Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5757164, 10 pages
https://doi.org/10.1155/2022/5757164

Research Article

Enhanced Security Against Volumetric DDoS Attacks Using Adversarial Machine Learning

Jugal Shroff,[1] **Rahee Walambe**,[1,2] **Sunil Kumar Singh**,[3] **and Ketan Kotecha**[1,2]

[1]*Symbiosis Institute of Technology (SIT), Symbiosis International (Deemed University), India*
[2]*Symbiosis Centre for Applied Artificial Intelligence, Symbiosis Institute of Technology (SIT), Symbiosis International (Deemed University), India*
[3]*School of Computer Science and Engineering, VIT-AP University, India*

Correspondence should be addressed to Rahee Walambe; rahee.walambe@sitpune.edu.in and Ketan Kotecha; drketankotecha@gmail.com

Received 7 November 2021; Accepted 24 February 2022; Published 11 March 2022

Academic Editor: Shalli Rani

With the increasing number of Internet users, cybersecurity is becoming more and more critical. Denial of service (DoS) and distributed denial of service (DDoS) attacks are two of the most common types of attacks that can severely affect a website or a server and make them unavailable to other users. The number of DDoS attacks increased by 55% between the period January 2020 and March 2021. Some approaches for detecting the DoS and DDoS attacks employing different machine learning and deep learning techniques are reported in the literature. Recently, it is also observed that the attackers have started leveraging state-of-the-art AI tools such as generative models for generating synthetic attacks which fool the standard detectors. No concrete approach is reported for developing and training the models which are not only robust in the detection of standard DDoS attacks but which can also detect adversarial attacks which are created synthetically by the attackers with harmful intentions. To that end, in this work, we employ a generative adversarial network (GAN) to develop such a robust detector. The proposed framework can generate and classify the synthetic benign (normal) and malignant (DDoS) instances which are very similar to the corresponding real instances as evaluated by similarity scores. The GAN-based model also demonstrates how effectively the malicious actors can generate adversarial DDoS network traffic instances which look like normal instances using feature modification which are very difficult for the classifier to detect. An approach on how to make the classifiers robust enough to detect such kinds of deliberate adversarial attacks via modifying some specific attack features manually is also proposed. This work provides the first step towards developing a generic and robust detector for DDoS attacks originating from various sources.

1. Introduction

According to [1], cybercrimes have increased by over 600% during the COVID-19 pandemic. Organizations have a huge amount of sensitive public data which needs to be protected. A cyber-attack can severely damage their reputation and consumer trust, leading to loss of customers and sales, thus resulting in financial losses. Further implications of this could result in harassment and cyberbullying of the individuals whose data is hacked or stolen. Additionally, there are legal consequences such as heavy fines imposed by the government that an organization might face after suffering a cyber-attack. Therefore, with the increasing number of Internet users, cybersecurity is becoming more and more critical. Cyber-attacks can be divided into two categories:

(1) Passive attacks: these cause damage to data confidentiality. In this kind of attack, an intruder monitors the system for information that can later be used for malicious purposes. The information remains unchanged, and the system has no impact. Some of the examples include an attacker trying to scan a

device or a server to find vulnerabilities such as open ports or an attacker trying to monitor a website's traffic [2, 3]

(2) Active attacks: active attacks cause damage to the integrity and availability of the system. In this kind of attack, an attacker uses the information gained during the passive attack to exploit a device or a server. Unlike passive attacks, the information can be changed, and system service may be harmed during an active attack. Some of the examples of active attacks are DoS/DDoS attacks [4], MITM attacks [5], and Trojan attacks [6]

Among all the attacks, DoS and DDoS attacks are two of the most common types of cyber-attacks [7]. Between the period of January 2020 and March 2021, DDoS attacks increased by 55% with the technology sector being the most impacted ones [8].

A denial of service (DoS) attack is the type of attack in which an attacker tries to make a website or a computer server unavailable to other users by flooding the website or the server with heavy traffic. The attacker sends much more traffic than the server or the website can accommodate. A distributed denial of service (DDoS) attack is a DoS attack originating from multiple sources on the same target. DDoS attacks can be divided into 3 types [9]:

(1) Volume-based attacks: in this type of attack, heavy traffic is sent to the server to consume all its network bandwidth

(2) Protocol attacks: in this type of attack, the aim is to exploit server resources such as firewalls and load balancers

(3) Application layer attacks: these types of attacks are considered as the most serious types of attacks and exploit the weaknesses present in the application layer

Figure 1 shows the frequency of different DDoS attacks from January 2020 through March 2021 [8].

From Figure 1, it can be observed that volumetric DDoS attacks have higher chances of occurring than other types of attacks. Therefore, security systems must be able to detect such volumetric DDoS attacks and raise an alarm at the right time to prevent any damage.

