Hindawi
Security and Communication Networks
Volume 2021, Article ID 9947059, 15 pages
https://doi.org/10.1155/2021/9947059

Research Article

A GAN and Feature Selection-Based Oversampling Technique for Intrusion Detection

Xiaodong Liu,[1] **Tong Li**,[1,2] **Runzi Zhang**,[3,4] **Di Wu**,[1] **Yongheng Liu**,[1] **and Zhen Yang**[1,2]

[1] *Faculty of Information Technology, Beijing University of Technology, Beijing, China*
[2] *Engineering Research Center of Intelligent Perception and Autonomous Control, Ministry of Education, Beijing, China*
[3] *NSFOCUS Technologies Group Co., Ltd., Beijing 100089, China*
[4] *Department of Automation, Tsinghua University, Beijing 100089, China*

Correspondence should be addressed to Tong Li; litong@bjut.edu.cn

Received 5 April 2021; Revised 10 June 2021; Accepted 26 June 2021; Published 6 July 2021

Academic Editor: Savio Sciancalepore

In recent years, there have been numerous cyber security issues that have caused considerable damage to the society. The development of efficient and reliable Intrusion Detection Systems (IDSs) is an effective countermeasure against the growing cyber threats. In modern high-bandwidth, large-scale network environments, traditional IDSs suffer from a high rate of missed and false alarms. Researchers have introduced machine learning techniques into intrusion detection with good results. However, due to the scarcity of attack data, such methods' training sets are usually unbalanced, affecting the analysis performance. In this paper, we survey and analyze the design principles and shortcomings of existing oversampling methods. Based on the findings, we take the perspective of imbalance and high dimensionality of datasets in the field of intrusion detection and propose an oversampling technique based on Generative Adversarial Networks (GAN) and feature selection. Specifically, we model the complex high-dimensional distribution of attacks based on Gradient Penalty Wasserstein GAN (WGAN-GP) to generate additional attack samples. We then select a subset of features representing the entire dataset based on analysis of variance, ultimately generating a rebalanced low-dimensional dataset for machine learning training. To evaluate the effectiveness of our proposal, we conducted experiments based on the NSL-KDD, UNSW-NB15, and CICIDS-2017 datasets. The experimental results show that our method can effectively improve the detection performance of machine learning models and outperform the baselines.

1. Introduction

The rapid development of network technology has dramatically improved people's daily lives, but it has also brought many threats. For example, Marriott International's Starwood network was maliciously breached, and the private information of some 500 million users was compromised (http://sn.people.com.cn/n2/2018/1204/c190199-32363619.html). Not only is there a risk of personal information being copied on the Internet, but corporate production is also under serious threat. In 2018, Taiwan Semiconductor Manufacturing Corporation (TSMC), the world's number one chip foundry, was compromised by the WannaCry ransomware virus, which led to a complete shutdown of all production lines and ultimately caused losses of approximately NTD 5.2 billion (https://english.cw.com.tw/article/article.action?id=2194). The 2018-2019 Global Application and Cyber Security Report released by Radware shows that 93% of respondents have suffered from network attacks in the past 12 months. Network security has become an issue that people cannot ignore.

Intrusion Detection Systems (IDSs) have been widely adopted as an effective method to detect and defend against network attacks in response to the growing network threats. It monitors network traffic in real-time, divides network records into normal records and malicious records, and provides essential information for the defense system. In the last few decades, machine learning has been used to improve intrusion detection [1]. Nevertheless, due to the sparsity of attack data, the training set for this type of approach is unbalanced, affecting analysis performance [2].

Oversampling techniques are commonly used to address the problem of unbalanced datasets. Traditional methods are used to generate samples among the nearest neighbors by interpolation, such as Synthetic Minority Oversampling Technique (SMOTE) [3], and Adaptive Synthetic Sampling Technique (ADASYN) [4]. Generative Adversarial Network (GAN) is a new generative model that provides a new framework for sample generation [5]. It allows the generator to learn the data features sufficiently by gaming between the generator and the discriminator to simulate data distributions. It shows its most advanced technology in the generation of images, sounds, and texts [6–8]. Moreover, researchers in other fields are regularly applying this method in their research direction.

This paper presents an oversampling technique based on Generative Adversarial Networks (GAN) and Feature Selection (GAN-FS) applied to intrusion detection from the perspective of data imbalance and high dimensionality. We construct an attack sample generation model based on an improved generative adversarial network WGAN-GP. In addition, considering the characteristics of large data volume and high dimensionality in intrusion detection, we use Analysis of Variance (ANOVA) for data dimensionality reduction. Effective data dimensionality reduction can remove redundant and irrelevant features to reduce the curse of dimensionality and thus improve classification accuracy [9]. Our contribution can be summarized as follows:

(1) We propose a new oversampling method, GAN-FS, to solve the class imbalance problem in intrusion detection. We construct an attack generation model based on WGAN-GP to generate attack samples. The data are then feature-selected using ANOVA to obtain a rebalanced low-dimensional dataset for training the intrusion detection model. We have modified this in our contribution as follows.

(2) Based on three popular intrusion detection datasets, we conducted experiments on several machine learning detection models. The experimental results show that our approach can effectively improve intrusion detection models' performance. Moreover, compared with multiple popular methods, our approach achieves better results.

(3) We discuss and analyze the impact of our approach on different datasets and different machine learning detection models.

The remainder of the paper is organized as follows. In Section 2, we provide an overview of the relevant studies. Section 3 presents the GAN-FS. The design, execution, and results are presented in Section 4. Finally, the paper is concluded in Section 5.

2. Related Work

Based on GAN and feature selection, we propose an oversampling technique applied to intrusion detection. Therefore, we discuss related work in the following four approaches: intrusion detection method (Section 2.1), feature selection (Section 2.2), oversampling technique (Section 2.3), and generative adversarial network (Section 2.4), respectively.

2.1. Intrusion Detection. As an essential tool of cyber security, IDSs are responsible for identifying and warning of cyber attacks. Since the first paper on IDSs [10] was published, there have been numerous research achievements in this field. In recent years, IDS has developed rapidly with the help of machine learning.

Aslahi-Shahri et al. proposed an intrusion detection algorithm based on a genetic algorithm and support vector machine [11]. They used a hybrid algorithm for feature selection and then ranked the selected features according to their importance and finally achieved good results. Elbasiony et al. proposed a hybrid network intrusion detection framework based on random forest and weighted k-means [12]. They combined random forest and k-means based intrusion detection models to construct a hybrid intrusion detection model, which effectively reduces the false alarm rate. Compared with traditional intrusion detection methods, Wang et al. proposed an intrusion detection method based on artificial neural networks and fuzzy clustering [13]. The above methods can effectively detect and prevent network attacks, but their research focuses on improving and combining existing methods, ignoring the intrusion detection dataset's imbalance. For example, in the widely used NSL-KDD dataset [14], the number of normal samples is 67,343, while the number of R2L and U2R attacks is only 995 and 52, respectively. In the UNSW-NB15 dataset [15], the number of normal samples is 37,000, while the number of Shellcode and Worms attacks is only 378 and 44. The imbalance in the intrusion detection dataset affects the detection accuracy and stability of IDSs.

2.2. Application of Feature Selection in Intrusion Detection. Feature selection is referred to as obtaining a subset from an original feature set according to a particular feature selection criterion, selecting the dataset's relevant features. In the field of intrusion detection, the datasets used are characterized by large numbers and high dimensionality. Feature selection reduces the computational difficulty and eliminates data redundancy, thus improving the detection rate of machine learning techniques and reducing false alarms [1].

Khammassi and Krichen propose a GA-LR wrapper approach for feature selection in network intrusion detection [16]. They used a genetic algorithm-based packing method as a search strategy and logistic regression as a learning algorithm to select the best subset of features. Moreover, the method effectively improves intrusion detection performance. Mohammadi et al. propose an intrusion detection method based on feature selection and clustering algorithm using filter and wrapper methods [17]. The filter and wrapper methods are named feature grouping based on linear correlation coefficient algorithm and cuttlefish algorithm. Based on this method, the performance of its intrusion detection model has been significantly improved.

In general, feature selection is an effective method for data dimensionality reduction and is widely used in intrusion detection. The feature selection-based data dimensionality reduction method can effectively improve intrusion detection performance.

