Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064833

ARTICLE

DEMGAN: A Machine Learning-Based Intrusion Detection System Evasion Scheme

Dawei Xu[1,2,3], Yue Lv[1], Min Wang[1], Baokun Zheng[4,*], Jian Zhao[1,3] and Jiaxuan Yu[5]

[1]College of Computer Science and Technology, Changchun University, Changchun, 130022, China
[2]School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, 100081, China
[3]Key Laboratory of Intelligent Rehabilitation and Barrier-free for the Disabled (Changchun University), Ministry of Education, Changchun, 130022, China
[4]School of Information Management for Law, China University of Political Science and Law, Beijing, 102249, China
[5]College of Artificial Intelligence, Nankai University, Tianjin, 300350, China

*Corresponding Author: Baokun Zheng. Email: zhengbk@cupl.edu.cn

Received: 25 February 2025; Accepted: 16 April 2025; Published: 09 June 2025

ABSTRACT: Network intrusion detection systems (IDS) are a prevalent method for safeguarding network traffic against attacks. However, existing IDS primarily depend on machine learning (ML) models, which are vulnerable to evasion through adversarial examples. In recent years, the Wasserstein Generative Adversarial Network (WGAN), based on Wasserstein distance, has been extensively utilized to generate adversarial examples. Nevertheless, several challenges persist: (1) WGAN experiences the mode collapse problem when generating multi-category network traffic data, leading to subpar quality and insufficient diversity in the generated data; (2) Due to unstable training processes, the authenticity of the data produced by WGAN is often low. This study improves WGAN to address these issues and proposes a new adversarial sample generation algorithm called Distortion Enhanced Multi-Generator Generative Adversarial Network (DEMGAN). DEMGAN effectively evades ML-based IDS by proficiently obfuscating network traffic data samples. We assess the efficacy of our attack method against five ML-based IDS using two public datasets. The results demonstrate that our method can successfully bypass IDS, achieving average evasion rates of 97.42% and 87.51%, respectively. Furthermore, empirical findings indicate that retraining the IDS with the generated adversarial samples significantly bolsters the system's capability to detect adversarial samples, resulting in an average recognition rate increase of 86.78%. This approach not only enhances the performance of the IDS but also strengthens the network's resilience against potential threats, thereby optimizing network security measures.

KEYWORDS: Adversarial attacks; intrusion detection; adversarial traffic examples; DEMGAN

1 Introduction

Network intrusion detection systems (IDS) function as a proactive defense mechanism aimed at identifying and addressing suspicious or malicious activities within networks and systems [1–3]. With the continuous development of adversarial sample generation technology, traditional signature-based intrusion detection systems are facing challenges in meeting the growing detection demands [4]. In recent years, machine learning (ML), particularly deep learning (DL), already widely used in the IDS field, however, they exhibit a level of susceptibility to inaccuracies, leading to a significant occurrence of false positives [5]. Moreover, the susceptibility of ML models to manipulation of input data by malicious actors has led to the

emergence of a novel form of network traffic attack known as adversarial attacks [6]. Adversarial attacks attempt to fool ML models by making subtle changes to input data.

Adversarial attacks can be categorized into two main types: black-box and white-box attacks. While certain traditional white-box attack methods [7–10] have been successful against neural networks and their ability to bypass defense mechanisms, some approaches may not be feasible in specific practical contexts due to implementation constraints. Therefore, black-box attack method is crucial in contemporary network security threats. State-of-the-art black-box attack strategies include techniques based on Generative Adversarial Networks (GANs) [11], transfer learning methods, and embedded adversarial sample generation approaches. GAN generates traffic with malicious intent to evade IDS through adversarial training between the generator and the discriminator.

Although GAN has made great achievements in the field of adversarial attacks, a key challenge persists, known as pattern collapse [12]. There are many reasons for mode collapse, one of which is insufficient generator capacity. Given that the GAN model primarily generates data through the interplay between the generator and the discriminator, the selection of the loss function directly influences the diversity and quality of the generated data [13,14]. Therefore, we still need to further explore more complex and advanced GAN technologies to increase the diversity and authenticity of adversarial samples.

This paper proposes an adversarial attack method called DEMGAN to cover up malicious traffic and evade IDS detection. This not only has a positive impact on network security but also plays an important role in privacy protection [15]. The main contributions of this paper are summarized as follows.

1. We use WGAN [16] as the base model and introduce multiple generators structure in DEMGAN to solve the mode collapse problem of traditional GAN, this approach enhances the ability to conceal malicious traffic and enhances the adaptability of the model. Experimental results demonstrate that DEMGAN, utilizing multiple generators structures, achieves an average evasion rate enhancement of 17.79% compared to WGAN with a single generator structure.
2. To enhance the ability to conceal malicious traffic data while preserving its malicious nature, a distortion rate is incorporated into DEMGAN to quantify the variance between the generated data and the initial data. Experimental results demonstrate that the evasion rate achieved by DEMGAN exhibits an average improvement of 21.34% in comparison to WGAN.
3. We performed experiments using the CICIDS2017 and CICIDS2018 datasets to evaluate the effectiveness of our proposed DEMGAN in evading ML-based IDS. Our findings show that DEMGAN achieved an average evasion rate increase of 22.89% compared to WGAN. In addition, retraining IDS with adversarial examples generated by DEMGAN can improve the detection ability of IDS.

2 Motivation

As network attacks become increasingly complex and diverse, IDS, as the first line of defense for network security, is of great importance. In recent years, intrusion detection algorithms based on machine learning have gradually become mainstream due to their powerful pattern recognition capabilities and adaptive characteristics. GAN, as a powerful generative model, can generate samples that are highly similar to real data by learning data distribution. Using GAN to generate adversarial samples can not only simulate the behavior of attackers and reveal potential vulnerabilities in intrusion detection systems, but also provide defenders with a new tool for enhancing the security of the system. By using the generated adversarial samples for retraining intrusion detection systems, the model's resistance to adversarial attacks can be significantly improved, thereby improving overall network security.