1.1. Detecting DDoS Attacks with Standard Approaches. One of the ways to mitigate a DDoS attack is to limit the number of requests a server or a device from a particular IP address can send. However, with this approach, even legitimate requests can be blocked in some cases such as a user trying to refresh a page multiple times. Another way includes filtering out network traffic based on certain features, but identifying those features is not an easy task. With the recent advancements in AI, many researchers have tried to apply various machine learning and deep learning algorithms to detect DDoS attacks. In [10], multiple linear regression [11] is employed to detect DDoS attacks using CIC-IDS 2017 dataset [12]. The authors shortlisted some important features using the information gain technique [13]. First, the top 16 features are used to train the model and predict the classes. The reported prediction accuracy is 73.79% for the Friday afternoon dataset. Further, 10 statistically insignificant attributes are eliminated, reducing the accuracy to 71.7% on the same dataset. Later on, the authors experimented with the ensemble model [14] and obtained an accuracy of 97.86%. The authors in [15] have used machine learning (ML) methods, namely, linear regression, *K*-nearest-neighbors (KNN), Naive Bayes (NB), decision tree (DT), random forest (RF), artificial neural network (ANN), and support vector machine (SVM), to detect DDoS attacks where the ANN outperforms the rest of the methods. Elsayed et al. proposed another method to detect DDoS attacks in [16] in which they have used a recurrent neural network (RNN) [17] along with an autoencoder [18] on the CICDDoS-2019 dataset [19]. They were able to outperform the previous models with an accuracy of 99%. In [20], a bidirectional RNN [21] along with long-short-term memory (LSTM) [22] and gated recurrent unit (GRU) [23] (to eliminate the vanishing gradient problem of RNN) to detect DDoS attacks is implemented. UNB ISCX Intrusion Detection Evaluation 2012 dataset [24] is used to demonstrate the approach, and maximum accuracy of 97.996% and 98.410% using two different datasets is reported.

1.2. Detecting DDoS Attacks Using Adversarial Machine Learning(AML) Paradigm. Various AI algorithms to detect DDoS attacks are proposed in [10, 15, 16, 20]. However, ML- and deep learning- (DL-) based classification models may perform poorly when there are changes in the input feature space [25]. This problem of generalization can be used by some malicious actors to trick the classifiers into making a wrong decision. This falls under the adversarial machine learning (AML) paradigm. AML techniques attempt to fool the AI detectors by supplying deceptive input with a primary reason to cause the malfunction in the machine learning model. Most ML models are designed to work on a specific dataset where the train and test data come from the same independent and identical distribution (IID). However, in the real-world scenario, if the data that is supplied to this model does not satisfy this statistical assumption and comes from a different IID, the results may get compromised [26, 27]. Adversarial attacks can be classified into two categories:

(1) Poisoning attacks: these types of attacks occur during the AI model training phase. In this type of attack, either the training dataset is poisoned with the malicious input data or the model training algorithm is modified by the attacker, thus changing the way the algorithm learns to classify input data

(2) Evasion attacks: they are the most prevalent type of attack, wherein the data is modified to be classified as legitimate and evade detection after deployment

To mitigate this problem, GANS [28] can be used to generate synthetic benign and DDoS instances and validate if the security systems are robust enough to identify those

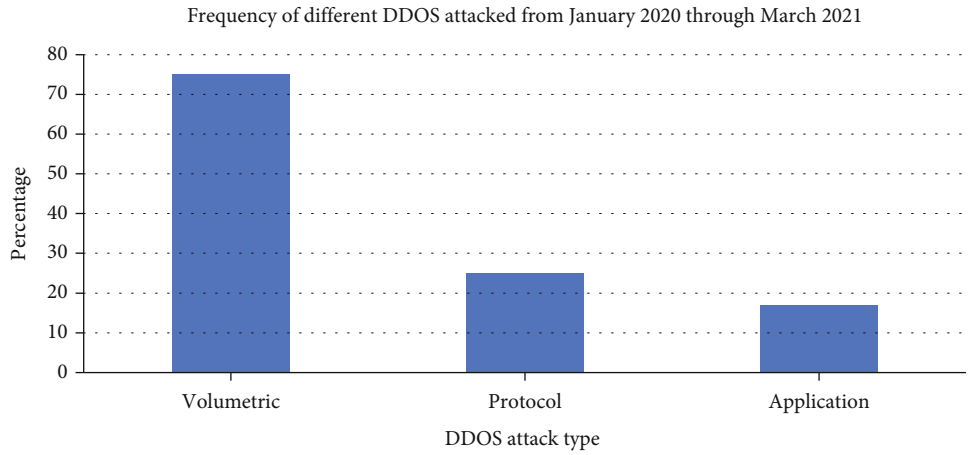

Figure 1: Frequency of different DDoS attacks from January 2020 through March 2021.

generated instances with high accuracy. In [29], the authors have introduced a "DoS-WGAN" model, which is used to generate DoS traffic that looks similar to normal traffic and can bypass a classifier trained using CNN [30], thus reducing the detection rate of the classifier. A "Wasserstein GAN with Gradient Penalty" (WGAN-GP) model [31] is used and implemented for the KDD CUP'99 dataset [32]. Although they were able to reduce the classifier's accuracy to approximately 46.27% from 97.34%, the dataset used is very old. It has several significant issues, such as the huge number of replicated records [33]. Charlier et al. [34] introduced a "SynGAN" framework that can generate synthetic network attacks of high quality using publicly available datasets like NSL-KDD [35] and CICIDS2017 [12]. A root mean square (RMS) error of 0.10 is reported on adversarial generated attacks, showing a close similarity between the artificially generated attacks and real attacks. An area under curve (AUC) score of 75% is also reported, proving that the evaluator cannot differentiate between the real data and the generated synthetic data. In [36], it is shown that even after defensive systems are developed which employ incremental learning, they can still be vulnerable to new attacks if the attack profile is changed. Another challenge while detecting DoS and DDoS attacks is to be able to differentiate between the flash crowds and the actual attacks. An unexpected increase in the number of visitors visiting a website due to some event is known as flash crowds. Gursun et al. [37] first described how to differentiate between DDoS attacks and Flash crowds by statistically characterizing certain traffic features. Later on, in the same paper, the authors proved that even DDoS attack instances could be made to look like flash crowds using AI techniques.