2.3. Oversampling Techniques. To improve the ability of machine learning models to judge and analyze minority samples in the presence of sample imbalance, researchers have proposed several rebalancing techniques at different levels, such as data-level and algorithm-level. The data-level oversampling technique increases the number of minority samples by artificial means to improve the dataset's balance. The typical methods mainly include random oversampling, SMOTE, and ADASYN [3, 4]. In recent years, researchers have proposed new oversampling methods, such as *K*-means SMOTE and G-SMOTE [18, 19]. These techniques have improved the data imbalance to varying degrees. Table 1 demonstrates the design rationale for the above methods.

Random oversampling increases the number of minority samples but can lead to severe overfitting. Also, if the minority sample is biased or noisy, this will increase the interference with the classifier. SMOTE uses interpolation to generate samples, avoiding sample overlap due to random oversampling. However, because SMOTE treats all minority samples equally and does not consider the category information of neighbor samples, it cannot effectively enhance decision boundaries, resulting in poor classification results. ADASYN takes into account information about the distribution of the data while generating samples by interpolation. However, network traffic complexity leads to a blurring of its category boundaries, and using this strategy may exacerbate the confusion of decision boundaries.

In order to solve the problems of the above methods, researchers have made continuous attempts. For few classes, *K*-means SMOTE generates a different number of samples based on the clustering density. Furthermore, the method does not consider the category labels, thus ensuring that the generated samples are in a safe region. G-SMOTE substitutes the data generation mechanism by defining a flexible geometric region around each minority sample. Then synthetic instances are generated inside the boundaries of the region.

2.4. Generative Adversarial Networks. GAN is a deep learning model that models complex high-dimensional distributions of real-world data. Inspired by the two-person zero-sum game in game theory, it consists of a Generator (G) and a Discriminator (D). G and D are both neural networks. G captures the potential distribution of real data samples and generates new data samples; D is a binary classifier, judging whether the input is real data or generated samples. Classification results will be passed back to G and D through the loss of weight updates. Both networks are trained until D can no longer distinguish real samples from generated samples. The optimization process is a minimax game problem. The optimization goal is to achieve a Nash equilibrium so that the generated network can estimate data samples' distribution. The objective function is defined as follows:

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

where P_{data} is defined as the real sample distribution, P_G is the sample distribution generated by the generator, $P_Z(z)$ is the noise variable distribution, $G(z)$ is defined as a function of mapping noise to data space, and $D(x)$ represents the probability that the sample x is real data rather than a generated sample. To distinguish between real data and generated samples, $D(x)$ should be maximized and $D(G(z))$ should be minimized. When $P_G = P_{\text{data}}$, the objective function obtains the global optimal solution.

In the security field, some researchers have applied GAN in their work. Ring et al. use GAN to generate high-quality data flow [20]. Rigaki and Garcia can successfully bypass detection by using malicious traffic generated by GAN [21]. Lee and Park proposed an intrusion detection method based on generative adversarial networks and random forests. They oversampled the intrusion detection dataset and then used the random forest for classification. The method achieved better performance on the CICIDS-2017 dataset compared to the original random forest method. However, their work did not consider the instability of GAN and the high dimensionality of the data and did not perform more validation on other datasets and models [22]. Yilmaz et al. proposed an intrusion detection method based on GAN and MLP [23]. They generated three types of attacks on the UGR 16 dataset to balance the dataset. The experimental results show that GAN's balanced attack sample dataset produces more accurate results than the unbalanced attack sample set. Vu and Nguyen proposed a method based on Auxiliary Classifier Generative Adversarial Network (ACGAN) to enhance the balance of the dataset [24]. The method achieved better performance than machine learning algorithms trained on the original dataset and other sampling techniques. Yin et al. proposed a framework for intrusion detection based on generative adversarial networks. The generative model in their framework is used to generate other complementary labeled samples for adversarial training, which helps the classifier perform classification [25].

In this paper, we propose a new oversampling method, GAN-FS. Compared with existing work, we design the GAN-FS oversampling method in terms of both the number and dimensionality of the data. We build an attack generation model based on WGAN-GP to generate higher quality samples. And, we introduce ANOVA for feature selection from the perspective of data high dimensionality to further reduce the learning difficulty of the classifier. We also

Table 1: Design rationale of different oversampling techniques.

Technique	Design rationale
Random oversampling	Minority samples are randomly selected and replicated to increase the number of samples
SMOTE	Samples are generated by interpolation between each minority sample and its surrounding minority samples
ADASYN	As with SMOTE, ADASYN generates new samples by interpolation; the difference is that the number of new samples that need to be synthesized for each minority sample is determined by the density of majority class instances around it
K-means SMOTE	*K*-means SMOTE will first cluster the data into multiple clusters; different samples are then generated for the clusters' density, with smaller densities generating a more significant number of samples
G-SMOTE	G-SMOTE generates synthetic samples in a geometric region of the input space, around each selected minority sample

perform experimental validation on several popular datasets and detection models and analyze the impact of different datasets and different machine learning detection models.

3. An Oversampling Technique GAN-FS

The imbalance of data affects the performance of machine learning-based IDSs. Due to the lack of analysis on the correlation of features, existing oversampling technology cannot effectively generate high-dimensional network traffic. We build an attack generation model based on WGAN-GP to generate higher quality samples. And, we introduce ANOVA for feature selection from the perspective of data high dimensionality to further reduce the learning difficulty of the classifier.

Figure 1 illustrates the workflow of GAN-FS. There are five steps in the framework: Data Preprocessing, Data Partition, Rare class Oversampling, Feature Selection, and Train & Test ML Model.

(i) Step 1: the dataset is preprocessed and then divided into a training set and a testing set.

(ii) Step 2: the training set is divided into rare class data and other class data by data partitioning.

(iii) Step 3: the GAN model uses rare class data to generate samples.

(iv) Step 4: the oversampled data is combined with the other class data obtained in step 2, and then, feature selection is performed. The optimal feature subset and the corresponding new low-dimensional training set are obtained in the feature selection step.

(v) Step 5: finally, the new training set is used to train the machine learning (ML) model, and the testing set is used to test the model.

3.1. Preprocessing. The dataset used in the context of intrusion detection contains different forms of features such as continuous, discrete, and symbolic with varying resolution and ranges. We need to process the data to make it suitable for our model. The preprocessing includes numeralization and normalization. We also need to partition the dataset if it does not provide a defined training and testing set.

3.1.1. Numeralization. Intrusion detection data usually contains nonnumeric features such as protocols and states. These nonnumeric features need to be converted to numeric features to suit our model. Nonnumeric features are mapped to integer values between 0 and S-1, where S is the number of symbols.

3.1.2. Normalization. The inconsistent feature scales of data in different dimensions will affect the results of intrusion detection. We need to normalize the data to eliminate the dimension influence between indicators. We scale all the features to [0, 1] except the attack type label. The min-max normalization is used to scale the data values linearly, as follows:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{2}$$

where x is the value before normalization, x' is the value after normalization, $x_{\max}$ is the maximum value of sample data, and $x_{\min}$ is the minimum value of sample data.

3.2. Data Partition and Rare Class Attack Oversampling. Attack data is a rare class of data in training set in the field of intrusion detection. We need to take rare class data from the training set and oversample them. Before generating samples, we train a generative model to model the distribution of attack data.

In GAN, the Jensen-Shannon divergence is used to measure the difference between two distributions, but it requires some overlap between the two distributions. When the discriminator is trained to be optimal, there is no overlap between the distribution of the real sample p_{data} and the distribution of the generated sample p_G, and two non-overlapping or negligible overlap distributions cause the generator's gradient to disappear. Wasserstein GAN (WGAN) [26] effectively improves GAN by increasing the Lipschitz limit and introducing Wasserstein distances. However, WGAN also suffers from the disappearing gradient problem, and WGAN-GP [27] introduces the gradient penalty to solve this problem. The structure of the WGAN-GP is the same as that of the GAN, as shown in Figure 2.