This study aims to explore the adversarial sample generation technology based on GAN and evaluate its effectiveness in evading intrusion detection systems. At the same time, we further study how to use these adversarial samples to enhance the robustness of intrusion detection systems. Through this study, we hope to provide a new defense idea for the field of network security and help build a more secure and reliable network environment.

3 Related Work

3.1 Network Intrusion Detection Technology

IDS is an important component of network security, which aims to identify potential attacks or abnormal activities by monitoring network traffic or system behavior. In recent years, with the increasing complexity and diversity of network attacks, traditional IDS methods face many challenges, such as insufficient detection capabilities for new attacks, high false alarm rates, and limited processing capabilities for encrypted traffic. To address these challenges, machine learning techniques are widely used in intrusion detection systems. For example, supervised learning methods (such as SVM and random forest) are used for network traffic classification and attack identification, while unsupervised learning methods are used for anomaly detection. In addition, deep learning techniques [17,18] excel in processing high-dimensional data and nonlinear relationships and are used to detect complex network attacks.

3.2 Adversarial Machine Learning

Adversarial machine learning is to attack machine learning models by generating adversarial samples, or to improve the model's ability to resist such attacks. Adversarial machine learning can be divided into two main categories: adversarial attacks and adversarial defenses. Adversarial attacks can be divided into white-box attacks and black-box attacks. In recent years, research [19] has employed white-box attack methods to carry out traffic attacks. However, in practical scenarios, attackers often face challenges in obtaining comprehensive internal information about the target system due to its confidential and sensitive nature. Consequently, black-box attacks have emerged as a more viable option.

We utilize GANs in our algorithms for two main reasons. Firstly, GANs have demonstrated strong performance in various domains, and the data produced by GANs can mislead recognition systems across different domains. Secondly, GANs possess a distinctive advantage in data generation by understanding the distributional characteristics of the data and creating samples that closely resemble real data. This proficiency is especially vital in areas where data scarcity or difficulty in obtaining data is a prevalent issue. GAN continuously optimize the generator's capacity to produce authentic samples through adversarial training. Consequently, GANs emerge as an optimal option for generating adversarial samples, particularly for intrusion detection systems [20–24].

4 Methodology

In this section, we propose a method called DEMGAN for generating adversarial samples of malicious traffic to evade ML-based IDS. We use WGAN as the base model and enhance it.

Overview: Fig. 1 illustrates our process of generating adversarial examples, which is divided into two parts: data preprocessing and adversarial example generation based on DEMGAN.

4.1 Data Preprocessing

The data preprocessing part is divided into three stages: data cleaning, feature extraction, and data normalization.

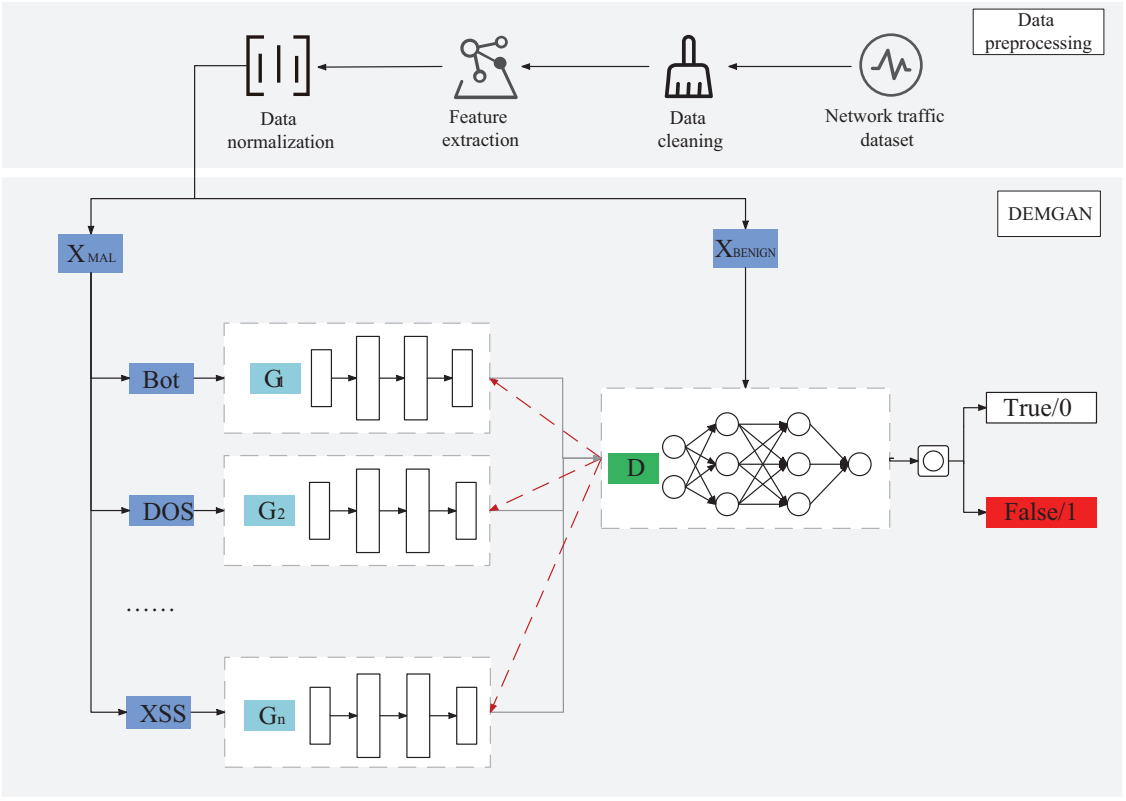


Figure 1: Adversarial example generation process

Data Cleaning: Data cleaning is one of the crucial steps in data preprocessing, ensuring that the model receives accurate and stable input during the training process. In numerous datasets, outliers like NaN or Infinity may exist, potentially impacting model training negatively. When dealing with data containing NaN or Infinity, the approach in this article is to remove the data samples containing these outliers. This practice helps prevent the outliers from interfering with model training, ensuring the quality and consistency of the dataset, as well as the accuracy and stability of model training.