Although [29, 34, 36, 37] have described and proved that an AI model can be trained to generate new synthetic instances and fool the security systems, they have not provided any concrete solution on how a classifier can be trained to detect such kind of generated synthetic adversarial instances. An attacker can use these generated synthetic instances to generate evasion attacks on the security systems to make the classifier misclassify those samples. Having undetected DDoS traffic can turn out to be very costly, and a robust classifier capable of detecting DDoS traffic instances even when there are some changes in the nonattack features of DDoS instances is essential. For this purpose, the GAN framework can generate new DDoS instances and check if the classifier is robust enough to detect such generated synthetic instances. [38] have implemented a GAN-based framework wherein they have used the discriminator model to detect DDoS attacks. Although the discriminator model in GANs can help make the system less sensitive to adversarial attacks, traditional GANs are known to suffer from problems like vanishing gradients and mode collapse. To that end, in this work, a framework consisting of a special type of GAN for the generation and detection of DDoS attacks is proposed.

The contribution of this work is:

(1) Development of a deep neural network- (DNN-) based classifier that can differentiate between DDoS instances and benign instances from the dataset

(2) Development of two separate GAN-based models to generate synthetic traffic instances

(a) First generator that is capable of generating the benign instances which look very similar to benign instances from the dataset

(b) Second generator that is capable of generating the DDoS instances, which look very similar to DDoS instances from the dataset

These synthetic traffic instances (benign and DDoS from a and b, respectively) are used to test the classifier and check if they can predict those generated instances correctly.

(3) Modifying the values present in the DDoS-specific features in the generated benign instances to convert them into DDoS instances. We test if the classifier can predict such adversarial instances as DDoS instances even though they look very similar to

benign ones. This is carried out to check if the trained classifier can detect evasion/adversarial attacks after deployment

(4) Development and demonstration of an approach to train the classifier to differentiate between benign and DDoS instances when both of them look very similar

The rest of the paper is divided as follows: Section 2 gives a brief background of the GAN framework implemented in this work. Section 3 describes the methodology proposed in this research for developing a GAN-based framework for generating and detecting attacks. The performance of the classifier at different stages is also discussed briefly in this section. Section 4 describes the experimentation and results, followed by their analysis. Finally, Section 5 concludes the paper and presents some future work ideas.

2. Materials and Methods

The work proposed in this research primarily employs the generative adversarial network- (GAN-) based framework. In this section, firstly, the basic GAN architecture is discussed. This is followed by the specific approach WGAN employed here.

2.1. Generative Adversarial Networks (GANS) [28]. A GAN model consists of 2 submodels: generator (G) and discriminator (D). The role of the generator is to generate new examples and make the discriminator classify them as the real ones. The role of the discriminator is to classify which examples are generated ones and which are real. This process works as a zero-sum game [39].

The accuracy of the generator is defined by how well it can fool the discriminator by making the discriminator classify its generated examples as real ones. The accuracy of the discriminator is measured by how well it can differentiate between the examples generated by the generator and the real examples. Essentially, both G and D networks strive to train better, and the model achieves convergence where further improvement in outcomes is not possible.

The methodology for training the GAN model (refer to Figure 2) is as follows:

(1) Initially, a random noise vector is given to the generator submodel

(2) The generator tries to produce some examples from the noise vector given to it

(3) The generated example is then passed on to the discriminator to classify it as real or generated

(4) Based on the output of the discriminator, the generator is trained to make it generate even better examples that can fool the discriminator

(5) Similarly, based on the discriminator's ability to classify the generated examples as real ones or fake ones, the discriminator is trained to classify the examples more correctly

Mathematically, the loss function of a GAN model can be defined as [28]:

$$\min_G \max_D V(D,G) = E_{x\sim p_{\text{data}}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))], \tag{1}$$

where $p_z(z)$ is the input noise variable; the goal of the generator is to generate new adversarial samples $G(z)$ that come from the same distribution of x. The discriminator model "D" returns the probability $D(x)$ that the given sample "x" is not generated by G and is actually from a real dataset. The goal of G is to maximize the probability of D predicting the generated data as a real one, whereas for D, the goal is to minimize this probability.

The GAN model that is mentioned in Section 3.1 faces the problem of vanishing gradients and mode collapse. To avoid this problem, a special type of loss function known as "Wasserstein Loss" is used by [40].

2.2. Wasserstein Generative Adversarial Networks (WGANS) [40, 41]. A WGAN is a type of GAN that uses Wasserstein Loss as the loss function. In GANS, the role of the discriminator is to identify the probability of the given sample being real or fake. But in the case of WGANS, instead of having a discriminator, a critic is present whose job is to identify how real or fake the given sample is instead of just predicting the probability of the given sample being real or fake. That is, the critic predicts the realness of the given sample.

