup quantifies the inherent security degradation of undefended classifiers subjected to constrained black-box evasion attempts.
- **Scenario 3 – Adversarial Training Defense:** Models are optimized using a robust training dataset augmented with adversarial examples, and subsequently evaluated against the same type of attack. This scenario measures defensive efficacy against expected, seen threats.
- **Scenario 4 – Cross-Attack Generalization:** Models undergo adversarial training against a specific perturbation strategy but are evaluated against a completely different, unseen adversarial attack. This scenario determines whether defensive robustness transfers across diverse threat vectors.
- **Scenario 5 – Robustness–Accuracy Trade-off:** Models are reinforced via adversarial training but evaluated on standard network streams containing benign and malicious samples. This configuration measures whether hardening the NIDS against malicious perturbations inadvertently compromises its nominal classification accuracy on standard non-adversarial traffic.

The underlying adversarial traffic across these scenarios is generated via two attack methods: Fast Gradient Sign Method (FGSM) [12] and Projected Gradient Descent (PGD) [12]. To preserve realism, these examples are strictly bounded by the Tabular Constraint Guaranteed Evasion (TCGE) algorithm [13], which ensures that all resulting adversarial variants constitute valid network flows.

C. Structure of the Paper

The remainder of this paper is organized as follows. Section II reviews the background on adversarial attacks and their defense mechanisms. Section III presents the proposed adversarial evaluation methodology, including the threat model, victim and attacker datasets, adversarial example generation, feature constraints, and evaluation scenarios. Section IV presents the experimental results and discussion. Finally, Section V concludes the paper and outlines directions for future work.

II. Background

This section provides the necessary background on adversarial attacks and their corresponding countermeasures.

A. Adversarial Attacks

Adversarial attacks exploit inherent vulnerabilities in AI models by adding very small modifications to the input data to deceive the model and cause it to make incorrect decisions [5]. In the context of cybersecurity, the impact of adversarial attacks extends beyond model performance degradation to pose significant security threats with serious consequences.

These threats can be classified according to several criteria, including:

- **Attacker Timing:** Training-time attacks corrupt the learning itself (e.g., data poisoning (label-focused), data manipulation (feature-focused), and model manipulation), and inference-time attacks manipulate inputs at deployment (e.g., evasion attacks, privacy attacks, and model extraction) [5]. This study focuses on inference-time evasion attacks.

- **Attacker Knowledge:** In a white-box setting, the attacker has full access to the model architecture, parameters, and gradients, and in a black-box setting, the attacker has no direct access to the target model and must rely on transferability from a surrogate model or query-based estimation [5]. In gray-box attacks, the attacker only knows part of the information about the targeted system. This study focuses on realistic black-box attacks.
- **Attacker Objective:** In targeted output attacks, the attackers have a specific objective, forcing the model to generate a predetermined false result. In untargeted output attacks, attackers are not concerned with a specific output; they only seek to degrade overall performance. In this work, we assume that the attacker aims to cause the model to misclassify malicious flows as benign, while remaining imperceptible to detection systems.
- **Attacker Capability:** Direct attacks occur when the adversary interacts directly with the target model, using either complete structural knowledge (white-box) or query responses (black-box) to craft specific perturbations. Transfer-based attacks exploit the cross-model portability of adversarial examples. Here, the attacker generates perturbations on a local, substitute model and deploys them against a completely separate, unseen target system. In this study, we adopt a transfer-based approach, where adversarial samples are generated using a surrogate Multi-Layer Perceptron (MLP) model and then applied to the victim models.

B. Adversarial Defenses

Adversarial defense can be classified into preventive, detection, reactive, and privacy-preserving approaches [5].

- **Preventive Approaches:** Aim to improve model robustness before or during the learning phase by reducing inherent architectural vulnerabilities to adversarial perturbations. For instance, *input purification* runs on 100% of data and removes adversarial disturbances from it before it reaches the model. *Adversarial training*, on the other hand, augments the training dataset with adversarial examples, thereby training the model to resist such attacks. In this work, we focus on preventive defense, specifically adversarial training.
- **Detection Approaches:** Function as an external monitoring layer to identify adversarial inputs at inference time before they can influence predictions, without modifying the underlying model weights.
- **Reactive Approaches:** Respond dynamically to detected attacks at runtime through real-time correction, model adaptation, or post-deployment recovery mechanisms.
- **Privacy-Preserving Approaches:** Focus on safeguarding the model's intellectual property, training data, and architecture from extraction attacks.

