Version of Record: https://www.sciencedirect.com/science/article/pii/S0167404823000275
Manuscript_eebd6c9bddccb22a516ce8325e680d7a

Detecting DDoS Attacks Using Adversarial Neural Network

Ali Mustapha[1], Rida Khatoun[1], Sherali Zeadally[2], Fadlallah Chbib[1], Ahmad Fadlallah [1], Walid Fahs[3], Ali El Attar[1]
Institut Polytechnique de Paris, Telecom Paris (INFRES), LTCI, France[1]
College of Communication and Information, University of Kentucky, Lexington, Kentucky, USA[2]
Faculty of Engineering, IUL, Lebanon[3]
(ali.mustapha, rida.khatoun, fadlallah.chbib, ali.elattar)@telecom-paris.fr, szeadally@uky.edu,
a.fadlallah@usal.edu.lb, walid.fahs@iul.edu.lb

***Abstract*—In a Distributed Denial of Service (DDoS) attack, a network of compromised devices is used to overwhelm a target with a flood of requests, making it unable to serve legitimate requests. The detection of these attacks is a challenging issue in cybersecurity, which has been addressed using Machine Learning (ML) and Deep Learning (DL) algorithms. Although ML/DL can improve the detection accuracy, but they can still be evaded - ironically - through the use of ML/DL techniques in the generation of the attack traffic. In particular, Generative Adversarial Networks (GAN) have proven their efficiency in mimicking legitimate data. We address the above aspects of ML/DL-based DDoS detection and anti-detection techniques. First, we propose a DDoS detection method based on the Long Short-Term Memory (LSTM) model, which is a type of Recurrent Neural Networks (RNNs) capable of learning long-term dependencies. The detection scheme yields a high accuracy level in detecting DDoS attacks. Second, we tested the same technique against different types of adversarial DDoS attacks generated using GAN. The results show the inefficiency of the LSTM-based detection scheme. Finally, we demonstrate how to enhance this scheme to detect adversarial DDoS attacks. Our experimental results show that our detection model is efficient and accurate in identifying GAN-generated adversarial DDoS traffic with a detection ratio ranging between 91.75% and 100%.**

***Keywords*—** Distributed Denial of Service (DDoS), Long Short Term Memory (LSTM), Generative Adversarial Networks (GANs), Intrusion Detection System (IDS), Machine Learning (ML)

I. Introduction

According to Nexusguard [1], in the first half of 2022, the total attack count increased by 75.60% compared to the second half of 2021. In its report [2], Amazon showed that in Q1 of 2020, the DDoS attack rate reached 2.3 Tbps. The costs associated with these attacks are also increased. Spamhaus has recently published its report called "Spamhaus Botnet Threat Update" and showed statistics about botnets and their sources where it is illustrated that most botnets originated from China, with over 590,000 bots, the US with 376,000 bots, and India, with 350,000 bots [3]. DDoS attack's average time is less than 10 minutes according to the Carero report [4]. DDoS attacks can cost enterprise organizations $50,000 in lost revenue from downtime and mitigation costs. As attack vectors, 71%, SYN floods, and DNS attacks remain the most popular DDoS attack vectors in Q3 in 2022 according to Cloudflare. HTTP DDoS attacks which aim to disrupt a web server, are also emerging attacks that can perturb servers' services as other DDoS attacks. As solutions against DDoS attacks, Internet Service Providers (ISP) and companies are using a lot of solutions and Scrubbing Centers such as Radware DefensePro, Radware Cloud DDoS Protection Service, Cloudflare DDoS Mitigation Services, Akamai Edge DNS, Arbor Cloud, F5 Silverline DDoS Protection, Nexusguard DDoS Mitigation Services, Oracle Dyn DDoS Protection, Azure DDoS Protection, etc. All these solutions offer features such as multi-layered protection, real-time threat detection, reporting, and analytics. However, actual DDoS attacks are faster with unprecedented rates, and more sophisticated. Botnets are more decentralized and very well-secured. Using intelligent scrubbing centers is a new direction to enhance center's automation and precision. Hence, machine learning-based scrubbing centers are considered Next Generation Scrubbing Centers (NGSC). Internet-based services can have different security requirements such as confidentiality, integrity, and availability. The latter, in particular, can be of high/critical importance for certain services. Attackers mainly target the availability of service through Denial of Service (DoS) attacks. A DoS attack occurs when legitimate users cannot access the systems and resources they need because of the malicious actions of cyber attackers [5]. A DoS attack can be launched in a distributed manner, in a so-called Distributed DoS (DDoS) attack, when multiple machines are operating together to attack a target (Fig. 1). DDoS attacks commonly flood the victim with a huge number of requests/packets to saturate its resources, so that it can no longer satisfy the requests of legitimate users.

DDoS can be detected and mitigated using Intrusion Detection Systems (IDS), which are designed to detect traffic anomalies associated with the attack strategy and implementation. While being efficient against "traditional" DDoS attacks, "traditional" IDS's fail to cope with complex DDoS attacks today. This IDSs commonly follow a signature-based approach which makes them unable to self-learn and thus take the necessary actions unless they are configured for the appropriate rules/patterns and the actions associated. To address such limitations, IDS techniques are being enhanced with Machine Learning (ML) or Deep Learning (DL) algorithms [6, 7, 8]. This is currently getting a lot of attention in the DDoS detection research field [9, 10, 11, 12].

The ML-based IDS could identify and defend against known DDoS indicative patterns. An ML-based intrusion detection system is made up of a feature extractor and a Machine Learning model that serves as the detection engine. The feature encoder organizes raw network data to extract features suitable for model inputs. The detection engine is trained using both DDoS and benign data to be able to categorize samples in real-time traffic. Numerous ML-based IDS [6, 7, 8, 13, 14] have already been proposed in the literature demonstrating high detection accuracy for DDoS attacks. However, many deep learning and machine learning algorithms are learned based on a single dataset, where the training and the testing sets are drawn from the same source. Therefore, if the input data originate from an external source where there is a small change in the input feature space, the performance of this type of algorithm degrades as a result [15]. Malicious actors might use this generalization problem to lead the classifiers to reach incorrect decisions. Attackers

© 2023 published by Elsevier. This manuscript is made available under the Elsevier user license
https://www.elsevier.com/open-access/userlicense/1.0/

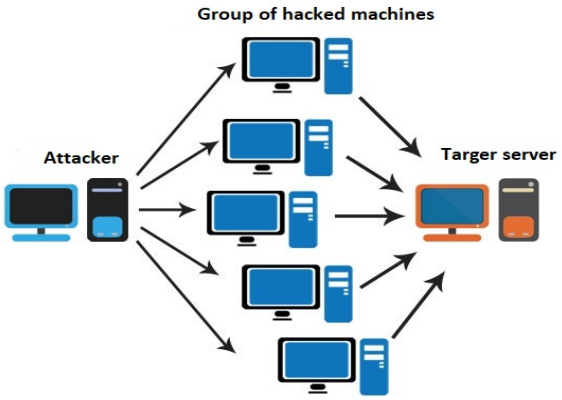

Fig. 1: DDoS attack

might adapt their attacks to prevent signature-based detection, which makes ML-based IDS vulnerable to adversarial attacks. This is referred to as Adversarial Machine Learning (AML), where the Generative Adversarial Networks (GAN) [16] are utilized to try to fool ML-based detectors by providing fake features as inputs for the model. GAN is a deep learning paradigm introduced in 2014, that can learn the pattern of the original dataset in order to create novel similar data. A GAN is composed of two neural network models: the generator and the discriminator, which have been trained to act against each other. Adversarial attacks (generated by GAN) can be further "enhanced" through perturbation. Adversarial perturbations are small, unnoticeable but targeted adjustments to the input of the ML model, resulting in incorrect behavior. They have proven to be highly effective in "fooling" ML-based classification algorithms [17].

In this paper, we present a GAN-based framework that can provide strong DDoS adversarial samples. These samples are then modified (i.e., perturbed) by replacing their features with values from the benign samples. We evaluate the perturbation technique against a high-precision IDS and observe a significant decrease in the IDS's detection accuracy. Thus, we propose to use GAN to improve the IDS precision, by training a new model based on the generated adversarial samples in order to be able to detect them later. The proposed IDS consists of two models: the first model is responsible for blocking the adversarial samples, and the second one distinguishes between the DDoS and the benign instances.