Feature Extraction: This paper utilizes mutual information [25] as a method for feature extraction to compute the correlation coefficient between each feature and the label associated with the data sample. By computing mutual information, the significance of each feature in predicting labels can be assessed, enabling the identification of the most beneficial features for the model's predictions. In mutual information, let X and Y denote two random variables that can represent any form of data, such as text, images, etc. Their joint probability distribution is denoted as $P(X, Y)$, and their marginal probabilities are $P(X)$ and $P(Y)$. The mutual information between X and Y is typically denoted as $(X; Y)$ and is defined by Eq. (1).

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} P(X,Y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right) \tag{1}$$

Data Normalization: This article employs the maximum-minimum normalization method to standardize the data. Max-min normalization is a widely used data preprocessing technique. It scales the original data to the range $[0, 1]$, ensuring a consistent scale for the data, which aids in model training and convergence. The formula for max-min normalization is presented in Eq. (2).

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

4.2 Adversarial Example Generation

This paper proposes the DEMGAN model for generating adversarial examples. The DEMGAN model comprises multiple generators and a discriminator. The generator creates adversarial example data, while the discriminator assesses the authenticity of the generated adversarial examples.

4.2.1 WGAN

WGAN was proposed by Martin et al. in 2017. It is a variant of GAN. The significance of Wasserstein distance is to measure the minimum moving cost in real space between two distributions. In WGAN, by minimizing the Wasserstein distance between the generated distribution and the real distribution, the generator can be prompted to generate samples that are closer to the real data distribution, thereby improving the performance and stability of the generated model. In WGAN, for a given real distribution (P_r) and generator distribution (P_g), the Wasserstein distance is defined as Eq. (3). Among them, $\prod(P_r, P_g)$ represents the set of all joint distributions $\gamma(x, y)$, whose margins are P_r and $P_g \cdot \gamma(x, y)$ represents that to convert the distribution P_r into P_g, it must be converted from the "quality" of transmission from x to y, and $W(P_r, P_g)$ is the cost of the optimal transmission solution, which is the "price" that must be paid.

$$\mathrm{W}\left(\mathrm{P_r}, \mathrm{P_g}\right) = \inf_{\gamma \in \Pi\left(\mathrm{P_r}, \mathrm{P_g}\right)} \mathrm{E}_{(\mathrm{x},\mathrm{y})\sim\gamma}[||\mathrm{x} - \mathrm{y}||] \tag{3}$$

Since the information in the Wasserstein distance cannot be solved directly, a known theorem can convert the Wasserstein distance into Eq. (4). Here, 1/K represents a normalization factor used to adjust the effect of the Lipschitz constant K, sup represents the supremum, $E_{x\sim P_r}[f(x)]$ represents the expected value of function f under the real data distribution, $E_{x\sim P_g}[f(x)]$ represents the expected value of the function f under the generated data distribution, $\|f\|_L \leq K$ is the Lipschitz condition, which means that the Lipschitz constant of function f does not exceed K, $f(x)$ is a K-Lipschitz continuous function, usually implemented by the discriminator.

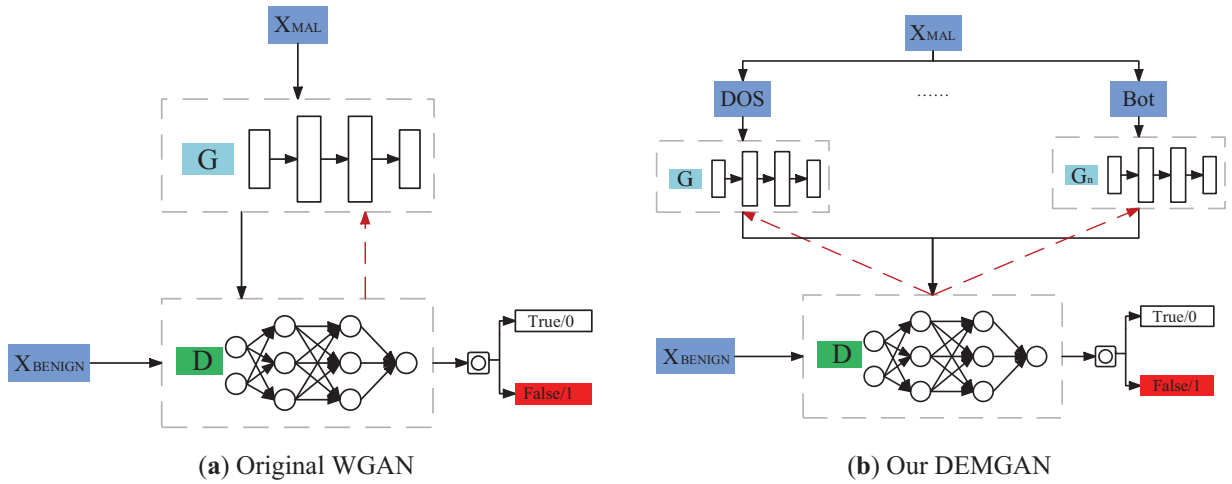
$$W\left(P_r, P_g\right) = \frac{1}{K} \sup_{\|f\|_L \leq K} E_{x\sim P_r}[f(x)] - E_{x\sim P_g}[f(x)] \tag{4}$$

4.2.2 DEMGAN

This article discusses the unresolved issues in WGAN, maintains the utilization of the Wasserstein distance method in WGAN, and integrates Transformer as the generator architecture, and RNN as the discriminator architecture, resulting in the development of a robust generative adversarial network called DEMGAN. We have mainly made two improvements based on WGAN, namely a multi-generator structure and loss function.

Multi-generator Structure: In WGAN, the task of the generator is often much more challenging than that of the discriminator. The generator must strive to produce highly realistic data to minimize the discriminator's errors. Although WGAN solves some problems inherent in GAN, the problem persists when dealing with large datasets. Drawing inspiration from the use of multi-generators in the field of image processing [26], we propose incorporating a multi-generator structure into WGAN to enhance generator diversity and improve the quality of generated samples.

