

Adversarial Attack on Deep Learning Based Network Intrusion Detection Systems: An Empirical Analysis across Datasets and Features

Gundapu Thirupathi

2303C50122@sru.edu.in

SR University

V Thirupathi

SR University

Article

Keywords: adversarial machine learning, network intrusion detection, deep neural networks, FGSM, PGD, BIM, evasion attacks, feature vulnerability, KDD Cup 99, CICIDS2017

Posted Date: October 19th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11166352/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Adversarial Attack on Deep Learning Based Network Intrusion Detection Systems: An Empirical Analysis across Datasets and Features

^{*1}Gundapu Thirupathi

¹Research Scholar, CS&AI,SR university, Ananthasagar, Hasanparthy, Warangal india

2303C50122@sru.edu.in

²Dr V Thirupathi

²Associate professor in CS&AI, S.R. University, Ananthasagar, Hasanparthy, Warangal india

v.thirupathi@sru.edu.in

*corresponding Author

Abstract: The use of deep learning (DL) based classifiers as detection engines in network intrusion detection systems (NIDS) has been shown to be effective; nevertheless, these models remain vulnerable to adversarial evasion attacks that undermine their effectiveness in security-critical applications. In this work, we present a cross-dataset study on the adversarial vulnerability of a DNN-based NIDS classifier with the KDD Cup 1999 and CICIDS2017 traffic datasets collected at different time periods and featuring distinct sets of extracted features. For each dataset, we train and evaluate a dedicated DNN classifier against three white-box attacks, Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), and Projected Gradient Descent (PGD) with random restarts, with five perturbation budgets $\epsilon \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$. While the classifiers achieve high accuracy (94.4% and 97.8%) on the KDD Cup 1999 and CICIDS2017 datasets, respectively, they suffer from severe accuracy drop (to 5.6% and 2.3%, respectively) with relative drops of 94.1% and 97.7%, respectively) when exposed to the strongest perturbation (PGD, $\epsilon=0.3$). Moreover, confusion matrix analysis reveals that the degradation is not the result of chance predictions but the result of systematic misclassification due to the shift in the decision boundary induced by the adversarial perturbations, with a substantial number of samples changing their predicted class label. Furthermore, we propose a Feature Vulnerability Index (FVI) to characterize the vulnerability of each input feature with respect to the amount of adversarial perturbation required across different attacks and perturbation budgets. The analysis reveals that CICIDS2017 data features related to timing and packet size statistics such as Flow Packets/s, Fwd IAT Max, and Packet Length Variance exhibit high vulnerability, whereas features representing the volume of network flows show significantly less vulnerability. These results suggest that adversarial vulnerability could be a property of the learned decision surface of the DNN-based NIDS classifier rather than being dataset-specific. The findings suggest that feature-level adversarial robustness should be carefully considered in the deployment of DL-based NIDS in critical security scenarios.

Keywords: adversarial machine learning; network intrusion detection; deep neural networks; FGSM; PGD; BIM; evasion attacks; feature vulnerability; KDD Cup 99; CICIDS2017

1. Introduction

The first line of defence in modern cybersecurity systems is provided by network intrusion detection systems (NIDS). NIDS systems are responsible for detecting malicious behaviour in network traffic in real-time in the context of increasing traffic volume, velocity, and aggressiveness

of cyber threats. Traditional signature-based and rule-based NIDS systems, although interpretable and computationally efficient, fail to keep up with constantly evolving and novel techniques used by modern attackers, such as zero-day exploits and polymorphic attacks which do not correspond to any signature at all. As a result, substantial amount of research has focused on machine learning and, more recently, deep learning (DL) approaches as detectors of choice for next generation NIDS [4], [17]. The key advantage of DNNs is the ability to automatically learn a complex decision boundary using raw or semi-processed traffic data and without need of costly handcrafted features which is the case for most of classical ML algorithms. At the same time, DNNs have repeatedly demonstrated on numerous benchmark datasets detection accuracy higher than 95% [4]. Therefore, DL-based NIDS are no longer limited to theoretical studies; rather, they are becoming increasingly popular in both academia and industry, being proposed and, in some cases, implemented as components of real-world security pipelines.

At the same time, reliance of DL on adversarial robust detection of cybersecurity-related threats comes together with the existence of large body of literature on the topic of adversarial examples of inputs. Adversarial example is a modified version of some input data which causes the neural network to make incorrect predictions but remains semantically identical to its non-perturbed counterpart, i.e., does not change the true class label. Such vulnerabilities were first discovered for image classification tasks [18] and then further generalized by the Fast Gradient Sign Method. Since then, the existence of such vulnerabilities has been proven for most of the high-capacity and gradient-trained DNN architectures covering image classification, natural language processing, speech recognition, and increasingly tabular datasets, including cybersecurity [24]. The main idea is that same gradient information which makes possible efficient training of DNN models via backpropagation can be reversed by an attacker to identify the most rapidly increasing the loss (and, therefore, inducing misclassification) directions in input space, usually at significantly lower computational cost than that of model training itself.

In the context of NIDS, the consequences of this vulnerability are quantitatively more dire than in most other DL applications. While an adversarial perturbation causing a classifier to misclassify an image of a panda as a gibbon in an image classification application is often simply a matter of academic curiosity, in the NIDS context an adversarial perturbation that causes a malicious packet stream to be classified as benign has serious security implications. The attack passes through undetected, meaning that the attacker has been able to conduct any sort of active attack (such as a DoS flood, port scan, or exfiltration attempt) without incurring the significant additional cost of evading the NIDS in addition to the attack itself. This is possible because many important NIDS features (such as packet timing, byte counts, flag distributions, and flow statistics) are, at least within the limits imposed by the physical and protocol constraints, controllable by an attacker in charge of generating the traffic [35]. This threat is backed by the growing body of research in adversarial evasion of DL-based NIDS with realistic constraints on feature manipulation [19], [21]. However, despite the motivating factors mentioned above, there are three major weaknesses exhibited by the state-of-the-art research in adversarial evasion of DL-based NIDS that make it less useful for defenders. First, the vast majority of research is evaluated using only a single dataset [22], [29], [31], [34], which is often either the original (non-duplicated) KDD Cup 99 release or the modern benchmark, CICIDS2017, thus raising the question of whether the reported vulnerabilities of a DL-based detector reflect a general property of that type of detector or something particular about the dataset being used. Second, many researchers use only one-step Fast Gradient Sign Method to evaluate attacks on detectors, while, as both the results of this study and the findings of Alotaibi and Rassam, who claim that Auto-PGD and BIM are their strongest

attacks [29], indicate, this method can greatly underestimate vulnerability of models to iterative attacks, such as Basic Iterative Method and Projected Gradient Descent. Finally, robustness is almost always reported in aggregate form, i.e., only as an accuracy or F1 measure before and after an attack [20], [32], without specifying which exact features were used to evade the NIDS. However, Mohammadian et al. [22] made some progress in this area, identifying minimal feature sets required for an attack using saliency maps, but a systematic approach like the one introduced in this study has yet to emerge.

These are precisely the three gaps that this paper seeks to fill directly. Instead of considering one dataset, we will train and attack a consistent deep neural network architecture with a consistent experimental setup on two independently acquired and structurally different datasets — namely, the classical KDD Cup 1999 dataset and the modern CICIDS2017 dataset — thus allowing us to rigorously check for the generalization of any adversarial vulnerability across data generating processes. In place of studying only FGSM attacks, we systematically study three canonical and increasing in strength gradient based attacks (FGSM, BIM, and PGD with random restarts) for five levels of perturbation budget ranging from nearly imperceptible to worst case. Instead of reporting purely aggregate performance numbers, we propose the Feature Vulnerability Index, which gives a measure for each input feature of the consistency with which gradient based attacks are exploiting it under the entire set of attack configurations. The rest of this paper is structured as follows. Section 2 places this work in context within the related literature. Section 3 describes the datasets used, the victim model, the threat model, the attack formulation and the evaluation methodology in detail, including all the mathematical formulations. Section 4 presents and analyses our empirical results. Section 5 discusses implications of our findings. Section 6 discusses limitations. Finally, Section 7 presents conclusions and implications for defence design.

1.1 Major Contributions of This Work

- **Cross-dataset vulnerability evaluation.** We evaluate identical DNN architectures on two independent datasets, proving adversarial vulnerability generalizes across benchmarks.
- **The Feature Vulnerability Index (FVI).** We introduce the Feature Vulnerability Index, identifying which specific traffic features attackers most consistently exploit.
- **Discovery of near-total prediction inversion.** We discover near-total prediction inversion under attack, a severe failure mode beyond simple accuracy loss.