The WGAN function is given by [31]:

$$\min_G \max_{D\in D} E_{x\sim P_r}[D(x)] - E_{\tilde{x}\sim P_g}[D(\tilde{x})]], \tag{2}$$

where P_r is original data distribution, P_g is the generative model distribution, $D(x)$ is the predictions made by the critic on original data distribution, and $D(\tilde{x})$ is the predictions made by the critic on generated data samples. The goal of the generator is to minimize the distance between $D(x)$ and $D(\tilde{x})$, whereas the goal of the critic is to maximize the distance between $D(x)$ and $D(\tilde{x})$.

2.3. Wasserstein Generative Adversarial Networks with Gradient Penalty (WGAN-GP) [31]. To further optimize WGAN, a gradient norm penalty method was introduced by [31] to generate samples of even high quality. The WGAN-GP function is given as [31]:

$$L = E_{\tilde{x}\sim P_g}[D(\tilde{x})] - E_{x\sim P_r}[D(x)] + \lambda E_{\hat{x}\sim P_\alpha}\left[\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1\right)^2\right]. \tag{3}$$

Wasserstein loss augmented with a gradient norm penalty for random samples $\hat{x} \sim P_\alpha$ to achieve Lipschitz continuity. The 2nd part of the equation is the applied gradient norm penalty function as discussed in this section. In this work, the WGAN-GP model is employed.

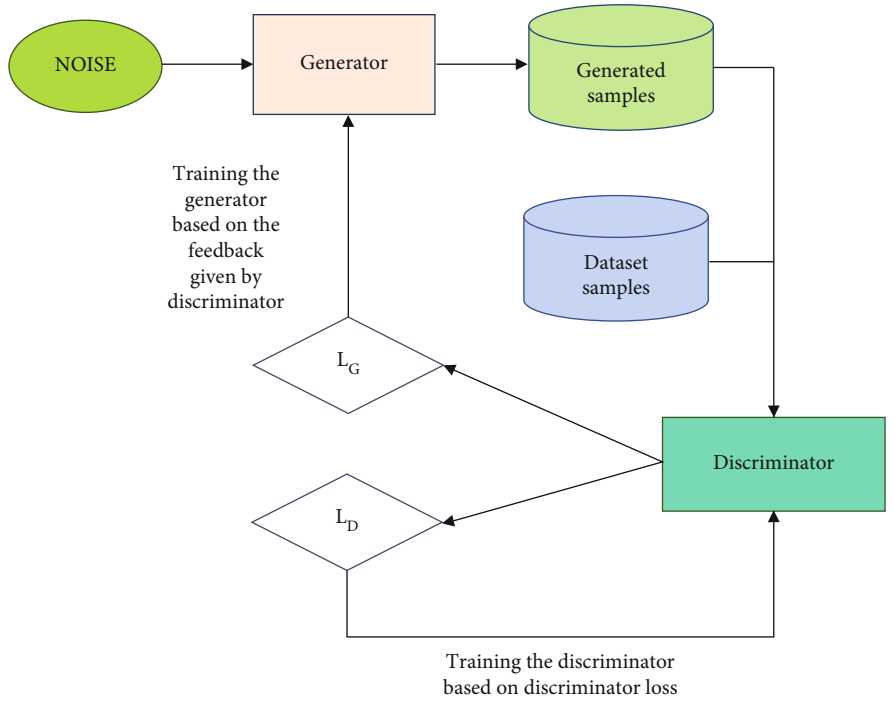

Figure 2: GAN training process.

Table 1: Abbreviations used in this and further sections.

b	Benign/nonmalicious samples from the dataset
m	DDoS/malicious samples from the dataset
bg	Generated benign instances which look like b
mg	Generated DDoS instances which look like m
bgm	Generated DDoS instances which look like b
clf-1	Classifier trained only on b and m
clf-2	Classifier trained on b, m, bg and bgm

3. Methodology

The proposed methodology based on the GAN-based model is described in this section. The performance of the classifier at various stages is also briefly mentioned. Detailed results are mentioned in Section 4.

For the simplicity purpose and to improve readability, the abbreviations shown in Table 1 are used throughout this work.

3.1. Datasets. Two different datasets are employed for experimentations and validation.

(i) CIC-DDoS2019 [19] dataset which contains the most common types of DDoS attacks. From this dataset, 533052 samples of UDP-based DDoS attacks were considered for the study. UDP-based DDoS attack is a type of volumetric DDoS attack. 3134 samples of benign traffic data are also considered. Since the data in this dataset was not balanced, we augmented the benign class data from another dataset, namely, CIC-IDS2017 [12]

(ii) CIC-IDS2017 [12] dataset contains benign traffic data and some other types of attack data. 529918 samples of benign data instances from this dataset are combined with the 3134 samples from the CIC-DDoS2019 [19] dataset

After collecting data from both datasets, the two-class data were merged based on common features. The features which were not common to both datasets were not considered. Finally, the combined dataset has a total of 533052 instances of UDP-based DDoS attack data, a total of 533052 instances of benign traffic data, and 79 features.