We organize the rest of the paper as follows. Section II presents an overview of related works on ML-based IDS and adversarial attacks against IDS. Section III presents the attack model. Section IV describes the experiments and results of the generation of the adversarial attack, and the perturbation of the fake-DDoS samples. We discuss the improvement of the IDS performance in Section V. Section VI describes the implementation of the IDS in the internet service providers (ISP). And section. Finally, section VII concludes the paper and presents our future work.

II. Related Work and Research Contributions

As we have previously mentioned, the use of machine learning and deep learning has attracted a lot of attention in the DDoS detection field. This section shed the light on the most known ML/DL-based DDoS detection schemes and emphasizes the main contributions of this work in comparison with the existing solutions.

A. Related Work

Over the last decade, several research efforts [10, 11, 12, 14, 18, 19, 20] have conducted research on ML-based IDS with high accuracy. In [13], the authors proposed a method that is an ensemble of feature selection utilizing information gained using the CIC-IDS2017 dataset [21]. According to the results, this ensemble method for the Friday morning dataset has an accuracy of 97.86%. But the prediction accuracy for the Friday afternoon log file is 73.79% for 16 features when utilizing an information gain-based feature selection and regression analysis-based ML model. Their approach's main disadvantage is its high computational complexity.
The authors of [14] examined many machine learning methods (Extreme Gradient Boosting (XGBoost), K-Nearest Neighbor (KNN), and a Convolutional Neural Network (CNN) deep learning architecture to identify and categorize DDoS attacks using the CIC-DDoS2019 dataset[22]. The results demonstrate that XGBoost obtains the maximum accuracy of 89.29%, while CNN and KNN also provide comparable results. The dataset used in this study was unbalanced, with a low percentage of normal data compared to attack data, which can lead to misdetection in some real-time scenarios with a variant of normal data.
To detect different types of attacks such as HTTP Flood attack, Smurf attack, and UDP Flood attack, the authors of [19] used machine learning algorithms such as K-Nearest-Neighbors (KNN), Naive Bayes (NB), Decision Tree (DT), Artificial Neural Network (ANN), and Support Vector Machine (SVM). They found that the ANN surpasses the rest of the techniques by 98.645% accuracy. The limitation is that all the classifiers could not detect the Smurf class of DoS attacks.
In [10], the authors proposed an IDS based on LSTM for detecting DoS attacks. They evaluated the proposed framework using the CICIDS-2017 [21], and NSL-KDS [23] datasets. The results obtained show that the LSTM can effectively detect DoS attacks with an accuracy of 99.2% with CICIDS-2017 and 98.6% with NSL-KDD. Their framework was tested and evaluated only on DOS attacks.
The authors of [8] combined the CIC-DDoS2019 [22] dataset with the generated DDoS data using BoNesi and SlowHTTPTest simulators. They evaluated the performance of the proposed DDoS detectors which are the convolution model and the LSTM-based model. The authors discussed the suitability of these models in the IoT network. The results show that the identification accuracy of the proposed LSTM is as high as 98.9%. Then, when they implemented the LSTM model in edge servers with computational capacities greater than those of personal computers, they discovered that the models satisfy the IoT's delay requirements. Despite the use LSTM model, they didn't investigate the vulnerabilities of ML-based IDS against adversarial attacks.
In [11], the authors propose an intrusion detection system based on deep learning algorithms. They evaluated various models such as Deep Neural Network (DNN), Convolutional Neural Networks (CNN), and Long Short Term Memory (LSTM) using the CIC-DDoS2019 [22] dataset. The results obtained show that the CNN model gave the best results among the proposed models. The authors obtained an accuracy of 99.99% for binary classification (normal/abnormal classes) and 99.30% for multi-class classification. The limitation here is that the model lacks generalization because of the use of some features with uneven distribution.

Section I has highlighted, the vulnerability of ML-based IDS against the adversarial perturbation has been explored and evaluated in [24, 25, 26, 27].
In [25], the authors investigated adversarial attacks targeting an anomaly-based IDS in the Software Defined Networks (SDN) environment. Payload size, packet rate, and volume of bidirectional traffic are the three features that the authors focused on perturbing. The attack was performed in SYN flood records that were classi-

fied by various ML models. The obtained results demonstrate the effectiveness of such an attack strategy by decreasing the detection accuracy of the targeted IDS from 100% to 0%. This paper focuses on generating new attacks without investigating a detection model for them.

In [24], the authors split the features in the KDDCup99 dataset [28] into unmodifiable ones that are required to keep the malicious function running and modifiable ones that could be modified without disrupting the malicious function. They used a Wasserstein GAN (WGAN) [29] based neural network to control the DDoS attack modifiable features. According to the reported results, the IDS's detection accuracy decreased by 50% which shows that the ML-based IDS are vulnerable to this type of attack.

To avoid detection, the authors of [26] devised a framework that uses GAN to build adversarial malware. The purpose of this work was to employ a black-box malware detector because attackers are ignorant of the detection techniques utilized in the detectors. Rather than directly assaulting the white-box detector, the researchers developed a model that can collect data from the black-box targeted system. Then, this model uses the gradient computation from the GAN to generate perturbed malware data. The authors were able to achieve a model accuracy of roughly 98% using the Drebin dataset [30]. The limitation is that the authors didn't investigate the stability of the process of training the GAN model.

The authors of [27] proposed a solution for avoiding malicious PDF file detection. The attacking approach can produce feature vectors similar to those produced by WGAN for benign PDF files. Allowing the features of the malicious PDF file to match the features of the benign PDF files, to avoid classification by the detector. The authors evaluated the performance of the proposed approach using the Contagio dataset[31]. The outcomes results demonstrate that the adversarial samples created by this method reach a 100% avoidance rate. This strategy is limited to PDF-specific formats.

The authors of [32] proposed a defense strategy based on ML ensemble models and adversarial training. The network traffic is only deemed normal if all models agree on this classification, forcing the attacker to develop samples that can bypass all models at the same time. Additionally, the authors used adversarial training to add adversarial samples to the training dataset, enhancing the robustness of individual models. The authors evaluated the defense strategy on the CICIDS2018 [33] dataset and found that it successfully lowers the success rate of adversarial attacks. Using the ensemble learning technique is high computational complexity.

The authors of [34] proposed a strategy to enhance the IDS against adversarial DDoS attacks. The authors trained two LUCID to citeLUCID models to detect SYN flood and HTTP GET flood attacks. They found that these high-performance models are vulnerable to adversarial perturbed DDoS samples. They proposed a defense technique using the GAN model. Then they used the generated perturbed data to train the LUCID model on it. As a result, they found that the models attained an F1 score of more than 98% and that the False Negative rate decreased to less than 1.8% on perturbed DDoS traffic. The limitation of their work is that the generated dataset used to enhance the LUCID model is based on a single perturbation technique by modifying a limited number of features.

Table III summarizes the main ideas, results, and limitations of the ML-based DDoS detection schemes described above.

B. Literature gap and research contributions

As we have previously explained, ML- and DL-based classification models (e.g., [10, 11, 13, 14, 19] perform poorly when the input feature space changes. This makes ML- or DL-based IDS vulnerable to cyber-attacks generated using other ML/DL models such as GANs with some feature modifications as works such as [24, 25, 26, 27] have demonstrated.

Recently, some research efforts [24, 25, 26, 27] have focused on generating adversarial attacks using GAN and investigated if these attacks can be detected by the IDS. It was discovered that most of these efforts do not prioritize training the IDS with adversarial data produced by the GAN and testing whether the IDS can identify the same types of attacks. Furthermore, the use of samples produced based on the victim model is a common drawback of adversarial training approaches [32, 34], where they used the generated dataset which is based on a single perturbation technique, or that is based on a single victim model. As a result, the attack model develops the ability to produce weak adversarial samples, and rather than training the IDS to protect against significant perturbation, it trains from weak attack samples, and it remains vulnerable to adversarial attacks. To address this major weakness, in this paper, we develop a new robust detection method in the IDS to detect DDoS samples accurately, regardless if the attack features have been replaced (i.e., perturbed) or not.