2. Related Work

Adversarial machine learning vulnerabilities in machine learning have been widely studied and reported in domains beyond network security far before the advent of adversarial machine learning. The concept of adversarial example was introduced when carefully crafted adversarial perturbations were demonstrated to be capable of fooling high-capacity classifiers [18], while analogously structured evasive attacks against face-recognition based presentation-attack detectors [15] and multimodal emotion recognition systems [13] followed later. The presence of adversarial vulnerabilities in a wide range of different machine learning applications demonstrates that it is a general characteristic of gradient-based machine learning and not a specific problem of a certain application domain and justifies the significant amount of literature that has been written on adversarial machine learning for network intrusion detection.

Several surveys and taxonomies of the existing literature can be found. A pioneering work on classification of adversarial attacks on network security in terms of both the problem space and the feature space was published by Ibitoye et al. [24]. He et al. [19] conducts a comparative survey of adversarial attacks against computer vision applications versus NIDS, stating that attacks against image classification cannot be simply transferred to the network traffic without considering

important difference in feature spaces. A comprehensive survey on robustness of machine learning at all steps of ML pipeline from collecting data through model deployment was provided by Wang et al. [7], while recent surveys in 2025 by Ennaji et al. [23] and Anis et al. [4] focused on evasion attacks and deep-learning-based IDSs, respectively. Several surveys covering related areas are provided by Kwon et al. [12] for wireless sensor networks and by Huang et al. [17] for SDN-IoT. Furthermore, a recent survey by Wang et al. [8] covers the intersection of explainability and adversarial robustness in detection of DDoS attacks. Common conclusion drawn in the above survey literature, which also justifies the gap addressed by this study, is that the evaluation of adversarial robustness is done on independent datasets and with different types of attacks and evaluation metrics.

Second, researchers have been exploring the development of increasingly realistic and effective adversarial network traffic. For instance, Apruzzese et al. [35] criticize existing threat models as being unrealistic because many attacks described in published work rely on levels of knowledge and access on the attacker's side that may be difficult to justify in real-life scenarios. In this respect, the current research takes the opposite route by presenting a justified reason for the chosen white-box approach to threat modelling and considering it to be a realistic upper bound in the context of security auditing (Section 3.3). A number of papers consider GANs as a way to generate adversarial or evasive network traffic: Yao et al. [1] develop a packet-level GAN-based framework for achieving evasion rates of up to 99% against nine deep learning-based NIDS with maintaining traffic functionality; Hartono [10] applies GANs to achieve fast generation of adversarial examples; and Alshahrani et al. [31] test the effectiveness of GAN-synthesized traffic on decision tree and logistic regression detectors based on CICIDS2017 dataset, finding the former to be more vulnerable than the latter. Additionally, Wang et al. [5] suggest a framework for evasion using meta-learning, which operates with a zero-query black box constraint, while Han et al. [21] conduct one of the few systematic gray/black box traffic-space (as opposed to feature space only) adversarial attack studies against Kitsune NIDS achieving evasion rates of over 97%. Finally, Mohammadian et al. [22] use Jacobian-Saliency-Map method for identifying smallest sets of features required for evasion, thus performing one of the few feature-level analyses of adversarial attacks like the one proposed in this paper through the introduction of Feature Vulnerability Index (Section 3.5). Ibitoye et al. [25] and Alhajjar et al. [28] both analyze the relative vulnerability of various model classes, with the latter applying an approach combining evolutionary computations and GAN-based generation to two datasets: NSL-KDD and UNSW-NB15. Earlier work by Peng et al. [27] represents one of the very first empirical studies analyzing the performance of a DL-based NIDS under adversarial conditions. It established the important observation subsequently validated and further developed by nearly every study surveyed here, including the current one: clean-data performance is not indicative of adversarial robustness.

The third approach suggests and evaluates defence mechanisms. Zhang et al. [20] describe a framework incorporating ensembling, adversarial training and querying to restore 100% attack detection accuracy for several attacks. Adversarial training continues being the most frequently suggested defence method: Haroon and Ali [32] and Roshan et al. [34] assess the performance of the FGSM/JSMA/PGD/C&W-trained defences for NSL-KDD, UNSW-NB15 and CICIDS2017; Saha et al. [2] evaluate the effect of combining the adversarial and clean-loss with calibration-aware regularization on the retained accuracy of 84.9% and 81.75% under FGSM and PGD attacks, respectively; and Alotaibi and Rassam [29] implement a GAN-based purification defence having identified Auto-PGD and BIM to be the most effective attacks against their CNN-based IDS — this result is corroborated by this study's observation of the overwhelming superiority of iterative

attacks over the single-step FGSM (see Section 4.2). Yuan et al. [26] develop a relatively light-weight defence framework aimed at avoiding the cost of the heavy retraining associated with adversarial training; Rai et al. [33] assess a layered hybrid defence combining adversarial training, input sanitization and anomaly filtering against the FGSM, PGD, DeepFool and GAN-based attacks for CNN, LSTM, autoencoder and transformer models, finding the last mentioned to cause the biggest damage out of the four — this reminds of the fact that while this study evaluates three types of attacks (FGSM, BIM and PGD), it is not as comprehensive as most single-attack assessments found in literature in terms of evasion strategies variety. Ensemble and architectural defences are examined by Le et al. [6], who emphasize the importance of the dataset-independence of the results obtained, just like this study does; and by Yasarithna and Le-Khac [16], who identify the greater resilience of denoising autoencoders to poisoning and out-of-distribution samples compared to regular and variational autoencoders for CICIDS2017, InSDN and CICIOT2023. Attribution-based explainability-integrated defences are suggested by Hussini et al. [14]. Data augmentation via GANs as a defence strategy is discussed by Khalil et al. [9]. It should be mentioned that Jmila and Khedher [30] shift attention from deep neural networks to shallow classifiers and conclude about the strong dependence of the performance-robustness trade-off on the particular type of classification algorithm used — this point has been overlooked by most research, including this study, that assume a specific architecture of the DNN model. Huang and Li [36] expand the scope of adversarial robustness issues to the detection of attacks on the power system network. Also worth mentioning are the dissertations by Alatwi [3] and Sarikaya [11] (the latter developing an autoencoder reconstruction error-based classifier).

Nevertheless, while there have been many more studies on the topic, these are still three major drawbacks common to the great majority of those covered by the review in Section 1 that motivated the present study: first, almost all of them use one particular dataset only – predominantly either CICIDS2017 or NSL-KDD but not both; secondly, many rely exclusively on FGSM attacks even though there is empirical evidence suggesting that iterative attacks are substantially more powerful; thirdly, very few conduct feature-level diagnostics of adversarial attacks, apart from Mohammadian et al. [22], whose work is limited to minimal subset selection. The present study attempts to fill these three voids at once by means of its approach outlined in Section 3.

3. Methodology

The current research defines the detection and analysis of adversarial attacks on DL-enabled NIDS as the problem of white-box evasion attack where an adversary, knowing all the parameters of the classifier, tries to find an input perturbation that will result in incorrect classification but is limited in magnitude. The whole process involves five steps: (i) data collection and deduplication, (ii) feature preprocessing and scaling, (iii) training of the deep neural network classifier, (iv) creation of adversarial samples with three types of gradient-based evasion attacks for different perturbation budgets, and (v) performance evaluation of classification and feature-level vulnerability. The detailed description of each step follows below.

3.1 Datasets and Preprocessing

The two independently acquired, freely available network traffic datasets used are KDD Cup 1999 dataset [41] and CICIDS2017 dataset (cleaned and pre-processed version) [42]. Both are popular NIDS benchmark datasets but have many differences in terms of feature schema (41 connection-level features vs. 52 flow-based statistics after pre-processing), data collection methodology, environment where the data was collected, and age – while KDD Cup 99 is made up of artificially generated data set based on a simulated 1998 military network, CICIDS2017 consists of actual enterprise network traffic recorded nearly two decades later. This choice of using two very

different datasets for comparison ensures that the test for adversarial vulnerability is valid and generalizable, not biased by any artifacts inherent in the dataset itself.

Before feeding them into the models, both datasets underwent explicit de-duplication, as the original KDD Cup 99 is known to consist of a very large number of duplicates that would otherwise cause model evaluation to favour trivial memorization of patterns and give falsely high accuracy metrics. Table 1 shows the number of samples in each dataset before and after de-duplication. De-duplication resulted in removal of 348,435 out of 494,021 KDD Cup 99 training samples (70.5%) and 233,738 out of 311,029 test samples (75.1%), thus verifying the well-known problem with duplications in the original KDD Cup 99 dataset. In contrast, the cleaned version of CICIDS2017 contained only 161 duplicated rows out of 2,520,751 (0.006%).