We train multiple generators simultaneously in DEMGAN. Each generator is responsible for generating a subset of the data or a specific pattern. In order to effectively input the output of these generators into the discriminator, we adopted the concatenation method, that is, concatenating the subsets generated by multiple generators into a set, and inputting the data in the generated set and the original real data into the discriminator together. The discriminator determines the authenticity of the generated data, and then feeds the result back to the generator to adjust the parameters so that the generator can generate high-quality data. This approach allows each generator to focus on generating a specific pattern, enhancing the simulation and generation of various types of data samples. Consequently, it increases the diversity and authenticity of the generated data. Fig. 2 illustrates the fundamental structural difference between WGAN and DEMGAN.



(**a**) Original WGAN (**b**) Our DEMGAN

Figure 2: Structure of WGAN and DEMGAN

Loss Function Improvement: In WGAN, we minimize the Wasserstein distance by constraining the parameters of the discriminator to be Lipschitz continuous and then optimizing a loss function with truncated gradients. Based on the generator loss function of WGAN, we incorporate the distortion rate to quantify the disparity between real data and generated data. This enhancement can assist the model in more precisely assessing the variance between generated data and real data, and aid the generator in finely tuning the attributes of the generated data, thereby enhancing the training efficacy of the generator and the quality of the generated samples. The final loss function of the DEMGAN generator is defined as Eq. (5). In the formula, $P_g^{(i)}$ represents the ith feature of the original data, $P_r^{(i)}$ represents the ith feature of the generated adversarial samples, and N represents the feature dimension, which is used to measure the strength of the correction. The main reason why we add the distortion rate on the basis of the WGAN loss function is that the distortion rate can measure the distance between the generated data and the real data, further improving the authenticity of the generated data.

$$\mathcal{L}_{\mathcal{G}} = -E_{x\sim P_g}\left[f_w\left(x\right)\right] + \sqrt{\sum_i \left(P_g^{(i)} - P_r^{(i)}\right)^2 / N} \tag{5}$$

According to the definition of DEMGAN, it can be concluded that the purpose of the generator is to generate more realistic data. Therefore, the final training objective of the generator is as depicted in Eq. (6). We consider Eq. (6) as having two parts. The first part pertains to the training objective of the original WGAN, which we will not delve into here. The second part involves the training objective introduced by

DEMGAN, which we will elaborate on. When the data generated by DEMGAN approaches the original data infinitely closely, which can be achieved under ideal conditions, Eq. (7) can be derived. At this point, the DEMGAN generator attains optimal performance.

$$\min\left(-E_{x\sim P_g}\left[f_w\left(x\right)\right]\right)+\min\left(\sqrt{\sum_i\left(P_g^{(i)}-P_r^{(i)}\right)^2/N}\right) \tag{6}$$

$$lim\sqrt{\sum_i\left(P_g{}^{(i)}-P_r{}^{(i)}\right)^2/N}=0 \tag{7}$$

4.3 Evading ML-Based IDS

This paper selects a variety of ML-based IDS algorithms for experiments to verify the effectiveness of DEMGAN in evading ML-based IDS. The specific operations are as follows:

- Use traffic data to train ML-based IDS algorithms.
- The adversarial examples generated by DEMGAN are input into the ML-based IDS algorithm for classification.
- If the ML-based IDS cannot identify the adversarial examples generated by DEMGAN as malicious traffic but benign traffic, it means that the adversarial examples generated by DEMGAN can evade the ML-based IDS, which also means that DEMGAN can evade the ML-based IDS.

5 Experiments and Results

In this section, we experimentally verify the effectiveness of the adversarial examples generated by DEMGAN. First, we provide detailed settings of the experimental process and an introduction to the dataset. Subsequently, we conduct a series of experiments on the proposed adversarial example generation method. Additionally, we perform ablation experiments to confirm the effectiveness of all improvement steps.

5.1 Experimental Setup

5.1.1 Experiment Equipment

Our experimental equipment consists of two parts: a computer equipped with an Intel Xeon Platinum 8352 V processor (Santa Clara, CA, USA) and 90 GB of memory, and an NVIDIA RTX 4090 graphics card (Santa Clara, CA, USA). The computer runs on the Ubuntu 20.04 operating system, and we utilize Python 3.8, PyTorch 2.0, and TensorFlow 2.0 framework to implement our deep learning model. In the experiment, we utilized the CPU for the initial data preprocessing and model training and then switched to the GPU to expedite the model calculations. In order to demonstrate the effectiveness of DEMGAN more intuitively and clearly, in our experiments, both the generator and discriminator models of WGAN and DEMGAN use the Transformer model and RNN model.

5.1.2 Datasets and Preprocessing

This article utilizes the standard data sets CICIDS2017 and CICIDS2018, commonly employed in analyzing traffic attacks. This article removes data containing NaN and Infinity from the dataset. We utilized all the data from the CICIDS2017 dataset for our experiments. For the CICIDS2018 dataset, only a portion of the data was selected for experimentation. Following data cleaning, the traffic categories and quantities for each dataset used in this article are presented in Table 1. Mutual information is used to calculate the

correlation coefficient between each feature in the experimental data and the data label. Table 2 lists the top 10 features with the highest correlation coefficients with traffic data labels in the two datasets.