We summarize the main research contributions of this paper as follows:

- We developed a GAN model generator capable of creating DDoS traffic that closely matches the DDoS instances from the dataset. We modified the values present in the DDoS-functional features in the generated DDoS traffic to make them look similar to the benign instances.
- We built a new dataset based on the combination of the generated and the original dataset, with two classes: real and fake.
- We trained a new model using the new dataset, to be able to detect the fake or generated data.
- We trained another model using the original dataset including only the DDoS's functional features, to be able to distinguish between DDoS and normal samples.

The proposed IDS in this research could be used by organizations to protect their networks from DDoS attacks. For example, it could be deployed by a company to protect its website from being taken down by a DDoS attack or by a government agency to protect its critical infrastructure from being disrupted by such an attack. The ability to detect adversarial DDoS attacks generated with a WGAN model provides a valuable improvement in network security and helps these organizations maintain the availability and reliability of their services.

III. Attack Model

The main goal of a DDoS attack is to disrupt the availability of a server, making it inaccessible to legitimate users. The traditional detection schemes (whether embedded in an IDS or as a separate module) rely on identifying the signature of DDoS attacks within the traffic monitored. Most DDoS detection schemes proposed a "traditional" DDoS attacker and ML-enhanced DDoS detection engines. In our scheme, we assume that an ML-enhanced attacker along with the possibility of generating "traditional" DDoS traffic, can create perturbed adversarial DDoS attack traffic, where the DDoS attack's features are manipulated to follow the normal features to avoid detection by the IDS model. The attacker's target is a network or a server where the network traffic is observed by an ML-based IDS, but the attacker does not have direct access to the model itself or the monitoring process of the IDS architecture. In other words, the target is a black-box ML-Based IDS. But given that DDoS attacks have been extensively studied in the literature, an attacker could take advantage of this knowledge to learn the fundamental traffic features used by the IDS to detect the attacks. These basic traffic features allow an attacker to control their values to mimic the distribution of benign network traffic, in a way that the DDoS attack remains valid.

A. Determination of DDoS functional features

The adversary model aims to generate DDoS attacks on the network traffic that remains undetected by the ML-based IDS.

TABLE I: Summary of related works

Reference	Category	Dataset	Main idea	Performance evaluation	Limitation
[11]	Ml-based IDS	CICIDS-2017, NSL-KDS	LSTM model	Accuracy of 99.2% with CICIDS-2017 and 98.6% with NSL-KDD	Limited to one type of attack which is DoS
[13]	ML-Based IDS	CIC-DDoS2019	Learned the ensemble of feature selection using information gain and regression analysis	Accuracy of 97.86%	The prediction accuracy for the Friday afternoon log file is 73.79% for 16 attributes
[14]	ML-Based IDS	CIC-DDoS2019	XGBoost, KNN, CNN, Random Forest, SVM, and ANN	XGBoost obtained high accuracy	The dataset is unbalanced, and non-attack samples were fewer than the attack samples
[11]	ML-Based IDS	CIC-DDoS 2019	DNN, CNN, and LSTM	The CNN model gave the best result, accuracy of 99.99% for binary classification, and 99.30% in multi-class classification	They used some features with uneven distributions
[19]	ML-Based IDS	CIC-DDoS2019	Linear regression, KNN, Naïve Bayes, Decision Tree, Random Forest, SVM, and ANN	ANN outperforms the rest of the method with an accuracy of 98.645%	All the classifiers were unable to detect the Smurf class
[24]	Adversarial attack	KDDCup99	WGAN	Accuracy decreased by 50%	Generated samples were based on the single victim model
[25]	Adversarial attack	CIC-DDoS2019	Attacking the anomaly-based IDS in SDN by perturbing some features using GAN	Accuracy dropped from 100% to 0%	The attack is only implemented in SYN flood traces and based on a single targeted victim/model
[26]	Adversarial attack	Drebin	GAN-based attack model against machine learning-based classifiers for malware detection	Fooling the classifier by up to 99%	GAN model which is hard to be trained
[27]	Adversarial attack	Contagio	Generates a set of feature vectors similar to benign PDF file features using WGAN	100% evasion attack rate	Limited to specific format which is PDF
[32]	Adversarial attack and defense strategy	CIC-DDoS2019	Defense using ensemble voting based on three different ML models	Decrease in the success rate of the adversarial attacks	The attack model is based on a single targeted system or model
[34]	Adversarial attack and defense strategy	CIC-DDoS2019	Defense strategy using GAN model to enhance the LUCID model on the IDS	The model achieved an F1 score of 98%, the false negative rate drops to below 1.8 %	The generated dataset used to enhance the LUCID model is based on a single perturbation technique
Our proposed approach	Adversarial attack and defense strategy	CIC-DDoS2019, CICIDS-2017	Enhancing the IDS by adding another LSTM model responsible for blocking the adversarial samples	The model achieved a detection accuracy of 91.75% against the perturbed adversarial samples	The complexity and the time consuming of this algorithm because it employs 2 models

Therefore, the generator model should be able to control the value of the features that are used by the IDS to decide whether to block the traffic or forward it to the server. To accomplish this, we need a qualitative awareness of the relationship between the model's prediction and the attributes of the data instance used to make that prediction. We need explainable ML techniques which clarify some of these aspects. The SHapley Additive exPlanations (SHAP) [35] is one of these methods. SHAP is used to explain how each feature affects the model and permits local and global analysis for the dataset.

SHAP is an individualized model-agnostic explainer. The assumption made by a model-agnostic approach is that the model being explained is a black-box [36], and it is unknown how the model functions internally. As a result, the model-agnostic approach can only access the input data and the model's prediction. The idea behind SHAP's feature importance is that the features with large absolute Shapley values [37] are important. The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each feature to the prediction. Shapley values are calculated with the SHAP explanation method using coalitional game theory. A data

instance's feature values participate in a coalition as players. We can fairly distribute the "payout" among the features by using Shapley values. The explanation of the SHAP is specified as:

$$g(z') = \phi_0 + \sum_{j=1}^{n} \phi_j z' j \tag{1}$$

where g is the explanation model, z' $\in \{0,1\}^n$ is the coalition vector, n is the maximum coalition size, and $\phi_j \in \mathbb{R}$ is the feature attribution for a feature j. These features act as the DDoS functional features that will be controlled by the attack model to avoid detection by the IDS.

B. Adversary model

Generative Adversarial Network (GAN) is a machine learning-based paradigm proposed in 2014 [38]. The idea behind GAN is that neural networks could be used for generating new examples from an existing distribution. However, GAN consists of two neural networks, the generator (G), and the discriminant (D).

To generate samples that are similar to the real data, the generator estimates the probability distribution of the real samples. To determine whether a sample comes from real data as opposed to being generated, the discriminator is trained. In this method, the two components are trained to compete against each other. By bringing both sides together, the discriminator and the generator play a min-max game. We have to improve the loss function (Equation (2)), where x is the real data, z is the latent vector (noise), E is the expected value, D represents a discriminator and G represents a generator, and D and G are both differential functions represented by a multi-layer perceptron. Meanwhile, L(G,D) represents the value of the loss function.

$$\begin{aligned} Min_G Max_C L(G,D) = E_{x \in P_{data}(x)}[\log(D(x))] \\ + E_{z \in P_z(z)}[\log(1 - D(z))] \end{aligned} \tag{2}$$

The GAN method has several limitations and the most common ones include [39, 40]:

1) Convergence: In subsequent cycles, as the Generator improves, the discriminator's classification performance decreases. The convergence problem occurs when the GAN is trained from this point because the generator is trained from less meaningful data.
2) Mode collapse: The GAN can generate a wide range of data. However, if a generator learns to make a specific collection of data, for the discriminator to classify them as the original, the generator will only produce these sets of data and easily fool the discriminator.