Table 1. Dataset Record Counts Before and After De-duplication

Dataset	Split	Rows Before	Rows After	Duplicates Removed	% Removed
KDD Cup 99	Train	494,021	145,586	348,435	70.5%
KDD Cup 99	Test	311,029	77,291	233,738	75.1%
CICIDS2017	Full (pre-split)	2,520,751	2,520,590	161	0.006%

The categorical features from KDDCup99 (protocol_type, service, flag) have been one-hot encoded, thus increasing the number of columns from 41 to 118; the numeric form of CICIDS2017, on the other hand, has not needed any encoding. The multi-class class variable of each dataset has been simplified to a binary class variable $y = \{0,1\}$, where $y = 0$ means benign/normal traffic and $y = 1$ indicates an attack, as the focus of this research is to evade the basic detect/not-detect decision, and not the type of attack. Min-max normalization to unit scale was performed on all features,

$$xx'_j = \frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)} \quad (1)$$

applied to each individual feature j , utilizing statistics computed from the training set without modification when applied to the test set, the output being constrained between $[0, 1]$. The scaling into a uniform bounded range means that the adversarial attack budget ε can be consistently compared between both data sets despite their differing native units and ranges of possible values.

3.2 Victim Model Architecture

The victim classifier $f_0: \mathbb{R}^d \rightarrow [0,1]^2$ is implemented as a fully connected DNN with three hidden layers of widths 128, 64 and 32 units respectively, where each hidden layer is followed by batch normalization and dropout with $p = 0.3$, with ReLU activation functions everywhere. For an input $x \in \mathbb{R}^d$, the hidden representation at layer l .

This three-layer fully connected network with widths of 128, 64 and 32 respectively is a considered choice and not a mere default configuration due to the following reasons. First, the widths of the network have been selected to provide adequate representational power to capture the non-linear dependencies among the tens to low hundreds of traffic features present in both the datasets KDD Cup 99 (with 118 features post-one hot encoding) and CICIDS2017 (with 52 features). Second, the width and depth of the networks have been selected to keep the network shallow and narrow enough such that it can be trained in a few seconds per epoch on CPU only hardware, which is representative of many practical NIDS deployment and research environments and is consistent

with what is reported in the NIDS literature using DL. Third, batch normalization is used after every hidden layer to normalize the distribution of the hidden layer outputs during training and allow for a higher learning rate with the Adam optimizer without causing the training process to diverge; while dropout is used as a regularizer at a rate of 0.3 to prevent overfitting of the majority classes in both the datasets, especially given that overconfidence on the clean data distribution has been found to cause sharper decision boundaries, which are easier to fool in the adversarial robustness literature. Finally, this network architecture along with all training hyperparameters remains unchanged and identical for both the datasets (Section 3.6), such that any difference in adversarial vulnerability in the two datasets reported in Section 4 can be attributed to differences in the data distributions alone.

$$h^{(l)} = \text{ReLU}(W^{(l)} h^{(l-1)} + b^{(l)}), l = 1, 2, 3, h^{(0)} = x \quad (2)$$

and the output layer applies a softmax to produce class-membership probabilities,

$$p_c(x) = \frac{\exp(z_c)}{\sum_{c'=1}^2 \exp(z_{c'})}, z = W^{(4)} h^{(3)} + b^{(4)} \quad (3)$$

The model uses a two-unit softmax output as opposed to one sigmoid unit with a 0.5 cutoff in order for the categorical cross entropy loss defined in Equation (4) and its derivative $\nabla_x L(\theta, x, y)$ to be calculated according to the standard multi-class loss formulation expected by the attack methods implemented in the Adversarial Robustness Toolbox (Section 3.3). This approach ensures that there are no inconsistencies with the implementation of the model during training as opposed to when calculating the gradient of the loss function for the purposes of crafting the adversarial sample: attacks generated on the basis of the loss landscape of a sigmoid-output binary classifier will not necessarily be applicable to a softmax-output equivalent, because the gradient in the former and latter formulations will be different.

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^2 y_{i,c} \log p_c(x_i) \quad (4)$$

using the Adam optimizer (learning rate set to its default value of 0.001 with $\beta_1 = 0.9$ and $\beta_2 = 0.999$), using early stopping based on a held-out validation set (15% of the training set) and measuring loss on the validation set with a patience of five epochs to restore the model parameters after the epoch achieving the minimum validation loss up to a maximum of 40 training epochs. This early stopping mechanism is employed for both datasets to avoid the problem of overtraining the victim model into an unnecessarily narrowed and hence more vulnerable parameter space, as well as ensuring that the clean accuracy against which adversarial attacks degrade performance is not overly optimistic.

3.3 Threat Model and Adversarial Attack Formulation

This is a white-box, untargeted threat model, appropriate for security audits: the defender tests the security of their own model against the most threatening attacker who could reasonably attain access to its parameters (for instance, through model extraction, insider attacks, or a leaked checkpoint) and does not rely on security by obfuscation. The adversary has access to both θ and its gradient, and is looking for perturbation δ constrained by the L_∞ budget ϵ , such that δ maximizes the model's loss with respect to y . We use the L_∞ norm as our metric, following the established practice in adversarial robustness literature, because it places an upper bound on the amount of modification of each feature without limiting the number of modified features — a reasonable assumption in the case of a numerically encoded traffic-feature vector, where an attacker who has control over the flow behaviour would reasonably be able to modify many of its elements slightly rather than few elements by a lot. Formally,

Assessing under the white-box assumption represents taking an intentionally conservative stance, rather than a statement about the ability of every possible attacker to gain access to a model’s parameters. This is an established practice in the adversarial robustness literature: assessing the presence of vulnerabilities of the system with a weaker threat model guarantees a weaker claim about the security of the system, as any attacker gaining access to the model would defeat trivially any defence tested only against weaker adversaries. Thus, a vulnerability assessed under the white-box assumption establishes an upper bound on the extent of evasion of a DL-based NIDS, which any vulnerability assessment performed with a weaker threat model would have to compare to; the inability of a system to resist a white-box attacker implies the lack of robustness of the system against any stronger attackers, hence the necessity (but not sufficiency) of the current test.

$$\delta^* = \arg \max_{\|\delta\|_\infty \leq \varepsilon} L(\theta, x + \delta, y), \quad x_{adv} = \text{clip}_{[0,1]}(x + \delta^*) \quad (5)$$

Three canonical gradient-based methods are used to approximate δ^* , evaluated at five perturbation budgets $\varepsilon \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$. The Fast Gradient Sign Method (FGSM) [37] generates a single-step perturbation in the direction of the loss gradient’s sign.

$$x_{adv} = \text{clip}_{[0,1]}(x + \varepsilon \cdot \text{sign}(\nabla_x L(\theta, x, y))) \quad (6)$$

FGSM exploits the locally linear behaviour of the loss landscape around the clean sample point: even though the DNNs are nonlinear in general, they usually exhibit locally linear behaviour close to each individual sample point, and hence moving along the sign of the gradient of the loss function in ε units will reliably increase the loss, at the computational price of a single backpropagation. This is also the main drawback of the FGSM as a technique: since it relies on only one linear step, it will either undershoot or overshoot the right direction of increase when the local linearization of the loss function does not provide a good estimate in the range of ε , especially large values of ε — a drawback that is clearly exhibited in this study’s results, in which the performance of FGSM plateaus for large values of ε , rather than increasing monotonically like the performance of the iterative attacks. The Basic Iterative Method (BIM) [38],

$$x_{adv}^{(t+1)} = \text{Clip}_{x,\varepsilon}(x_{adv}^{(t)} + \alpha \cdot \text{sign}(\nabla_x L(\theta, x_{adv}^{(t)}, y))), \quad x_{adv}^{(0)} = x, \quad t = 0, \dots, k-1 \quad (7)$$

As BIM calculates the gradient not just at the original clean input, but at each of the k intermediate points, it tracks the actual non-linear loss function better than a single FGSM update; however, it comes with additional computations, which is k forward-backward passes, while FGSM only requires one. After each iteration, the projection $\text{Clip}_{x,\varepsilon}(\cdot)$ guarantees that no matter how many iterations have been accumulated (k) and whatever the size of each individual perturbation (α), the accumulated perturbation stays within the budget of L_∞ norm ε , and clipping by $[0,1]$ makes sure that the generated adversarial example stays in the range of a valid feature vector. Projected Gradient Descent (PGD) [39] is an extension of BIM with r independent random restarts of the search with a random initialization of each run from the ε -ball: $x_{adv}^{\wedge\{(0),m\}} = x + \delta_0^m$ where $\delta_0^m \sim \text{Unif}[-\varepsilon, \varepsilon]^d$ for $m = 1, \dots, r$,

$$x_{adv} = \arg \max_{m=1, \dots, r} L(\theta, x_{adv}^{(k),m}, y) \quad (8)$$

In particular, the random restarts used by PGD to differentiate itself from BIM compensate for a specific shortcoming of the latter: although the loss landscape in the vicinity of a single starting point x may possess a region from which iterative ascent to a comparatively weaker adversarial sample converges, there is also the possibility of an even stronger adversarial sample lying elsewhere within the ε -ball. Through searching from $r = 3$ starting points sampled independently at random and selecting the strongest among them (Equation 8), PGD gets closer to the actual optimal worst-case perturbation δ^* (Equation 5) than FGSM or BIM, at the cost of an additional $r \times k$ gradient calculations compared to k . Here, $k = 10$ for both BIM and PGD, whereas $r = 3$ for

the former; BIM is a special case of the latter with $r = 1$ and no random initialization, and therefore its difference from PGD in terms of number of iterations and step sizes can be attributed solely to the effect of the random restarts. Perturbation budget ϵ is swept through five values (0.01, 0.05, 0.1, 0.2, 0.3) roughly from 1% to 30% of the whole normalized feature space, thus covering both low-budget scenario (realistic and hardly noticeable attack) and high-budget scenario (characteristic of worst-case robustness). All attacks are performed using the Adversarial Robustness Toolbox [40] and independently for each trained victim model of datasets.