3.2. Preprocessing. The first step of data preprocessing is to change the target label. Label "0" for benign data and "1" for UDP-based DDoS attack data is used. Followed by this, some of the features were omitted, either because they were unnecessary or because the data distributions in those features were highly uneven. Further, all the infinite values were replaced with the maximum value of that feature. Finally, to scale all the data evenly, a min-max scaler was used. After carrying out all the preprocessing steps, our final dataset has 533052 instances of UDP-based DDoS attack data, 533052 instances of benign traffic data, and 54 features. Figure 3 describes the preprocessing steps.

3.3. Model Architecture. The proposed framework consists of one classifier and two generator models. The primary goal of the classifier is to classify the given input data as a benign one or a DDoS attack. The classifier is a DNN-based model with 5 layers consisting of 128, 64, 32, 16, and 1 neuron[s], respectively. Rectified linear unit (ReLU) activation [42] is employed in the first four layers, followed by a sigmoid

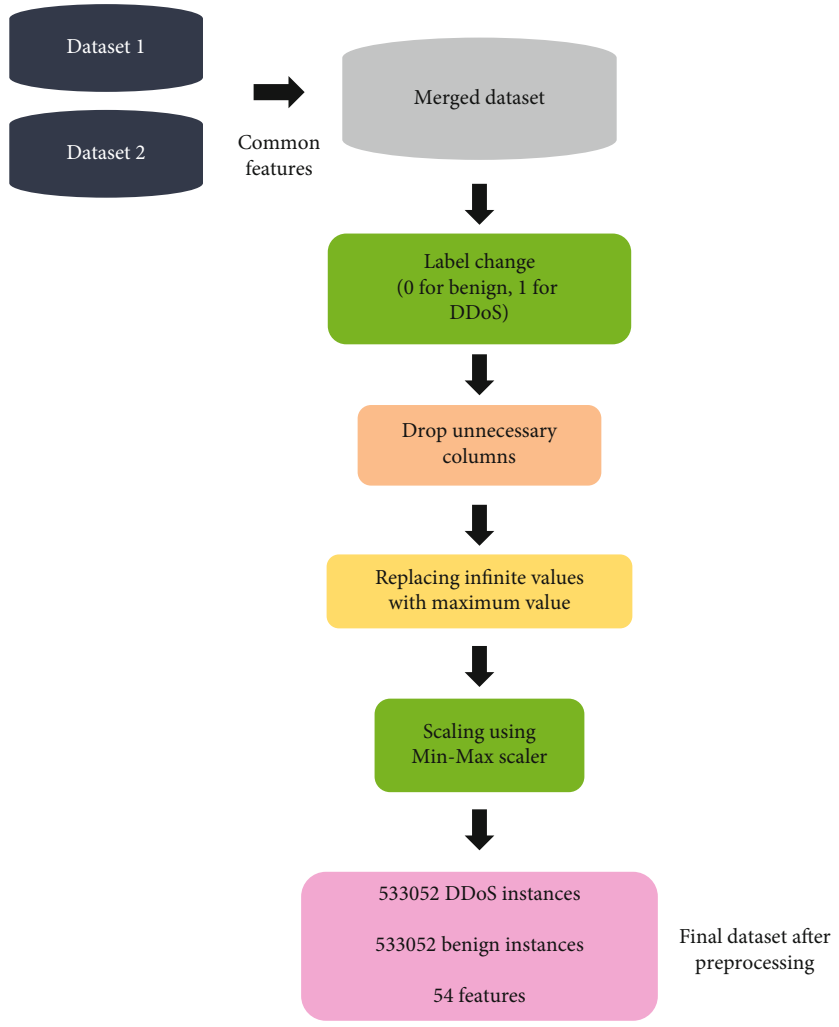

Figure 3: Preprocessing steps.

Table 2: List of DDoS specific attack features (functional features).

FlowDuration	Duration of flow in a millisecond
FwdPacketLengthMax	Max packet size sent in a forward direction
AvgFwdSegmentSize	Average segment size sent in a forward direction
TotalLengthofFwdPackets	The total length of packets sent in the forward direction
BwdPacketLengthStd	Standard packet size sent in the backward direction
AveragePacketSize	The average size of the packet while in transmission
AvgBwdSegmentSize	Average segment size sent backward direction
PacketLengthStd	The standard deviation of packet length
FlowIATStd	The standard deviation of interarrival time between two flows
ACKFlagCount	Packet counts with ACK
BwdPacketLengthMean	Mean of number of packets sent in the backward direction

activation [43] at the last layer. Since the classifier is trying to find the probability of an instance being benign or malicious independently, the sigmoid activation function is implemented instead of the standard Softmax function [43] in the last layer. The binary cross-entropy loss function with ADAM optimizer is employed. Our trained classifier will be able to detect the following two types of attacks after deployment:

(1) Automated adversarial DDoS attacks generated using the methodology similar to the ones suggested by [29, 34, 36, 37]

Table 3: Cosine similarities of bg, mg, and bgm with that of dataset instances.

Instances [sample 1, sample 2]	Mean cosine similarity (20 samples of sample 1 with the whole sample 2 dataset)
[bg, b]	85.40
[mg, m]	95.37
[bgm, b]	77.74

(2) Instances that look very similar to the benign ones where some DDoS specific attack features were manually manipulated to make classifier misclassify them as benign instances

Along with detecting these two types of attacks, our classifier will also be able to differentiate the benign instances from the malignant DDoS instances, thus reducing the number of false positives after deployment.