III. Adversarial Evaluation Methodology

This section outlines our empirical NIDS evaluation framework. We define the zero-knowledge threat model (III-A), detail the asymmetric datasets (III-B), present the gradient-based attack methods (III-C), and outline the protocol constraints applied to preserve network flow integrity (III-D). Finally, these components are structured into five scenarios (III-E) to answer our core research questions.

A. Threat Model

We adopt a black-box threat model because it represents the most realistic attack scenario for deployed AI-based NIDS [6]. In this setting, the attacker aims to craft malicious network flows that evade detection during the inference phase, while having no access to the victim model's architecture, parameters, or training data. Instead, after the victim models have been trained and deployed, the attacker performs a reconnaissance phase by submitting network flow samples and observing only their final hard-label predictions, namely benign (0) or malicious (1).

Each queried network flow is paired with the hard-label prediction returned by the deployed victim models to construct a surrogate training dataset. These hard-label predictions serve as pseudo-labels for training a local MLP surrogate model following the model extraction strategy of [14].

MLP is selected as the surrogate model because its differentiable architecture enables efficient gradient computation required by gradient-based attacks such as FGSM and PGD [12]. Moreover, MLP-based surrogate models are widely adopted in transfer-based black-box adversarial attacks [3], [15].

Following the model extraction strategy of [14], we limit the attacker to 5,000 oracle queries to train the surrogate model. Prior work has demonstrated that effective surrogate models can be learned through black-box oracle queries, while reducing the number of queries remains a key goal of model extraction attacks [14], [16], [17]. Therefore, this query budget represents a practical and stealth-conscious attacker that limits probing activity while collecting enough information to approximate the victim model's decision boundary. In particular, we consider an attacker that has two levels of knowledge:

- **Partial Knowledge Attack:** Assumes that both the surrogate and victim models are trained on the same dataset. This scenario evaluates the transferability of adversarial samples when the attacker has partial knowledge of the victim's data distribution.
- **Strict Black-Box Attack:** Assumes that the surrogate model is trained on an independent dataset from the victim dataset. Specifically, the victim models are trained on the NF-UQ-NIDS-v2 dataset [10], whereas the attacker trains the surrogate model using the CICIDS2017 dataset [11], collected from a different network environment. This scenario evaluates attack transferability under a more realistic black-box setting.

B. Victim and Attacker Datasets

The first dataset, NF-UQ-NIDS-v2 (NetFlow University of Queensland Network Intrusion Detection System) [10], serves as the native training environment for the four victim models for binary benign and malicious classification, while

the second dataset, CICIDS2017 (Canadian Institute for Cybersecurity Intrusion Detection Evaluation Dataset) [11], is used by the attacker to train the surrogate model and generate adversarial examples.

*1) **Victim Dataset**:* NF-UQ-NIDS-v2 [10] was constructed by merging four standardised NetFlow-based datasets [10]: *i*) NF-UNSW-NB15-v2, *ii*) NF-BoT-IoT-v2, *iii*) NF-ToN-IoT-v2, and *iv*) NF-CSE-CIC-IDS2018-v2. Out of its 75.9 million total records, 33.12% represent benign traffic while the remaining 66.88% constitute malicious network flows, making it the largest publicly available NIDS dataset exceeding the scale of individual datasets.

Unlike traditional NIDS datasets that rely on complex and handcrafted features, this dataset is based on NetFlow features, which are lightweight, scalable, and deployable in real-world network environments without extensive feature engineering. These features primarily represent Layer 3 (e.g., `MIN_TTL`, `MAX_TTL`) and Layer 4 (e.g., `PROTOCOL`, `TCP_FLAGS`), with a limited number of Layer 7 features (e.g., `DNS_QUERY_TYPE`, `DNS_TTL_ANSWER`).