To address these issues, the authors of [29] proposed the Wasserstein GAN method (WGAN). WGAN provides a better approximation of the distributed data in the training set. WGAN employs a generator and a critic to provide a score indicating how real and fake the generated data is. The WGAN value function using the Kantorovich-Rubinstein duality is as follows:

$$Min_G Max_{C \in A} E_{x \sim P\_data(x)}[C(x)] - E_{z \sim P\_z(Z)}[C(G(z))] \tag{3}$$

where (A) is the set of 1-Lipschitz functions [41]. The generator (G) creates a sample data point and sends it to the critic (C) after receiving a latent variable (Z) from a standard multivariate normal distribution. Once the generator and critic have been trained together, the generator will eventually resemble a deterministic transformation that yields data that is similar to the real data.

The generation of the adversarial DDoS attack is divided into two stages. In the first stage, we use WGAN to generate adversarial data that closely follows the DDoS instances from the input dataset.

We then evaluate the generator model by the similarity scores and the visualization of the cumulative sums per features between the generated and the original dataset. In the second stage, we replace the values presented in the DDoS function features with values from the benign samples, and the perturbed samples are input to the ML-based IDS models as Fig. 2 shows.

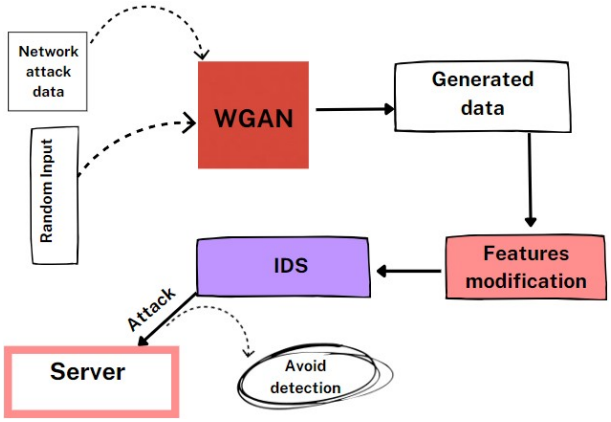

Fig. 2: Generating adversarial DDoS attack

IV. Experiments: Generation of the Adversarial Attack

In the first set of experiments, we investigate the effectiveness of the model in generating DDoS attacks data which follow the data distribution of the samples in the dataset. In addition, we evaluate the proposed perturbation technique in the generation of the adversarial DDoS attack that aims to evade the detection by the IDS by building the standard IDS and testing it with both the original and the adversarial DDoS attacks.

A. Datasets

In this research, we use two distinct datasets, namely the CIC-DDoS2019 [22] and the CIC-IDS2017 [21] datasets. In the CIC-DDoS2019 dataset, the victim network is a high-security infrastructure with a firewall, router, switches, and several common operating systems. The network attack is a completely separate third-party infrastructure that executes different types of DDoS attacks including the most prevalent forms of DDoS assault. CICFlowMeter-V3 [42] was used to extract more than 80 traffic features such as destination port, flow duration, and the number of packets per second of traffic flows. The resulting dataset contains 251723 samples of attacks and 3134 samples of benign traffic. For this reason, to avoid the unbalanced classes problem, we combined the CIC-DDoS2019 dataset with the benign class samples from the CIC-IDS2017 dataset based on the shared features. The updated dataset contains 251723 samples of DDoS attacks and 251723 samples of benign traffic and 80 features.

We pre-processed (Fig. 3) the dataset by removing null and infinite values, and dropping some features (21 features), either because they were unused (e.g., source IP address, source port, timestamps), or because the data distributions in those features were uneven (e.g., acknowledgment flag count, down/up ratio). Next, to normalize and scale the dataset, we applied the min-max scaling (4) to it in order to put all of our features in the same scale. After performing the pre-processing steps, we split the dataset into 70% for training, 15% for validation and 15% for testing to evaluate the models.

$$x = \frac{x - min(x)}{max(x) - min(x)} \tag{4}$$

We applied the SHAP analysis on the dataset, and Fig. 4 shows the result of the global effect of the features in the output. In this plot, the y-axis which represents the features that are listed from the

highest to the lowest effect on the prediction is based only on the x-axis that represents the mean of the absolute SHAP value. Thus, it is irrelevant whether the feature has a positive or negative impact on the output, the value of the absolute Shapley values per feature across the data is described in equation(5):

$$\phi_j = \frac{1}{n}\sum_{j=1}^{n} |\phi_j^{(i)}| \tag{5}$$

This information is not enough to understand the data importance and the impact of each contributed feature on the result. For this reason, another plot can be used, namely the bee swarm plot Fig. 5. In the bee swarm plot, the features are also ordered by their effect on the prediction, but we can also see how the higher or the lower values of the feature will affect the result. Here, the x-axis represents the SHAP value, the y-axis represents the features, and the color of the point shows if that observation has a higher or a lower value. As an example, a lower value of the total number of bytes sent in the initial window in the backward direction (Init_win_bytes_backward) has a positive impact, but a higher value has a negative impact on the result.

These features act as functional DDoS features that are modified in a way that look like normal traffic to avoid detection by the IDS.

B. Generating DDoS attacks using WGAN model

In this step, we use WGAN to generate adversarial data that follow the DDoS data distribution from the input dataset. Then, we evaluate the generator model by the similarity scores and the visualization of the cumulative sums per feature between the generated and the original dataset.

The WGAN model consists of two models, generator and critic, and Fig. 6 shows the network that connects these two models that are designed with gradient penalty [43].

The generator is a model consisting of a fully connected layer of $Fc_{le \to Nf}$, with a ReLU activation function, and the hidden layer is formed by the concatenation of multiple vectors that could form data similar to the transformed original data with the same dimension le.

