IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 28, 2020, accepted October 13, 2020, date of publication October 19, 2020, date of current version October 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031892

Network Intrusion Detection Based on Conditional Wasserstein Generative Adversarial Network and Cost-Sensitive Stacked Autoencoder

GUOLING ZHANG, XIAODAN WANG, RUI LI, YAFEI SONG, (Member, IEEE), JIAXING HE, AND JIE LAI
College of Air and Missile Defense, Air Force Engineering University, Xi'an 710051, China

Corresponding author: Xiaodan Wang (afeu_wang@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61876189, Grant 61806219, Grant 61503407, Grant 61703426, and Grant 61273275; and in part by the Young Talent Fund of the University Association for Science and Technology, Shanxi, China, under Grant 20190108.

ABSTRACT In the field of intrusion detection, there is often a problem of data imbalance, and more and more unknown types of attacks make detection difficult. To resolve above issues, this article proposes a network intrusion detection model called CWGAN-CSSAE, which combines improved conditional Wasserstein Generative Adversarial Network (CWGAN) and cost-sensitive stacked autoencoders (CSSAE). First of all, the CWGAN network that introduces gradient penalty and L2 regularization is used to generate specified minority attack samples to reduce the class imbalance of the training dataset. Secondly, the stacked autoencoder is used to intelligently extract the deep abstract features of the network data. Finally, a cost-sensitive loss function is constructed to give a large misclassification cost to a minority of attack samples. Thus, effective detection of network intrusion attacks can be realized. The experimental results based on KDDTest$^+$, KDDTest-21, and UNSW-NB15 datasets show that the CWGAN-CSSAE network intrusion detection model improves the detection accuracy of minority attacks and unknown attacks. In addition, the method in this article is compared with other existing intrusion detection methods, excellent results have been achieved in performance indicators such as accuracy and F1 score. The accuracy on the above datasets reached 90.34%, 80.78% and 93.27% respectively. The accuracy of U2R on the KDDTest$^+$ and KDDTest-21 datasets both reached 42.50%. The accuracy of R2L on the KDDTest$^+$ and KDDTest-21 datasets reached 54.39% and 52.51%, respectively. And the F1 score on the above datasets reached 91.01%, 87.18% and 93.99% respectively.

INDEX TERMS Intrusion detection, conditional Wasserstein GAN, stacked autoencoder, imbalanced classification, cost-sensitive, regularization, deep learning.

I. INTRODUCTION

Based on the rapid development of emerging technologies such as cloud computing, big data, and the Internet of Things, cyberspace has become the fifth-largest space besides land, sea, air and sky [1]. However, hundreds of millions of network access points and user equipment access networks have brought huge difficulties and challenges to cyberspace security. Recently, various network security incidents and network attacks have occurred frequently. For example, the Portuguese multinational energy giant Energias de Portugal (EDP) was attacked by RagnarLocker ransomware in April 2020. The attacker claimed to have obtained 10TB of sensitive data files from the EDP Company and demanded a ransom of 1,580 bitcoins [2]. In addition, in June 2020, Honda's car server was attacked by EKANS (snake spelled backwards) ransomware, which caused part of the production system to be interrupted [3]. In summary, it is of great strategic significance to detect and respond to network intrusions into a timely and accurate manner to ensure the security of cyberspace.

Intrusion detection technology uses an active defense method. Through continuous monitoring and analysis of network activities, it detects intrusions and responds promptly.

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

Now, this technology has become an important means of maintaining cyberspace security. Due to the different execution locations of intrusion detection, intrusion detection systems (IDS) can be divided into network-based intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS). This article studies the network-based intrusion detection system. Currently, the scale of network data is gradually increasing and network attack technologies are rapidly updated. Intrusion detection models based on shallow machine learning algorithms exhibit problems such as difficulty in processing large-scale network intrusion data, poor recognition of various new types of attacks, high false alarm rates, and excessive reliance on researchers for feature design and feature selection [4], [5].

In recent years, deep learning technologies such as GAN (Generate Adversarial Network) [6], SAE (Stacked Autoencoder) [7], DBN (Deep Belief Network) [8], DNN (Deep Neural Network) [9], LSTM (Long-Term Short-Term Memory) [4] are widely used in the field of intrusion detection. Deep learning technologies can automatically extract high-level abstract features of network data and realize accurate identification of network attacks, which can overcome the limitations of shallow learning algorithms. Although the application of deep learning in the field of intrusion detection has achieved the expected research results, there are still many problems to be solved. On the one hand, a good dataset plays a vital role in model training. However, the network intrusion detection data obtained in the real network environment often contains a large amount of normal behavior data onto minority attacks behavior data. Also, the number of different types of attacks is imbalanced. It results in the model's poor recognition of minority intrusions [10]. On the other hand, as the rapid development of network technology, network attack methods are changing rapidly and more unknown attacks are threatening the security of cyberspace, which bring new challenges to the research of deep learning on intrusion detection [11].

So far, the methods to solve the problem of class imbalance can be divided into two categories: data level and algorithm level. The data-level method is mainly to reduce the degree of class imbalance by changing the original sample distribution. The general methods are to increase minority samples, i.e., oversampling techniques. For example, Random Over Sampler (ROS) [12], Synthetic Minority Oversampling Technique (SMOTE) [13], Adaptive Synthetic (ADASYN) [14]. However, excessive sampling of minority classes is likely to cause overfitting problems [6]. From the algorithm, considering the difference in the cost of different misclassification situations, introducing cost-sensitive factors and designing a cost-sensitive classification algorithm is one of the methods to solve the problem of class imbalance. At present, the use of cost-sensitive algorithms to solve the class imbalance problem in the field of intrusion detection is also involved, such as the literature [15], [16].

Regarding solving the problem of data imbalance and unknown attack detection in intrusion detection, this article proposes a novel network intrusion detection model called CWGAN-CSSAE, which combines improved CWGAN and a cost-sensitive stacked autoencoder. CWGAN-CSSAE uses a combination of data and algorithms to solve the problems of imbalanced class distribution, which improves the accuracy of the model's detection of minority attacks and unknown attacks. The advantages of the network intrusion detection model proposed in this article are as follows:

1) CWGAN can capture the real data distribution, and further, generate specified types of attack samples based on preset labels, which reduces the imbalance of the training set. Also, the newly generated samples simulate the unknown network attack. It is helpful to improve the accuracy of the model's detection of minority attacks and unknown attacks.

2) CWGAN introduces a gradient penalty term and L2 regularization. It overcomes the problems of mode collapse and gradient convergence. Thus, it effectively enhances the stability of network training. Besides, it can alleviate the problem of overfitting efficiently.

3) This article proposes a cost-sensitive loss function to improve stacked autoencoder (SAE) based on the number of different attacks. Give a larger misclassification cost to the minority attacks, and give a smaller misclassification cost to the majority attacks. It can not only improve the overall performance of the model but also improve the accuracy of the model's detection of minority attacks.

4) CSSAE can extract high-level abstract features of network data by extracting features layer by layer, as well as fine-tune network parameters to obtain the optimal model, which can better deal with large-scale and complex network attacks.

5) CWGAN-CSSAE uses a combination of data level and algorithm level to solve the problem of data imbalance. On the one hand, it avoids the overfitting problem caused by only using CWGAN to over-generate minority samples. On the other hand, it overcomes the disadvantage of giving too large misclassification cost to very few classes when only using cost-sensitive classification algorithms. Compared with the existing intrusion detection models, better results have been achieved on the NSL-KDD (KDDTest-21), NSL-KDD (KDDTest$^+$) and UNSW-NB15 datasets.

The remainder of this article is structured as follows. Section II introduces the research of deep learning in the field of intrusion detection and related research to solve the problem of class imbalance. Section III introduces relevant background knowledge and proposes an improved model. Besides, Section IV describes the proposed intrusion detection model in detail. Section V describes the experimental results and performance analysis. Finally, Section VI presents some conclusions and further work.

II. RELATED WEORKS

A. RELATED RESEARCH ON DEEP LEARNING IN NIDS

With the further development of research in deep learning, many researchers apply deep learning to network intrusion detection. Its advantage of automatically extracting high-level abstract features helps to complete the classification of large-scale and complex network data. The application of deep learning in network intrusion detection has made some remarkable research results [17]–[22].

Li *et al.* [17] proposed a network intrusion detection method based on deep learning. First of all, an image conversion method is proposed to process NSL-KDD data, as well as convolutional neural networks ResNet and GoogLeNet are used to learn and recognize the features of the converted graphics. Experimental results show that the CNN model is pretty sensitive to the images transferred from the attack data.

Tama *et al.* [18] proposed a two-stage classifier ensemble for intelligent anomaly-based Iintrusion detection system (TSE-IDS). First, the hybrid feature selection technique is used to reduce the feature amount of the training dataset, and then the two-level classifier is used for classification. TSE-IDS has achieved good classification accuracy on KDDTest$^+$, KDDTest-21 and UNSW-NB15 datasets.