The two generators are WGAN-GP models. The role of the first generator is to generate bg instances, and the role of the second generator is to generate mg instances. A WGAN-GP model consists of 2 submodels: a generator and a critic. The generator is responsible for generating the required instances. The role of the critic is to give feedback to the generator on these generated instances. Based on the feedback given by the critic, the generator learns to generate instances of even higher quality. The generator in model 1 will generate benign instances, and the generator in model 2 will generate the malignant DDoS instances.

The generators in both the models consist of 5 layers with 128, 64, 64, 32, and 54 neurons. ReLU activation function is used in all the layers except the last layer, where the LeakyReLU [44] activation function with the alpha value of -0.01 is employed. A negative alpha value is used in the last layer since a network packet can never have negative value data inside it and should preferably have a nonzero positive value.

The critics in both the models have five layers with 128, 64, 64, 32, and 1 neuron(s). For layers 1, 2, and 3, the ReLU activation function is employed. No activation function is used for the last two layers to allow the critic to use negative values in its output to demonstrate its predicted realness to the given input [45].

3.4. Model Training

(1) First, both b and m datasets are split into training and testing sets. The training sets are denoted as b_train and m_train and testing sets as b_test and m_test. The classifier is trained using b_train and m_train. After training, the classifier is tested on b_test and m_test. The classifier can predict the instances with high accuracy

(2) Next, two generators are trained: the first generator is trained to generate bg instances that look very similar to b, and the second generator is trained to generate mg instances that look very similar to m. After generating mg instances, the values present in DDoS-specific attack features are modified with the values of the DDoS instances from the dataset (m). This is done to validate the attack

The DDoS specific attack features are shown in Table 2.

(3) The trained classifier is retested on bg and mg and found to be able to correctly predict those instances. From this, we conclude that our trained classifier can detect adversarial attacks generated using AI models when there are no manual changes made in any of the feature values in the generated synthetic instances

(4) Further, we test if the classifier will be able to detect the attack when there are some manual changes made in the attack features of the generated benign instances. For this, bg instances are considered, and the values present in DDoS-specific attack features are modified with the values of DDoS instances (m) from the dataset. This way, the generated benign instances get converted to DDoS instances, and they look very similar to b. These new instances are "bgm" as described in Table 1

The cosine similarities of bg, mg, and bgm with that of the original dataset are mentioned in Table 3.

(5) The classifier is tested on bgm. The classifier is predicting them as "benign" even though they are DDoS instances. From this, it can be concluded that the classifier trained only using the dataset cannot correctly predict the instances based on DDoS-specific attack features. As mentioned in Table 1, at this stage, the classifier is termed as "clf-1"

(6) Next, the bg and bgm instances are split into training and testing sets. To make the classifier robust enough, further training of the classifier is carried out on bg_train and bgm_train. This way, the classifier learns to differentiate between DDoS and benign instances giving more weightage to the attack features

(7) The classifier is tested on bg_test, bgm_test, b, and m. The classifier is correctly able to predict all the instances. At this stage, the classifier is termed as "clf-2"

Figure 4 shows the workflow of the implementation.

The detailed results and comparison between the working of clf-1 and clf-2 are mentioned in Section 4.

4. Results and Discussion

Since most of the bg, mg, and bgm instances were used up in training the classifier, new bg and mg instances are generated to check the performance of both the classifiers "clf-1" and "clf-2" on newly generated data. New bgm instances are also generated using the same approach previously employed to generate bgm instances. These newly generated

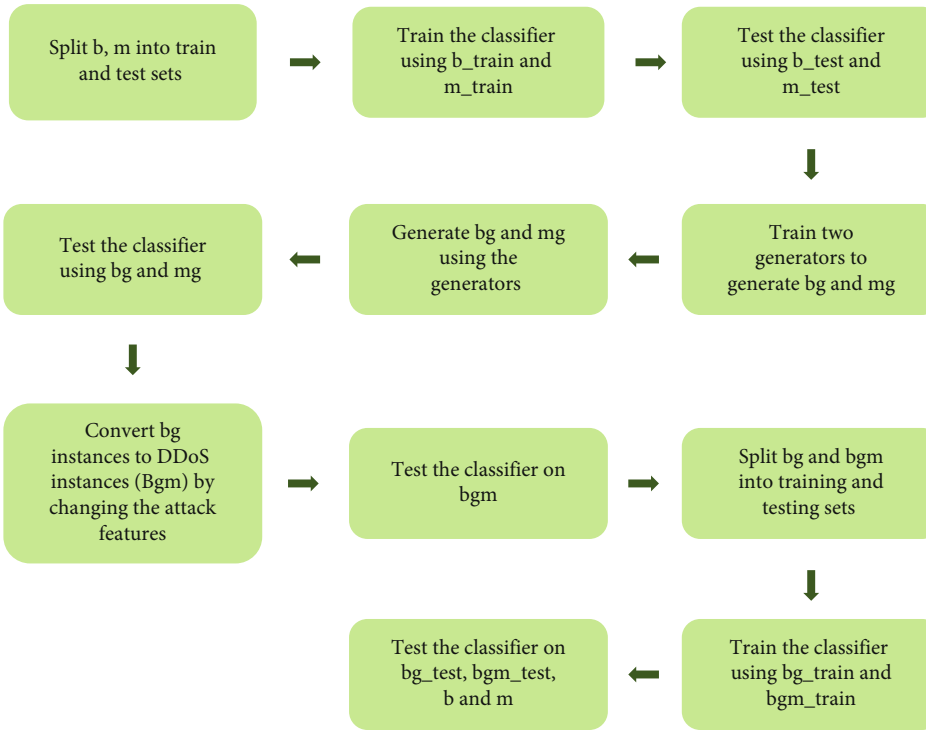

Figure 4: Workflow of the implementation.