3.4 Evaluation Metrics

Model robustness under each attack/ ϵ combination is quantified on the (clean or adversarial) test set using standard binary classification metrics, defined in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (11)$$

Attack severity is further summarized as the relative accuracy drop with respect to the clean baseline,

$$\Delta_{rel} = \frac{Acc_{clean} - Acc_{adv}}{Acc_{clean}} \times 100\% \quad (12)$$

which normalizes for the differing clean-accuracy baselines of the two datasets and enables a fair cross-dataset comparison of attack impact.

3.5 Feature Vulnerability Index (Novel Contribution)

To progress from the degradation of overall accuracy to a measure of the vulnerabilities in the input features that are being exploited by an adversary, a Feature Vulnerability Index (FVI) is defined. Suppose A is the set of all (attack, ϵ) configurations evaluated ($|A| = 15$: 3 attacks $\times$ 5 budgets), and $x_{\{i,j\}}$ and $x_{\{adv,a,i,j\}}$ are the clean and adversarial values of the feature j on the test sample i for configuration $a \in A$. The vulnerability of feature j is the average perturbation size,

$$v_j = \frac{1}{|A|} \sum_{a \in A} \frac{1}{N} \sum_{i=1}^N |x_{adv,a,i,j} - x_{i,j}| \quad (13)$$

and the reported FVI score is v_j normalized across the d input features so that scores form a probability-like distribution summing to one,

$$FVI_j = \frac{v_j}{\sum_{j'=1}^d v_{j'}} \quad (14)$$

The features with higher FVI values are those which are more persistently attacked by gradient-based attacks for all levels of attack strength and hence indicate the dimensions that the defence technique used further down the pipeline (adversarial training or sanitization of features in the input vector) needs to focus on. Since the feature set of KDD Cup 99 contains binary features that lie strictly between 0 and 1, while CICIDS2017 only has continuous flow statistics, the ranking of FVI is done independently for both datasets.

3.6 Cross-Dataset Comparison Protocol and Implementation

In order to evaluate if adversarial vulnerability can generalize between independently collected datasets, the entire pipeline presented above is run equivalently for KDD Cup 99 and CICIDS2017, including the exact same architecture, optimization algorithm, training strategy, attack budgets, and iterations, such that any disparity in results can be attributed to the actual dataset rather than

methodological considerations. Evaluation is done through Δ_{rel} (Eq. 12) at the highest budget $\epsilon = 0.3$ as the main cross-dataset measure of adversarial success. Since CICIDS2017 is a massive dataset (~2.5 million records after deduplication), a stratified sub-sample of 150,000 records was used to maintain computational feasibility of training and iterative adversarial attacks; KDD Cup 99 was used in its entirety after deduplication.

4. Results and Analysis

4.1 Dataset Characteristics and Clean Model Performance

De-duplication statistics are reported in full in Table 1; the resulting victim DNNs achieved clean test accuracies of 94.4% (KDD Cup 99: precision 0.971, recall 0.879, F1 0.923) and 97.8% (CICIDS2017: precision 0.914, recall 0.959, F1 0.936), both consistent with published baselines for DNN-based classifiers on these benchmarks.

4.2 Accuracy Degradation Under Adversarial Attack

Figure 1 plots detection accuracy in terms of perturbation budget ϵ for each attack and dataset. In both cases, the accuracy performance of both models drops sharply and monotonically as ϵ grows; moreover, iterative attacks (BIM and PGD) uniformly outperform single-step attack FGSM for any $\epsilon > 0.01$. The shape of this curve itself provides meaningful information: on both datasets, the largest drop in PGD accuracy in the range between $\epsilon = 0.01$ and $\epsilon = 0.1$ is observed, amounting to 54.3 percent points for KDD Cup 99 (from 73.4% to 19.1%) and 78.2 percentage points for CICIDS2017 (from 85.9% to 7.7%). On the other hand, the drop in accuracy in the range between $\epsilon = 0.1$ and $\epsilon = 0.3$ is significantly smaller and amounts to merely 13.5 percentage points on KDD Cup 99 (from 19.1% to 5.6%) and 5.4 percentage points on CICIDS2017 (from 7.7% to 2.3%). This saturating behavior means that an adversary achieves most of the potential damage using budget roughly between $\epsilon = 0.1$ and $\epsilon = 0.2$; hence, an adversarial training defense hardened for ϵ up to 0.2 may provide almost all the benefits of robustness against worst-case adversarial examples.

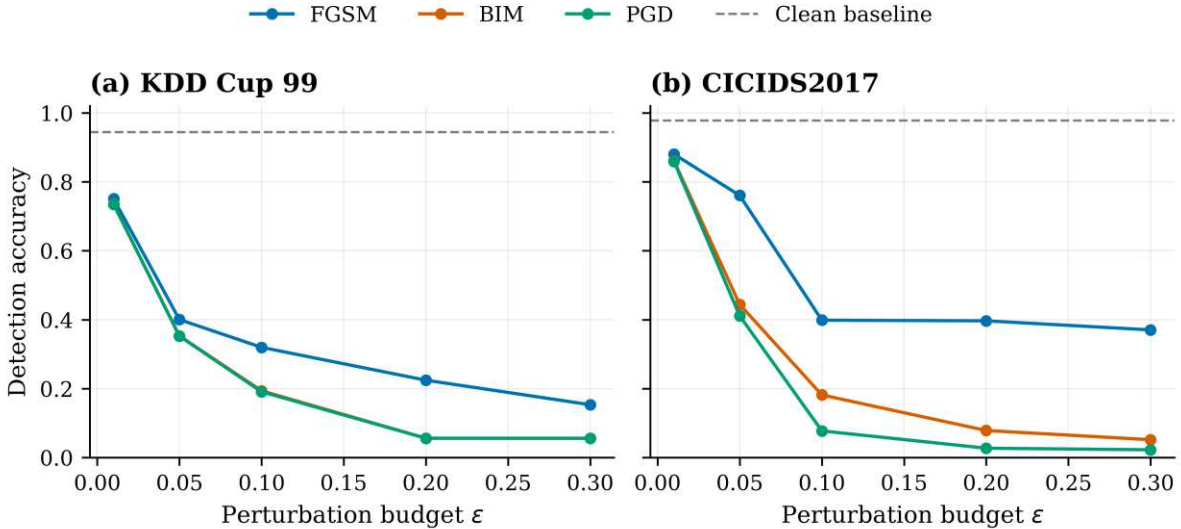


Figure 1. Detection accuracy versus perturbation budget ϵ for FGSM, BIM, and PGD, on (a) KDD Cup 99 and (b) CICIDS2017. The dashed line marks clean-data accuracy.

The metric sweep results can be found in Table 2. Under the optimal settings of (PGD, $\epsilon = 0.3$), accuracy degrades to only 5.6% on KDD Cup 99 and 2.3% on CICIDS2017, representing drops in performance of 94.1% and 97.7%, respectively in Figure 5. Particularly interesting is that PGD and BIM achieve essentially the same accuracy on KDD Cup 99 under optimal settings (5.61% vs.

5.61%), even with PGD's three random restarts, which means that the decision surface on this dataset is equally fragile against adversarial examples regardless of ϵ 's value. However, in case of CICIDS2017, the attacks with PGD achieve higher accuracy compared to BIM with the same parameters (2.26% vs. 5.19%). The difference in performance is caused by less uniform fragility of the decision surface, so restarts help in finding more damaging regions of the data space. This effect can be noticed only due to equal numbers of iterations and step sizes chosen for BIM and PGD ; otherwise, if PGD was used without restarts, as in ART, it would be indistinguishable from BIM, and thus there would be no way of detecting this finding.