Yin *et al.* [19] proposed a network intrusion detection method based on recurrent neural network RNN-IDS. They have studied the binary classification performance and multivariate classification performance of RNN-IDS. Experiments based on the NSL-KDD dataset show that RNN-IDS is an intrusion detection model with excellent performance and high accuracy. It has achieved 83.28% and 68.55% detection precision on the KDDTest$^+$ and KDDTest-21 test sets, respectively.

Aygun and Yavuz [20] proposed an anomaly detection attack model based on autoencoder and an anomaly detection attack model based on denoising autoencoder. The experiment was conducted on the NSL-KDD (KDDTest$^+$) dataset, as well as the recognition accuracy rates reached 88.28% and 88.65% respectively, which effectively improved the detection accuracy of zero-day attacks.

Ma *et al.* [21] proposed an intrusion detection method for detecting malicious network traffic based on the combination of spectral clustering (SC) and deep neural network algorithm, called SCDNN. First of all, according to the sample similarity, the dataset is divided into k subsets using clustering centers. Then, the similarity feature is used to measure the distance between the data points in the test set and the training set, as well as it is used as the input of the deep neural network algorithm. The recognition accuracy of this method on the KDDTest$^+$ and KDDTest-21 datasets reached 72.64% and 44.55%, respectively.

Khan *et al.* [22] proposed a new network intrusion detection method based on two-stage deep learning (TSDL) model. The model includes two decision-making stages: the first stage is to use a probability score value to classify network traffic as normal or abnormal traffic, and the second stage is to use the probability score value as an additional feature in the testing phase. Experimental results show that the recognition rate of the model on the KDD99 and UNSW-NB15 datasets reached 99.996% and 89.134%, respectively.

TABLE 1. The recognition accuracy, detection rate, and F1 score (%) of the methods on NSL-KDD (KDDTest$^+$), NSL-KDD (KDDTest-21) and UNSW-NB15 datasets.

Datasets	Models	Acc	DR	F1
NSL-KDD (KDDTest+)	SAVAER-DNN [11]	89.36	95.98	90.08
	ResNet [17]	79.14	69.41	79.12
	GoogLeNet [17]	77.04	65.64	76.50
	TSE-IDS [18]	85.79	86.80	/
	RNN-IDS [19]	83.28	73.12	83.22
	AE [20]	88.28	87.68	89.51
	DAE [20]	88.65	83.08	89.28
	SCDNN [21]	72.64	57.48	/
NSL-KDD (KDDTest-21)	SAVAER-DNN [11]	80.30	95.19	86.92
	TSE-IDS [18]	72.52	72.50	/
	RNN-IDS [19]	68.55	/	/
UNSW-NB15	SAVAER-DNN [11]	93.01	91.94	93.54
	TSE-IDS [18]	91.27	91.30	/
	TSDL [22]	89.13	/	/

The recognition accuracy, detection rate, and F1 score of the above methods on the benchmark NSL-KDD (KDDTest$^+$), NSL-KDD (KDDTest-21) and UNSW-NB15 datasets is summarized in Table 1.

The above-mentioned deep learning methods have achieved satisfactory results in the network intrusion detection system.

However, they pay too little attention to the problem of data imbalance and unknown attack detection, resulting in poor detection of minority attacks and unknown attacks.

B. RELATED RESEARCH ON IMBALANCED DATA

The intrusion detection model trained with the imbalanced training set has serious biases, which means that the model will pay too much attention to normal behaviors, which will reduce the recognition effect of attack behaviors with a small number of samples. Thus, how to achieve excellent results on imbalanced datasets is a major challenge in the application of deep learning to the field of intrusion detection. In recent years, domestic and foreign scholars have researched the problem of class imbalance in network intrusion detection.

To solve the problem of data imbalance, Lee and Park [6] proposed an intrusion detection method GAN-RF based on generative adversarial network and random forest algorithm. GAN successfully solves the problems of over-fitting and class overlap in traditional over-sampling technology. Experimental results also show that the model shows good performance on imbalanced intrusion detection data.

In order to improve the detection accuracy of low-frequency attacks and unknown attacks, Yang *et al.* [11] proposed a new network intrusion detection model SAVAER-DNN. The model uses SAVAER's decoder to synthesize new low-frequency and unknown attack samples, as well as uses SAVAER's encoder to extract high-level abstract features of the original sample to initialize DNN. DNN is used as a classifier to complete the classification of intrusion detection data. Through multi-classification experiments on the benchmark NSL-KDD (KDDTest$^+$), NSL-KDD (KDDTest-21) and UNSW-NB15 datasets, it is concluded that compared with other state-of-the-art intrusion detection models, the SAVAER-DNN algorithm has a better detection effect on low-frequency attacks and unknown attacks.

Kim *et al.* [23] proposed a zero-day attack detection model based on Transfer Generative Adversarial Network (tGAN). At the same time, they creatively use the autoencoder structure to pre-train GAN, which enhances the stability of GAN in the training process. And use the t-SNE algorithm to visualize the clustering mode of malware. From the experimental results, we can know that the model has better performance than machine learning algorithms.

Long *et al.* [24] proposed an active cost-sensitive learning method for intrusion detection. A cost-sensitive learning method is a basic classifier and the sampling criterion of the maximum misclassification cost. The experimental results based on the KDDCUP 99 intrusion detection dataset show that the method is effective.

To solve the problem of the class imbalance that is common in industrial intrusion detection systems, Li *et al.* [15] proposed a cost-sensitive online learning algorithm. Experiments conducted on the two test data of the natural gas pipeline system and power system have proved that the algorithm can effectively improve the detection rate of network attacks in industrial control systems.

Zhang *et al.* [25] proposed an intrusion detection model, named parallel cross convolutional neural network (PCCN). PCCN extracts flow features by fusing two branched convolutional neural networks, thereby improving the detection performance of unbalanced data. Based on the experimental results on the CICIDS2017 dataset, the model not only has a good detection performance on unbalanced data, but also reduces the detection time.

It can be seen that the above method has achieved satisfactory results on network intrusion datasets with imbalanced categories. Inspired by previous research, this article proposes a new hybrid intrusion detection model called CWGAN-CSSAE. The model solves the problem of class imbalance from the data level and algorithm level at the same time. Firstly, CWGAN that introduces gradient penalty and L2 regularization is used to generate specified minority attacks, as well as merge the generated attacks into the original training set to construct a new training dataset. It increases the diversity of samples and reduces the imbalance of the dataset. Secondly, based on the new training dataset, the misclassification cost is set according to the imbalanced proportion of samples and a cost-sensitive SAE network is proposed. On the one hand, CSSAE can extract high-level abstract features to process large-scale network data. On the other hand, it can improve the detection of minority attacks. Finally, the AdaBound [26] algorithm is used to optimize the training process of CSSAE.

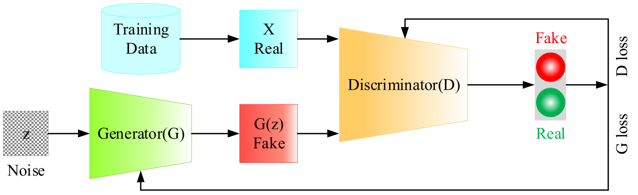


FIGURE 1. The network structure of GAN.

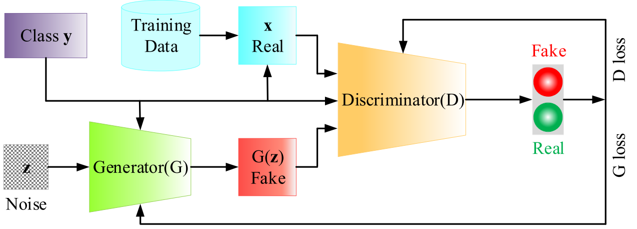


FIGURE 2. The network structure of CGAN.

III. METHODOLOGY

This section mainly introduces two parts, one part is an introduction to GAN and its improved versions CGAN and WGAN. This article puts forward the improved CWGAN based on it. The other part introduces SAE and the proposed CSSAE based on the number of samples.

A. REVIEW OF GANs

The Generative Adversarial Network (GAN) is a new type of generative model proposed by Goodfellow *et al.* in 2014 [27]. Inspired by the zero-sum game, GAN is composed of a generator and a discriminator. Its network structure is shown in Figure 1. The objective function of the GAN is:

$$\min_G \max_D V(G, D) = \min_G \max_D E_{\boldsymbol{x}\sim p_r}[\log D(\boldsymbol{x})] + E_{z\sim p_z}[\log(1 - D(G(z)))] \quad (1)$$

In the above equation, z is random noise, p_z is the distribution of noise samples z, p_r is the distribution of real data $\boldsymbol{x}$, p_g is the distribution of attack samples generated by G, $G(z)$ is the pseudo data generated by generator G, as well as $E(\cdot)$ is the expected value. The two networks confront each other and iteratively optimize, optimize D to maximize the accuracy of discriminating data sources, and optimize G to generate more realistic fake samples to deceive the discriminator D.