Before model training, the dataset undergoes a multi-stage preprocessing pipeline to guarantee data integrity and maximize classifier performance:

- **Data Sampling:** Due to physical computational constraints (specifically RAM and GPU limitations), a stratified subset of 2,000,000 records is selected. This downsampling strategy minimizes memory overhead while explicitly preserving the structural diversity and native class distributions of the original dataset.
- **Feature Filtering:** The raw dataset is initially composed of 43 features describing distinct network flow characteristics. Four metadata features (`L4_SRC_PORT`, `L4_DST_PORT`, `Dataset`, and `Attack`) are systematically removed because they introduce non-generalizable target leakage and do not contribute to semantic feature learning, yielding a final set of 39 features for training and evaluation.
- **Data Splitting:** The sampled dataset is partitioned using an 80/20 division rule, allocating 80% of the records for model optimization and training, and reserving the remaining 20% as a completely disjoint holdout test set.
- **Data Scaling:** A standard z-score (`StandardScaler`) normalization is applied to the feature space to center the data around a zero mean and unit variance, stabilizing gradient propagation prior to model training.

Given its curated construction, NF-UQ-NIDS-v2 [10] requires no cleaning or anomaly imputation. Data preparation was therefore restricted to data sampling and feature filtering.

*2) **Attacker Dataset**:* CICIDS2017 [11] contains approximately 2.8 million network flow records captured over a five-day duration, encompassing both benign activity and a wide range of attack vectors, such as DoS, DDoS, brute-force, web attacks, and botnets. The dataset provides 78 flow-level features extracted via the CICFlowMeter tool [11], covering packet length statistics, inter-arrival times, flow duration, and TCP flag counts.

This dataset is collected from a completely distinct network environment than the victim dataset (NF-UQ-NIDS-v2), reflecting a realistic scenario in which the attacker has no access to the victim's proprietary training data. The following preprocessing steps are applied to the attacker dataset:

- **Data Cleaning:** Records containing infinite or missing values are systematically removed.
- **Data Sampling:** To maintain computational feasibility and balance storage overhead, a representative subset of 225,745 network flow records is selected for this study, consisting of 97,718 benign samples (43.29%) and 128,027 active attack samples (56.71%).
- **Data Splitting:** The cleansed dataset is partitioned at an 80/20 ratio, allocating 80% of the traffic to local surrogate optimization and 20% for adversarial evaluation.
- **Data Scaling:** A standard z-score normalization is applied across the feature space.

*3) **Feature-Space Alignment**:* Since CICIDS2017 [11] and NF-UQ-NIDS-v2 [10] datasets use different feature representations, we apply a semantic feature-mapping procedure to align corresponding traffic features with the victim feature space. Of the 39 victim features, 28 are mapped to corresponding CICIDS2017 features, while the remaining 11 victim features have no direct counterparts and are therefore zero-filled to obtain the required 39-dimensional input representation.

C. Adversarial Example Generation

We utilize two standard gradient-based attack methods to generate adversarial examples. This allows us to quantify the vulnerability and structural robustness of the target AI models under two different evaluation environments.

- **Fast Gradient Sign Method (FGSM) [12]:** A foundational, single-step attack designed for rapid adversarial generation. It computes the gradient of the loss function with respect to the input data and applies a single-step perturbation in the direction of the sign of the gradient, effectively maximizing the model's classification error with minimal computational overhead.
- **Projected Gradient Descent (PGD) [12]:** An iterative optimization-based extension of FGSM that applies multiple gradient updates with smaller step sizes. After each iteration, the perturbed input is projected back onto a specified ϵ-bounded constraint space. This iterative approach generates significantly more aggressive adversarial examples that can successfully bypass complex defenses.

D. Adversarial Feature Constraints

To ensure that all generated adversarial examples remain realistic and feasible within production environments, we enforce domain-aware constraints on feature perturbations using the TCGE algorithm [13].

These constraints are derived from real-world network protocol specifications rather than arbitrary dataset properties. Accordingly, features are classified into three primary groups:

- **Immutable Features (e.g., protocol types, TCP flags, and DNS/ICMP fields):** These must remain completely

unchanged because they are determined by the underlying network protocol stack. Altering these values would break the state machine of the communication session, causing immediate session termination or packet drops by intermediate routers, which defeats the attacker's purpose of stealthy evasion.