Table 4: Cosine similarity table for newly generated samples used for testing the classifier.

Instances [sample 1, sample 2]	Mean cosine similarity (20 samples of sample 1 with the whole sample 2 dataset)
[bg_new, b]	85.34
[mg_new, m]	95.63
[bgm_new, b]	78.07
[bgm_new, bg_new]	78.62

Table 5: Confusion matrix for clf-1 on [bg_new, mg_new].

	Predicted benign	Predicted DDoS
Actual benign	533052	0
Actual DDoS	0	533052

Table 6: Confusion matrix for clf-1 on [bg_new, bgm_new].

	Predicted benign	Predicted DDoS
Actual benign	533052	0
Actual DDoS	533052	0

Table 7: Confusion matrix for clf-2 on [bg_new, mg_new].

	Predicted benign	Predicted DDoS
Actual benign	532880	172
Actual DDoS	0	533052

instances are termed as "bg_new," "mg_new," and "bgm_new." The cosine similarities of newly generated instances are mentioned in Table 4.

From Tables 3 and 4, it can be concluded that both the generators can generate instances that look very similar to benign and DDoS instances from the dataset. The predictions made by the classifiers are mentioned below:

4.1. B, M. B is the combination of bg_new and b, and M is the combination of mg_new, bgm_new and m.

As can be observed from Table 5, the classifier trained only using the dataset can correctly classify the generated benign and DDoS instances when they look like benign and DDoS instances from the dataset. Therefore, this classifier, which is trained only using the dataset, will be able to classify both generated adversarial attack instances and benign instances correctly, similar to the discriminator model trained by [38] as long as there are no manual changes made in the input features of the generated instances. But from Table 6, it is concluded that such a classifier will not be able to predict the attack if the values of DDoS-specific attack features are changed in benign instances to make them malicious. From Tables 7–10, we conclude that the classifier trained using the approach suggested in this work is correctly able to differentiate between DDoS attack instances and benign instances using the attack features and will be able to detect DDoS attacks with high accuracy even if someone tries to make them look as benign as possible by manually changing some features.

Table 8: Confusion matrix for clf-2 on [bg_new, bgm_new].

	Predicted benign	Predicted DDoS
Actual benign	532880	172
Actual DDoS	27	533025

Table 9: Confusion matrix for clf-1 on [B, M].

	Predicted benign	Predicted DDoS
Actual benign	1066097	7
Actual DDoS	533053	1066103

Table 10: Confusion matrix for clf-2 on [B, M].

	Predicted benign	Predicted DDoS
Actual benign	1065164	940
Actual DDoS	28	1599128

5. Conclusion and Future Work

With the recent advancements in AI, many attackers have started using AI to generate adversarial attacks and bypass the security systems. Therefore, it is necessary to build the security systems that are robust enough so that they can correctly identify different types of adversarial attacks, thus helping in preventing them and minimizing the damage.

In this work, we first described how one can generate synthetic instances using GANs. We have used a special type of GAN framework called "WGAN-GP" for generating both benign and DDoS instances. These generated instances can be used to test if the classifiers are robust enough to detect automated attacks generated using AI techniques. Later on, we proved how a classifier, which is correctly able to detect the automated attacks generated using GANs, can be made to misclassify the samples by using other techniques described in adversarial AI paradigm. Lastly, we suggested an approach on how a classifier can be trained to detect evasion attacks by changing the DDoS attack features in the generated benign instances manually. This approach can also be used to test the classifier after deployment.

Recently many new AI techniques and frameworks are being introduced. Therefore, it is important to understand how one can leverage these new techniques in the cybersecurity domain. This work only focuses on volumetric DDoS attacks, since they are one of the most common types of attacks. However we believe that this approach can be used to detect other types of attacks. Similar approach can also be used in malware detection as well.

Data Availability

All the datasets used for training the models are publicly available and their links are provided in the reference section. The preprocessed and the generated datasets can be accessed via the link provided below: https://drive.google.com/drive/folders/1lu-cf0RLj0R7AiooLmGeMh8b1_nCRZS1?usp=sharing.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Symbiosis Centre for Applied Artificial Intelligence (SCAAI) and Symbiosis International University (SIU) under its research support fund.

References

[1] PURPLESEC, "Cyber-security-statisticsPURPLESECSeptember 2021, https://purplesec.us/resources/cyber-security-statistics/.

[2] A. Bhattacharya, "Active-and-passive-attacks," Encryption Consulting, 2021, September 2021, https://www.encryptionconsulting.com/active-and-passive-attacks/.

[3] N. Hassan, "What-active-attack-vs-passive-attack-using-encryption," Venafi, 2020, September 2021, https://www.venafi.com/blog/what-active-attack-vs-passive-attack-using-encryption.