Table 1: Data distribution in the data set after data cleaning

Dataset	Category	Quantity	Dataset	Category	Quantity
	Benign	2,271,319		Benign	1205,106
	Bot	1956	CICIDS2018	Bot	286,191
	Brute Force	1507		DoSHulk	461,912
	DDoS	128,025		SlowHTTPTest	139,886
CICIDS2017	XSS	625	–	–	–
	SSH Patator	5897	–	–	–
	SQL Injection	21	–	–	–
	PortScan	158,804	–	–	–
	Infiltration	36	–	–	–
	Heartbleed	11	–	–	–
	FTP Patator	7935	–	–	–
	DoS	251,712	–	–	–

Table 2: Characteristics of the top 10 correlation coefficients with data labels in the CICIDS2017 and CICIDS2018 datasets

Ranking	CICIDS2017	CICIDS2018
1	Total length of Fwd packets	Dst port
2	Subflow Fwd bytes	Fwd IAT max
3	Subflow Bwd bytes	Fwd IAT mean
4	Total length of Bwd packets	Flow Pkts/s
5	Bwd packet length mean	Fwd IAT tot
6	Avg Bwd segment size	Flow IAT mean
7	Destination port	Fwd Pkts/s
8	Fwd header length	Flow IAT max
9	Average packet size	Flow duration
10	Bwd packet length max	Fwd IAT min

When using DEMGAN to generate adversarial samples, we considered the balance between malicious intent and evasion detection to ensure that the generated adversarial examples can effectively deceive the intrusion detection system while maintaining the malicious nature of the data, thereby enhancing adversarial attacks. Table 3 illustrates the modifications made to some features in the two datasets. Changed indicates that the feature data modified by DEMGAN can serve as the final data of the adversarial example, unchanged indicates that the feature data still requires the use of the original data that has not been altered by DEMGAN. To provide clarity, we will use specific feature data modification conditions as examples. The source address and destination address are crucial identifiers for network communications, used to denote the sender and receiver of a data packet. Therefore, even if DEMGAN alters the content of malicious traffic, the source address and destination address in the data packet must remain unaltered. The protocol field of the traffic data specifies the transport protocol utilized by the data packet, such as TCP, UDP, or ICMP. This field dictates

how the data packet is processed, the format of the data packet header, and cannot be modified. ACK Flag Cnt denotes the number of acknowledgment flags in a TCP packet, indicating whether the packet includes an acknowledgment message. These flags are defined by the TCP protocol, and their number and significance are vital for the accurate processing of the TCP protocol, hence they cannot be altered. Once information such as ACK Flag Cnt is changed, the authenticity and validity of the traffic data cannot be ensured, and it will not be able to play a role in adversarial attacks, affecting the final experimental results.

Table 3: Modification status of feature data in two sets of datasets

CICIDS2017		CICIDS2018	
Feature	Operate	Feature	Operate
Total length of Fwd packets	Changed	Dst Port	Unchanged
Subflow Fwd bytes	Changed	Flow Duration	Changed
Subflow Bwd bytes	Changed	Total Fwd Packets	Changed
Total length of Bwd packets	Changed	Fwd Packet Length Max	Changed
Destination port	Unchanged	PSH Flag Cnt	Unchanged
Total Fwd packets	Changed	ACK Flag Cnt	Unchanged
URG Flag count	Unchanged	Flow IAT Std	Changed
CWE Flag count	Unchanged	Bwd Header Length	Changed

5.1.3 Evaluation Metrics

We use evasion rate (ER) to measure the success rate of evading ML-based IDS after using DEMGAN for adversarial example generation. The evasion rate is a common evaluation indicator in traffic attacks. It can reflect the effectiveness and practicality of the adversarial example generation algorithm, as well as the difficulty of detecting the adversarial examples. Let P be the total number of malicious traffic, and L be the total number of traffic that the ML-based IDS detection result is benign. The calculation of ER is as shown in Eq. (8).

$$ER = \frac{L}{P} \tag{8}$$

We use accuracy to verify the improvement effect of adversarial samples generated by DEMGAN on the intrusion detection algorithm after retraining. The specific mathematical representation is shown in Eq. (9). TP represents true positive examples, TN represents true negative examples, FP represents false positive examples, and FN represents false negative examples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

5.1.4 IDS Algorithm Selection

This paper evaluates the effectiveness of the proposed DEMGAN using various ML-based intrusion detection systems (IDS) [27]. We selected a range of intrusion detection algorithms such as decision tree (DT), naive Bayes (NB), logistic regression (Logistic), multi-layer perceptron (MLP), and we also selected more complex neural networks such as convolutional neural network (CNN), recurrent neural network (RNN) and CNN-BiLSTM. By adopting these different intrusion detection algorithms, we can

comprehensively evaluate the performance and impact of DEMGAN, as well as its applicability to various types of intrusion detection algorithms.

5.2 DENGAN's Improvement Proof

In this section, we use experiments to prove the effectiveness of DEMGAN's two improvements.

5.2.1 Solve the Mode Collapse Problem

Table 4 presents the experimental results of evasion rates using WGAN and DEMGAN for five categories of malicious traffic data. The results indicate that malicious traffic disguised using DEMGAN is more successful in evading ML-based IDS compared to traffic disguised using WGAN. Consequently, the adversarial examples produced by DEMGAN exhibit greater diversity than those generated by WGAN.

Table 4: Comparative experiment on multi-category evasion rate

		Doshulk	DoSGoldenEye	DoSGoldenEye	FTP-Patator	PortScan
DT	WGAN	51.04%	58.79%	54.44%	61.56%	56.88%
	DEMGAN	95.65%	97.15%	100%	100%	100%
Logistic	WGAN	48.22%	48.15%	48.25%	47.68%	48.01%
	DEMGAN	100%	100%	100%	100%	93.7%
MLP	WGAN	86.73%	80.18%	100%	72.36%	88.64%
	DEMGAN	100%	100%	100%	100%	100%
NB	WGAN	90.90%	91.11%	90.19%	90.20%	90.96%
	DEMGAN	100%	100%	100%	100%	93.7%

5.2.2 Addressing Data Diversity Lack

We conducted multiple experiments using DEMGAN and recorded the generator loss each time, as shown in Fig. 3. The experimental results indicate that as the number of training rounds increases, the generator loss gradually decreases. In Fig. 3, lines of different colors represent the changes in the generator's loss function in different rounds of experiments as the number of training times increases.