- **Perturbable Numerical Features (e.g., byte counts, packet counts, flow durations, and retransmission rates):** These features can be dynamically modified within valid operational bounds because they represent traffic volume and timing characteristics. An attacker can realistically influence these values at runtime (e.g., by introducing artificial delays or injecting dummy packets) while still strictly complying with protocol semantics.
- **Derived Features (e.g., throughput and packet distributions):** These features must remain consistent with underlying traffic properties because they are computed directly from raw traffic measurements. Modifying them independently would produce internally inconsistent flow records that cannot exist in any real network capture, making the attack instantly detectable by anomaly-based parsers.

Following the adversarial perturbation step, all samples are projected back onto a constrained domain to enforce non-negativity, integrality constraints, and feature boundary limits. This projection preserves intrinsic logical dependencies, such as valid packet size ordering, Time-to-Live (`TTL`) inequalities (`MIN_TTL` $\leq$ `MAX_TTL`), and bounded retransmission rates that cannot exceed total traffic volume.

E. Evaluation Scenarios

We evaluate four representative machine learning and deep learning classifiers (Logistic Regression, Random Forest, XGBoost, and Multi-Layer Perceptron) across five experimental configurations.

- **Scenario 1 – Baseline performance:** Target models are optimized and evaluated on standard network traffic consisting of benign and non-adversarial malicious samples (NF-UQ-NIDS-v2 [10]). This undefended and unattacked configuration establishes the nominal performance baseline under standard operational conditions, serving as the primary gauge against which subsequent scenarios are interpreted.
- **Scenario 2 – Vulnerability To Realistic Adversarial Attacks:** Target classifiers are optimized on standard network traffic consisting of benign and non-adversarial malicious samples (NF-UQ-NIDS-v2 [10]) but evaluated against adversarial test streams generated using both FGSM and PGD (detailed in Case 1 and Case 2 below). This setup quantifies the inherent security degradation of undefended models subjected to constrained black-box evasion attempts, which addresses **RQ1**.
  To evaluate varying degrees of adversarial difficulty, this configuration is divided into two contexts:
  - ***Case 1 – Partial Knowledge Attack:*** The local surrogate model is trained on the native dataset (NF-UQ-NIDS-v2 [10]), representing a scenario where the attacker lacks architectural insights but operates within the victim's exact training feature distribution to generate FGSM and PGD attacks.
  - ***Case 2 – Strict Black-Box Attack:*** The victim models remain trained on NF-UQ-NIDS-v2 [10], while the black-box adversary must train their surrogate model and extract responses using a disjoint dataset (CICIDS2017 [11]), which mirrors how a real-world attacker operates without any access to the victim's data.
- **Scenario 3 – Adversarial Training Defense**: This configuration measures the empirical efficacy of adversarial training as a defense mechanism against malicious feature perturbations encountered during the optimization phase, which addresses **RQ2**. To implement this setup, the attacker leverages the disjoint dataset (CICIDS2017 [11]) to train the surrogate architecture, generating protocol-constrained FGSM and PGD adversarial variants. Target classifiers are trained on data augmented with FGSM examples (NF-UQ-NIDS-v2 [10]) and subsequently evaluated against an FGSM-perturbed test stream. For the second evaluation, an identical pipeline is mirrored independently for PGD, which tests the resilience against expected attacks.
- **Scenario 4 – Cross-Attack Generalization:** This scenario evaluates whether defensive robustness built against an expected perturbation strategy transfers successfully to a completely unseen attack (e.g., training against a weak attack but evaluating against a strong attack, and vice versa). This scenario addresses **RQ3**, which represents a critical real-world example since adversaries are rarely bound to anticipated defense strategies. Two cases are considered. In the first one, models are trained using FGSM adversarial examples and evaluated against PGD. In the second case, models are trained using PGD and evaluated against FGSM.
- **Scenario 5 – Robustness–Accuracy Trade-off:** Target classifiers undergo adversarial training using both FGSM and PGD examples and are subsequently evaluated on standard network traffic consisting of benign and non-adversarial malicious samples (NF-UQ-NIDS-v2 [10]). This configuration measures whether hardening the NIDS introduces an operational penalty that compromises its nominal classification accuracy on standard traffic, which addresses **RQ4**.