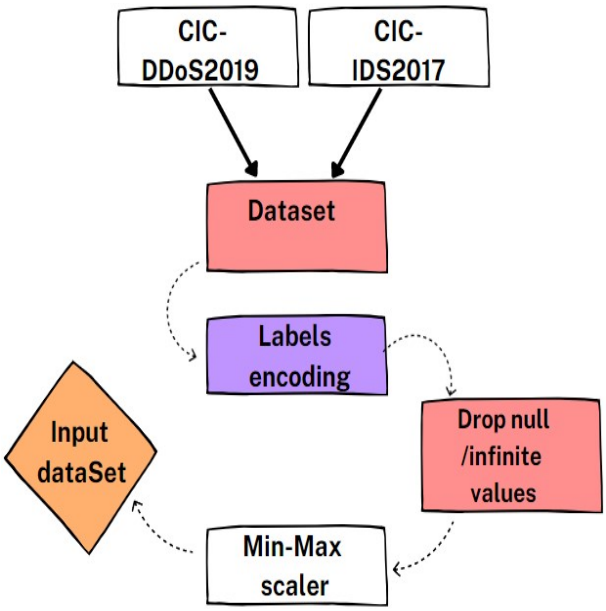

Fig. 3: Dataset pre-processing

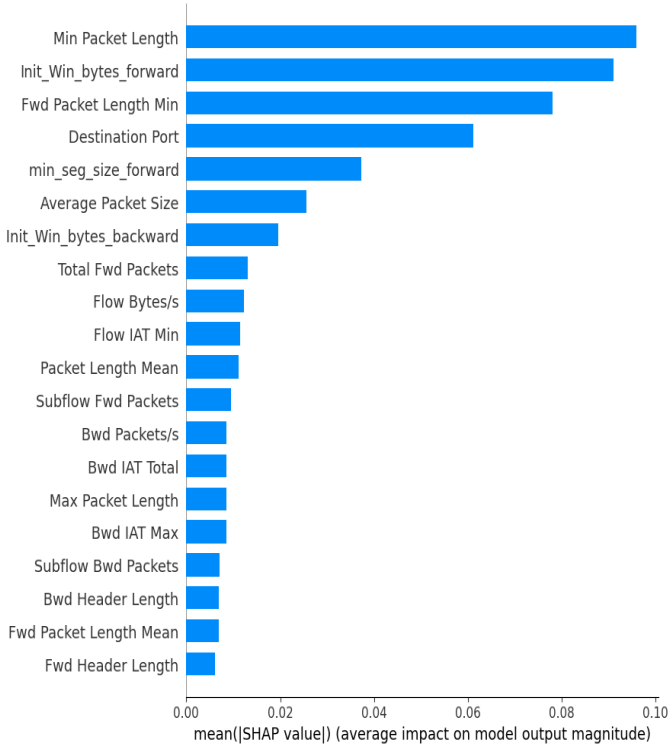

Fig. 4: Feature importance using SHAP

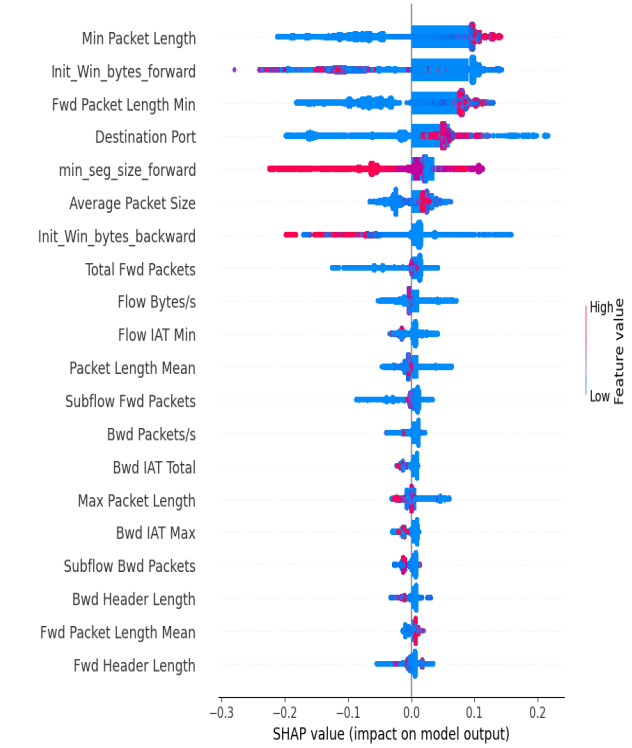

Fig. 5: Feature impact using SHAP

This architecture is formally described in (IV-B):

$$\begin{cases} h0 = Z(latent\ vector) \\ h1 = ReLu(Fc_{le \to h0}) \\ h2 = ReLu(Fc_{le \to h1}) \end{cases}$$

where $Fc_{x \to z}$ is the fully connected layer with the input size (x) and output size z, and $Relu(x)$ denotes applying Relu activation on x.
The critic consists of two fully connected layers with the LeakyRelu [44] activation function and is described in Equatilooks(6):

$$\begin{cases} h0 = output\ of\ the\ generator \\ h1 = LeakyReLu_{0.01}(Fc_{le \to h0}) \\ h2 = LeakyReLu_{0.01}(Fc_{le \to h1}) ndcases \end{cases} \quad (6) where$$

LeakyReLu$_r(X)$ is applying the LeakyRelfeaturetion function with slop r on X.

After building the generator and the discriminator models, we initialize the training parameters. We use a batch size of 256, Adam optimizer[45] with α = 0.0002, $\beta 1$ = 0.5, and $\beta 2$ = 0.999. We load the dataset, and we train the model for 100 epochs. The generator model can sometimes become too powerful and exploit the Critic model, leading to gradient vanishing problems and resulting in invalid adversarial samples. To prevent this from occurring, we have trained the Critic model for longer periods to build a stronger model that is better able to measure the distance between real and fake samples. This helps to ensure that the generated adversarial samples are valid and accurately reflect the statistical properties of the real traffic data. By training the Critic model for longer periods, we can improve its ability to discriminate between real and fake samples and reduce the likelihood of generating invalid adversarial samples. To this end, for each epoch, we train the critic 4 times and the generator 1 time. In the Critic's training, each time we use the generator model to produce fake samples from noise input. We randomly select samples from the original dataset, and based on the prediction of the critic for these two classes of samples, we update the critic weights by decreasing the gradient. In the generator training, we generate fake samples, and use them as input for the critic model, and based on the output, we update the weights of the generator model by decreasing the gradient.

When the training process ends, we use the generator model to generate a new fake dataset. To evaluate the validity of the generated adversarial samples we followed the evaluation approach used in the literature by comparing the cumulative sums per feature. We compare the cumulative sums of the adversarial samples with those of the real samples, and we draw a graph that shows the difference between the generated dataset and the original data. Fig. 7 presents the result of this comparison which shows that the cumulative sums of the adversarial samples are similar to those of the real samples and that most of the features match closely with the features of the real data. This show that the adversarial samples are "valid" and follow the statistical distribution of the real samples. Additionally in the next section, we compare the performance of different machine learning algorithms on the generated data which helps us determine if the models trained on the original data are able to generalize to the generated data and provides insight into whether the distribution of the generated data is similar to that of the original data. By conducting this comparison, we can assess the quality of the generated data.

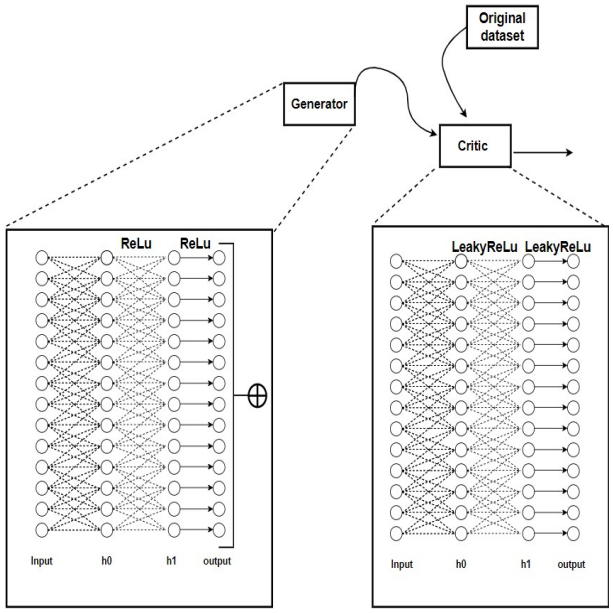

Fig. 6: WGAN model architecture

C. Standard IDS model Architecture

After evaluating the generation of the DDoS samples and comparing them with the original data, our next goal is to evaluate the perturbation technique on a high performance ML-based IDS.
To meet this objective, we need to build the IDS. We used different machine learning algorithms that are widely used in the literature resulting in high accuracy in detecting a DDoS attack. The algorithms include decision tree classifier [46], extreme gradient boosting (XGBoost), [47], multilayer perceptron classifier (MLPClassifier) [48], random forest classifier [49], and deep learning algorithms namely LSTM [50]. We used LSTM in our evaluation and it consists of a Simple LSTM model with DNN layers. We used the ReLU activation function in all layers added to the Sigmoid activation function in the last layer, and we use binary cross-entropy as the loss function with the ADAM optimizer.
The best model is chosen to act as the baseline detector in the detection engine. We train these models using the original data and we perform tests to evaluate them. Additionally, we test all these models with the generated DDoS attack as a second way of evaluating the GAN model. Table II shows the results. F1 real represents the F1 score for these algorithms on the original dataset, and F1 fake represents the F1 score 7 on the fake dataset, show that the LSTM outperformed the other algorithms with F1 score of 0.99.
Most models used in the experiments achieved similar results with both the real and adversarial datasets, demonstrating that the adversarial samples are "valid" and accurately reflect the statistical properties of the real samples. This suggests that the WGAN model used to generate the adversarial samples successful in mimicking the statistical distribution of the real traffic data. As a result, the adversarial samples can be used to effectively evaluate the performance of the IDS in detecting DDoS attacks.

$$F1 = \frac{TP + TN}{TP + FN + FP + TN} \quad (7)$$

Our ML-based IDS, so far, consists of the features encoder, detection engine (the pre-trained LSTM), and the decision makers (alert System) as Fig. 8 illustrates.