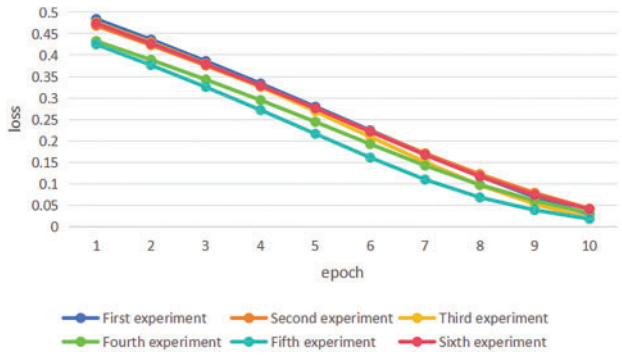


Figure 3: Changes in generator loss from multiple experiments

5.3 Attacking ML-Based IDS

In this section, we detected the evasion rate on the preprocessed CICIDS2017 dataset and uniformly labeled all malicious traffic data as attack samples. Fig. 4 displays the classification confusion matrix results after injecting the adversarial examples generated by DEMGAN into normal traffic. The green section represents the volume of malicious traffic, which, when camouflaged by DEMGAN, leads to the IDS detection outcome being benign traffic. Table 5 presents the evasion rate results of traffic attacks against five ML-based IDS after applying DEMGAN to disguise malicious traffic.

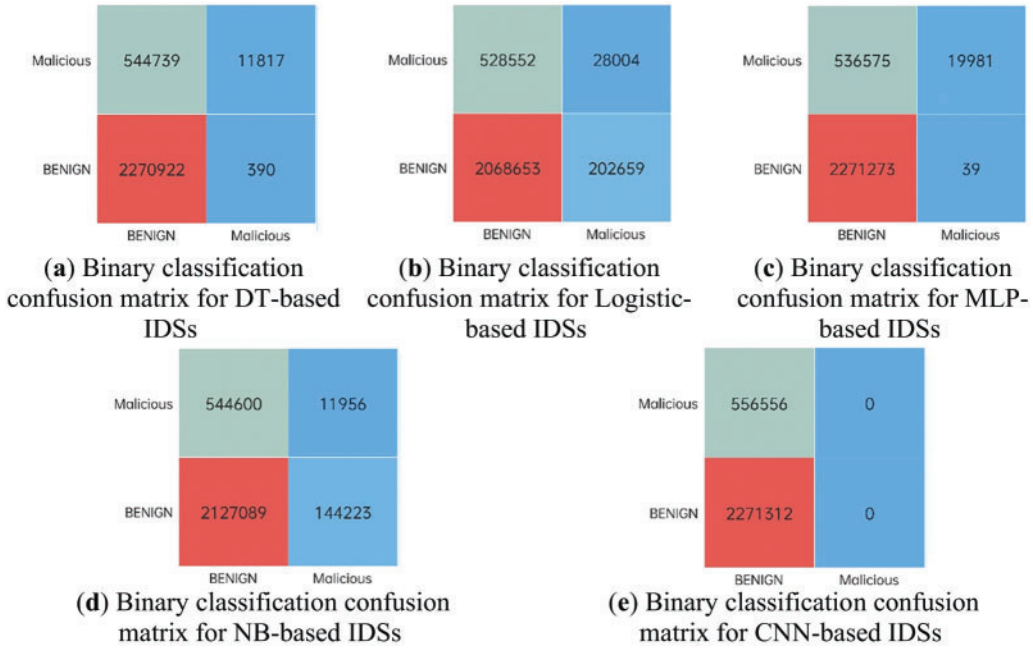


Figure 4: Results of dichotomous confusion matrices based on different IDSs

Table 5: The escape rate results of binary classification of adversarial examples generated by DEMGAN

	DT	Logistic	MLP	NB	CNN	RNN	CNN-BiLSTM
WGAN	61.15%	48.94%	83.96%	90.77%	35.99%	41.06%	36.64%
DEMGAN	97.88%	94.97%	96.41%	97.85%	100%	97,.61%	96.41%

5.4 Comparison with GAN-Based Attacks

We simulated the threat model specified in the ADVGAN [28] paper on the CICIDS2017 dataset. The results are presented in Table 6. The experimental results indicate that ADVGAN attains a high evasion rate among MLP methods but demonstrates poor performance in the other IDS, particularly in CNN, where the evasion rate is less than 10%.

5.5 Validation of DEMGAN Applicability

To verify the universality of the DEMGAN method, we selected some data from the CICIDS2018 dataset for experiments. Table 7 displays the experimental results of the evasion rate in comparison to the original WGAN. The findings indicate that DEMGAN can effectively evade ML-based IDS. Our approach achieves a

100% evasion rate on MLP, NB, and CNN models. However, the results are slightly less favorable for logistic regression and decision tree models, which have now become a focal point for our future research.

Table 6: ADVGAN attack results

	DT	Logistic	MLP	NB	CNN	RNN	CNN-BiLSTM
ADVGAN	84.59%	67.24%	95.15%	18.86%	1.7%	18.06%	17.81%
DEMGAN	97.88%	94.97%	96.41%	97.85%	100%	97,.61%	96.41%

Table 7: The binary evasion rate results of DEMGAN attack on ML-based IDS

	DT	Logistic	MLP	NB	CNN	RNN	CNN-BiLSTM	Macro-ER
WGAN	68.26%	58.07%	65.84%	100%	82.84%	53.47%	42.81%	67.33%
DEMGAN	72.22%	65.34%	100%	100%	100%	97.51%	98.72%	90.54%

5.6 Ablation Experiment

In this section, we verify the effectiveness of various improvements to the model through ablation experiments. Fig. 5 intuitively demonstrates the efficacy of various improvements. The experimental results show that although there is a significant improvement in the effect when only one of the two improvements is made, when the two improvements are applied to DEMGAN at the same time, the effect of DEMGAN reaches the best and the two improvements do not have a counter-effect.