CGAN is a conditional version of GAN, and its network structure is shown in Figure 2. In CGAN, both the generator G and the discriminator D are added with an implicit label $\boldsymbol{y}$, and through $\boldsymbol{y}$ can generate samples of the specified label (type). Therefore, according to equation (1), the objective function of CGAN is:

$$\min_G \max_D V(G, D) = \min_G \max_D E_{\boldsymbol{x}\sim p_r}[\log D(\boldsymbol{x}|\boldsymbol{y})] + E_{z\sim p_z}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))] \quad (2)$$

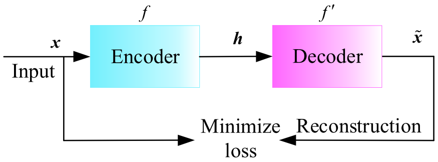


FIGURE 3. Auto-Encoder architecture.

In the equation, $\boldsymbol{y}$ is the one-hot code of the specified attack type. The optimization of CGAN is similar to that of GAN. Due to the use of Jensen-Shannon (JS) divergence as the metric for generating samples, both GAN and CGAN have problems such as mode collapse and training instability (gradient disappearance) [28].

In 2017, Arjovsky *et al.* [29] proposed WGAN that theoretically solved the problems of gradient disappearance during the training of GAN and CGAN. In addition, experiment results showed that WGAN effectively solved the problems of mode collapse. The core idea of WGAN is to change the JS divergence when calculating the loss function to Wasserstein distance (also called Earth-Mover (EM) distance), which is defined as follows:

$$W(p_r, p_g) = \inf_{\gamma \in \prod(p_r, p_g)} E_{(\boldsymbol{x},\boldsymbol{y})\sim\gamma}[\|\boldsymbol{x}-\boldsymbol{y}\|] \tag{3}$$

In the equation, $\prod(p_r, p_g)$ is the set of all possible joint distributions of p_r and p_g. $W(p_r, p_g)$ can be regarded as the "minimum consumption" of transforming distribution p_r into distribution p_g under "optimal path planning". The objective function of WGAN is given by

$$\min_G \max_D V(G, D) = \min_G \max_D E_{\boldsymbol{x}\sim p_r}[D(\boldsymbol{x})] - E_{z\sim p_z}[D(G(z))] \tag{4}$$

B. STACKED AUTO-ENCODER

Autoencoder (AE) is a fully connected unsupervised neural network that extracts features by reconstructing unlabeled data [30]. AE network is composed of two parts: encoder and decoder, its network structure is shown in Figure 3.

1) The encoder uses a deterministic mapping function to map input $\boldsymbol{x}$ to hidden layer representation $\boldsymbol{h}$. Usually, this mapping function f is non-linear. The encoding process is as follows:

$$\boldsymbol{h} = f(\boldsymbol{Wx} + \boldsymbol{b}) \tag{5}$$

In the equation, $\boldsymbol{W}$ is the weight between the input and the hidden layer representation, and $\boldsymbol{b}$ is the bias.

2) The decoder reconstructs the hidden layer representation $\boldsymbol{h}$ to obtain output $\tilde{\boldsymbol{x}}$. The decoding process is:

$$\tilde{\boldsymbol{x}} = f'(\boldsymbol{W'h} + \boldsymbol{b'}) \tag{6}$$

In the equation, $\boldsymbol{W'}$ is the weight between the hidden layer representation and the output, and $\boldsymbol{b'}$ is the deviation. $\tilde{\boldsymbol{x}}$ is the reconstruction output of $\boldsymbol{y}$.

The autoencoder is trained by minimizing the reconstruction error. Assuming the training set $D = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$, the loss function of AE is:

$$L(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \frac{1}{N}\sum_{x\in D}\|\boldsymbol{x}-\tilde{\boldsymbol{x}}\|^2 \tag{7}$$

The autoencoder adjusts the network parameters through the backpropagation algorithm of the error, and makes the loss function reach the minimum through iterative training, thereby extracting the abstract features in the data.

Stacked Autoencoder (SAE) is a deep neural network composed of multi-layer autoencoders. SAE is a deep learning model, which has powerful feature extraction capabilities, that is why it can better handle large-scale and complex data classification problems. SAE uses a greedy layered training method for network training, and uses the hidden layer output of the previous layer of AE as the input of the next layer of AE. And the AdaBound algorithm is used to fine-tune the entire network.

C. PROPOSED METHODOLOGY

1) CWGAN

This article makes full use of the advantages of CGAN and WGAN to propose a conditional Wasserstein Generative Adversarial Network (CWGAN), which can improve the quality of newly generated minority attack samples. It can solve the problems of model collapse and training instability during training. Also, it generates data for the specified model. The objective function of CWGAN is:

$$L_{CWGAN} = E_{\boldsymbol{x}\sim p_r}[D(\boldsymbol{x}|\boldsymbol{y})] - E_{z\sim p_z}[D(G(z|\boldsymbol{y}))] \tag{8}$$

However, due to the use of weight clipping to force the Lipschitz constraint to be satisfied, CWGAN still produces poor samples or does not converge in some cases. Therefore, this article add a gradient penalty term to its input to replace the original weight reduction method [31]. Its objective function is:

$$L_{GP} = E_{\hat{\boldsymbol{x}}\sim p_{x,z}}[(\left\|\nabla D(\hat{\boldsymbol{x}})\right\|_2 - 1)^2] \tag{9}$$

In the equation, $\|\cdot\|_2$ represents the 2-norm; $\hat{\boldsymbol{x}}$ is obtained by random interpolation sampling on the line between the real sample $\boldsymbol{x}$ and the generated sample $G(z|\boldsymbol{y})$. The calculation equation is $\hat{\boldsymbol{x}} = \varepsilon\boldsymbol{x} + (1-\varepsilon)G(z|\boldsymbol{y})$. ε obeys a uniform distribution on [0, 1].

In order to enable generator G to generate data closer to the real sample and prevent overfitting, this article introduce a traditional loss $L2$ distance into CWGAN. The equation is:

$$L_{L_2} = E_{\boldsymbol{x},z\sim p_{x,z}}[\|\boldsymbol{x} - G(z|\boldsymbol{y})\|_2] \tag{10}$$

Finally, the objective function is a combination of CWGAN's original objective function, gradient penalty term and $L2$ regularization.

$$L_{\text{total}} = \min_G \max_D L_{CWGAN} + \lambda_1 L_{GP} + \lambda_2 L_{L_2} \tag{11}$$

In the equation, λ_1 is the coefficient of gradient penalty; λ_2 is the coefficient of $L2$ regularization.

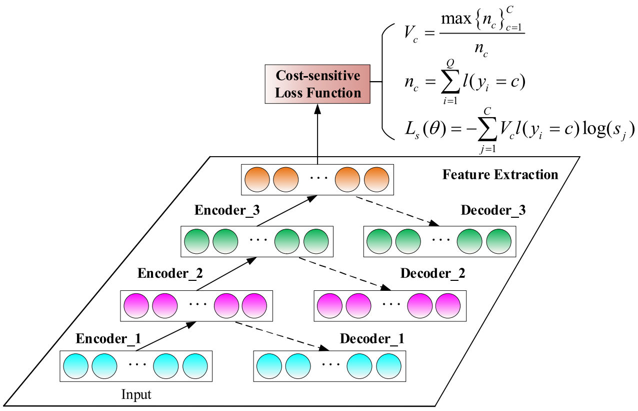

Cost-sensitive Loss Function
$V_c = \frac{\max\{n_c\}_{c=1}^{C}}{n_c}$
$n_c = \sum_{i=1}^{Q} l(y_i = c)$
$L_s(\theta) = -\sum_{j=1}^{C} V_c l(y_i = c)\log(s_j)$
Feature Extraction
Encoder_3
Decoder_3
Encoder_2
Decoder_2
Encoder_1
Decoder_1
Input

FIGURE 4. CSSAE architecture.

During the training process, the discriminator D and the generator G perform alternating confrontation training, and the Adam [32] algorithm is used to optimize the parameter update of the entire network.

2) CSSAE

Since SAE defaults that the cost of misclassification for each class is the same without considering the problem of data imbalance, it has a low recognition rate for minority classes. To solve the above problem, this article gives a larger misclassification cost to minority samples and a smaller misclassification cost to most samples during the training process of SAE. As a result, a cost-sensitive stacked autoencoder (CSSAE) is proposed. Its network structure is shown in Figure 4.

The stacked three-layer AE is mainly used for feature extraction. The following Softmax classifier is mainly used for classification. Also, a cost-sensitive loss function was introduced to improve the detection accuracy of minority samples.