The WGAN-GP objective function is expressed as follows:

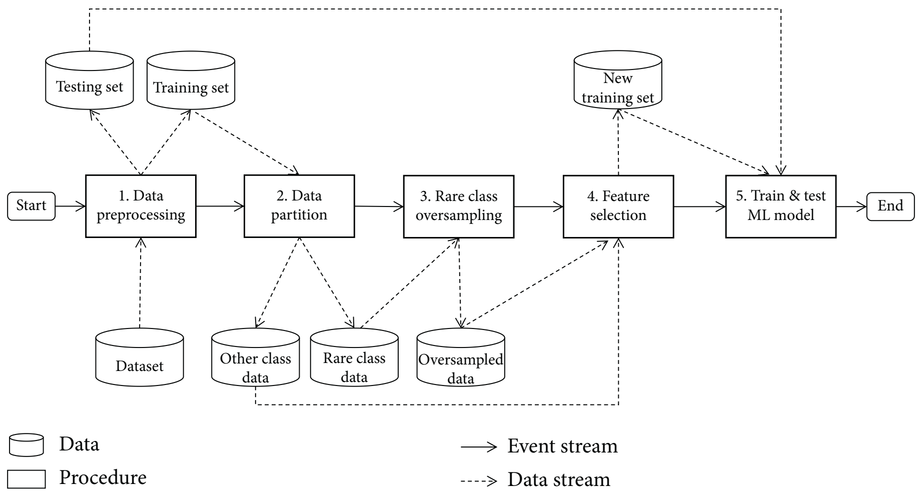


FIGURE 1: The workflow of GAN-FS.

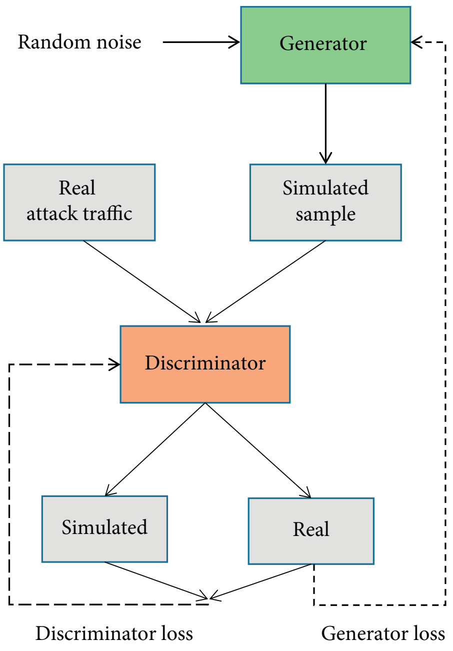


FIGURE 2: The architecture of WGAN-GP.

$$L = \underset{\tilde{x}\sim P_G}{E}[D(\tilde{x})] - \underset{x\sim P_{\text{data}}}{E}[D(x)] + \lambda \underset{\hat{x}\sim P_{\hat{x}}}{E}\left[\left(\left\|\nabla_{\hat{x}} D(\hat{x})\right\|_2 - 1\right)^2\right], \quad (3)$$

where P_{data} is the data distribution, P_G is the model distribution implicitly defined by $\tilde{x} = G(z), z \sim p(z)$ (the input z to the generator is sampled from some simple noise distribution, such as the uniform distribution or a spherical Gaussian distribution), and $P_{\hat{x}}$ defines the uniform sampling along a straight line between the point pairs sampled from the data distribution P_{data} and the generated distribution P_G. A penalty on the gradient norm is enforced for random samples $\hat{x} \sim P_{\hat{x}}$. In this way, the generator and frequency discriminator can be improved at the same speed to avoid mode-collapse, which leads to the optimization of training effect and the weight of neural network for the poor and improves WGAN's training to a certain extent.

In the process of generating the sample, noise and rare class attacks are used to train WGAN-GP. The training process begins with fixing the discriminator and training the generator to model the distribution of real data. When the discriminator cannot correctly distinguish whether the samples are coming from the real attack set or the generator, fix the generator and start training the discriminator. When the discriminator can correctly distinguish between samples through continuous training, fix the discriminator and the training generator. Follow this process for iterative training, and finally, use the generator to generate attack samples. The generated attack samples are eventually added to the training set.

3.3. Feature Selection. Feature selection is a data dimensionality reduction method and is often used to deal with high-dimensional and complex data. Feature selection is referred to the process of obtaining a subset from an original feature set according to a particular feature selection criterion, which selects the relevant features of the dataset [28]. Feature selection is a process of selecting n most valuable features from the m existing original features to reduce the dimensionality of the dataset. The filter method is a common feature selection method, and the selection of features is separate from any machine learning technique. It selects features based on scores in various statistical tests and on indicators of relevance.

ANOVA (analysis of variance) F-test is a commonly used method of feature selection [29, 30]. It uses the F-test to determine whether the means of some groups are different and to test statistically whether the means are equal. More specifically, for each feature x_i, we assume that x_i has the same mean value in the positive and negative category samples, i.e., $H_0\colon \mu_{S+} = \mu_{S-}$, where μ denotes the mean value, $S+$ denotes the set of x_i values belonging to the positive category samples, and $S-$ denotes the set of x_i values belonging to the negative category samples; then, the f_value is calculated according to the following equation:

$$f\_\text{value} = \frac{S_A/(r-1)}{S_E/(n-r)}, \tag{4}$$

where S_A and S_E denote the component and intragroup deviation, respectively, n is the total number of samples, and r is the number of categories, where r is 2. f_value of each feature is calculated separately according to the above steps. Finally, the optimal subset is obtained by ranking the features according to their importance.

3.4. Train and Test Machine Learning Model. The imbalanced dataset affects the machine learning-based intrusion detection model's analysis capability, making its classification results biased towards normal activities and leading to a high false alarm and missed alarm rate. We oversample rare classes of attacks in the training set based on WGAN-GP and then downsample the training set using the ANOVA feature selection method to obtain a low-dimensional rebalanced training set finally. In this step, we use the rebalanced low-dimensional dataset to train the machine learning model. When the model training is completed, we use a test set based on a subset of features to test its performance.

4. Evaluation

In this section, we systematically design and conduct a series of experiments and analyze the results.

4.1. Research Questions

(1) Q1: can our proposed method effectively improve the detection performance of machine learning models?

(2) Q2: is the combination of GAN and feature selection effective?

(3) Q3: is our proposed method better than other oversampling methods?

4.2. Datasets

NSL-KDD: the earliest IDSs dataset was created by the Defense Advanced Research Projects Agency in 1998 and was named the DARPA 1998 dataset. Subsequently, the KDD99 dataset was created from the DARPA 1998 dataset and has become one of the most widely used datasets [31]. The presence of many duplicate instances in the KDD99 dataset can affect the detection performance of machine learning methods by biasing them towards normal instances. Tavallaee et al. built the NSL-KDD dataset in 2009 based on the KDD99 dataset to solve the above problem by eliminating duplicate records [14]. The NSL-KDD training set consists of 125,973 records, and the test dataset contains 22,544 records. The NSL-KDD dataset includes four types of attacks and 41 attributes. The four types of attacks are DoS, Probe, R2L, and U2R. However, the number of attack instances in this dataset is much lower than normal instances, with only 995 and 52 for R2L and U2R attacks, respectively.

UNSW-NB15: in recent years, the Cyber Range Lab of the Australian Centre for Cyber Security has created the UNSW-NB15 dataset. This dataset contains a variety of novel attacks and is therefore widely used for intrusion detection. There are nine types of attacks to simulate the real network environment, namely, Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. The UNSW-NB15 dataset contains a training set and a testing set. The training set has 82332 records, and the testing set has 175341 records. The training set is unbalanced, with the number of normal data being much higher than the number of attacks.

CICIDS-2017: the dataset was created in a 5-day simulation environment containing network traffic in packet-based and bidirectional stream formats. The authors extracted more than 80 attributes for each stream and provided additional metadata about IP addresses and attacks. Compared to NSL-KDD and UNSW-NB15, CICIDS-2017 includes a wide range of attack types such as SSH brute force, heartbleed, botnet, DoS, DDoS, web, and penetration attacks. Moreover, with nearly three million data, CICIDS-2017 can evaluate the performance of IDS in large-scale scenarios.

We increase the number of attacks by oversampling the rare attack categories in the dataset. Tables 2–4 show the data distribution of the dataset before and after oversampling.

4.3. Experimental Settings. In this section, we describe the relevant experimental setup, including the selection of models, the setting of parameters, and the selection of evaluation metrics.

Table 2: Distribution of records in NSL-KDD before and after oversampling.

Type	Before oversampling	After oversampling
Normal	67343	67343
DoS	45927	45927
Probe	11656	11656
R2L	995	10995
U2R	52	10052

Table 3: Distribution of records in UNSW-NB15 before and after oversampling.