IV. Empirical Evaluation and Discussion

This section presents the experimental results across the five evaluation scenarios and interprets the findings with respect to the four RQs. Three main metrics are considered in this study:

- **Accuracy:** The ratio of correctly predicted instances over the total number of instances.
- **Evasive Rate (ER) [13]:** Quantifies the proportion of generated adversarial examples (AE) that successfully

bypass the target classifier and cause a misclassification, calculated as follows:

$$\text{ER} = \frac{|AE_{\text{evade}}|}{|AE_{\text{total}}|} \tag{1}$$

where AE_{total} represents the complete set of generated adversarial examples and AE_{evade} denotes the subset that successfully misclassifies the target model.

- **Valid Evasive Rate (VER) [13]:** Measures the ratio of successful adversarial evasions that maintain network protocol validity and satisfy all domain-enforced constraints.

$$\text{VER} = \frac{|AE_{\text{evade}} \cap AE_{\text{valid}}|}{|AE_{\text{total}}|} \tag{2}$$

To guide the interpretation of our empirical results, the relationship between ER and VER serves as a diagnostic tool. If the tables reveal an absolute identity (ER $=$ VER), this demonstrates that the TCGE algorithm functioned effectively as an absolute realism filter, ensuring every successful evasion remains a valid network flow. Conversely, if a condition where VER $<$ ER was found, it would indicate that a portion of the adversarial noise capable of fooling the models breaks underlying protocol rules.

A. Scenario 1 – Baseline Performance

Table I presents the standard detection performance of the four victim models trained on NF-UQ-NIDS-v2 [10], providing the baseline performance in the absence of any adversarial perturbation.

TABLE I
SCENARIO 1 – BASELINE PERFORMANCE (NO DEFENSE AND NO ADVERSARIAL ATTACK).

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	98.36%	98.39%	98.36%	98.36%
XGBoost	97.98%	98.00%	97.98%	97.98%
Logistic Regression	84.03%	84.09%	84.03%	84.02%
MLP	95.47%	95.49%	95.47%	95.47%

Among the evaluated models, Random Forest achieves the highest accuracy (98.36%), followed by XGBoost (97.98%) and MLP (95.47%), indicating that tree-based ensemble methods and deep neural networks effectively capture the complex pattern present in network intrusion data. In contrast, the performance of Logistic Regression (84.03%) reflects the limited capacity of linear classifiers to capture the complex behavior of malicious network traffic.

B. Scenario 2 – Vulnerability To Realistic Adversarial Attacks

Scenario 2 evaluates the vulnerability of the four models under adversarial attacks in the absence of any defense mechanism. Two cases are considered:

1) ***Partial Knowledge Attack****:* In this case, the attacker trains the surrogate model using the same dataset (NF-UQ-NIDS-v2 [10]) as the victim models, while having no access to their architecture or parameters. Table II compares the baseline clean detection accuracy of each victim model with its accuracy under FGSM and PGD attacks generated by the surrogate model.

TABLE II
SCENARIO 2 (CASE 1) – DETECTION ACCURACY UNDER PARTIAL KNOWLEDGE FGSM AND PGD ATTACKS.

Model	Clean Acc (%)	FGSM Acc (%)	PGD Acc (%)
Random F.	98.36	73.00	64.76
XGBoost	97.98	72.16	68.24
Logistic R.	84.03	83.71	83.70
MLP	95.47	90.63	91.36

Table II reveals two key observations. First, tree-based models exhibit substantially greater performance degradation than the MLP under both attacks, indicating higher vulnerability to transfer-based adversarial examples. Second, PGD consistently causes a larger reduction in detection accuracy than FGSM across all victim models, which reflects the advantage of iterative optimization in generating more effective adversarial perturbations.

Table III reports the Evasive Rate (ER) and Valid Evasive Rate (VER) for both attacks across all victim models. The equality of ER and VER across all configurations indicates that the TCGE algorithm successfully preserves traffic validity while generating effective adversarial examples.

TABLE III
SCENARIO 2 (CASE 1) – EVASIVE RATE (ER) AND VALID EVASIVE RATE (VER) UNDER PARTIAL KNOWLEDGE FGSM AND PGD ATTACKS.