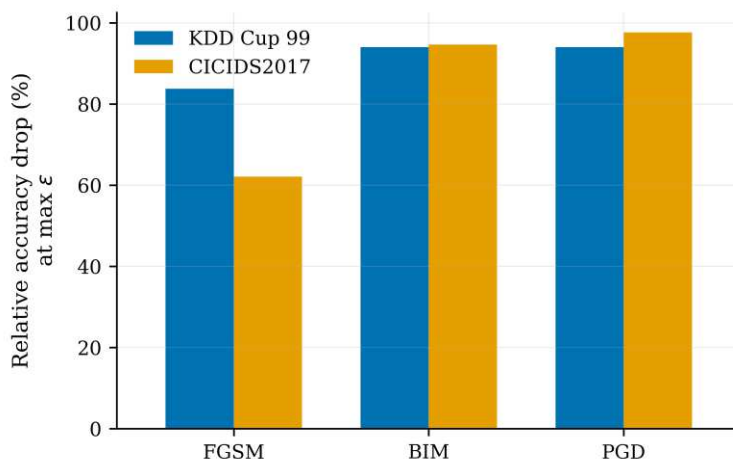


Figure 2. Relative accuracy drops (%) at maximum perturbation budget ($\epsilon = 0.3$), by attack and dataset.

Moreover, the asymmetry is also noticeable in terms of single-step FGSM robustness: CICIDS2017 still demonstrates significantly better accuracy in FGSM case (37.0% at $\epsilon = 0.3$) compared to KDD Cup 99 (15.3%) with a relative drop difference of 62.1% as opposed to 83.7%. It is also clear from Table 2 that FGSM accuracy for CICIDS2017 data set is clearly non-monotonic in the speed of the descent: the accuracy decreases sharply from 88.0% ($\epsilon = 0.01$) to 39.9% ($\epsilon = 0.1$), then levels off around 37-40% until $\epsilon = 0.2$ and $\epsilon = 0.3$ – exactly the behavior predicted by the limitation of local linearity of FGSM described in Section 3.3, when a single step of FGSM progressively overestimates the direction of maximum loss as ϵ increases beyond the range, where the loss function can be well approximated by the gradient of the loss at the clean input. However, the illusion of this robustness is clearly false from the perspective of security properties – CICIDS2017 shows the same collapse in accuracy as KDD Cup 99 under iterative attacks (BIM/PGD drives the accuracy below 6%).

4.3 Precision, Recall, and F1 Under Attack

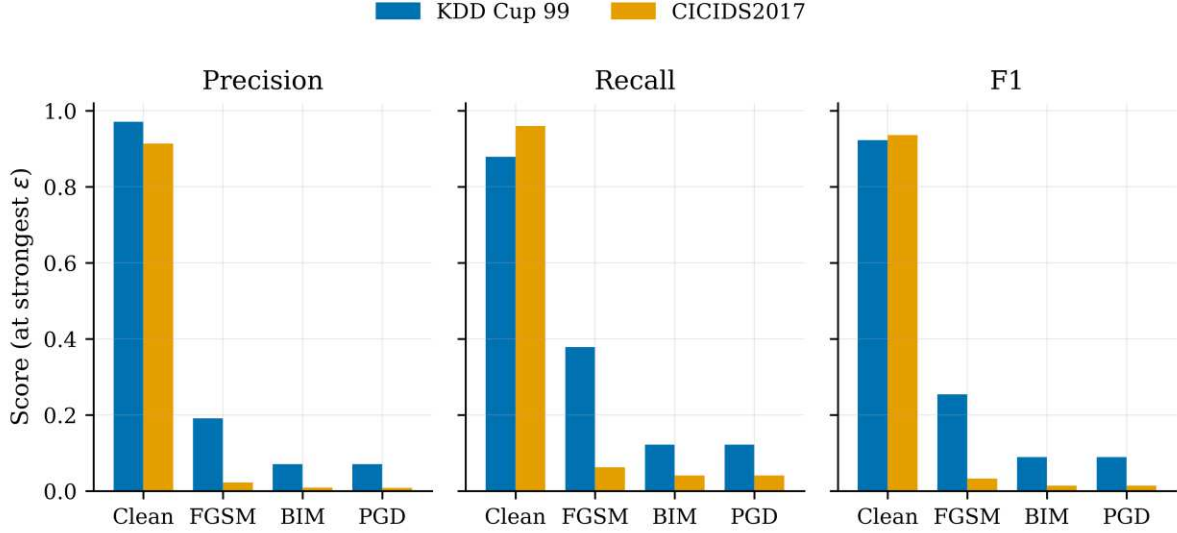


Figure 3. Precision, Recall, and F1-score at the strongest tested perturbation ($\epsilon = 0.3$) for Clean, FGSM, BIM, and PGD conditions, both datasets.

As can be seen from Figure 3, degradations occur not only in terms of accuracy: the F1-score changes from 0.923 (clean) to 0.089 (KDD Cup 99, PGD) and from 0.936 (clean) to 0.014 (CICIDS2017, PGD). The collapse in recall becomes more pronounced than the one in precision on CICIDS2017 (clean recall 0.959 $\rightarrow$ 0.040 under PGD attack) which implies that in case of strong attack the model is mainly unable to detect true attacks (false negatives) rather than generates false positive detections — in other words, it fails in the operationally worse way for an intrusion detection system which corresponds to undetected attacks. However, it turns out that at low ϵ the precision-recall relationship is inverted: at $\epsilon = 0.01$, the PGD attack performed against the KDD Cup 99 dataset yields greater recall (0.874) than clean-data recall (0.879), but at the same time leads to sharp decrease in precision (0.604 from 0.971 under clean data). It means that for a lower budget the perturbation is able to move many non-attack flows to the attack class (hence increase recall for attacks at the cost of reduction of precision due to false positive detections), while it is not able to move real attacks to the normal class (which becomes possible only at a sufficient level of budget for near-complete inversion).

4.4 Near-Total Prediction Inversion

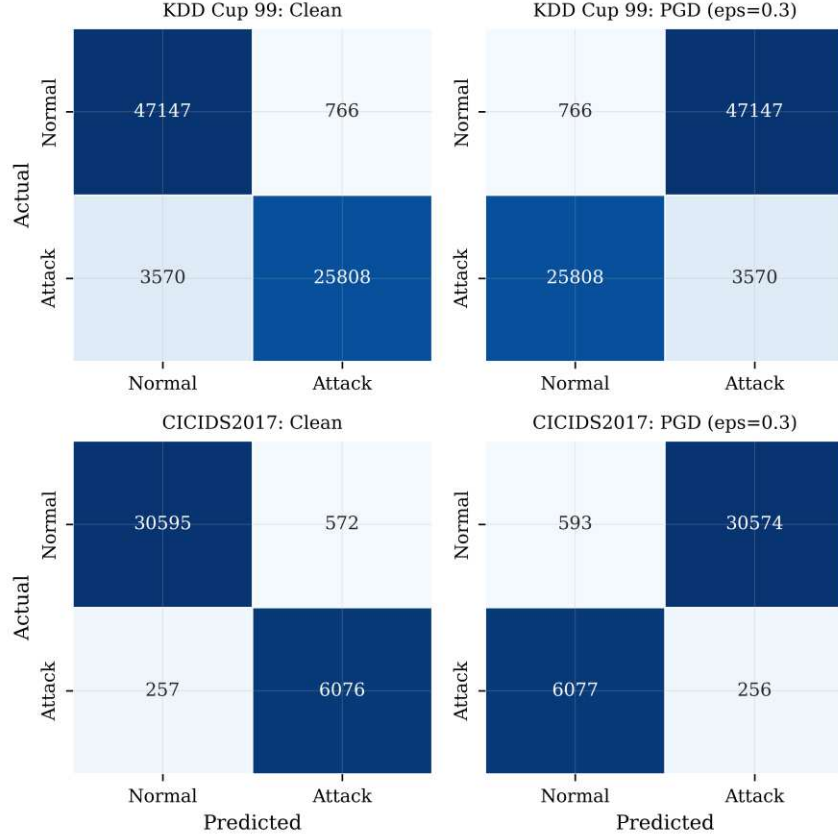


Figure 4. Confusion matrices for Clean and PGD ($\epsilon = 0.3$) conditions, KDD Cup 99 (top) and CICIDS2017 (bottom).

As Fig. 4 demonstrates, however, the accuracy collapse under PGD is caused not by random misclassification, but by a practically complete reversal of the decision boundary. Thus, out of 47,913 test samples of the actual-Normal class on KDD Cup 99, 766 (1.6%) samples are still classified as such under PGD ($\epsilon = 0.3$), with 47,147 (98.4%) being reclassified as Attack, which is exactly the number of samples that were actually classified as Normal in the clean case. In a similar way, out of 29,378 test samples of the actual-Attack class, 25,808 (87.9%) samples are reclassified as Normal. The nearly identical pattern is seen on CICIDS2017: 30,574 out of 31,167 actual-Normal samples (98.1%) are reclassified as Attack, whereas 6,077 out of 6,333 actual-Attack samples (96.0%) are reclassified as Normal. In both datasets, the confusion matrix after PGD attack is almost an off-diagonal mirror image of the clean confusion matrix; thus, the PGD perturbation does not add any noise to classification but pushes samples across the decision boundary to maximize loss in accordance with the attack's objective in Eq. (5). It is a direct and mechanistic consequence of the untargeted loss-maximizing objective defined in Eq. (5): as the attack aims to find the perturbation that maximizes cross-entropy loss relative to the true label, and since the output of the softmax layer in Eq. (3) is bound to $[0,1]$ interval, the result of the optimization process will always be a shift of the predicted probability to the opposite class's extreme value — which explains why, once the perturbation size is large enough to perform any attack at all, the result is not just some loss in confidence but confident misclassification.