Type	Before oversampling	After oversampling
Normal	37000	37000
Generic	18871	18871
Exploits	11132	11132
Fuzzers	6062	6062
DoS	4089	4089
Reconnaissance	3496	3496
Analysis	677	10677
Backdoor	583	10583
Shellcode	378	10378
Worms	44	10044

Table 4: Distribution of records in CICIDS-2017 before and after oversampling.

Type	Before oversampling	After oversampling
Normal	1363935	1363935
DoS	151735	151735
PortScan	95135	95135
DDoS	76878	76878
Patator	8290	8290
Web attack	1281	11281
Bot	1163	11163
Infiltration	20	10020
Heartbleed	8	10008

4.3.1. *Machine Learning Model Selection*

Naive Bayes (NB) is a classification technique based on Bayes' theorem, which assumes that predictors are independent of each other [32]. Simply put, the Naive Bayes classifier assumes that a feature in a category is independent of the presence of other features. Naive Bayes models are easy to build, and, in addition to being simple, Naive Bayes outperforms even highly complex classification methods.

Decision Tree (DT) is widely used in intrusion detection [33, 34]. A decision tree is a tree-like structure with leaves, which represent classifications, and branches, representing the combination of features that lead to those classifications. An example is the classification of nodes of a decision tree by testing their feature values against each other. Moreover, in the work of Mishra et al., decision trees are the single classifier with the best performance [35].

Random Forest (RF) is commonly used as an integration algorithm [36]. The integration of classifiers provides a more robust generalization capability than a single base learner and is also widely used in intrusion detection [37–39]. RF integrates multiple weak classifiers, and the final result is obtained by voting or taking the mean. This allows the overall model to have a high degree of accuracy and generalization.

Gradient Boosting Decision Tree (GBDT) is a robust integrated learning algorithm that extends and augments a categorical regression tree model based on gradient augmentation [40]. The GBDT iteratively constructs decision trees, and in each iteration, a decision tree is trained from the residuals of the previous tree. The final result is then obtained cumulatively from the predictions of all trees.

Support Vector Machine (SVM) is also one of the most widely used machine learning algorithms [31]. SVM is a supervised learning method [41]. It performs classification by constructing an N-dimensional hyperplane that optimally classifies the data into different classes.

K-Nearest Neighbors (K-NN) is a data mining algorithm that is theoretically mature and less complex [42]. The basic idea is that, in the sample space, if most of the nearest neighbor samples belong to a class, then the samples belong to the same class.

Artificial Neural Networks (ANN) is a form of distributed computing inspired by biology. It has a strong self-learning capability and is suitable for solving nonlinear problems, so it is also commonly used in intrusion detection [43].

4.3.2. Model Settings. WGAN-GP is implemented through the deep learning framework Kears. Both generator and discriminator are feedforward neural networks. The learning rate of the generator and discriminator is 0.0001. The dimension of the noise vector is 100. Furthermore, the weight clipping threshold of discriminator training is set to 0.01. Root Mean Square Error (RMSE) can characterize the degree of fit between the generated samples and the real samples. As shown in Figure 3, the model is close to convergence when trained to 150 rounds. Therefore, the training epoch for WGAN-GP is set to 150.

Scikit-learn is used to implement the feature selection process and the construction of the machine learning models. After feature selection, the feature subsets of the NSL-KDD and UNSW-NB15 datasets are shown in Table 5. For a detailed explanation of the features, see [14, 15]. The four machine learning models are also implemented by scikit-learn.

4.3.3. Evaluation Metrics. IDSs are a vital tool to ensure network security. It is necessary not only to identify attacks accurately but also to avoid false alarms. Therefore, we use recall and precision as metrics. Also, we need to consider the overall accuracy, so we use accuracy as a metric. Besides, we introduce F-measure as a metric to fully evaluate the

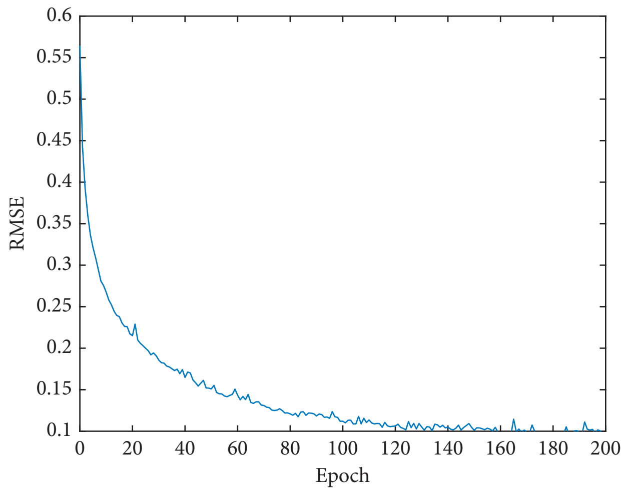


Figure 3: The convergence curve of WGAN-GP.

Table 5: Subsets of features selected based on ANOVA.

Dataset	Selected features	Quantity
NSL-KDD	protocol_type, flag, logged_in, count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate	16
UNSW-NB15	Proto, dttl, dloss, sinpkt, swin, stcpb, dtcpb, dwin, dmean, ct_state_ttl, ct_dst_ltm, ct_src_dport_ltm, is_sm_ips_ports	13
CICIDS-2017	Destination Port, Flow Duration, Fwd Packet Length Max, Fwd Packet Length Min, Fwd Packet Length Mean, Bwd Packet Length Max, Bwd Packet Length Min, Bwd Packet Length Mean, Bwd Packet Length Std, Flow Packets/s, Flow IAT Mean, Flow IAT Std, Flow IAT Max, Fwd IAT Total, Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, Bwd IAT Std, Bwd IAT Max, Fwd PSH Flags, Min Packet Length, Max Packet Length, Packet Length Mean, Packet Length Std, Packet Length Variance, FIN Flag Count, SYN Flag Count, PSH Flag Count, ACK Flag Count, URG Flag Count, Down/Up Ratio, Average Packet Size, Avg Fwd Segment Size, Avg Bwd Segment Size, Init_Win_bytes_backward, Idle Mean, Idle Std, Idle Max, Idle Min	39

detection performance of machine learning models. After classification, the data can be divided into four categories: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). The calculation formula is as follows:

$$\begin{aligned}\text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TP} + \text{FP} + \text{FN}},\\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}},\\ \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}},\\ F - \text{measure} &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.\end{aligned} \tag{5}$$

4.4. Experiments and Results.

4.4. Experiments and Results. To answer the three questions mentioned earlier, we design three separate sets of experiments and analyze the results.

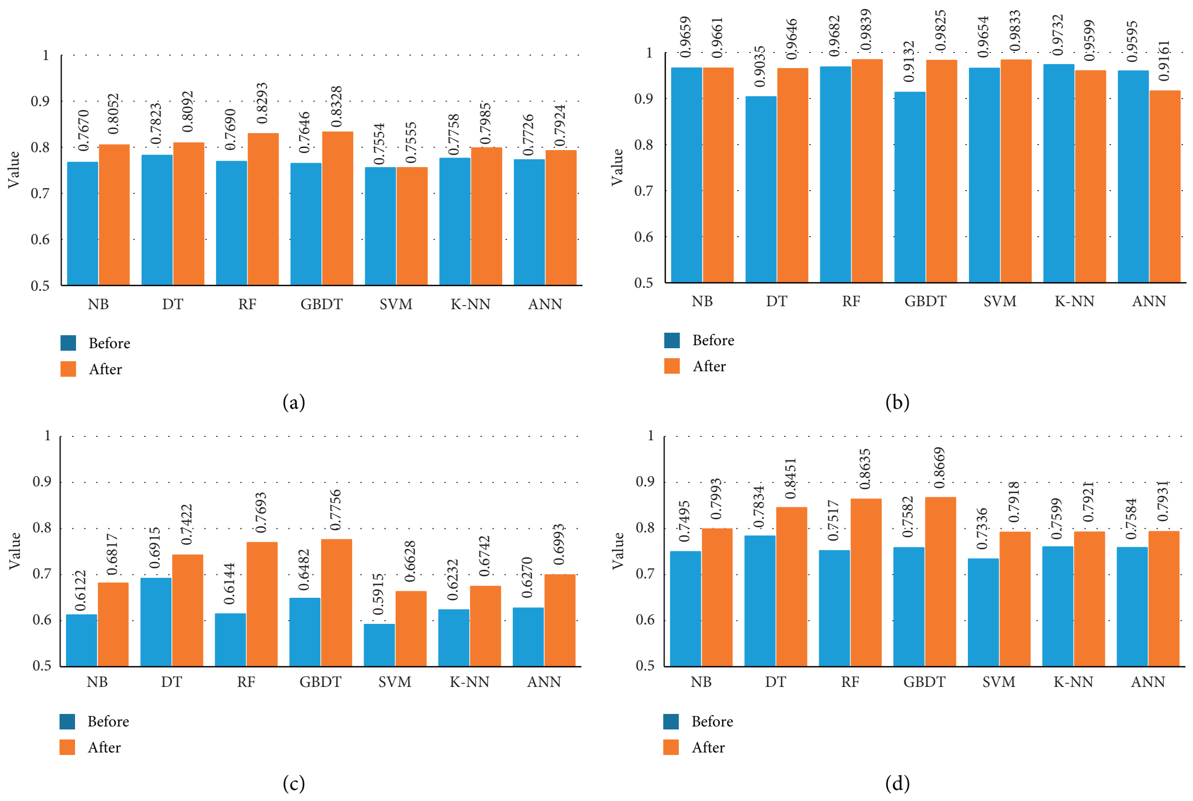
4.4.1. Experiment I. To investigate GAN-FS's effectiveness (Q1), we will train machine learning models using the original training set and the training set oversampled by GAN-FS, respectively. The experiments are arranged as follows:

(1) Train the machine learning models using the dataset that is not oversampled

(2) Train the machine learning models using the dataset oversampled by GAN-FS

4.4.2. Results. The comparison results before and after oversampling using GAN-FS are shown in Figures 4–6. The results show that the performance of the classifiers is improved to different degrees after oversampling using our proposed method.

As shown in Figure 4(a), the Accuracy of each detector is improved on the NSL-KDD dataset. Such results indicate that our method can improve the overall performance of the detectors. In terms of the performance of individual detectors, GBDT shows the best detection performance. Its Accuracy was improved by about 6% to 83.28%. Figure 4(c) shows that the Recall of all detectors also improves to varying degrees, which indicates that the samples we generated improve the diversity of attack samples and thus enhance the generalization of the knowledge learning of the detectors. The improved Recall indicates that the classifier



(a) (b) (c) (d)

Figure 4: Results of Experiment I on the NSL-KDD dataset. (a) Accuracy. (b) Precision. (c) Recall. (d) *F*-measure.

can identify attacks more accurately and reduce the attack miss rate, which is of great practical importance in intrusion detection. Figure 4(b) shows the precision of the detectors. It can be seen from the figure that the precision of individual classifiers has declined. This is also because while the detector's generalization ability improves after adding the generated samples, it misclassifies some normal samples as attack samples. Since different detectors have different learning abilities, there is some difference in the degree of impact. Finally, the F-measure of the detectors shows that the overall performance of the detectors is effectively improved. The same conclusion can be drawn from the experiments on the UNSW-NB15 and CICIDS datasets. Figures 5(a) and 5(a) show that our approach can effectively improve the Accuracy of the classifier and more significantly improve the Recall of the classifier. Ultimately, the F-measure of each classifier has different degrees of improvement. Notably, in the experiments of CICIDS-2017, the F-measure of GBDT without oversampling has been as high as 99.36%. In this case, our method does not allow further improvement of the classifier, but the impact is also tiny.

In addition, analyzing the performance of individual classifiers, RF and GBDT achieved quite good performance after oversampling. GBDT showed optimal performance on both NSL-KDD and UNSW-NB15 datasets and showed better performance on the CICIDS-2017 dataset. RF showed optimal performance on the CICIDS-2017 dataset with a 99.6% F-measure. This demonstrates the powerful generalization ability of RF and GBDT as integrated learning and highlights the effectiveness of our proposed oversampling method.

4.4.3. *Experiment II.*

GAN-FS is based on GAN and feature selection oversampling methods. This experiment is to verify the effectiveness of the combination of GAN and feature selection. In this experiment, we will process the training set using GAN and feature selection separately, evaluate it using machine learning models, and then compare it with our method. The experiments are arranged as follows:

(1) Oversampling the dataset using WGAN-GP (w/o ANOVA)

(2) Reducing the dimensionality of the dataset using ANOVA (w/o WGAN-GP)

(3) Oversampling the dataset using GAN-FS

4.4.4. *Results.*

Figures 7–9 show the results of using WGAN-GP and ANOVA and GAN-FS alone. The experimental results show that the combination of WGAN-GP and ANOVA is effective. GAN and feature selection can improve the detection performance of machine learning models to some extent. However, from the comparison in Figure 7, we can see that the overall performance of using only WGAN-GP for oversampling or only ANOVA for dimensionality reduction is lower than that of GAN-FS. We can draw the same conclusion from the comparison in Figure 8. This is because WGAN-GP can learn the distribution of the attack samples

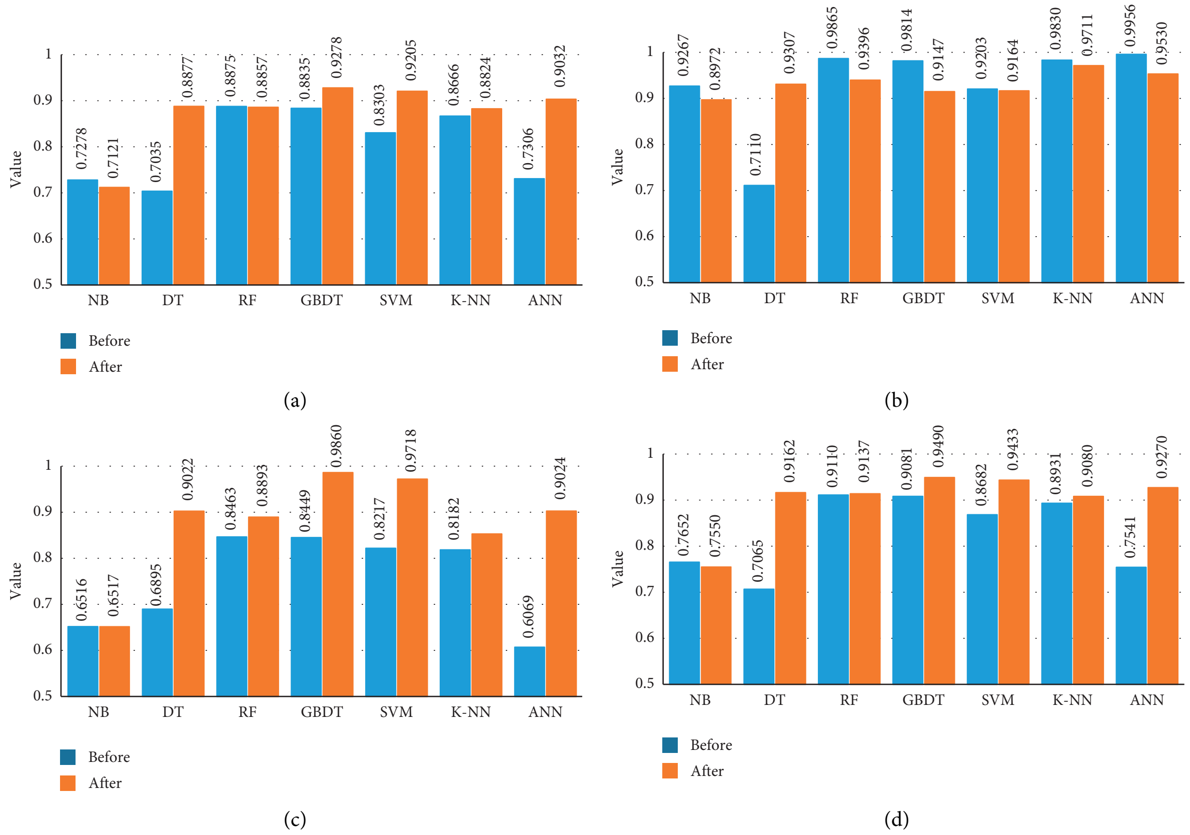


Figure 5: Results of Experiment I on the UNSW-NB15 dataset. (a) Accuracy. (b) Precision. (c) Recall. (d) *F*-measure.

and thus generate samples. The generated samples can increase the diversity of attack samples, enhance the learning of attack features by the detector, and improve the detection performance. However, the high dimensionality of the original data also affects the analytical performance of the detector, and the use of feature selection can delete some of the features that are not important for analytic learning, thus improving the classifier's performance. However, when the detector performance is high, the improvement from feature selection is subtle. As shown in Figure 9, the lead of our method is more subtle. This is mainly because the detector already achieves very high performance when feature selection is not performed. Performing feature selection, in this case, destroys the original combination of features and does not result in a very significant improvement in the detector's performance. Combining the above results, the combination of WGAN-GP and ANOVA, i.e., GAN-FS, is compelling.