Victim Model	FGSM Attack		PGD Attack	
	ER (%)	VER (%)	ER (%)	VER (%)
Random F.	27.00	27.00	35.24	35.24
XGBoost	27.84	27.84	31.76	31.76
Logistic R.	16.29	16.29	16.30	16.30
MLP	9.37	9.37	8.64	8.64

2) ***Strict Black-Box Attack****:* In this case, the attacker trains the surrogate model using CICIDS2017 [11], while the victim models remain trained on NF-UQ-NIDS-v2 [10]. This setting represents a strict black-box scenario in which the attacker has no access to the victim model, their training data, or feature distribution.

Table IV compares the baseline clean detection accuracy obtained on both datasets and the corresponding accuracy under FGSM and PGD attacks. Specifically, Column 3 (CICIDS2017, Acc %) presents the baseline performance under out-of-distribution conditions (optimizing on NF-UQ-NIDS-v2 and evaluating on CICIDS2017), while Column 4 (resp. Column 5) reports the classification accuracy under transferred FGSM (resp. PGD) attacks.

TABLE IV
SCENARIO 2 (CASE 2) – DETECTION ACCURACY UNDER STRICT BLACK-BOX FGSM AND PGD ATTACKS.

Model	NF-UQ-NIDS-v2 Acc (%)	CICIDS2017 Acc (%)	FGSM Acc(%)	PGD Acc (%)
Random F.	98.36	43.28	43.28	43.31
XGBoost	97.98	55.91	46.60	38.10
Logistic R.	84.03	56.72	56.72	56.72
MLP	95.47	59.86	60.68	61.97

Compared with the baseline results presented in Column 2 (NF-UQ-NIDS-v2, Acc %) (i.e., Scenario 1), all victim models

experience substantial accuracy degradation when evaluated on CICIDS2017 (Scenario 2, case 2), and this in the absence of adversarial perturbations, which indicates that distribution shift alone significantly affects detection performance. Therefore, the adversarial results are interpreted relative to the clean CICIDS2017 baseline rather than attributing the entire performance degradation to FGSM or PGD. Random Forest and Logistic Regression exhibit no additional degradation beyond the cross-dataset baseline, whereas XGBoost remains susceptible to transferred perturbations. In contrast, MLP shows slight improvement relative to the cross-dataset baseline.

Table V reports the corresponding ER and VER values. Consistent with the previous scenario, ER = VER across all configurations.

TABLE V
SCENARIO 2 (CASE 2) – EVASIVE RATE (ER) AND VALID EVASIVE RATE (VER) UNDER STRICT BLACK-BOX FGSM AND PGD ATTACKS.

Victim Model	FGSM Attack		PGD Attack	
	ER (%)	VER (%)	ER (%)	VER (%)
Random F.	56.76	56.76	56.35	56.35
Logistic R.	43.28	43.28	43.28	43.28
XGBoost	48.53	48.53	60.89	60.89
MLP	41.96	41.96	38.25	38.25

Answer to RQ1: Undefended ML and DL-based NIDS are vulnerable to realistic black-box adversarial attacks that strictly comply with real-world network protocols. Our empirical findings reveal that strict cross-dataset black-box attacks are significantly more effective at evading detection than partial-knowledge setups.

C. Scenario 3 – Adversarial Training Defense

In this scenario, we evaluate adversarial training as a defense mechanism against seen attacks. Each victim model is trained on an augmented training dataset (NF-UQ-NIDS-v2 [10]) with adversarial examples generated by the surrogate model using FGSM and PGD (CICIDS2017 [11]).

Table VI compares the detection accuracy of the victim models before and after adversarial training under FGSM and PGD. We notice that adversarial training improves detection accuracy under both FGSM and PGD attacks across all victim models. Overall, FGSM adversarial training achieves a slightly higher detection accuracy than PGD across most models. This difference reflects the simpler perturbation pattern produced by FGSM, making it easier for the models to learn during adversarial training.

TABLE VI
SCENARIO 3 – DETECTION ACCURACY UNDER SEEN FGSM AND PGD ATTACKS.

Model	FGSM (%) Undefended	FGSM (%) Defended	PGD (%) Undefended	PGD (%) Defended
Random F.	73.00	99.98	64.76	99.96
XGBoost	72.16	99.78	68.24	99.67
Logistic R.	83.71	81.91	83.70	81.96
MLP	90.63	99.63	91.36	99.48

Table VII reports the corresponding ER and VER values. Consistent with previous scenarios, ER remains equal to VER.