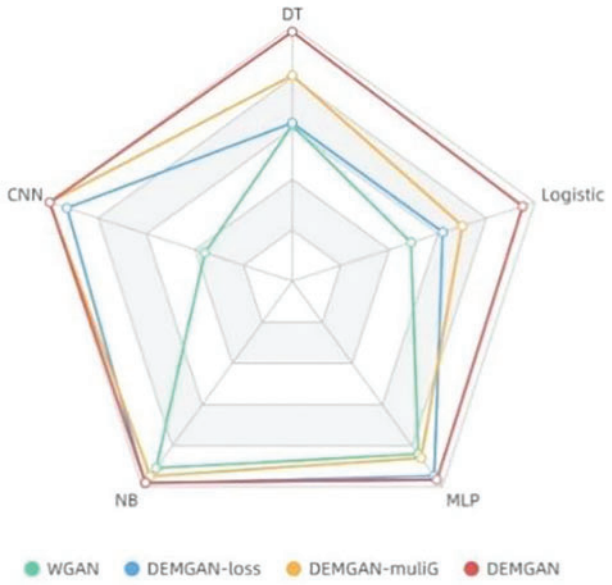


Figure 5: Escape rate results under various improvement

In the ablation experiment, there are four combinations.

- Combination 1 (WGAN): Indicates the case where the loss function of the original WGAN is used, the Transformer is used as the generator, and RNN is used as the discriminator.
- Combination 2 (DEMGAN-loss): Indicates the use of our improved loss function with distortion rate added, Transformer as the generator, and RNN as the discriminator.
- Combination 3 (DENGAN-muliG): Indicates the case of using the loss function of the original GAN, using multiple Transformer generator structures, and using RNN as the discriminator.
- Combination 4 (DEMGAN): Indicates the use of our improved loss function that adds distortion rate and multiple Transformer generator structures.

5.7 Comparative Efficiency Analysis

In order to comprehensively evaluate the performance of DEMGAN in practical applications, we conducted a systematic efficiency comparison experiment on WGAN and DEMGAN in the same experimental environment as Section 5.1.1. The experiment used 100,000 sample data from 7 randomly selected traffic categories, and obtained reliable performance indicator data by taking the average value of 10 independent repeated experiments. The specific experimental results show that under the condition of generating the same number of adversarial samples, the total time consumed by WGAN is 5248.72 s, while DEMGAN only takes 5124.93 s. In terms of resource usage, the peak memory usage of both methods is 456 MB, showing a comparable level of resource requirements. These experimental results verify that DEMGAN has achieved significant time efficiency improvement while maintaining the same resource usage as WGAN.

5.8 Enhancements to IDS

We further validated the enhancement effect of the DEMGAN algorithm on Intrusion Detection Systems (IDS) by using its generated adversarial examples to retrain the ML-based IDS. Fig. 6 illustrates the change in recognition accuracy of the adversarial examples generated by DEMGAN before and after retraining the IDS.

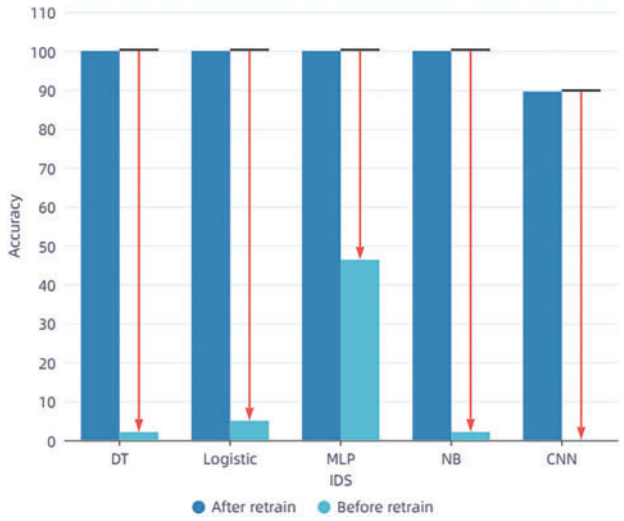


Figure 6: Recognition accuracy of ML-based IDS before and after retraining using adversarial examples

According to the experimental results, it can be observed that when the Intrusion Detection System (IDS) is retrained without using adversarial examples, the adversarial examples generated by DEMGAN can successfully evade the intrusion detection algorithm. This evasion is particularly effective when employing techniques such as decision trees, linear regression, Naive Bayes, and CNN, as the recognition rate of adversarial examples can be decreased to below 5%.

By utilizing adversarial examples for retraining, an Intrusion Detection System (IDS) learns the counterattack mechanism against adversarial examples. This process can effectively enhance its ability to identify adversarial examples and enable it to better distinguish between normal data and malicious intrusion behaviors. Consequently, it further improves the overall performance and security of the IDS, safeguarding the network from potential intrusion threats.

6 Conclusion

This paper introduces an adversarial traffic attack method called DEMGAN, designed for attackers to evade intrusion detection systems. We have made two improvements based on WGAN. The improved algorithm can generate real adversarial malicious traffic and evade a variety of ML-based IDS. We conducted ablation experiments and algorithm applicability experiments, and the results show that the adversarial examples generated by our attack algorithm can achieve the purpose of evading IDS on multiple datasets. We also conducted experiments to demonstrate that retraining using adversarial examples generated by DEMGAN can effectively improve the performance of IDS. In the future, we will focus on addressing DEMGAN's low evasion rate on linear regression algorithms and decision tree algorithms. At the same time, we still need to continuously improve the algorithm to achieve better results, make it more meaningful in practical applications, and improve the level of network security.

Acknowledgement: The authors would like to express their appreciation to the anonymous referees for their valuable suggestions and comments.

Funding Statement: This work is supported by the National Defense Basic Scientific Research Program of China under grant No. JCKY2023602C026.