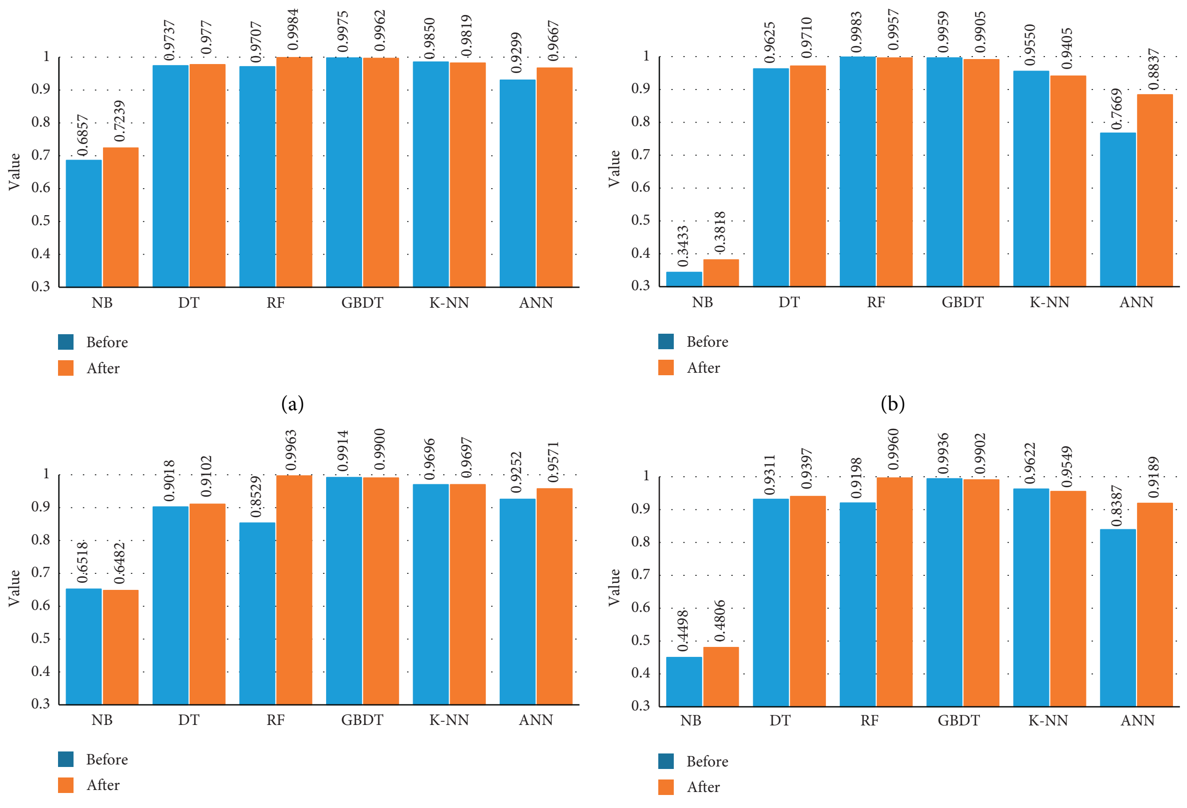
4.4.5. Experiment III. As we talked about in Section 2, SMOTE [3] and ADASYN [4] are two classical oversampling methods, while K-means SMOTE [19] and G-SMOTE [18] are two newer methods proposed in recent years. We will compare these four methods. In addition, we also compare them with the GAN-based methods of Vu and Nguyen [24] and Lee and Park [22]. The F-measure is an overall evaluation of the Precision and Recall, which we use to measure the methods' performance. The experiments are arranged as follows:

(1) Oversampling the dataset using baseline separately

(2) Oversampling the dataset using GAN-FS

4.4.6. Results. The experimental results compared to the baseline are shown in Tables 6–8. From the table, we can see that the detectors' performance based on our proposed method tends to be higher than the other baselines compared to the other methods. From the perspective of individual detectors, we can see that our method performs better on DT, RF, GBDT, and ANN. This is related to the principle of the detector. The idea of the tree model is to construct a tree with the fastest decreasing entropy using information entropy as a metric. Our method generates samples based on the original distribution, which increases the diversity of samples and removes unnecessary features using feature selection. Therefore, after oversampling, the decision tree can better classify the samples. Compared with DT, RF and GBDT are integrated learning algorithms with more excellent learning capability and can improve classification performance. The experimental results also show that the performance of RF and GBDT are generally higher than DT. As a neural network structure, ANN is also a model with higher learning ability, and its performance is improved by increasing the number of attack samples.

However, due to different algorithm principles, there are differences in the performance of the classifiers after oversampling. For the K-NN model, we find that the advantage



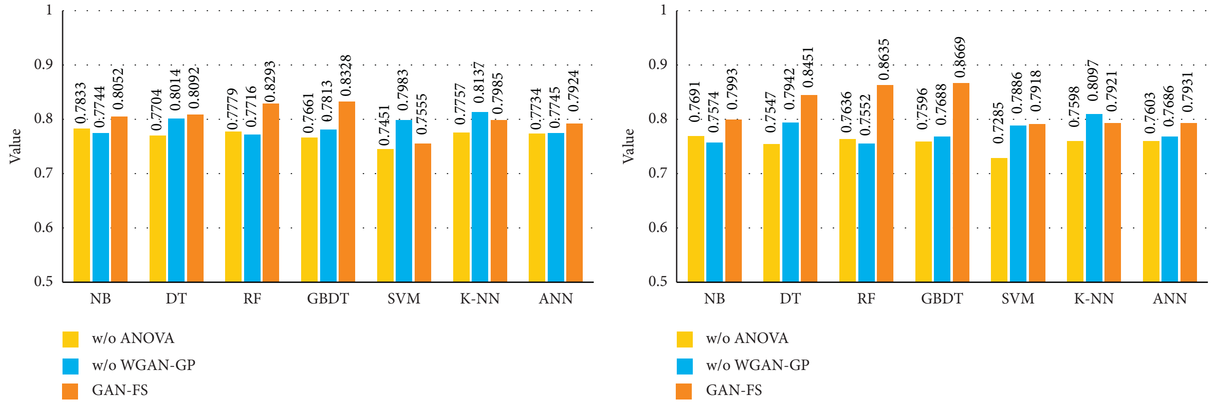
(a)

(b)

(c)

(d)

FIGURE 6: Results of Experiment I on the CICIDS-2017 dataset. (a) Accuracy. (b) Precision. (c) Recall. (d) *F*-measure.



(a)

(b)

FIGURE 7: Results of Experiment II on NSL-KDD dataset. (a) Accuracy. (b) *F*-measure.

of our method is not very prominent. This is because traditional oversampling methods such as SMOTE are based on K-NN to generate samples. Generating samples based on this method and then training the K-NN algorithm can improve the classifier's performance. In addition, although our method improves the performance of the SVM, in the experimental results of NSL-KDD (Table 6), our method is not optimal. This is because SVM solves the maximum partition hyperplane using support vectors. When the original SVM classification is not optimal, generating samples in a safe geometric space (G-SMOTE) will optimize the classification hyperplane more effectively.