D. Generation of the perturbed DDoS adversarial attack

In this step, we generate DDoS instances using the same WGAN models (Fig. 2) but we keep some of the DDoS functional features that have been extracted using the SHAP model (Fig. 4) to follow

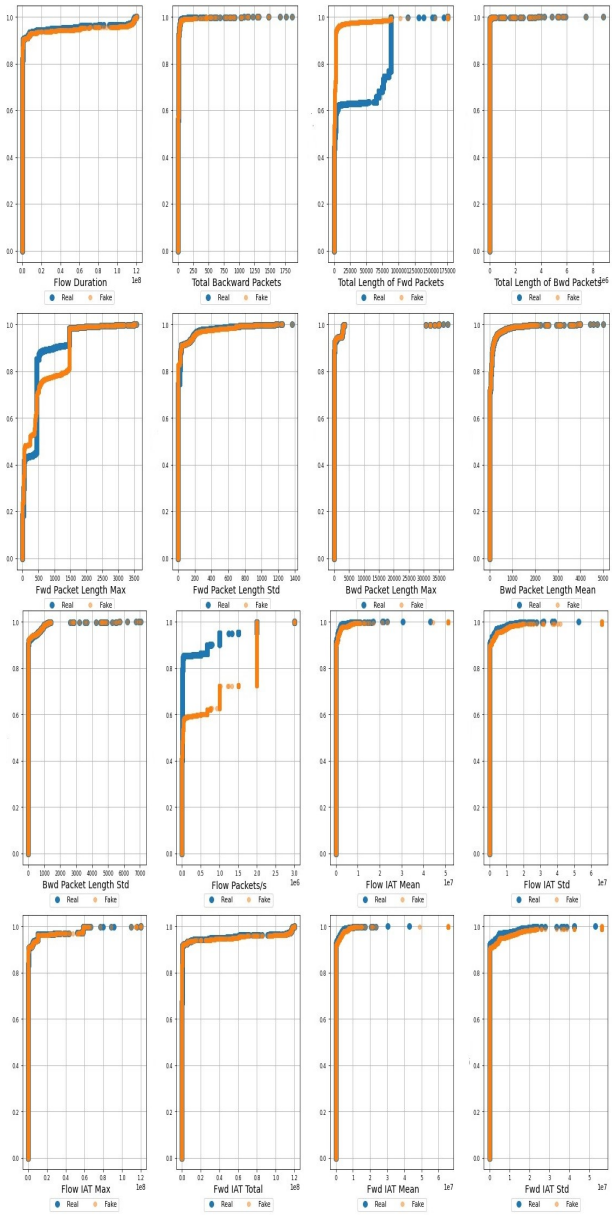

Fig. 7: Cumulative sums per features

Classifier	f1 real	f1 fake
Decision tree classifier	0.9570	0.9440
MLP classifier	0.9175	0.8470
XGBoost	0.9513	0.9065
Random forest classifier	0.9722	0.9625
LSTM	0.99	0.9825

TABLE II: Result of the machine learning classifiers.

the distribution of the benign samples' features. Every time we apply a new scenario, we change a different amount for the features (8, 16, 20). Next, we evaluate the IDS's ability to accurately classify them as well as if the accuracy decreases. We conducted tests for these different scenarios, and the confusion matrix (Fig. 9) shows the results. The confusion matrix is a 2 x 2 matrix which is used to evaluate the performance of a classification model. The

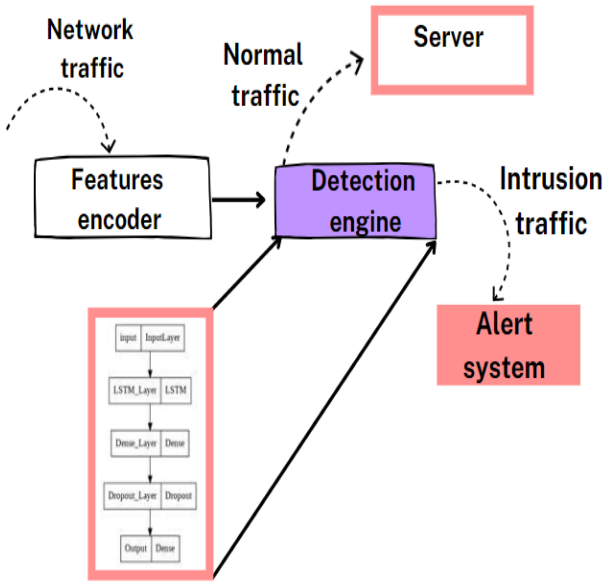

Fig. 8: IDS model architecture

(a)

	Normal	DDoS
Normal	100 %	0 %
DDoS	0%	100 %

(b)

	Normal	DDoS
Normal	99.3 %	0.7 %
DDoS	77.2 %	22.8

(c)

	Normal	DDoS
Normal	99.4 %	0.6 %
DDoS	79.5 %	20.5 %

Fig. 9: Confusion matrix of the predicted network traffic on the IDS in different scenarios: (a) On the original data, (b) With only 16 modifications in the functional features, (c) Modified all the functional features.

matrix compares the actual target values with those predicted by the machine learning model.

By comparing the 3 confusion matrices, we conclude that the IDS was able to classify all the data accurately (Fig. 9(a)) when there are no perturbations on the input features, but on the other hand, there is a significant decrease in this performance when there is a perturbation in the input features (Fig. 9(b,c)).

Additionally, to better understand the results, we show the AUC-ROC curve. The AUC-ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is at predicting benign classes as benign and DDoS classes as DDoS.

By analogy, the higher the AUC score, the better the model is at distinguishing between network traffic: if it is DDoS or benign. The ROC curve is plotted with True Positive Rate (TPR) against the False Positive Rate (FPR) where TPR is on the y-axis and FPR is on the x-axis. Based on these outcomes, by comparing the results of the AUC-ROC curve (Fig. 10), we found that the IDS was unable

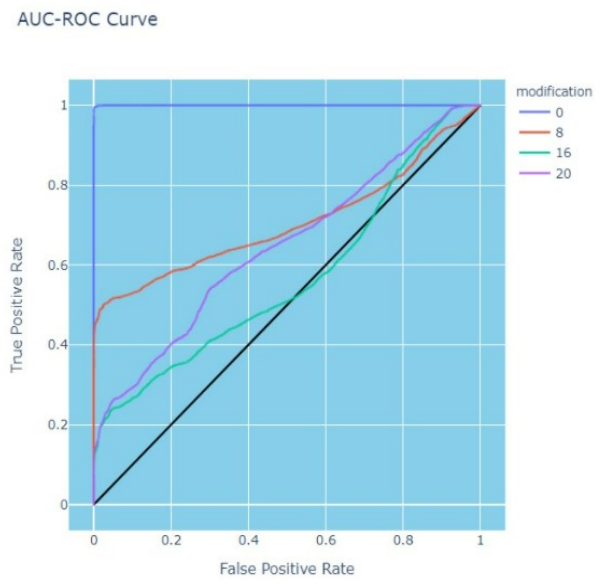

Fig. 10: ROC-AUC Curve

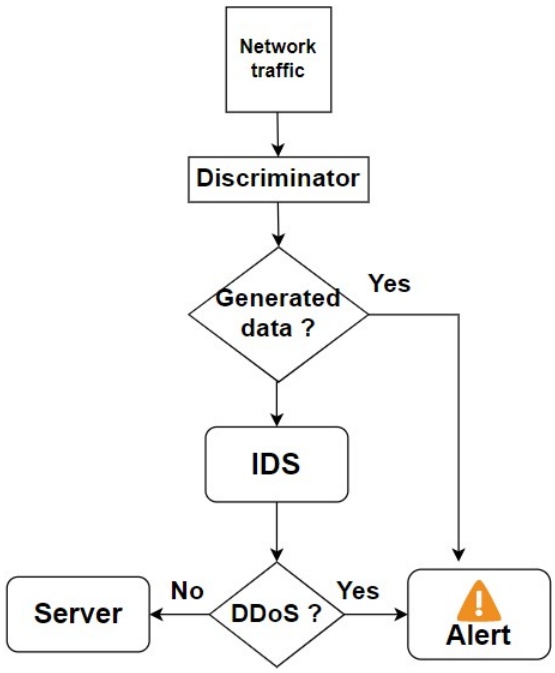

Fig. 11: IDS model architecture

to categorize this sort of attack, and the performance of the IDS decreases when the number of perturbed features increases.