[4] Paloalto networks, "What-is-a-denial-of-service-attack-dos-Palo Alto NetworksSeptember 2021, https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos.

[5] Imperva, "Man-in-the-middle-attack-mitm/ImpervaSeptember 2021, https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/.

[6] Kaspersky, "TrojansKasperskySeptember 2021, https://www.kaspersky.co.in/resource-center/threats/trojans.

[7] J. Melnick, "Top-10-most-common-types-of-cyber-attacks," Netwrix Blog, 2018, September 2021, https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/.

[8] D. Warburton, "Ddos-attack-trends-for-2020," F5 Labs, 2021, September 2021, http://www.f5.com/labs/articles/threat-intelligence/ddos-attack-trends-for-2020.

[9] S. M. Poremba, "Types-of-ddos-attacks," eSecurity Planet, 2017, September 2021, https://www.esecurityplanet.com/networks/types-of-ddos-attacks/.

[10] S. Sambangi and L. Gondi, "A machine learning approach for DDoS (distributed denial of service) attack detection using multiple linear regressioN," *Proceedings*, vol. 63, no. 1, p. 51, 2020.

[11] J. Brownlee, "Linear-regression-for-machine-learning/," Machine Learning Mastery, 2020, September 2021, https://machinelearningmastery.com/linear-regression-for-machine-learning/.

[12] "Ids-2017.htmlUniversity of New Brunswick - Canadian Institute for CybersecuritySeptember 2021, https://www.unb.ca/cic/datasets/ids-2017.html.

[13] J. Brownlee, "Information-gain-and-mutual-information/," Machine Learning Mastery, 2020, September 2021, https://machinelearningmastery.com/information-gain-and-mutual-information/.

[14] J. Brownlee, "Ensemble-methods-for-deep-learning-neural-networks/," Machine Learning Mastery, 2019, September 2021, https://machinelearningmastery.com/ensemble-methods-for-deep-learning-neural-networks/.

[15] 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, Bhubaneswar, India, December 2018.

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

[17] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, article 132306, 2020.

[18] J. Jordan, "Autoencoders/," 2018, September 2021, https://www.jeremyjordan.me/autoencoders/.

[19] "Ddos-2019.htmlUniversity of New Brunswick - Canadian Institute for CybersecuritySeptember 2021, https://www.unb.ca/cic/datasets/ddos-2019.html.

[20] 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, Hong Kong, China, May 2017.

[21] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.

[24] "ids.htmlUniversity of New Brunswick - Canadian Institute for CybersecuritySeptember 2021, https://www.unb.ca/cic/datasets/ids.html.

[25] 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, 2019.

[26] A. Kurakin, G. Brain, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," https://arxiv.org/pdf/1611.01236.pdf.

[27] I. Goodfellow, P. McDaniel, and N. Papernot, "Making machine learning robust against adversarial inputs," *Communications of the ACM*, vol. 61, no. 7, pp. 56–66, 2018.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," 2014, https://arxiv.org/pdf/1406.2661.pdf.

[29] 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.

[30] J. Brownlee, "Convolutional-layers-for-deep-learning-neural-networks/," Machine Learning Mastery, 2020, September 2021, https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/.

[31] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," https://arxiv.org/pdf/1704.00028.pdf.

[32] "kddcup99.html," UCI, 1999, September 2021, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[33] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT," *Procedia Computer Science*, vol. 167, pp. 1561–1573, 2020.

[34] J. Charlier, A. Singh, G. Ormazabal, R. State, and H. Schulzrinne, "SynGAN: towards generating synthetic network attacks using GANs," September 2021, https://arxiv.org/pdf/1908.09899.pdf.

[35] "nsl.htmlUniversity of New Brunswick - Canadian Institute for CybersecuritySeptember 2021, https://www.unb.ca/cic/datasets/nsl.html.

[36] R. Chauhan and S. Shah Heydari, "Polymorphic adversarial DDoS attack on IDS using GAN," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, Montreal, QC, Canada, October 2020.

[37] G. Gursun, M. Sensoy, and M. Kandemir, "On context-aware DDoS attacks using deep generative networks," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, Hangzhou, China, 2018.

[38] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proença Jr., "Adversarial deep learning approach detection and defense against DDoS attacks in SDN environments," *Future Generation Computer Systems*, vol. 125, pp. 156–167, 2021.

[39] "Zero-sum_gameWikipediaSeptember 2021, https://en.wikipedia.org/wiki/Zero-sum_game.

[40] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," https://arxiv.org/pdf/1701.07875.pdf.

[41] J. Brownlee, "How-to-implement-wasserstein-loss-for-generative-adversarial-networks/," Machine Learning Mastery, 2019, September 2021, https://machinelearningmastery.com/how-to-implement-wasserstein-loss-for-generative-adversarial-networks/.

[42] J. Brownlee, "Rectified-linear-activation-function-for-deep-learning-neural-networks/," Machine Learning Mastery, 2020, September 2021, https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

[43] "Activations/Keras.ioSeptember 2021, https://keras.io/api/layers/activations/.

[44] "Leaky-relupaperswithcodeSeptember 2021, https://paperswithcode.com/method/leaky-relu.

[45] "Improved_wgan.pyGitHubSeptember 2021, https://github.com/keras-team/keras-contrib/blob/master/examples/improved_wgan.py.