According to the proportion of the sample size, the misclassification cost of different types of samples are as follows:

$$V_c = \frac{\max\{n_c\}_{c=1}^{C}}{n_c} \tag{12}$$

$$n_c = \sum_{i=1}^{Q} l(y_i = c) \tag{13}$$

In the equation: Q is the total number of samples; n_c is the number of samples of different classes, which reflects the imbalanced distribution of the dataset. The misclassification cost of each type of sample is adaptively calculated by equation (12). When the samples of each category are balanced in the dataset, $V_c = 1$. According to equation (12), the cost-sensitive loss function is defined as:

$$L_{CS}(\theta) = -\sum_{j=1}^{C} V_c l(y_i = c)\log(s_j) \tag{14}$$

$$s_j = \frac{e^{x_j}}{\sum_{k=1}^{C} e^{x_k}} \tag{15}$$

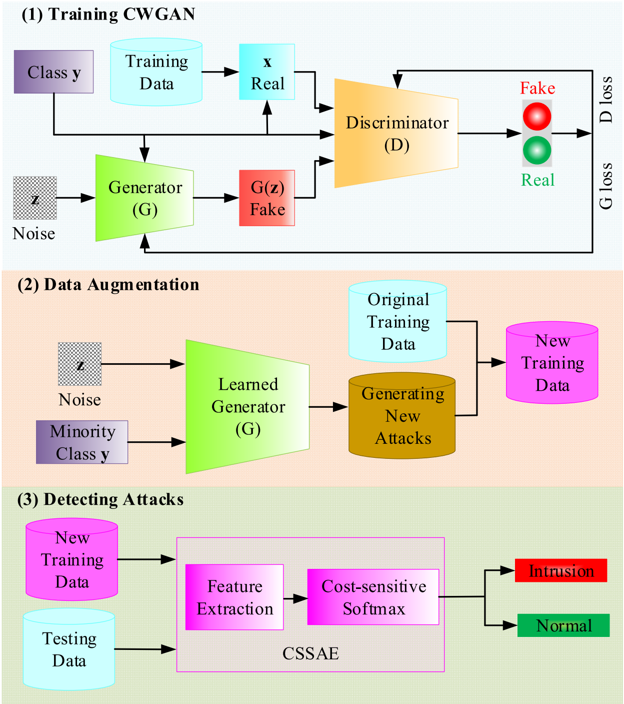


FIGURE 5. The proposed network intrusion detection framework.

In the equation, x_j is the j-th output value of the output layer; s_j is the probability of the j-th output value of the output layer; C is the number of neurons in the output layer; c is the actual category; y_i is the predicted category of the network output. If the equation in the $l(\cdot)$ brackets is true, the function value is 1, otherwise it is 0.

During the model training process, the AdaBound algorithm is used to optimize the model parameters. Besides the parameter θ is updated layer by layer to minimize $L_{CS}(\theta)$ so that the model has the highest recognition rate.

IV. PROPOSED NETWORK INTRUSION DETECTION FRAMEWORK

This section introduces the proposed CWGAN-CSSAE network intrusion detection model in details. The model mainly includes 4 steps: 1) data preprocessing; 2) training CWGAN network, discriminator network and generator network for alternating training; 3) the generation of minority attack samples and construction of a new training dataset; 4) training of CSSAE Network to classify the test set. The framework of CWGAN-CSSAE network intrusion detection model is shown in Figure 5.

A. DATA PREPROCESSING

The NSL-KDD and UNSW-NB15 datasets contain two types of features, numeric and string type. Since CWGAN-CSSAE cannot train string type data, it is necessary to convert string type features into numeric type. Here, one-hot encoding is used to map character attributes to binary values. For example, the protocol types TCP, UDP, and ICMP in the NSL-KDD dataset are represented as [1, 0, 0], [0, 1, 0], [0, 0, 1] after feature mapping.

After numerical processing, the data in the dataset is converted to numerical data. However, the numerical difference in the numerical data is large. For example, the range of the feature attribute duration (connection duration) in the NSL-KDD dataset is 0∼58329. Large differences in values are likely to cause problems such as slower network convergence and saturation of neuron output. Thus, it's vital to normalize the data. Here, the maximum-minimum normalization method is used to limit the data to [0, 1]. The equation is as follows:

$$\mathrm{x} = \frac{x - M_{\min}}{M_{\max} - M_{\min}} \tag{16}$$

In the equation, x is the value to be normalized; $M_{\min}$ is the minimum value of the dimension; $M_{\max}$ is the maximum value of the dimension.

B. TRAINING CWGAN

In order to further improve the stability of training and the generalization performance of the model, this article adopts a small batch training method in the game process of discriminator D and generator G. The training process of CWGAN is as follows:

(1) The random noise vector z that obeys the Gaussian distribution and the category label $\boldsymbol{y}$ are spliced as the input of the generator G. And the output is new attack samples $G(z)$ generated by training G. At this time, the similarity between the new attack samples and the real samples is very low.

(2) Fix the generator G and train the discriminator D. The real samples and the new attack samples generated by G are mixed as the input of the discriminator. The output is the classification probability value of the samples belonging to the real samples $\boldsymbol{x}$ and the pseudo samples $G(z)$. Then the probability value is converted into a predicted label through the activation function. The objective function of the discriminator D is $\max\limits_{D} L_{CWGAN} + \lambda_1 L_{GP} + \lambda_2 L_{L_2}$.

(3) The generator G is trained through the concatenation of $G - D$. After the training in step (2), the discriminating ability of the discriminator D is improved. At this time, the generator G is trained to generate pseudo samples with higher simulation. The objective function of generator G is $\min\limits_{G} L_{CWGAN} + \lambda_1 L_{GP} + \lambda_2 L_{L_2}$.

(4) Before the set number of cycles or the loss value reaches the threshold, the steps (2) and (3) are executed cyclically. The discriminator D and the generator G are trained alternately to make the generated pseudo samples are getting closer to the real samples. Also, the Adam algorithm is used to optimize the gradient update process to continuously optimize the loss value L_{total}.

C. BUILDING NEW TRAINING DATASETS

After the training, CWGAN is used to generate new attack samples of specified categories according to the number of attack samples of different types in the original dataset. Then, the new attack samples are merged into the original training dataset to construct a new training dataset, so as to alleviate the problem of imbalanced training datasets and increase the diversity of training samples. Furthermore, the accuracy of CWGAN-CSSAE's detection of minority attacks and unknown attacks is improved.

D. DETECTING ATTACKS

In the attacks detection stage, this article introduces a cost-sensitive loss function based on stacked autoencoder to construct CSSAE as a classifier of the intrusion detection model. The CSSAE network uses a small batch training mode. In the pre-training phase, the three AEs in CSSAE are trained separately, and then the three single-layer AEs are stacked in the manner shown in Figure 4, and then the cost-sensitive Softmax is pre-trained. After the pre-training phase, the three autoencoders in CSSAE and the cost-sensitive Softmax are introduced as a whole to fine-tune their free parameters. The AdaBound algorithm is used to optimize the parameter update process in the training process. Use the new training dataset as the input of CSSAE for model training. After CSSAE training is completed, the test set is input into the trained CSSAE for classification to realize network attacks detection.

The proposed CWGAN-CSSAE intrusion detection model is detailed in Algorithm 1:

V. EXPERIMENTS

A. PERFORMANCE EVALUATION METRICS

In order to effectively evaluate the performance of the proposed network intrusion detection model, this article selects 6 common indicators to evaluate the classification performance of the proposed CWGAN-CSSAE intrusion detection system: accuracy, precision, recall, DR (detection rate), FPR (false positive rate), F1 score, and G-mean.