TABLE VII
SCENARIO 3 – EVASIVE RATE (ER) AND VALID EVASIVE RATE (VER) UNDER SEEN FGSM AND PGD ATTACKS.

Model	FGSM / FGSM		PGD / PGD	
	ER (%)	VER (%)	ER (%)	VER (%)
Random F.	0.02	0.02	0.04	0.04
XGBoost	0.22	0.22	0.33	0.33
Logistic R.	18.09	18.09	18.04	18.04
MLP	0.37	0.37	0.52	0.52

Answer to RQ2: Adversarial training proves highly effective in improving robustness of network classifiers against attacks already seen during training. Both FGSM and PGD adversarial training restore detection accuracy, with FGSM achieving a slightly higher accuracy across most victim models.

D. Scenario 4 – Cross-Attack Generalization

This scenario evaluates whether robustness learned from a seen adversarial attack generalizes to an unseen attack. Two cross-attack configurations are considered. In the FGSM-PGD scenario, the models are trained using FGSM and evaluated on PGD attacks. In the second PGD-FGSM scenario, models are trained using PGD and evaluated against FGSM.

Table VIII compares the detection accuracy of victim models under seen and unseen attacks for both adversarial training strategies. The results show a clear asymmetry in cross-attack generalization. Models trained using PGD retain most of their robustness against unseen FGSM attacks, whereas models trained using FGSM show a larger reduction in detection accuracy when evaluated against unseen PGD attacks. These behaviors reflect the broader perturbation space explored during PGD adversarial training, which improves generalization to weaker FGSM attacks. In contrast, FGSM training does not prepare models for the stronger perturbations of PGD.

Table IX reports the corresponding ER and VER values, which remain equal.

TABLE IX
SCENARIO 4 – EVASIVE RATE (ER) AND VALID EVASIVE RATE (VER) UNDER UNSEEN FGSM AND PGD ATTACKS.

Model	Train FGSM-Test PGD		Train PGD-Test FGSM	
	ER (%)	VER (%)	ER (%)	VER (%)
Random F.	12.72	12.72	4.39	4.39
XGBoost	7.32	7.32	2.61	2.61
Logistic R.	18.12	18.12	18.18	18.18
MLP	6.24	6.24	3.01	3.01

Answer to RQ3: Defensive robustness against unseen attacks improves only when models are trained on stronger adversarial attacks, which indicates that robustness transfers from stronger attacks to weaker ones, but not the opposite.

E. Scenario 5 — Robustness–Accuracy Trade-off

This scenario evaluates the impact of adversarial training on standard detection accuracy. Table X compares the detection accuracy of the victim models before and after adversarial training. The results show that adversarial training introduces a negligible change in detection accuracy. Across all victim

TABLE VIII
SCENARIO 4 – CROSS-ATTACK GENERALIZATION.

Model	Train FGSM Test FGSM (%)	Train FGSM Test PGD (%)	Train PGD Test PGD (%)	Train PGD Test FGSM (%)
Random F.	99.98	87.28	99.96	95.61
XGBoost	99.78	92.68	99.67	97.39
Logistic R.	81.91	81.88	81.96	81.82
MLP	99.63	93.76	99.48	96.99

models, the accuracy difference relative to the baseline performance remains within $\pm 2.10\%$, which indicates that improved robustness is achieved with little or no loss in clean detection accuracy.

TABLE X
SCENARIO 5 – DETECTION ACCURACY BEFORE AND AFTER ADVERSARIAL TRAINING.

Model	Baseline Acc (%)	Training Acc (%)	ΔAcc (%)
Random F.	98.36	98.36	0.00
XGBoost	97.98	97.62	-0.36
Logistic R.	84.03	81.93	-2.10
MLP	95.47	95.60	+0.13

Answer to RQ4: The integration of adversarial training does not introduce a detrimental trade-off that compromises the nominal classification accuracy of standard non-adversarial traffic. The models preserve their detection accuracy while benefiting from improved robustness.