V. Enhanced Security Against Perturbed DDoS Attack

After the evaluation of the attack model, and discovering that the IDS model was unable to categorize this sort of attack, our next objective is to enhance the IDS's performance in the detection of the DDoS attack of this nature, regardless if there are replacements or perturbations in the features of the attack, or not. To detect it, we hypothesize that GAN can be used to enhance IDS performance by training a new model based on the generated adversarial samples. To address this need, we update the previous IDS architecture (Fig. 8) so that instead of using a single model to determine whether the input traffic is DDoS or not, it uses two different models. The first model prevents the perturbed (generated) network traffic from fooling the IDS model, and the real network traffic is then forwarded to the second model. The second model blocks abnormal network traffic (DDoS), and the benign traffic is forwarded to the target system or machine as Fig. 11 shows.
The next section describes the details of the first model.

A. Discriminator Model

To detect the generated data and enhance the security against the type of DDoS, we build the first model, namely the discriminator. While building the discriminator we kept the architecture of this model to be the same as the previous IDS model. The only difference is that this model will be trained to detect the perturbed traffic, and to avoid the vulnerability of DL against the adversarial attack, we trained this model based on a combination of datasets between the original and the generated fake dataset, including all the features except those contributing to the DDoS function. The generated dataset that is used here is the result of the same GAN model mentioned before.

After building the model and at the end of the training phase, we evaluated the model with perturbed data generated from the same GAN model. Fig. 12 shows the results of our tests. From 12000 samples of network traffic, the discriminator model was able to detect 95.4% of the data correctly as fake, and 88.2% of data correctly as real data.

	Real	Fake
Real	88.2 %	11.8 %
Fake	4.6 %	95.4 %

Fig. 12: Confusion matrix of the predicted network traffic on the discriminator(Real/Fake)

B. IDS detection model

After building and testing the discriminator which achieved a high accuracy of 91.75%, we updated the second model that deals only with the real data forwarded using the discriminator. We trained LSTM using the original dataset, including only DDoS functional features.
As a result, this model was able to detect all the samples correctly between benign or DDoS as the confusion matrix (Fig. 13) shows.

	Normal	DDoS
Normal	100 %	0 %
DDoS	0%	100 %

Fig. 13: Confusion matrix of the predicted network traffic on the IDS(DDoS/Normal)

Finally, by combining both models, the discriminator, and the detection engine of the IDS, we evaluated the performance of the IDS against the perturbed adversarial attack. Based on the results of the ROC-AUC curve for the performance of the IDS in the detection of the DDoS attack regardless if there are perturbations or not, our updated IDS was able to detect, and block the DDoS attack with increased robustness against the perturbed DDoS samples as

(Fig. 14) shows in anticipation of an attacker who can craft packets and perturb malicious flows to mimic those characteristics of benign flows.

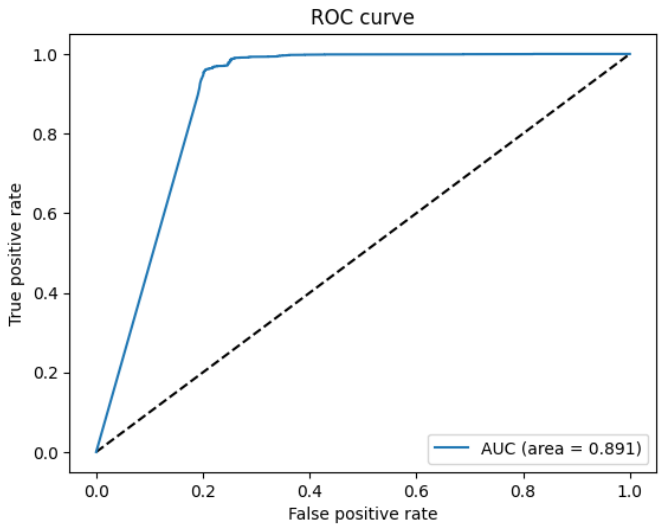

Fig. 14: ROC-AUC Curve for the enhanced IDS

VI. Deployment on the Internet Service Providers

As solutions against DDoS attacks, Internet Service Providers (ISP) and companies are using a lot of solutions and Scrubbing Centers such as Radware DefensePro, Radware Cloud DDoS Protection Service, Cloudflare DDoS Mitigation Services, Akamai Edge DNS, Arbor Cloud, Oracle Dyn DDoS Protection, Azure DDoS Protection, etc. All these solutions offer features such as multi-layered protection, real-time threat detection, reporting, and analytics. However, actual DDoS attacks are faster with unprecedented rates, and more sophisticated. Botnets are more decentralized and very well-secured. Using intelligent scrubbing centers is a new direction to enhance centers automation and precision. Hence, machine learning-based scrubbing centers are considered the Next Generation Scrubbing Centers (NGSC), deploying a machine learning-based IDS on an ISP requires carefully plan and executing the deployment of a machine learning-based IDS on an ISP's network, in order to ensure that it provides effective security and does not negatively impact network performance or disrupt users.

In our case, The proposed IDS is capable of detecting live traffic in real-time, along with the flow meters to extract the necessary features for decision-making, in our tests, we found that using the "Tesla V100-PCIE-16GB GPU" we were able to analyze a single packet in about 40ms including both models prediction and without the features extraction phase. This allows for efficient and effective real-time detection of potential security threats.

To mimic an ISP scenario in the context of the power needed, a simple multi-threading system should be sufficient to handle the processing of multiple samples in parallel. This will help ensure that the IDS can process and analyze traffic in real time without significant delays. As shown in Figure 15, our proposed IDS can be deployed in the following manner:

1) The first router forwards all network packets to the ISP.
2) The CICFlowmeter extracts the features needed by the IDS and stores them in a memory queue.
3) The IDS dequeues a packet from the queue and processes it every 40ms.

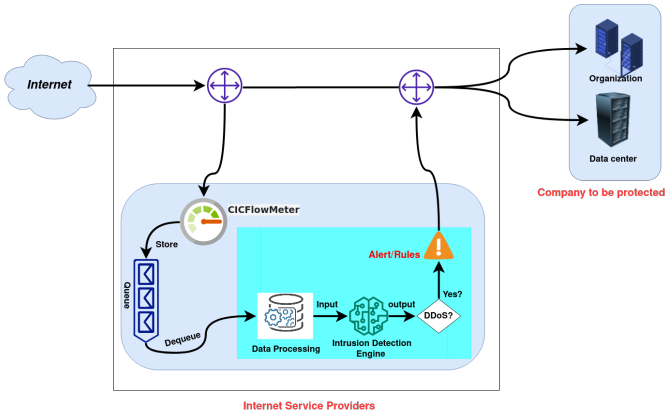

Fig. 15: IDS deployment on the ISP

4) Based on the IDS's decision, an alert system and corresponding rules are applied.

This deployment strategy allows for efficient and effective real-time monitoring of network traffic for security threats."

VII. Conclusion and Future Work

DDoS detection remains a challenging problem in cybersecurity. Recently, we have witnessed increasing interest in DDoS detection using machine learning (ML) and deep learning (DL) algorithms. Ironically, although ML/DL can increase detection accuracy, they can still be evaded by using ML/DL techniques to create attack traffic. This paper addresses the above aspects of ML-based DDoS detection and anti-detection techniques. First, we build a DDoS detection method based on the Long Short-Term Memory (LSTM) model. The detection scheme showed a high accuracy level in detecting DDoS attacks with an accuracy of 100%. Second, we tested the same technique against adversarial DDoS attacks generated using GAN. Based on the comparison of the ROC curves and the accuracy of different scenarios, the results obtained showed a decrease in the performance of the IDS. Finally, we demonstrated how to enhance this scheme to detect adversarial DDoS attacks by building two different models. The first one is used to detect if the network input traffic is fake to block it, otherwise, forward it to the IDS that is responsible for detecting if it is DDoS or normal traffic. Our experimental results show that our detection model is efficient and accurate in identifying GAN-generated adversarial DDoS traffic with a detection ratio ranging between 91.75% and 100%. As part of our future work, it would be necessary to evaluate the performance of our IDS on data generated by another model such as the auto-encoder. In addition, further work is required to study and investigate the use of online learning algorithms, which allow the IDS to update its model in real-time as it processes new data. By incorporating an incremental update capability, the IDS could maintain its effectiveness even in the face of evolving attack methods.