Accuracy is the proportion of test samples correctly predicted to all test samples. The value is in the range of [0, 1], and the larger the value, the better the classification performance of the model. The definition of Accuracy is:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

Precision is the proportion of all predicted attack samples that are actually attack samples. The value is in the range of [0, 1], and the larger the value, the better the classification performance of the model. The definition of Precision is:

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

DR (also known as Recall) is the proportion of attack samples predicted by the model to the total number of actual attack samples. If the DR is larger, the classification performance of the model is better ($DR \in [0, 1]$). DR is defined as:

$$DR = Recall = \frac{TP}{TP + FN} \tag{19}$$

Algorithm 1 CWGAN-CSSAE Network Intrusion Detection Model

```
**Input:** Training dataset $(\boldsymbol{x}, \boldsymbol{t})$; random noise $z$; specified generation category $\boldsymbol{y}$; number of generated samples $n$; CWGAN's learning rate $\alpha_0$, number of training cycles $epoch_0$ and batch size $batchsize_0$; CSSAE's learning rate $\alpha_1$, number of training cycles $epoch_1$ and batch size $batchsize_1$.
**Output:** The final network attack recognition result
1: Data preprocessing. The character features and classification labels are coded by one-hot, and all features are normalized.
2: Parameters setting. CWGAN: Learning rate $\alpha_0$ is 0.0001. The coefficient of gradient penalty $\lambda_1$ is 10; the coefficient of $L2$ regularization $\lambda_2$ is 100. The number of training cycles $epoch_0$ is 6000. The batch size $batch\ size_0$ is 500. The Generator of network structure is 400-200-100-122. The Discriminator of network structure is 400-100-20-1. CSSAE: Learning rate $\alpha_1$ is 0.001. Training cycle number $epoch_1$ is 200. Batch size $batch\ size_1$ is 200. The network structure is 500-200-100-X (X represents the number of categories).
3: CWGAN is trained on the original training set. The random noise $z$ and the minority class label $\boldsymbol{y}$ are input into CWGAN. And the discriminator $D$ and generator $G$ are trained in alternating confrontation.
4: Random noise $z$ and minority class label $\boldsymbol{y}$ are input to generate new minority class attack samples, which are merged into the original training set to form new training dataset $(\boldsymbol{x}_{new}, \boldsymbol{t}_{new})$.
5: CSSAE training. The new training dataset $(\boldsymbol{x}_{new}, \boldsymbol{t}_{new})$ is used as input. Firstly, 3 AEs will be trained individually. Then the trained 3 AEs are stacked according to Figure 4. And Softmax is trained. Then the entire network will be fine-tuned to optimize the parameters of each AE and Softmax classifier in CSSAE to obtain the final training model.
6: The test dataset is input into the trained CSSAE to detect the attack.
7: Return the final network attack recognotion result.
```

FPR is the proportion of all normal samples that are incorrectly classified as attack samples. The smaller the FPR value, the better the classification performance of the model ($FPR \in [0, 1]$). FPR is defined as:

$$FPR = \frac{FP}{TN + FP} \tag{20}$$

F1 score is the harmonic average of Precision and detection rate. The value is in the range of [0, 1], and the larger the value, the better the classification performance of the model. Compared with accuracy, F1 score is more suitable for performance evaluation of the classification results of imbalanced datasets. The definition of F1 score is:

TABLE 2. Confusion matrix.

	Predicted normal	Predicted attack
Actual normal	TN	FP
Actual attack	FN	TP

$$\begin{aligned} F1 &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \\ &= \frac{2 \times TP}{2 \times TP + FP + FN} \end{aligned} \tag{21}$$

G-mean is a comprehensive index of the accuracy of the positive class and the accuracy of the negative class. The larger the G-mean value, the better the classification performance of the model ($G{-}mean \in [0, 1]$). The definition of G-mean is:

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{22}$$

These performance evaluation indicators are calculated on the basis of the confusion matrix [33], as shown in Table 2. Among them, TP (True Positive) is the number of samples that are correctly classified as attacks, FP (False Positive) is the number of normal samples that are incorrectly classified as attacks, TN (True Negative) is the number of samples that are correctly classified as normal, FN (False Negative) is to incorrectly classify the attack sample as a normal number.

B. THE BENCHMARK DATASETS

1) NSL-KDD DATASET

Tavallaee *et al.* [34] proposed an improved version of the NSL-KDD dataset on the basis of the KDDCUP99 dataset, which not only eliminated a large amount of redundant data in KDDCUP99, but also adjusted the training set and test set so that the NSL-KDD dataset can be more suitable for network intrusion detection experiments. Its attack behavior includes four types: denial of service (Dos), port scanning and detection (Probe), illegal access to local super users (U2R), at the same time, illegal access to remote machines (R2L). The normal access data is Normal. Each piece of data includes 41 features and 1 label. We use the KDDTrain$^+$_20 dataset as the training set, as well as KDDTest$^+$ and KDDTest-21 [35] as the test sets. The detailed information of the NSL-KDD dataset is shown in Table 3.

The data in Table 3 shows that the NSL-KDD training set KDDTrain_20percent has a serious imbalance in the category distribution, as well as the number of U2R and R2L attack samples is seriously low. In addition, in the test datasets KDDTest$^+$ and KDDTest-21, there are a large number of attack types that did not appear in KDDTrain_20percent.

2) UNSW-NB15 DATASET

UNSW-NB15 dataset [36] is a comprehensive network attack traffic dataset created by ACCS, which is more suitable

TABLE 3. Class distribution of the NSL-KDD dataset.

Class	Attack	Training set	Testing set	
		KDDTrain+_20	**KDDTest+**	**KDDTest-21**
Normal	normal	13449	9711	2152
Dos	neptune	8282	4657	1579
	smurf	529	665	627
	back	196	359	359
	teardrop	188	12	12
	pod	38	41	41
	land	1	7	7
	apache2	0	737	737
	mailbomb	0	293	293
	processtable	0	685	685
	udpstorm	0	2	2
sum		9234	7458	4342
Probe	ipsweep	710	141	141
	satan	691	735	727
	portsweep	587	157	156
	nmap	301	73	73
	saint	0	319	309
	mscan	0	996	996
sum		2289	2421	2402
U2R	buffer_overflow	6	20	20
	rootkit	4	13	13
	loadmodule	1	2	2
	perl	0	2	2
	httptunnel	0	133	133
	ps	0	15	15
	sqlattack	0	2	2
	xterm	0	13	13
sum		11	200	200
R2L	guess_passwd	10	1231	1231
	warezmaster	7	944	944
	imap	5	1	1
	multihop	2	18	18
	phf	2	2	2
	ftp_write	1	3	3
	spy	1	0	0
	warezclient	181	0	0
	named	0	17	17
	sendmail	0	14	14
	xlock	0	9	9
	xsnoop	0	4	4
	worm	0	2	2
	snmpgetattack	0	178	178
	snmpguess	0	331	331
sum		209	2754	2754
total		25192	22544	11850

for the research of intrusion detection system. UNSW-NB15 dataset is divided into training set (175,341 records) and test set (82,332 records). And it contains one normal and nine attacks. Attack records are less than normal records, especially Worms and Shellcode. The detailed information of the UNSW-NB15 dataset is shown in Table 4.

TABLE 4. Class distribution of the UNSW-NB15 dataset.

Class	Training set	Testing set
Normal	56000	37000
Generic	40000	18871
Exploits	33393	11132
Fuzzers	18184	6062
DoS	12264	4089
Reconnaissance	10491	3496
Analysis	2000	677
Backdoor	1746	583
Shellcode	1133	378
Worms	130	44
Sum	175341	82332

TABLE 5. CWGAN-CSSAE parameter settings.

	$epoch_0$	6000		$epoch_1$	200
	$batch\ size_0$	500		$batch\ size_1$	200
	α_0	0.0001		α_{10}	0.001
CWGAN	β_{01}	0.5	CSSAE	α_{1f}	0.1
Adam	β_{02}	0.9	Adabound	β_{11}	0.9
[32]	ε_0	10^{-8}	[26]	β_{12}	0.999
				ε_1	10^{-7}
				γ	10^{-3}

C. EXPERIMENTAL SETUP

1) PARAMETERS SETTING

Since there is no automated parameter optimization algorithm currently, we have to rely on researchers to conduct a large number of parameter tuning experiments. Then, the performance of models under different parameters are compared and analyzed. The final parameters of the model can be determined by adjusting the parameters repeatedly. All hidden layer activation functions of the proposed CWGAN-CSSAE model are ReLU [37], and the activation function of the CSSAE output layer is Softmax. The network structures of generator and discriminator are 400-200-100-122 and 400-100-20-1, respectively. The network structure of CSSAE is 500-200-100-X (X represents the number of categories). The specific training parameter settings are shown in Table 5.

2) EXPERIMENTALS SETTING

The experimental environment in this article is Tensorflow with a 64-bit Windows 10 operating system, and the computer is configured with Intel(R) Core(TM) i7-4790 CPU 3.60GHz, 64GB RAM, and RTX 2080Ti GPU.

In order to verify the performance of the proposed CWGAN-CSSAE network intrusion detection model, the following experiments are designed:

Experiment 1: Model training experiment. Figure 6(a) shows the loss curves of the discriminator and the generator of CWGAN on the NSL-KDD dataset. Figures 6(b) and 6(c) respectively show the loss curve and detection accuracy curve of CSSAE on the KDDTest$^+$ dataset. Figures 6(d) and 6(e) respectively show the loss curve and detection accuracy curve of CSSAE on the KDDTest-21 dataset. Figure 7(a) show the loss curves of the discriminator and the generator on the UNSW-NB15 dataset, respectively. Figures 7(b) and 7(c) respectively show the loss curve and detection accuracy curve of CSSAE on the UNSW-NB15 dataset. Figure 8 shows the confusion matrix of CWGAN-CSSAE on the KDDTest$^+$, KDDTest-21 and UNSW-NB15 datasets.