Finally, as shown in Figure 10, we count the number of times that different methods achieve the optimal performance. The detectors trained based on our method achieved 12 optimal Accuracy and 14 optimal F-measure,

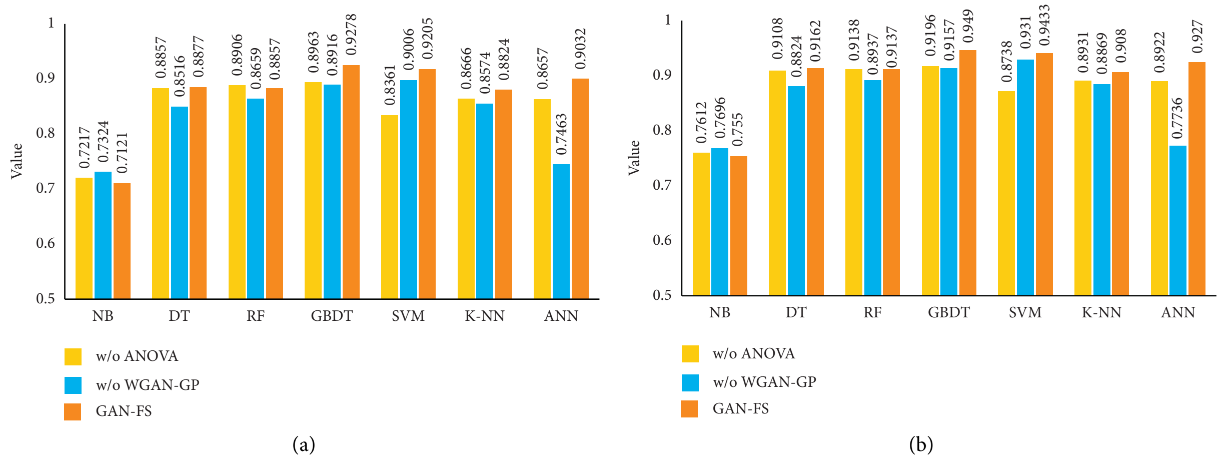


Figure 8: Results of Experiment II on UNSW-NB15 dataset. (a) Accuracy. (b) *F*-measure.

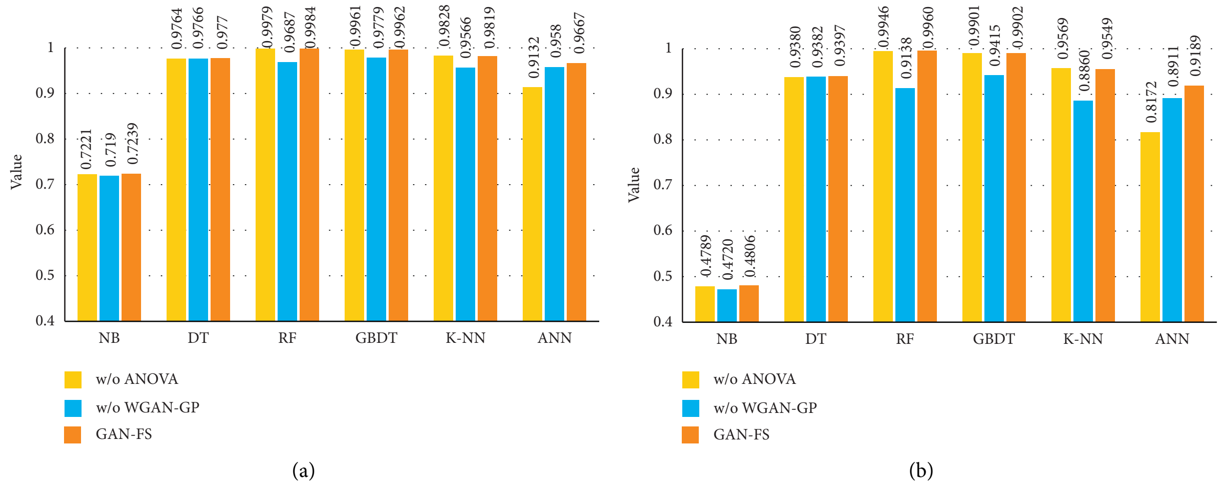


Figure 9: Results of Experiment II on CICIDS-2017 dataset. (a) Accuracy. (b) *F*-measure.

Table 6: Performance of different methods on the NSL-KDD dataset.

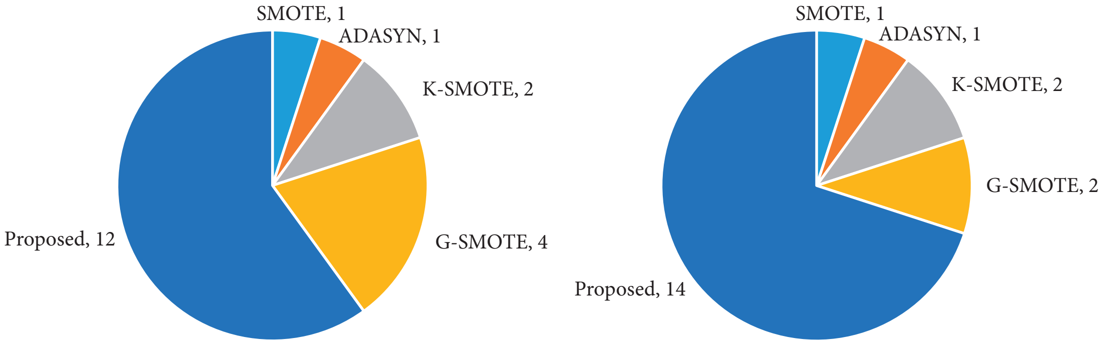
		SMOTE	ADASYN	K-SMOTE	G-SMOTE	ACGAN-SVM	GAN	Proposed
NB	Acc	**0.8475**	0.8090	0.8246	0.8027	0.7595	0.7771	0.8052
	F1	**0.8552**	0.8132	0.8307	0.7973	0.7395	0.7600	0.7993
DT	Acc	0.7856	0.7583	0.7736	**0.8101**	0.7983	0.7646	0.8092
	F1	0.7746	0.7392	0.7624	0.8047	0.7946	0.7564	**0.8451**
RF	Acc	0.7571	0.7494	0.7837	0.7822	0.7571	0.7618	**0.8293**
	F1	0.7363	0.7249	0.7715	0.7694	0.7365	0.7424	**0.8635**
GBDT	Acc	0.7800	0.7780	0.7840	0.7749	0.7741	0.7785	**0.8328**
	F1	0.7751	0.7646	0.7774	0.7684	0.7686	0.7737	**0.8669**
SVM	Acc	0.8080	0.8025	0.7821	**0.8407**	0.7781	0.7798	0.7555
	F1	0.8023	0.7963	0.7686	**0.8425**	0.7632	0.7654	0.7918
K-NN	Acc	0.7924	0.7890	0.7921	0.7921	0.7758	0.7758	**0.7985**
	F1	0.7823	0.7812	0.7821	0.7835	0.7599	0.7599	**0.7921**
ANN	Acc	0.7661	0.7850	0.7802	0.7850	0.7625	0.7587	**0.7924**
	F1	0.7577	0.7865	0.7759	0.7814	0.7455	0.7377	**0.7931**

Table 7: Performance of different methods on the UNSW-NB15 dataset.

		SMOTE	ADASYN	K-SMOTE	G-SMOTE	ACGAN-SVM	GAN	Proposed
NB	Acc	0.7241	**0.7378**	0.7265	0.7228	0.7246	0.7243	0.7121
	F1	0.7627	**0.7719**	0.7643	0.7619	0.7628	0.7629	0.7550
DT	Acc	**0.8904**	0.8837	0.6534	0.8854	0.8779	0.7269	0.8877
	F1	0.9156	0.9110	0.6670	0.9112	0.9040	0.7569	**0.9162**
RF	Acc	0.9077	0.8993	0.8820	**0.9097**	0.8872	0.8629	0.8857
	F1	0.9295	0.9229	0.9090	**0.9313**	0.9109	0.8896	0.9137
GBDT	Acc	0.9086	0.9200	0.8870	0.8997	0.8862	0.8847	**0.9278**
	F1	0.9308	0.9408	0.9146	0.9232	0.9101	0.9088	**0.9490**
SVM	Acc	0.8937	0.9098	0.8679	0.8993	0.8302	0.8309	**0.9205**
	F1	0.9171	0.9314	0.8954	0.9220	0.8682	0.8688	**0.9433**
K-NN	Acc	0.8723	0.8738	0.8744	0.8694	0.8666	0.8666	**0.8824**
	F1	0.8983	0.8997	0.9002	0.8956	0.8931	0.8931	**0.9080**
ANN	Acc	0.8630	0.7583	0.7397	0.8780	0.8325	0.8712	**0.9032**
	F1	0.8893	0.7848	0.7649	0.9028	0.8616	0.8970	**0.9270**

Table 8: Performance of different methods on the CICIDS-2017 dataset.