V. CONCLUSIONS AND FUTURE WORK

This paper presented a realistic black-box adversarial evaluation framework for AI-based NIDS, combining a cross-dataset surrogate attack strategy with the use of the TCGE algorithm to generate valid adversarial network traffic. The framework was evaluated through five experimental scenarios across four victim models (Random Forest, Logistic Regression, XGBoost, and MLP) trained on the NF-UQ-NIDS-v2 dataset and attacked by a surrogate MLP model trained on an independent dataset (CICIDS2017). The experimental results show that AI-based NIDS are vulnerable to realistic transfer-based black-box attacks, although the level of vulnerability depends on both the victim models and the attacker's knowledge. Adversarial training substantially improves robustness and restores detection accuracy, generalizes to unseen attacks only when models are trained on stronger adversarial perturbations, and provides an effective defense without compromising normal detection performance.

As future work, we plan to *i*) investigate adaptive, stateful black-box attacks engineered to circumvent robustly trained NIDS architectures and *ii*) extend the proposed adversarial evaluation framework beyond traditional MLP to include modern deep-learning architectures.

REFERENCES

[1] A. Alshammari and A. Aldribi, "Apply machine learning techniques to detect malicious network traffic in cloud computing," *Journal of Big Data*, vol. 8, no. 1, p. 90, 2021.

[2] O. F. Isife, K. Okokpujie, I. P. Okokpujie, R. E. Subair, A. A. Vincent, and M. E. Awomoyi, "Development of a malicious network traffic intrusion detection system using deep learning." *International Journal of Safety & Security Engineering*, vol. 13, no. 4, 2023.

[3] M. S. Haroon and H. M. Ali, "Adversarial training against adversarial attacks for machine learning-based intrusion detection systems." *Computers, Materials & Continua*, vol. 73, no. 2, 2022.

[4] A. Alghuried, A. Alkinoon, A. Alghamdi, S. Choi, S. Camarato, M. Mohaisen, and D. Mohaisen, "Comprehensive evaluation of adversarial perturbations against ml-based ethereum phishing detection systems," *Distributed Ledger Technologies: Research and Practice*, 2025.

[5] Y. Qiao, N. B. Sathyanarayana, C. Shi, Z. He, T. Wang, and T. Hou, "A survey on adversarial machine learning: Attacks, defenses, real-world applications, and future research directions," *Neurocomputing*, p. 132670, 2026.

[6] G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, and M. Colajanni, "Modeling realistic adversarial attacks against network intrusion detection systems," *Digital Threats: Research and Practice (DTRAP)*, vol. 3, no. 3, pp. 1–19, 2022.

[7] J. Li, J. Zhou, L. Li, B. Cheng, S. Li, and W. He, "Robust detection for adversarial attacks-based on a graph neural network for web application security protection system," in *Proceedings of the 4th International Conference on Artificial Intelligence and Computer Engineering*, 2023, pp. 927–932.

[8] K. Roshan, A. Zafar, and S. B. U. Haque, "A novel deep learning based model to defend network intrusion detection system against adversarial attacks," in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2023, pp. 386–391.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[10] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile networks and applications*, vol. 27, no. 1, pp. 357–370, 2022.

[11] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, no. 2018, pp. 108–116, 2018.

[12] W. Villegas-Ch, A. Jaramillo-Alcázar, and S. Luján-Mora, "Evaluating the robustness of deep learning models against adversarial attacks: An analysis with fgsm, pgd and cw," *Big Data and Cognitive Computing*, vol. 8, no. 1, p. 8, 2024.

[13] N. Alhussien, G. Agrawal, and A. Aleroud, "Augmented tabular adversarial evasion attacks with constraint satisfaction guarantees," in *International Conference on Availability, Reliability and Security*. Springer, 2025, pp. 365–386.

[14] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4954–4963.

[15] D. C. Asimopoulos, P. Radoglou-Grammatikis, P. Fouliras, K. Panitsidis, G. Efstathopoulos, T. Lagkas, V. Argyriou, I. Kotsiuba, and P. Sarigiannidis, "Surrogate-guided adversarial attacks: Enabling white-box methods in black-box scenarios," in *2025 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2025, pp. 950–956.

[16] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–34, 2023.

[17] A. Shafran, I. Shumailov, M. A. Erdogdu, and N. Papernot, "Beyond labeling oracles: What does it mean to steal ml models?" *arXiv preprint arXiv:2310.01959*, 2023.