Experiment 2: Experiment with generating minority samples. Table 6 and 7 respectively describe the category distribution of the new training dataset on the NSL-KDD and UNSW-NB15 datasets. Figure 9 and 10 shows the original training dataset and the new training dataset enhanced with newly generated minority samples on the NSL-KDD and UNSW-NB15 datasets, respectively.

Experiment 3: Comparative experiment of different data enhancement algorithms. Figures 11 to 15 show the comparison results of the proposed CWGAN-CSSAE and other data enhancement algorithms.

Experiment 4: Comparative experiment of different classification algorithms. Figures 16 to 20 show the comparison results of the proposed CWGAN-CSSAE and other classification algorithms.

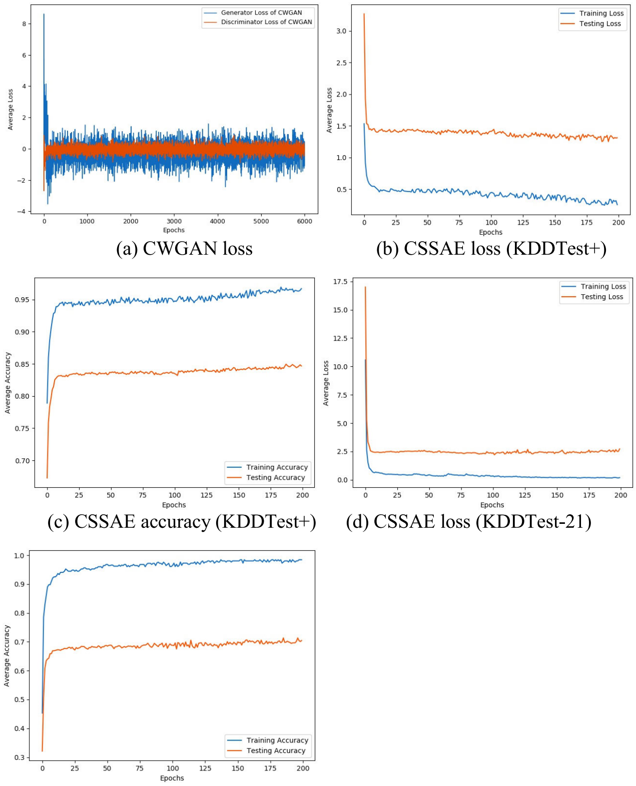
Experiment 5: The performance comparison experiment with the intrusion detection model reported in the existing intrusion detection literature. Tables 8, 9 and 10 compare the accuracy, precision, DR, F1 score, G-mean, FPR of the proposed CWGAN-CSSAE and previous intrusion detection methods on the KDDTest$^+$, KDDTest-21 and UNSW-NB15 datasets.

D. EXPERIMENT RESULTS AND ANALYSIS

1) MODEL TRAINING

CWGAN-CSSAE intrusion detection model training is divided into two parts. The first part is to start training the CWGAN network. Discriminator D and generator G conduct separate alternating iterative training. Discriminator D determines the true and false of the sample by minimizing the loss function. Generator G enhances the simulation degree of generated sample by minimizing the loss function. The training loss curves of the discriminator D and the generator G are shown in Figures 6(a) and 7(a).

After the training of the CWGAN network is completed, the generated attack samples are mixed with the original training dataset to form a new training dataset as the input of the CSSAE network. CSSAE training is divided into two stages: the pre-training stage and the fine-tuning stage. In the pre-training stage, the three layers of AE in CSSAE are studied individually in an unsupervised way, then the three layers of AE are stacked up and the Softmax is pretrained. In the fine-tuning stage, 3 layer of AE and the introduction of the cost-sensitive Softmax classifier are regarded as a whole to adjust the free parameters. The process of parameter tuning is intuitively manifested as the process of loss reduction, which means that the parameters obtained when the loss converges are the most excellent. The loss curves of CSSAE are shown in Figures 6(b), 6(d) and 7(b), and the recognition accuracy curves of CSSAE are shown in Figures 6(c), 6(e) and 7(c).

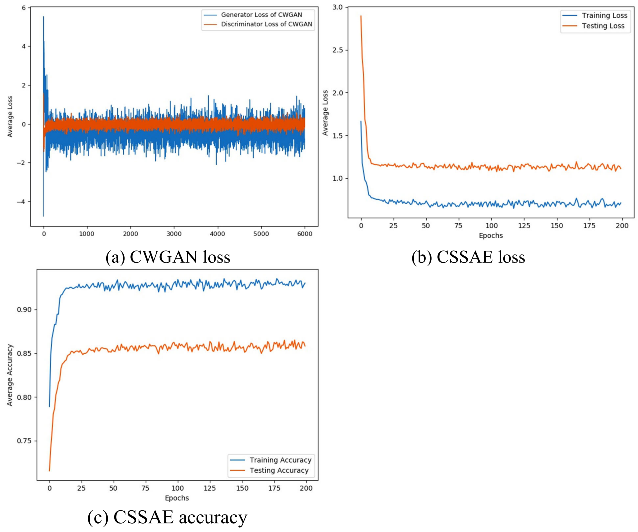


(a) CWGAN loss
(b) CSSAE loss (KDDTest+)
(c) CSSAE accuracy (KDDTest+)
(d) CSSAE loss (KDDTest-21)
(e) CSSAE accuracy (KDDTest-21)

FIGURE 6. Training graphs of CWGAN-CSSAE on the NSL-KDD dataset.

The data in Figures 6(a) and 7(a) shows that during network training, the loss of CWGAN discriminator and generator gradually decrease and tend to be minimal. It can be seen that the loss curve of the two fluctuates greatly, because the discriminator and generator are constantly fighting against each other during the training process. After game training, CWGAN reaches the Nash equilibrium point, which means that at this time, the CWGAN generator can generate highly simulated attack samples.

The data in Figures 6(b),6(d) and 7(b) shows that in the initial training stage, the training loss of CSSAE decreases rapidly with the increase of the number of iterations, as well as in the later stage of training it tends to stabilize and the training error is low. It can be seen from Figures 6(c) and 6(e) that the accuracy of the NSL-KDD training set has risen rapidly and stabilized to reach about 98.00%. The data in Figure 7(c) shows that the accuracy of UNSW-NB15 training set reaches about 93.00%. It shows that the CSSAE network has fast convergence speed, small training error and high accuracy. However, the classification accuracy on the KDDTest$^+$, KDDTest-21 and UNSW-NB15 test sets are only 84.61%, 70.07% and 85.90%, respectively. The reason may be that the model cannot detect the unknown types of attacks in the test set well.

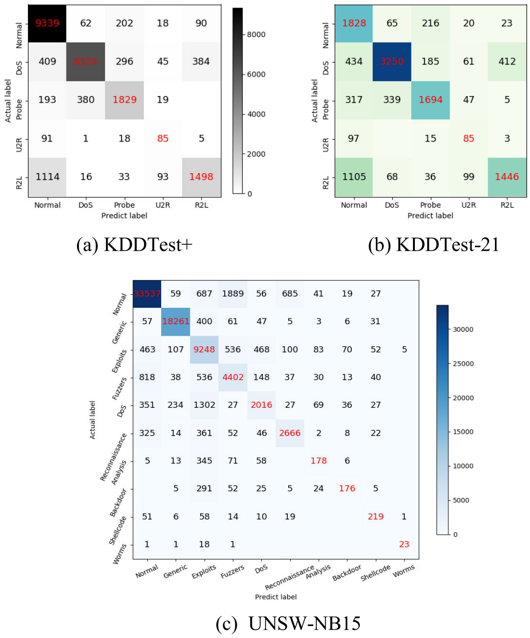


(a) CWGAN loss

(b) CSSAE loss

(c) CSSAE accuracy

FIGURE 7. Training graphs of CWGAN-CSSAE on the UNSW-NB15 dataset.



(a) KDDTest+

(b) KDDTest-21

(c) UNSW-NB15

FIGURE 8. Confusion matrix on the KDDTest$^+$, KDDTest-21 and UNSW-NB15 datasets.

The confusion matrix of CWGAN-CSSAE intrusion detection model is shown in Figure 8.

It can be seen from Figure 8 that CWGAN-CSSAE has a good recognition effect on normal traffic and various attack behaviors, and can well identify normal and attacks. However, the recognition effect of minority attacks needs to be further improved.

2) MINORITY CLASS DATA GENERATION

The data in Tables 3 and 4 shows that both the NSL-KDD training set and the UNSW-NB15 training set have a serious category distribution imbalance. For example, the number of samples for U2R and R2L attacks is seriously low.

TABLE 6. The newly generated training set on the NSL-KDD dataset.