		SMOTE	ADASYN	K-SMOTE	G-SMOTE	ACGAN-SVM	GAN	Proposed
NB	Acc	0.6652	0.6816	0.7238	**0.7543**	0.7263	0.7227	0.7239
	F1	**0.4854**	0.2653	0.3105	0.4425	0.4836	0.4797	0.4806
DT	Acc	0.9393	0.9585	0.9459	0.2490	0.9746	0.9721	**0.9770**
	F1	0.8220	0.8895	0.8454	0.3414	0.9331	0.9264	**0.9397**
RF	Acc	0.9969	0.9950	0.9684	0.5847	0.9692	0.9703	**0.9984**
	F1	0.9921	0.9872	0.9130	0.4856	0.9154	0.9187	**0.9960**
GBDT	Acc	0.9921	0.9947	0.9928	0.9779	0.9888	0.9719	**0.9962**
	F1	0.9801	0.9867	0.9817	0.9468	0.9710	0.9333	**0.9902**
K-NN	Acc	0.9839	0.9823	**0.9850**	0.9694	0.9847	0.9830	0.9819
	F1	0.9599	0.9562	**0.9622**	0.9267	0.9615	0.9613	0.9549
ANN	Acc	0.8161	0.3498	0.8925	0.5665	0.9294	0.9227	**0.9667**
	F1	0.6811	0.3774	0.7153	0.4334	0.8185	0.8311	**0.9189**



(a)

(b)

Figure 10: Number of times the optimal value is obtained for each method.

respectively, which are much higher than other methods. From these results, we can see that our method can better improve the detection performance of the intrusion detection model.

5. Conclusion and Future Work

In this paper, we take the perspective of imbalance and high dimensionality of datasets in intrusion detection and propose an oversampling intrusion detection technique based on GAN and feature selection. For one thing, our approach proposes to focus on oversampling the rare classes of attack samples in order to improve the effectiveness of intrusion detection. For another thing, we concentrate on only imperative features of attack samples using the ANOVA feature selection method. Then, the obtained low-dimensional rebalanced dataset is used to train intrusion detection classifiers. Experimental results show that our approach improves the performance of detecting intrusion detection models and outperforms other baselines.

As for future work, we first plan to explore the conjunction between our approach and deep learning. In addition, we will try to assign different weights to features to better reflect the significance of each feature in classification.

Data Availability

We use public datasets, which are deposited in public repositories NSL-KDD (https://www.unb.ca/cic/datasets/nsl.html), UNSW-NB15 (https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo4ys?path=%2FUNSW-NB15%20-%20CSV%20Files), and CICIDS-2017 (https://www.unb.ca/cic/datasets/ids-2017.html).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science of Foundation of China (nos. 61902010 and 61671030), CCF-NSFocus Funding (no.2020003), Beijing Excellent Talent Funding-Youth Project (no.201800 0020124G039), and Project of Beijing Municipal Education Commission (no.KM202110005025).

References

[1] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[2] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: a survey," *Applied Sciences*, vol. 9, no. 20, p. 4396, 2019.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.

[4] H. He, B. Yang, E. A. Garcia, and S. L. Adasyn, "Adaptive synthetic sampling approach for imbalanced learning," in *Proceedings of the Neural Networks, 2008 IJCNN 2008 (IEEE World Congress on Computational Intelligence)*, IEEE International Joint Conference, Hong Kong, China, September 2008.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672–2680, 2014.

[6] C. Ledig, L. Theis, F. Huszár et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4681–4690, Washington, DC, USA, July 2017.

[7] H. Su, X. Shen, P. Hu, W. Li, and Y. Chen, *Dialogue Generation with Gan*, AAAI, Menlo Park, CA, USA, 2018.

[8] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, Venice, Italy, October 2017.

[9] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[10] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.

[11] B. M. Aslahi-Shahri, R. Rahmani, M. Chizari et al., "A hybrid method consisting of ga and svm for intrusion detection system," *Neural Computing & Applications*, vol. 27, no. 6, pp. 1–8, 2016.

[12] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, and M. M. Fahmy, "A hybrid network intrusion detection framework based on random forests and weighted k-means," *Ain Shams Engineering Journal*, vol. 4, no. 4, pp. 753–762, 2013.

[13] J. Ma, L. Huang, J. Hao, and G. Wang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert Systems with Applications*, vol. 37, 2010.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, IEEE, Ottawa, Canada, July 2009.

[15] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *Proceedings of the Military Communications and Information Systems Conference (MilCIS) 2015*, Canberra, Australia, November 2015.

[16] C. Khammassi and S. Krichen, "A ga-lr wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255–277, 2017.

[17] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *Journal of information security and applications*, vol. 44, pp. 80–88, 2019.

[18] G. Douzas and F. Bacao, "Geometric smote a geometrically enhanced drop-in replacement for smote," *Information Sciences*, vol. 501, pp. 118–135, 2019.

[19] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on *k*-means and smote," *Information Sciences*, vol. 465, pp. 1–20, 2018.

[20] M. Ring, D. Schlör, D. Landes, and A. Hotho, "Flow-based network traffic generation using generative adversarial networks," *Computers & Security*, vol. 82, 2018.

[21] M. Rigaki and S. Garcia, "Bringing a gan to a knife-fight: adapting malware communication to avoid detection," in *Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW)*, pp. 70–75, San Francisco, CA, USA, May 2018.

[22] J. H. Lee and K. H. Park, "Gan-based imbalanced data intrusion detection system," *Personal and Ubiquitous Computing*, vol. 25, pp. 1–8, 2019.

[23] I. Yilmaz, R. Masum, and A. Siraj, "Addressing imbalanced data problem with generative adversarial network for intrusion detection," in *Proceedings of the 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pp. 25–30, IEEE, Las Vegas, NV, USA, August 2020.

[24] L. Vu and Q. U. Nguyen, "Handling imbalanced data in intrusion detection systems using generative adversarial networks," *Journal of Research and Development on Information and Communication Technology*, vol. 2020, no. 1, pp. 1–13, 2020.

[25] C. Yin, Y. Zhu, S. Liu, J. Fei, and H. Zhang, "Enhancing network intrusion detection classifiers using supervised adversarial training," *The Journal of Supercomputing*, vol. 76, pp. 1–30, 2019.

[26] A. Martin, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223, Sydney, Australia, August 2017.

[27] I. Gulrajani, F. Ahmed, A. Martin, D. Vincent, and A. Courville, "Improved training of wasserstein gans," in *Proceedings of the 31st Conference on Advances in Neural Information Processing Systems*, pp. 5767–5777, Red Hook, NY, USA, December 2017.

[28] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: a new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[29] S. Kumar Dey and M. Rahman, "Effects of machine learning approach in flow-based anomaly detection on software-defined networking," *Symmetry*, vol. 12, no. 1, p. 7, 2020.

[30] S.-Y. Ji, B.-K. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors," *Journal of Network and Computer Applications*, vol. 62, pp. 9–17, 2016.

[31] A. Özgür and H. Erdem, "A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *PeerJ Preprints*, vol. 4, Article ID e1954v1, 2016.

[32] M. Panda and M. R. Patra, "Network intrusion detection using Naive Bayes," *International journal of computer science and network security*, vol. 7, no. 12, pp. 258–263, 2007.

[33] A. Ahmed, L. Maglaras, M. Amine Ferrag, M. Derdour, and H. Janicke, "A novel hierarchical intrusion detection system based on decision tree and rules-based models," in *Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 228–233, IEEE, Santorini, Greece, May 2019.

[34] K. Peng, V. Leung, L. Zheng, S. Wang, C. Huang, and T. Lin, "Intrusion detection system based on decision tree over big data in fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 4680867, 10 pages, 2018.

[35] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686–728, 2018.

[36] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[37] N. Farnaaz and M. A. Jabbar, "Random forest modeling for network intrusion detection system," *Procedia Computer Science*, vol. 89, pp. 213–217, 2016.

[38] L. Li, Y. Yang, S. Bai, J. Cheng, and X. Chen, "Towards effective network intrusion detection: a hybrid model integrating gini index and gbdt with pso," *Journal of Sensors*, vol. 2018, Article ID 1578314, 9 pages, 2018.

[39] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, CRC Press, Boca Raton, FL, USA, 2012.

[40] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, pp. 1189–1232, 2001.

[41] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, *A Practical Guide to Support Vector Classification*, National Taiwan University, Taipei, Taiwan, 2003.

[42] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[43] B. Ingre and A. Yadav, "Performance analysis of nsl-kdd dataset using ann," in *Proceedings of the International Conference on Signal Processing and Communication Engineering Systems*, pp. 92–96, IEEE, Guntur, India, January 2015.