Author Contributions: The first author, Dawei Xu, proposed the main idea and writing of this article. The second author, Yue Lv, performed experiments and wrote part of the article. The third author, Min Wang, performed part of the experiments. The fourth author, Baokun Zheng, gave critical comments. The fifth author, Jian Zhao, gave critical comments. The sixth author, Jiaxuan Yu, gave critical comments. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The two datasets supporting the experiments in this article are public datasets, and their links are https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 16 April 2025) and https://www.unb.ca/cic/datasets/ids-2018.html (accessed on 16 April 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Lan J, Liu X, Li B, Sun J, Li B, Zhao J. MEMBER: a multi-task learning model with hybrid deep features for network intrusion detection. Comput Secur. 2022;123:102919. doi:10.1016/j.cose.2022.102919.

2. Talukder M, Islam MM, Uddin MA, Hasan KF, Sharmin S, Alyami SA, et al. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. J Big Data. 2024;11:1–44. doi:10.1186/s40537-024-00886-w.
3. Tao Z, Xu C, Lian Y, Tian H, Kang J, Kuang X, et al. When moving target defense meets attack prediction in digital twins: a convolutional and hierarchical reinforcement learning approach. IEEE J Sel Areas Commun. 2023;41:3293–305. doi:10.1109/JSAC.2023.3310072.
4. Zhang C, Luo X, Liang J, Liu X, Zhu L, Guo S. POTA: privacy-preserving online multi-task assignment with path planning. IEEE Trans Mob Comput. 2024;23(5):5999–6011. doi:10.1109/TMC.2023.3315324.
5. Zhang T, Xu C, Zou P, Tian H, Kuang X, Yang S, et al. How to mitigate DDoS intelligently in SD-IoV: a moving target defense approach. IEEE Trans Ind Inform. 2023;19:1097–106. doi:10.1109/TII.2022.3190556.
6. Alatwi HA, Morisset C. Adversarial machine learning in network intrusion detection domain: a systematic review. arXiv:2112.03315. 2021.
7. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv:1412.6572. 2014.
8. Papernot N, Mcdaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P); 2016 Mar 21–24; Saarbruecken, Germany.
9. Moosavi-Dezfooli SM, Fawzi A, Frossard P. DeepFool: a simple and accurate method to fool deep neural networks. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA. p. 2574–82.
10. Carlini N, Wagner DA. Towards evaluating the robustness of neural networks. In: Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP); 2017 May 22–26; San Jose, CA, USA. p. 39–57.
11. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. Commun ACM. 2016;63:139–44. doi:10.1145/3422622.
12. Li W, Fan L, Wang Z, Ma C, Cui X. Tackling mode collapse in multi-generator GANs with orthogonal vectors. Pattern Recognit. 2021;110:107646. doi:10.1016/j.patcog.2020.107646.
13. Kurach K, Lucic M, Zhai X, Michalski M, Gelly S. The GAN landscape: losses, architectures, regularization, and normalization. arXiv:1807.04720. 2018.
14. Mescheder LM, Geiger A, Nowozin S. Which training methods for GANs do actually converge? Proc Mach Learn Res. 2018;80:3481–90.
15. Zhang C, Zhao M, Liang J, Fan Q, Zhu L, Guo S. NANO: cryptographic enforcement of readability and editability governance in blockchain databases. IEEE Trans Dependable Secur Comput. 2024;21(4):3439–52. doi:10.1109/TDSC.2023.3330171.
16. Arjovsky M, Chintala S, Bottou L. Wasserstein GAN. arXiv:1701.07875. 2017.
17. Ding H, Sun Y, Huang N, Shen Z, Cui X. TMG-GAN: generative adversarial networks-based imbalanced learning for network intrusion detection. IEEE Trans Inf Forensics Secur. 2024;19:1156–67. doi:10.1109/TIFS.2023.3331240.
18. Park C, Lee J, Kim Y, Park J, Kim H, Hong D. An enhanced AI-based network intrusion detection system using generative adversarial networks. IEEE Internet Things J. 2023;10:2330–45. doi:10.1109/JIOT.2022.3211346.
19. He D, Dai J, Liu X, Zhu S, Chan S, Guizani M. Adversarial attacks for intrusion detection based on bus traffic. IEEE Netw. 2022;36:203–9. doi:10.1109/MNET.105.2100353.
20. Zhu Y, Cui L, Ding Z, Li L, Liu Y, Hao Z. Black box attack and network intrusion detection using machine learning for malicious traffic. Comput Secur. 2022;123:102922. doi:10.1016/j.cose.2022.102922.
21. Alshahrani E, Alghazzawi DM, Alotaibi RM, Rabie OBJ. Adversarial attacks against supervised machine learning based network intrusion detection systems. PLoS One. 2022;17(10):e0275971. doi:10.1371/journal.pone.0275971.
22. Zolbayar BE, Sheatsley R, Mcdaniel P, Weisman M, Zhu S, Krishnamurthy SV. Generating practical adversarial network traffic flows using NIDSGAN. arXiv:abs/2203.06694. 2022.
23. Sun P, Si C, Li S, Cheng Z, Zhao S, Liu Q. A targeted adversarial attack method for multi-classification malicious traffic detection. In: Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2024 Apr 14–19; Seoul, Republic of Korea.

24. Liu Z, Hu J, Liu Y, Roy K, Yuan X, Xu J. Anomaly-based intrusion on IoT networks using AIGAN—a generative adversarial network. IEEE Access. 2023;11:91116–32. doi:10.1109/ACCESS.2023.3307463.
25. Cover TM, Thomas JA. Elements of information theory. 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2005.
26. Hoang Q, Nguyen TD, Le T, Phung D. Multi-generator generative adversarial nets. arXiv:1708.02556. 2017.
27. Hu C, Zhang C, Lei D, Wu T, Liu X, Zhu L. Achieving privacy-preserving and verifiable support vector machine training in the cloud. IEEE Trans Inf Forensics Secur. 2023;18:3476–91. doi:10.1109/TIFS.2023.3283104.
28. Xiao C, Li B, Zhu JY, He W, Liu M, Song D. Generating adversarial examples with adversarial networks. arXiv:1801.02610. 2019.