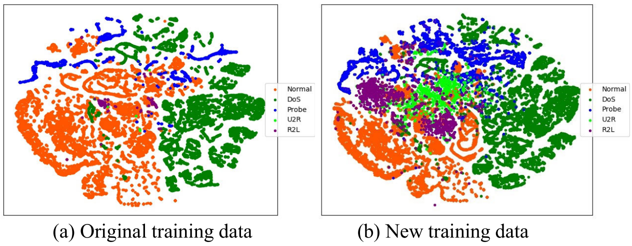
Class	number of original samples	Number of samples generated	Sum
Normal	13449	0	13449
DoS	9234	4215	13449
Probe	2289	3000	5289
U2R	11	1000	1011
R2L	209	2000	2209
Sum	25192	10215	35407

TABLE 7. The newly generated training set on the UNSW-NB15 dataset.

Class	number of original samples	Number of samples generated	Sum
Normal	56000	0	56000
Generic	40000	0	40000
Exploits	33393	0	33393
Fuzzers	18184	0	18184
DoS	12264	0	12264
Reconnaissance	10491	0	10491
Analysis	2000	8000	10000
Backdoor	1746	8000	9746
Shellcode	1133	8000	9133
Worms	130	4000	4130
Sum	175341	124423	203341

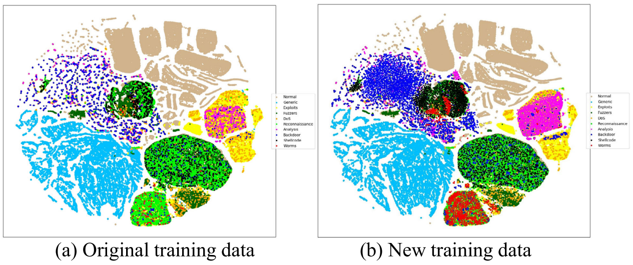
Because the imbalanced category distribution tends to make the classifier learn too much from the samples of the majority class, the detection accuracy of the minority class attack is very low. In addition, a large number of unknown types of attacks appeared in the NSL-KDD test sets. The training set KDDTrain$^+$_20 contained 22 attack types, while the test sets KDDTest$^+$ and KDDTest-21 appeared 17 unknown types of attacks. Data enhancement algorithm is an important method to solve the problem of class imbalance. On this basis, we propose an improved CWGAN to generate designated minority attack samples. The generated samples are added to the original training dataset to construct a new training dataset to reduce the imbalance of the training set and increase the diversity of training samples.

If the number of new samples generated by CWGAN far exceeds the number of samples in the original dataset, the quality of the new samples will decrease and the over-fitting phenomenon will be serious. For example, a large number of U2R attack samples are generated. Therefore, comprehensively consider the quality of the newly generated samples and the degree of imbalance in the categories of the dataset to determine the number of samples generated for each type. Then the newly generated attack samples are mixed with the original training set to get a new training set. The new NSL-KDD training set and the new UNSW-NB15 training set are shown in Tables 6 and 7, respectively.



(a) Original training data (b) New training data

FIGURE 9. t-SNE visualization of original and newly generated training data based on the NSL-KDD Dataset.



(a) Original training data (b) New training data

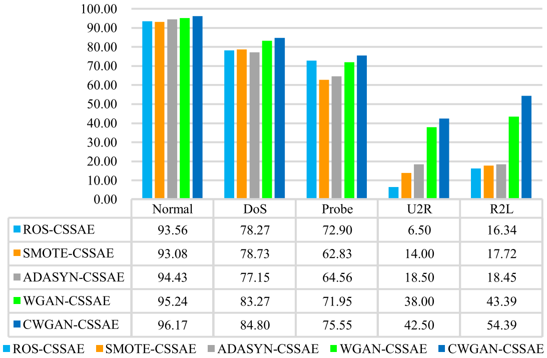
FIGURE 10. t-SNE visualization of original and newly generated training data based on the UNSW-NB15 Dataset.

In order to observe the newly generated minority samples more intuitively, the t-SNE (t Distributed Random Neighbor Embedding) [38] method is used to visualize the original dataset and the new training dataset. The results are shown in Figures 9 and 10.

The data in Figures 9(a) and 10(a) shows that the original training dataset is non-linearly separable. The data in Figures 9(b) and 10(b) shows that the newly generated minority attack samples have high similarities with the corresponding minority attack samples in the original dataset, such as U2R, R2L and Worms. However, some categories overlap. The reason is that t-SNE visualization is two-dimensional and cannot completely represent the real spatial distribution of the sample. Therefore, samples that seem to overlap may not overlap in real space. The classification results verify the separability of different attacks. It means that it is feasible and effective to use improved CWGAN to generate minority attacks to reduce the problem of class imbalance.

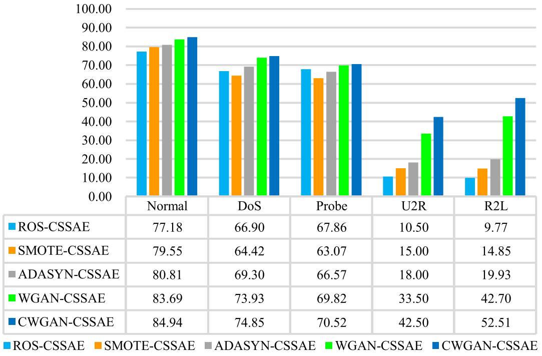
3) COMPARISON OF DATA AUGMENTATION ALGORITHMS

In order to balance the training dataset, the CWGAN-CSSAE model is proposed, which can improve the detection accuracy of minority attacks and unknown attacks. In order to verify the superiority of the CWGAN data enhancement algorithm, comparative experiments were designed. The selected comparison algorithms include classic data enhancement algorithms, such as ROS (Random Over Sampler) [12], SMOTE (Synthetic Minority Oversampling Technique) [13], and ADASYN (Adaptive Synthetic) [14], as well as the currently popular WGAN (Wassertein GAN) [29]. Based on the above data enhancement algorithm, four classification models are constructed: ROS-CSSAE, SMOTE-CSSAE, ADASYN-CSSAE, and WGAN-CSSAE. Note that the types and numbers of attacks generated by WGAN are consistent with those generated by CWGAN. The experimental results are shown in Figures 11 to 15.



	Normal	DoS	Probe	U2R	R2L
ROS-CSSAE	93.56	78.27	72.90	6.50	16.34
SMOTE-CSSAE	93.08	78.73	62.83	14.00	17.72
ADASYN-CSSAE	94.43	77.15	64.56	18.50	18.45
WGAN-CSSAE	95.24	83.27	71.95	38.00	43.39
CWGAN-CSSAE	96.17	84.80	75.55	42.50	54.39

FIGURE 11. Comparison of detection accuracy of different data augmentation methods on the KDDTest$^+$ dataset (%).



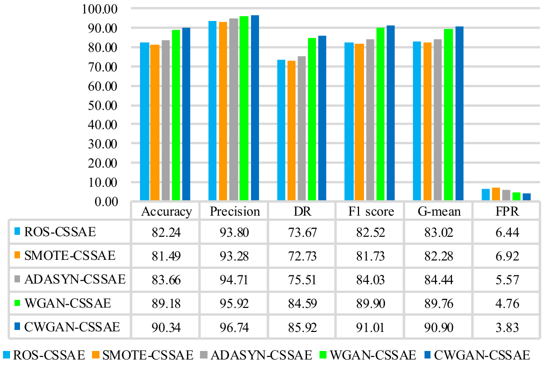
	Normal	DoS	Probe	U2R	R2L
ROS-CSSAE	77.18	66.90	67.86	10.50	9.77
SMOTE-CSSAE	79.55	64.42	63.07	15.00	14.85
ADASYN-CSSAE	80.81	69.30	66.57	18.00	19.93
WGAN-CSSAE	83.69	73.93	69.82	33.50	42.70
CWGAN-CSSAE	84.94	74.85	70.52	42.50	52.51

FIGURE 12. Comparison of detection accuracy of different data augmentation methods on the KDDTest-21 dataset (%).

Figures 11 and 12 respectively show the detection accuracy of different data enhancement methods on the KDDTest$^+$ and KDDTest-21 datasets for five types of attacks. It can be concluded that, compared with the other four data enhancement methods, CWGAN-CSSAE has the highest detection accuracy of the five types of attacks, especially the detection accuracy of U2R and R2L. The comparison experiment results show that the proposed CWGAN-CSSAE improves the detection accuracy of minority attacks. The data in Table 3 shows that there are a large number of attack types in the KDDTest$^+$ and KDDTest-21 datasets that do not appear in the training set, but the detection accuracy is still high. To a certain extent, it shows that CWGAN-CSSAE has a good ability to identify unknown attacks.

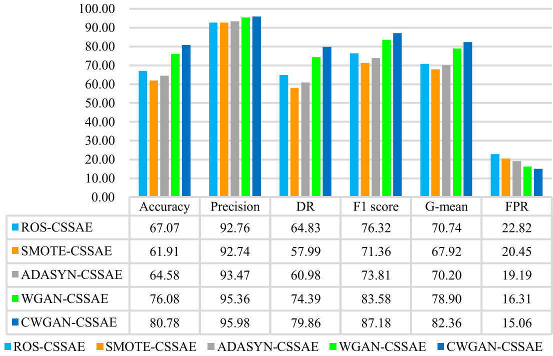
Figures 13, 14 and 15 show the overall classification performance of different data enhancement methods on the KDDTest$^+$, KDDTest-21 and UNSW-NB15 datasets, respectively. It can be seen that CWGAN-CSSAE has achieved the best results in accuracy, precision, DR, FPR, F1 score and G-mean. The experimental results show that CWGAN is an effective data enhancement method.