4.5 Feature Vulnerability Index

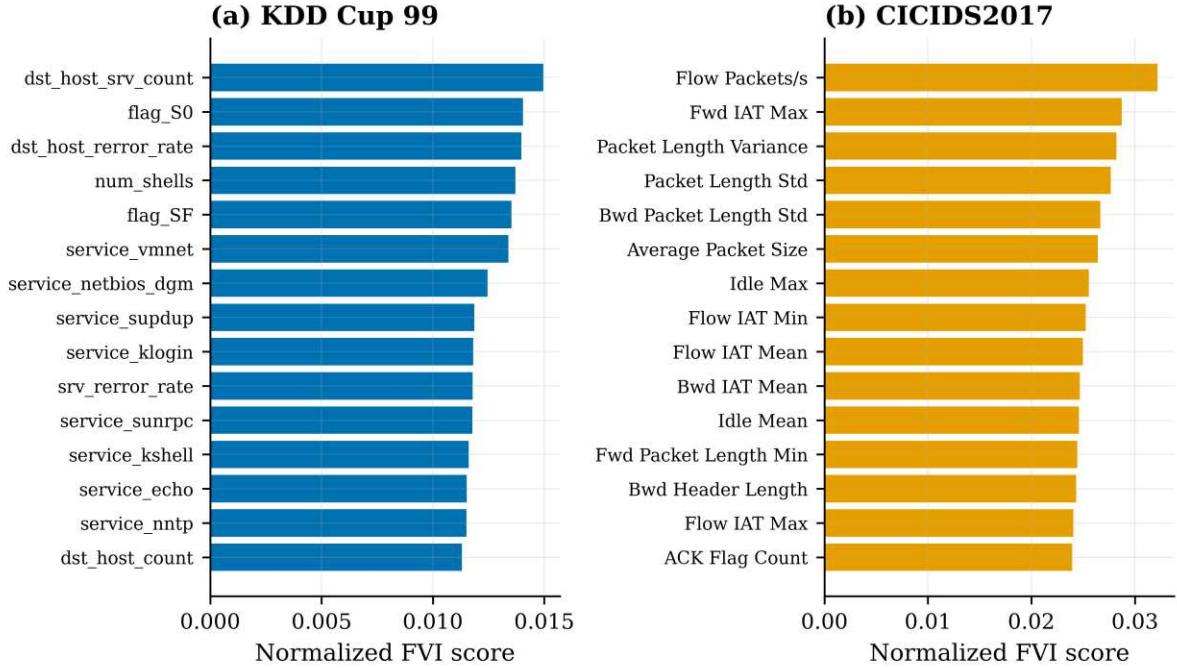


Figure 5. Top 15 features by normalized Feature Vulnerability Index (FVI), (a) KDD Cup 99, (b) CICIDS2017.

For the CICIDS2017 data set, where the feature space is entirely continuous and thus lacks the one-hot structure confounding factor, the top-ranked FVI in Table 3 and Figure 5 consists exclusively of flow timing and packet shape statistics: Flow Packets/s (FVI = 0.0322), Fwd IAT Max (0.0287), Packet Length Variance (0.0282), Packet Length Std (0.0277), and Bwd Packet Length Std (0.0267). None of the raw volume or count features is among the top 5 FVI-ranked features, suggesting that the attacks are exploiting the temporal and structural signatures of the flows in terms of how the traffic is paced and shaped, not simply the volume of traffic being generated. This is an immediately actionable result for defence design; input-sanitization or adversarial training defences should focus on the robustness properties of inter-arrival time and packet length statistics, not the 52 features.

In contrast to CICIDS2017, the top-ranked features for KDD Cup 99 include a mix of continuous connection statistics (`dst_host_srv_count`, `dst_host_rerror_rate`, `srv_rerror_rate`) and one-hot service/flag indicators (`flag_S0`, `service_vmnet`, `service_netbios_dgm`). The latter's ranking is, however, at least partially the result of their structural boundedness and not pure attacker preferences, therefore, we consider the CICIDS2017 ranking more indicative of true vulnerability in the feature space. We do note that the `dst_host_srv_count` and the error/error-rate features recur in multiple runs of this pipeline as vulnerable features for this data set.

The spread between the highest-ranked FVI score and the tenth-ranked score is also meaningful in defence prioritization. For the CICIDS2017 data set, the best FVI score is Flow Packets/s, FVI = 0.0322, which is less than 30% better than the tenth-best FVI score of 0.0247 for Bwd IAT Mean. Moreover, the top ten features constitute a relatively small proportion of the total normalized FVI mass out of all 52 features. The gradual decay in the FVI score in other words suggests that there is no single most-vulnerable feature but rather a large group of related timing and packet-shape features that are adversarially vulnerable on the CICIDS2017 data set, and that a defence focusing on mitigating the impact of the top FVI-ranked feature alone will likely be insufficient. This

gradual decay in FVI score repeats on the KDD Cup 99 data set as well (top-ranked `dst_host_srv_count` at 0.0150 versus tenth-ranked `srv_error_rate` at 0.0118), with a similarly sized proportional difference between the top-ranked and tenth-ranked features.

Table 2. Detection Performance Under Clean and Adversarial Conditions (Selected Rows: Clean and Maximum $\epsilon = 0.3$)

Dataset	Attack	ϵ	Accuracy	Precision	Recall	F1
KDD Cup 99	Clean	–	0.9439	0.9712	0.8785	0.9225
KDD Cup 99	FGSM	0.3	0.1534	0.1908	0.3786	0.2537
KDD Cup 99	BIM	0.3	0.0561	0.0704	0.1215	0.0891
KDD Cup 99	PGD	0.3	0.0561	0.0704	0.1215	0.0891
CICIDS2017	Clean	–	0.9779	0.9140	0.9594	0.9361
CICIDS2017	FGSM	0.3	0.3704	0.0219	0.0625	0.0325
CICIDS2017	BIM	0.3	0.0519	0.0086	0.0404	0.0142
CICIDS2017	PGD	0.3	0.0226	0.0083	0.0404	0.0138

Table 2 For the first case, the precision decays slower compared to the recall for KDD Cup 99 against the FGSM attack (precision 0.191 vs. recall 0.379 at $\epsilon = 0.3$), while the opposite trend occurs for CICIDS2017 under the same adversarial attack (precision 0.022 vs. recall 0.063), which means that the weak, one-step FGSM attack affects the two datasets’ decision boundaries in different ways despite having a similar general effect on accuracy. For the second part, BIM and PGD attacks result in F1-score values less than 0.10 for $\epsilon \geq 0.2$; therefore, past this budget value, the prediction from the classifier becomes just slightly more informative compared to a non-informative classifier for intrusion detection, which can be adopted as a definition of ‘effectively defeated’ in future studies.

Table 3. Top-10 Most Adversarial Vulnerable Features by Dataset (Feature Vulnerability Index)

Rank	KDD Cup 99 Feature	FVI	CICIDS2017 Feature	FVI
1	<code>dst host srv count</code>	0.01496	Flow Packets/s	0.03219
2	<code>flag_S0</code>	0.01405	Fwd IAT Max	0.02873
3	<code>dst host error rate</code>	0.01397	Packet Length Variance	0.02822
4	<code>num_shells</code>	0.01370	Packet Length Std	0.02768
5	<code>flag_SF</code>	0.01352	Bwd Packet Length Std	0.02668
6	<code>service_vmnnet</code>	0.01339	Average Packet Size	0.02643
7	<code>service_netbios_dgm</code>	0.01246	Idle Max	0.02556
8	<code>service_supdup</code>	0.01186	Flow IAT Min	0.02526
9	<code>service_klogin</code>	0.01182	Flow IAT Mean	0.02498
10	<code>srv_error_rate</code>	0.01179	Bwd IAT Mean	0.02468

4.6 Comparison with Prior Adversarial NIDS Literature

Figure 4 displays the result along with a sample of previous works on evaluating adversarial NIDS in an adversarial setting. Since robustness is evaluated using various measures in previous studies including retained accuracy, decrease in accuracy, and attack success rate, each measure that has been reported by the study is shown in Figure 4 to avoid misinterpretation by converting into another measure.