VIII. Appendix

The symbols used in the equations are described in Table III

References

[1] "Ddos statistical report for 1hy 2022." https://blog.nexusguard.com/threat-report/ddos-statistical-report-for-1hy-2022, 2022.

[2] A. Shield, "Threat landscape report – q1 2020." https://aws-shield-tlr.s3.amazonaws.com/2020-Q1-AWS-Shield-TLR.pdf, 2020.

TABLE III: Notation and definition:

Symbol	Definition
x	Real data
z	Latent vector
$D()$	Discriminator's evaluation of real or fake data
$C()$	Critic's evaluation of real or fake data
$G()$	Generator's evaluation of real or fake data
$P_{data}(x)$	Data distribution over real sample x
$P_z(z)$	Data distribution over fake sample z
E_x	Expected value over the original data
E_z	Expected value of the random data inputs to the Generator
Fc	Fully connected layer
h	Hidden layer

[3] Spamhaus, "Spamhaus botnet threat update." https://www.spamhaus.com/custom-content/uploads/2022/07/2022-Q2-Botnet-Threat-Update.pdf, 2022.

[4] corero.

[5] CyberSecurity and I. S. A. (CISA), "Security tip (st04-015). understanding denial-of-service attacks." Online, Nov. 2019.

[6] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Ddosnet: A deep-learning model for detecting network attacks," *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 391–396, 2020.

[7] L. Yong and Z. Bo, "An intrusion detection model based on multi-scale cnn," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 214–218, 2019.

[8] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, "Flowguard: An intelligent edge defense mechanism against iot ddos attacks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9552–9562, 2020.

[9] Y.-Y. Zhu, J.-F. Wu, and Z. Ming, "Research on intrusion detection based on network events and deep protocol analysis," *Journal of China Institute of Communications*, vol. 32, no. 8, pp. 171–178, 2011.

[10] K. O. Adefemi Alimi, K. Ouahada, A. M. Abu-Mahfouz, S. Rimer, and O. A. Alimi, "Refined lstm based intrusion detection for denial-of-service attack in internet of things," *Journal of Sensor and Actuator Networks*, vol. 11, no. 3, p. 32, 2022.

[11] D. Akgun, S. Hizal, and U. Cavusoglu, "A new ddos attacks intrusion detection model based on deep learning for cybersecurity," *Computers & Security*, vol. 118, p. 102748, 2022.

[12] X. Yuan, C. Li, and X. Li, "Deepdefense: identifying ddos attack via deep learning," in *2017 IEEE international conference on smart computing (SMARTCOMP)*, pp. 1–8, IEEE, 2017.

[13] S. Sambangi and L. Gondi, "A machine learning approach for ddos (distributed denial of service) attack detection using multiple linear regression," *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 63, no. 1, p. 51, 2020.

[14] G. Usha, M. Narang, and A. Kumar, "Detection and classification of distributed dos attacks using machine learning," in *Computer Networks and Inventive Communication Technologies*, pp. 985–1000, Springer, 2021.

[15] C. Yinka-Banjo and O.-A. Ugot, "A review of generative adversarial networks and its application in cybersecurity," *Artificial Intelligence Review*, vol. 53, no. 3, pp. 1721–1736, 2020.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[17] K. Liu, H. Yang, Y. Ma, B. Tan, B. Yu, E. F. Y. Young, R. Karri, and S. Garg, "Adversarial perturbation attacks on ml-based cad: A case study on cnn-based lithographic hotspot detection," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 25, aug 2020.

[18] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks," *Expert systems with Applications*, vol. 29, no. 4, pp. 713–722, 2005.

[19] K. S. Sahoo, A. Iqbal, P. Maiti, and B. Sahoo, "A machine learning approach for predicting ddos traffic in software defined networks," in *2018 International Conference on Information Technology (ICIT)*, pp. 199–203, IEEE, 2018.

[20] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[21] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, vol. 1, pp. 108–116, 2018.

[22] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–8, IEEE, 2019.

[23] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, pp. 1–6, Ieee, 2009.

[24] Q. Yan, M. Wang, W. Huang, X. Luo, and F. R. Yu, "Automatically synthesizing dos attack traces using generative adversarial networks," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 12, pp. 3387–3396, 2019.

[25] J. Aiken and S. Scott-Hayward, "Investigating adversarial attacks against network intrusion detection systems in sdns," in *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 1–7, IEEE, 2019.

[26] M. Shahpasand, L. Hamey, D. Vatsalan, and M. Xue, "Adversarial attacks on mobile malware detection," in *2019 IEEE 1st International Workshop on Artificial Intelligence for Mobile (AI4Mobile)*, pp. 17–20, IEEE, 2019.

[27] J. Zhang, Q. Yan, and M. Wang, "Evasion attacks based on wasserstein generative adversarial network," in *2019 Computing, Communications and IoT Applications (ComComAp)*, pp. 454–459, IEEE, 2019.

[28] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, pp. 1–6, Ieee, 2009.

[29] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, PMLR, 2017.

[30] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket.," in *Ndss*, vol. 14, pp. 23–26, 2014.

[31] S. Chenette, "Malicious documents archive for signature testing and research-contagio malware dump," 2011.

[32] C. Zhang, X. Costa-Pérez, and P. Patras, "Tiki-taka: Attacking and defending deep learning-based intrusion detection systems," in *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pp. 27–39, 2020.

[33] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward

generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, vol. 1, pp. 108–116, 2018.
[34] M. Abdelaty, S. Scott-Hayward, R. Doriguzzi-Corin, and D. Siracusa, "Gadot: Gan-based adversarial training for robust ddos attack detection," in *2021 IEEE Conference on Communications and Network Security (CNS)*, pp. 119–127, 2021.
[35] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.
[36] C. Moreira, Y.-L. Chou, M. Velmurugan, C. Ouyang, R. Sindhgatta, and P. Bruza, "Linda-bn: An interpretable probabilistic approach for demystifying black-box predictive models," *Decision Support Systems*, vol. 150, p. 113561, 2021.
[37] L. Merrick and A. Taly, "The explanation game: Explaining machine learning models using shapley values," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 17–38, Springer, 2020.
[38] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
[39] Z. Zhang and M. Li, "Jun yu. on the convergence and mode collapse of gan," *SIGGRAPH Asia 2018 Technical Briefs*, p. 21, 2018.
[40] H. Xie, K. Lv, and C. Hu, "An effective method to generate simulated attack data based on generative adversarial nets," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 1777–1784, IEEE, 2018.
[41] Z. Zhou, J. Liang, Y. Song, L. Yu, H. Wang, W. Zhang, Y. Yu, and Z. Zhang, "Lipschitz generative adversarial nets," in *International Conference on Machine Learning*, pp. 7584–7593, PMLR, 2019.
[42] A. H. Lashkari, A. Seo, G. D. Gil, and A. Ghorbani, "Cicab: Online ad blocker for browsers," in *2017 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–7, IEEE, 2017.
[43] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.
[44] X. Zhang, Y. Zou, and W. Shi, "Dilated convolution neural network with leakyrelu for environmental sound classification," in *2017 22nd international conference on digital signal processing (DSP)*, pp. 1–5, IEEE, 2017.
[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.
[46] P. H. Swain and H. Hauska, "The decision tree classifier: Design and potential," *IEEE Transactions on Geoscience Electronics*, vol. 15, no. 3, pp. 142–147, 1977.
[47] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.
[48] T. Windeatt, "Accuracy/diversity and ensemble mlp classifier design," *IEEE Transactions on Neural Networks*, vol. 17, no. 5, pp. 1194–1211, 2006.
[49] M. Pal, "Random forest classifier for remote sensing classification," *International journal of remote sensing*, vol. 26, no. 1, pp. 217–222, 2005.
[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.