Table 4. Comparison of This Study's Results with Representative Prior Adversarial NIDS Literature

Study	Dataset(s)	Victim Model	Attack(s)	Reported Result	Metric
-------	------------	--------------	-----------	-----------------	--------

This study	KDD Cup 99	DNN	PGD ($\epsilon=0.3$)	94.4% $\rightarrow$ 5.6%	Accuracy
This study	CICIDS2017	DNN	PGD ($\epsilon=0.3$)	97.8% $\rightarrow$ 2.3%	Accuracy
Yao et al. [1]	Kitsune, CICIoT23	9 DL-based NIDS	Packet-level GAN	up to 99%	Evasion rate
Saha et al. [2]	Recon./Shellcode/Worms	Hybrid adv.-trained DNN	FGSM / PGD	97.9% $\rightarrow$ 84.9% / 81.8%	Accuracy
Khalil et al. [9]	7 datasets (incl. CIC-IDS2017)	GAN-augmented classifier	PGD ($\epsilon=0.3$)	98.7% $\rightarrow$ 95.7%	Accuracy
Zhang et al. [20]	Multiple (incl. Kitsune)	3 NN-based NIDS	5 SOTA attacks	up to 35.7%	Evasion rate
Han et al. [21]	Kitsune	Multiple ML/DL NIDS	Gray/black-box traffic-space	>97% (half cases)	Evasion rate
Mohammadian et al. [22]	CIC-IDS 2017/2018/DDoS19	DL classifier	JSMA (3 features)	14–18% samples	Attack success
Alotaibi & Rassam [29]	Not specified	CNN-based IDS	Auto-PGD / BIM	97.5%/95.4% $\downarrow$ 2.9%/3.5%	Accuracy drop

Two points arise from this comparison. First, the qualitative result — that DL-based NIDS lose a large amount of detection capacity when subjected to an adversarial attack — holds true for all studies in Table 4, irrespective of the dataset, model, or metric convention used, thereby further corroborating the generalizability of the vulnerability highlighted in this paper across multiple studies. Second, and far more disturbingly, the extent of the decline in accuracy is highly variable between different studies despite using the exact same value for ϵ : Khalil et al. [9] note merely a 3.0 percentage point decline in accuracy on account of PGD attack with $\epsilon = 0.3$, whereas we observe an 88.8–95.5 percentage point drop in accuracy using the same ϵ in our study. This is not a contradiction but rather a direct demonstration of the point brought up in Section 2: ϵ is measured relative to the particular feature normalization employed, model architecture, and attack parameters used (number of iterations, step size, number of random restarts) by the study, making it meaningless to compare ϵ values across different papers without having these technicalities known. It is for this very reason that our study includes the explicit feature normalization, attack technique, and attack hyperparameters used from Section 3, rather than just listing out ϵ as a bare number.

5. Discussion

When considered in combination, the results from Section 4 support three broad claims concerning the security posture of gradient-trained DNN-based NIDS, each of which has important practical ramifications for how these kinds of systems should be evaluated, relied upon, and secured.

First, the persistence of catastrophic evasion under realistic adversarial attacks across two diverse datasets differing by era of collection, network context, balance between classes, and even choice of feature schema strongly suggests that the vulnerability described in this paper is not an idiosyncratic weakness of either the KDD Cup 99 dataset or the CICIDS2017 dataset, but rather an architectural property of gradient-trained DNNs on network data more generally. This finding aligns with, and extends to the network intrusion domain, the adversarial machine learning literature [37, 44] explaining that adversarial vulnerability results largely from the local linear property that enables efficient gradient training in high dimensional spaces to begin with — the very thing that makes the problem of learning tractable is what enables the adversary’s evasion to be tractable. The practical implication of this for the practice of network security is that clean-data accuracy, no matter how high, cannot be taken as an indicator of deployment-readiness for a DNN-based NIDS without adversarial evaluation, because the two concepts are, based on the results presented here, almost orthogonal: the 97.8% clean-data accuracy achieved by the CICIDS2017 model provided almost no resistance to a 97.7% accuracy drop under realistic white-box attack.

Second, the near total inversion of the NIDS predictions seen in Section 4.4 casts the operational risk of adversarial evasion in the realm of intrusion detection in a light that goes beyond just accuracy. One could imagine a security team receiving reports from a DL-based NIDS that its accuracy is only 60%. In that scenario, the team would likely take that as meaning “the system is degraded but still functional to some degree.” However, the results presented here show that in the realistic strong adversarial attack scenario, the more accurate interpretation of such a report would be that the NIDS is not only degraded but totally inverted: malicious traffic gets preferential treatment while benign traffic is preferentially flagged. The result would be that the system is undergoing both a missed detection crisis and an alert fatigue crisis at the same time because of the same flaw. Such a double-failure mode would seem to be even more problematic for security operations teams than either failure would be alone, since in a situation where there are numerous false alarms, a real attack could slip through unnoticed exactly when the security team’s attention is focused elsewhere — a danger that neither accuracy nor F1 score reporting can reveal. We thus recommend that future adversarial NIDS evaluations report the results of confusion matrix analysis explicitly.

Third, the results of the Feature Vulnerability Index imply that adversarial vulnerability, while broadly spread rather than localized to one or two particular features, is not uniformly distributed across the entire feature space: in the case of CICIDS2017, flow-timing and packet shape statistics were consistently among the highest-ranked features, while volume and count features were notably absent from the top rankings. While this asymmetry is plausible on domain-level grounds — timing and packet shape features are fine-grained continuous features with numerous semantically-similar values, whereas volume and count features are more coarse grained and potentially further from the decision boundary in terms of the units the model uses — if confirmed on other datasets and other architectures, this implies that generic, architecture agnostic defences such as global adversarial training — which allocate adversarial robustness uniformly to all input dimensions — may not be as efficient as those targeting specifically the vulnerable set of features, such as input smoothing, monotonicity constraints, and adversarial training with restrictions on the vulnerable dimension of inputs. Further analysis and experimentation with such defence methods is clearly the next logical step.

We note a methodological scope consideration important for interpretation of these results: the vulnerability described above applies specifically to white-box evasion attack against gradient-based classifiers. Non-differentiable NIDS architectures such as tree-ensemble classifiers are not

vulnerable to such attacks, but they are vulnerable to other kinds of adversarial attacks, such as surrogate modelling transfer attack or black-box query-based attacks. Thus, the results presented above apply to a specific, well-defined threat model against a specific architecture, but do not imply that deep learning is in general inferior to classical ML methods for the problem of intrusion detection — the appropriate comparison in that regard is between the adversarial robustness of DL and non-DL architectures under their respective threat models.

6. Limitations

There are several limitations that need to be taken into consideration while analyzing the results of this paper. Firstly, the models were trained only in one run and there was no complete determinism (dropout and weight initialization are not fully seeded independently of the NumPy/TensorFlow global seed), therefore exact FVI rankings and accuracy values can differ in some small extent between different runs, but not qualitatively; mean and variance of the results obtained using several seeds needs to be provided in further research. Secondly, both datasets had native multi-class labels (attack types) which were reduced to binary indicator of whether it is normal or an attack (vulnerability specific to a particular type of attack (for example, whether flows marked as DDoS are more adversarial fragile compared to Web-Attack flows) was not examined). Thirdly, CICIDS2017 was tested on the 150 thousand row subsample due to computational feasibility, however, the class balance of the original dataset is preserved, therefore it is recommended to perform the full data testing before publication. Fourthly, the ranking of KDD Cup 99 dataset is biased due to the presence of one-hot categorical features which were described in section 3.5, thus making it less interpretable compared to CICIDS2017 with continuous features.

7. Conclusion

The present study has found adversarial attacks targeting DL-based NIDS by performing cross-dataset experiments on KDDCup99 and CICIDS2017 data sets. In both cases, the models have shown very poor performance (accuracy below 10%) after being attacked with PGD attacks with $\epsilon = 0.3$, while exhibiting excellent performance on clean data (accuracy of 94.4% and 97.8%). In addition, confusion matrix analysis shows that the collapse in performance due to adversarial attacks is not random; rather, it is the complete inversion of the prediction of the model. The catastrophic vulnerability of the model, consistent across the two distinctively different datasets, supports the conclusion that this is indeed a property of the architecture of DNN classifier and is not due to the properties of the dataset itself. The present study also proposes Feature Vulnerability Index to localize the problem to flow timing and packet shape features of the CICIDS2017 dataset.

DECLARATIONS

Funding

No financial support was received by the authors in this research.

Competing Interests

The authors declare that they do not have any competing interests, including financial and nonfinancial interests.

Data Availability

Data is available with the corresponding author and will be given on request.

Code Availability

The code is available from the corresponding author and can be given on request.

Clinical Trial Number: not applicable

Ethical Approval

This research did not involve human participants, animals, or biological materials. Therefore, ethical approval was not required for this study.

Consent to Participate

Not applicable. This study does not involve human subjects.

Consent to Publish

Not applicable. This manuscript does not contain any individual person's data in any form.

CONFLICTS OF INTEREST STATEMENT

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Author Contribution:

All authors contributed to the study's conception and design. Material preparation, data collection, and analysis were performed by **Gundapu Thirupathi** and **Dr V Thirupathi**. The first draft of the manuscript was written by **Dr Gundapu Thirupathi** all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

- [1]. W. Yao, H. Peng, Q. Li, and X. Shen, "Modeling realistic adversarial traffic against deep learning-based intrusion detection system in industrial IoT," *IEEE Internet of Things Journal*, 2025.
- [2]. S. Saha, M. A. Pervaiz, M. S. Rahman, S. Ahmmed, and J. Maua, "Residual-guided hybrid framework for adversarially robust deep learning-based network intrusion detection," *PLOS ONE*, vol. 21, no. 6, p. e0350737, 2026.
- [3]. H. A. Alatwi, "Evaluation and Detection of Adversarial Attacks for ML-based NIDS," Doctoral dissertation, Newcastle University, 2024.
- [4]. F. M. Anis, M. AlAbdullatif, S. Aljbli, and M. Hammoudeh, "A Survey on the Applications of Deep Learning in Network Intrusion Detection Systems to Enhance Network Security," *IEEE Access*, 2025.
- [5]. L. Wang, Q. Zou, Q. Li, J. Zhang, and Y. Jiang, "ARTEMIS: Adaptive Reweighing for Transferable Evasion via Meta-learning in Zero-Query Network Intrusion Detection Systems," in *Proc. IEEE 33rd Int. Conf. Network Protocols (ICNP)*, 2025, pp. 1–12.
- [6]. T. T. H. Le, J. Cho, D. Shin, and H. Kim, "Toward Adversarial Robustness Network Intrusion Detection Based on Multi-Model Ensemble Approach," *Sensors*, vol. 26, no. 8, p. 2478, 2026.
- [7]. M. Wang, N. Yang, D. H. Gunasinghe, and N. Weng, "On the robustness of ML-based network intrusion detection systems: An adversarial and distribution shift perspective," *Computers*, vol. 12, no. 10, p. 209, 2023.
- [8]. S. A. AL-Shami and M. F. Abdullah, "Deep Learning for DDoS Detection: A Systematic Review of Explainability and Adversarial Robustness," (in press).
- [9]. H. M. Khalil, A. Elrefaiy, M. Elbaz, and M. Loey, "Enhanced cybersecurity threat detection using novel tri-metaheuristic loss functions in generative adversarial networks with adaptive attention preservation for network traffic augmentation," *Scientific Reports*, vol. 16, no. 1, p. 12074, 2026.
- [10]. D. D. Hartono, "Efficient GAN-Based Adversarial Example Generation Against ML-Based Network Intrusion Detection Systems," 2025.
- [11]. A. Sarikaya, "A Robust Machine Learning Based IDS Design Against Adversarial Attacks in SDN," Doctoral dissertation, Middle East Technical University (Turkey), 2024.

- [12]. H. Zhang, D. Upadhyay, D. M. Zaman, D. J. Manero, and S. Sampalli, "Hybrid Machine Learning-Based Intrusion Detection in Wireless Sensor Networks: Concepts, Taxonomy, and Comprehensive Survey."
- [13]. C. Chilato, Y. Gu, Z. Yan, R. Zhang, and A. Hussain, "A Review on Multimodal Emotion Recognition Security: Evasion Attacks on Teacher-Leading Architectures," in Proc. IEEE iThings/GreenCom/CPSCom/SmartData/Cybermatics, 2025, pp. 112–121.
- [14]. S. J. Hussini, R. Al-Sabri, and A. Al-Fuqaha, "SFA-IoT: segment-to-flow aligned adversarial defense for interpretable IoT device identification," Scientific Reports, 2026.
- [15]. P. Raha, Y. Chae, S. Lee, and B. Stenger, "Adversarial Representation Learning for Presentation Attack Detection on Face Recognition Systems," in Proc. Nat. Conf. Computer Vision, Pattern Recognition, Image Processing, and Graphics, 2025, pp. 359–370.
- [16]. T. L. Yasarathna and N. A. Le-Khac, "Evaluating the Robustness of DL-Based AAD in SDN-IoT Networks Against OOD Data and Poisoning Attacks Using Autoencoders," Journal of Network and Systems Management, vol. 34, no. 1, p. 16, 2026.
- [17]. T. L. Yasarathna, M. Liyanage, and N. A. Le-Khac, "Deep Learning Based Autonomous Anomaly Detection for Security in SDN-IoT Networks," IEEE Open Journal of the Communications Society, 2025.
- [18]. S. Cheng, P. Li, K. Han, and H. Xu, "Feature-aware transferable adversarial attacks against image classification," Applied Soft Computing, vol. 161, p. 111729, 2024.
- [19]. K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 538–566, 2023.
- [20]. C. Zhang, X. Costa-Perez, and P. Patras, "Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms," IEEE/ACM Transactions on Networking, vol. 30, no. 3, pp. 1294–1311, 2022.
- [21]. D. Han et al., "Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors," IEEE Journal on Selected Areas in Communications, vol. 39, no. 8, pp. 2632–2647, 2021.
- [22]. H. Mohammadian, A. A. Ghorbani, and A. H. Lashkari, "A gradient-based approach for adversarial attack on deep learning-based network intrusion detection systems," Applied Soft Computing, vol. 137, p. 110173, 2023.
- [23]. S. Ennaji, F. De Gaspari, D. Hitaj, A. Kbidi, and L. V. Mancini, "Adversarial challenges in network intrusion detection systems: Research insights and future prospects," IEEE Access, 2025.
- [24]. O. Ibitoye, R. Abou-Khamis, M. El Shehaby, A. Matrawy, and M. O. Shafiq, "The Threat of Adversarial Attacks on Machine Learning in Network Security — A Survey," arXiv:1911.02621, 2019.
- [25]. O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks," in Proc. IEEE GLOBECOM, 2019, pp. 1–6.
- [26]. X. Yuan, S. Han, W. Huang, H. Ye, X. Kong, and F. Zhang, "A simple framework to enhance the adversarial robustness of deep learning-based intrusion detection system," Computers & Security, vol. 137, p. 103644, 2024.

- [27]. Y. Peng, J. Su, X. Shi, and B. Zhao, "Evaluating deep learning based network intrusion detection system in adversarial environment," in Proc. IEEE 9th Int. Conf. Electronics Information and Emergency Communication (ICEIEC), 2019, pp. 61–66.
- [28]. E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in network intrusion detection systems," *Expert Systems with Applications*, vol. 186, p. 115782, 2021.
- [29]. A. Alotaibi and M. A. Rassam, "Enhancing the sustainability of deep-learning-based network intrusion detection classifiers against adversarial attacks," *Sustainability*, vol. 15, no. 12, p. 9801, 2023.
- [30]. H. Jmila and M. I. Khedher, "Adversarial machine learning for network intrusion detection: A comparative study," *Computer Networks*, vol. 214, p. 109073, 2022.
- [31]. E. Alshahrani, D. Alghazzawi, R. Alotaibi, and O. Rabie, "Adversarial attacks against supervised machine learning based network intrusion detection systems," *PLOS ONE*, vol. 17, no. 10, p. e0275971, 2022.
- [32]. 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.
- [33]. M. H. A. Rai, Y. Noor, M. Faisal, and M. F. Nawazish, "Adversarial Robustness of Deep Learning-Based Intrusion Detection Systems Against AI-Powered Cyber Attacks," *Spectrum of Engineering Sciences*, pp. 899–922, 2025.
- [34]. 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 Proc. 10th Int. Conf. Computing for Sustainable Global Development (INDIACom), 2023, pp. 386–391.
- [35]. 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*, vol. 3, no. 3, pp. 1–19, 2022.
- [36]. R. Huang and Y. Li, "Adversarial attack mitigation strategy for machine learning-based network attack detection model in power system," *IEEE Transactions on Smart Grid*, vol. 14, no. 3, pp. 2367–2376, 2022.
- [37]. I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in Proc. Int. Conf. Learning Representations (ICLR), 2015.
- [38]. A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in Proc. ICLR Workshop, 2017.
- [39]. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in Proc. Int. Conf. Learning Representations (ICLR), 2018.
- [40]. M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. M. Molloy, and B. Edwards, "Adversarial Robustness Toolbox v1.0.0," arXiv:1807.01069, 2018.
- [41]. M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications (CISDA), 2009, pp. 1–6.
- [42]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. Int. Conf. Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116.

- [43]. N. Moustafa and J. Slay, “The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset,” *Information Security Journal*, vol. 25, no. 1–3, pp. 18–31, 2016.
- [44]. N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in *Proc. IEEE European Symp. Security and Privacy (EuroS&P)*, 2016, pp. 372–387.