From this point of view, the proposed CWGAN-CSSAE performs better than ROS-CSSAE, SMOTE-CSSAE, and ADASYN-CSSAE. The reason is that ROS-CSSAE is only



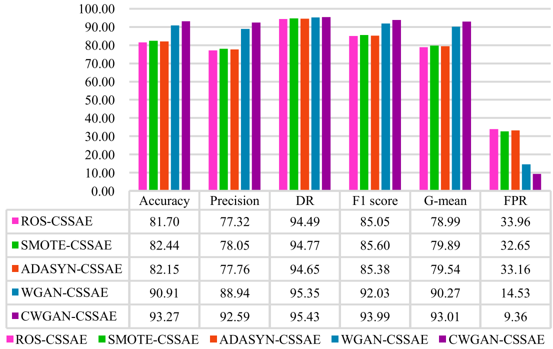
	Accuracy	Precision	DR	F1 score	G-mean	FPR
ROS-CSSAE	82.24	93.80	73.67	82.52	83.02	6.44
SMOTE-CSSAE	81.49	93.28	72.73	81.73	82.28	6.92
ADASYN-CSSAE	83.66	94.71	75.51	84.03	84.44	5.57
WGAN-CSSAE	89.18	95.92	84.59	89.90	89.76	4.76
CWGAN-CSSAE	90.34	96.74	85.92	91.01	90.90	3.83

FIGURE 13. Comparison of detection performance of different data augmentation methods on the KDDTest$^+$ dataset (%).



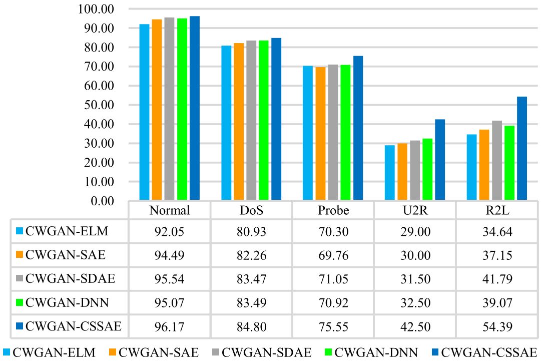
	Accuracy	Precision	DR	F1 score	G-mean	FPR
ROS-CSSAE	67.07	92.76	64.83	76.32	70.74	22.82
SMOTE-CSSAE	61.91	92.74	57.99	71.36	67.92	20.45
ADASYN-CSSAE	64.58	93.47	60.98	73.81	70.20	19.19
WGAN-CSSAE	76.08	95.36	74.39	83.58	78.90	16.31
CWGAN-CSSAE	80.78	95.98	79.86	87.18	82.36	15.06

FIGURE 14. Comparison of detection performance of different data augmentation methods on the KDDTest-21 dataset (%).



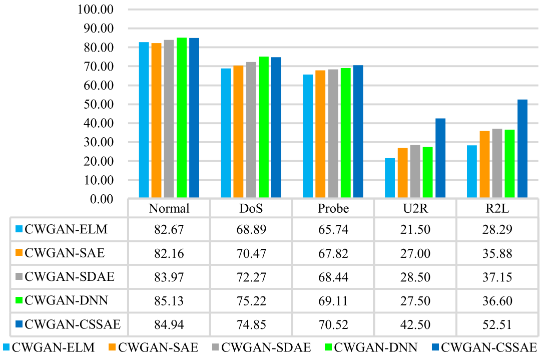
	Accuracy	Precision	DR	F1 score	G-mean	FPR
ROS-CSSAE	81.70	77.32	94.49	85.05	78.99	33.96
SMOTE-CSSAE	82.44	78.05	94.77	85.60	79.89	32.65
ADASYN-CSSAE	82.15	77.76	94.65	85.38	79.54	33.16
WGAN-CSSAE	90.91	88.94	95.35	92.03	90.27	14.53
CWGAN-CSSAE	93.27	92.59	95.43	93.99	93.01	9.36

FIGURE 15. Comparison of detection performance of different data augmentation methods on the UNSW-NB15 dataset (%).



	Normal	DoS	Probe	U2R	R2L
CWGAN-ELM	92.05	80.93	70.30	29.00	34.64
CWGAN-SAE	94.49	82.26	69.76	30.00	37.15
CWGAN-SDAE	95.54	83.47	71.05	31.50	41.79
CWGAN-DNN	95.07	83.49	70.92	32.50	39.07
CWGAN-CSSAE	96.17	84.80	75.55	42.50	54.39

FIGURE 16. Comparison of detection accuracy of different classification models on the KDDTest$^+$ dataset (%).



	Normal	DoS	Probe	U2R	R2L
CWGAN-ELM	82.67	68.89	65.74	21.50	28.29
CWGAN-SAE	82.16	70.47	67.82	27.00	35.88
CWGAN-SDAE	83.97	72.27	68.44	28.50	37.15
CWGAN-DNN	85.13	75.22	69.11	27.50	36.60
CWGAN-CSSAE	84.94	74.85	70.52	42.50	52.51

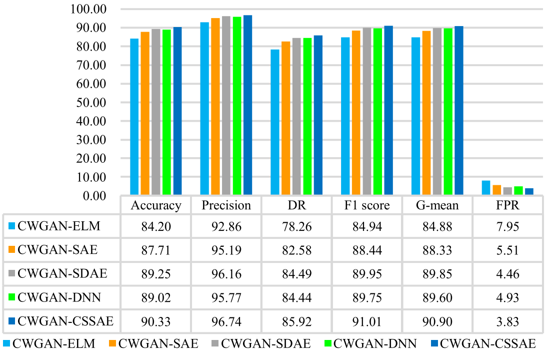
FIGURE 17. Comparison of detection accuracy of different classification models on the KDDTest-21 dataset (%).

an over-sampling of the original data, while SMOTE-CSSAE and ADASYN-CSSAE are based on the k-nearest neighbor principle to randomly synthesize the original data, which means that it lacks to learn the deep nature of the original data. CWGAN-CSSAE belongs to the category of deep learning. It can obtain the potential distribution of the original data. The detection performance of the proposed CWGAN-CSSAE is better than that of WGAN-CSSAE. Because CWGAN-CSSAE introduces a gradient penalty item, it solves the problems of model collapse and training instability during training. At the same time, the addition of L2 distance loss effectively alleviates overfitting, enabling CWGAN-CSSAE to generate attacks with better quality and higher similarity.

4) COMPARISON OF CLASSIFICATION ALGORITHMS

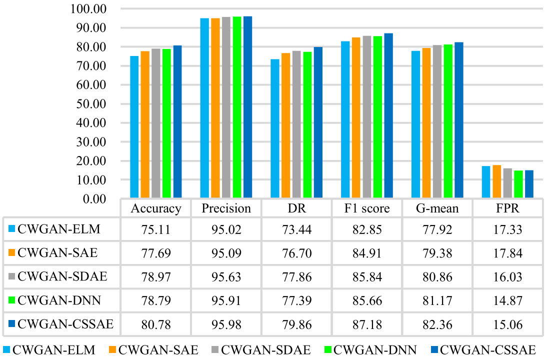
In order to improve the detection accuracy of minority attacks, at the algorithm level, this article introduces the cost-sensitive loss function into SAE to construct the CWGAN-CSSAE model. In order to verify that the proposed CSSAE classifier can better classify imbalanced datasets and improve the detection rate of minority attacks, comparative experiments are designed. According to ELM (Extreme Learning Machine) [39], SAE (Stacked Autoencoder) [40], SDAE (Stacked Denoising Autoencoder) [41], DNN (Deep Neural Networks), four classification models have been established: CWGAN-ELM, CWGAN-SAE, CWGAN-SDAE, CWGAN-DNN. The comparative experimental results are shown in Figures 16 to 20.

The data in Figure 16 shows that the detection accuracy of CWGAN-CSSAE on the KDDTest$^+$ for the five types of attacks is higher than that of CWGAN-ELM, CWGAN-SAE, CWGAN-SDAE and CWGAN-DNN. Figure 17 shows that CWGAN-CSSAE's detection accuracy of Normal and DoS on the KDDTest-21 dataset is slightly lower than CWGAN-DNN, but it has a higher detection accuracy for U2R and R2L attacks. From this point of view, CSSAE has a better classification effect on the imbalanced datasets. The reason is that ELM, SAE, SDAE, and DNN do not consider the problem of



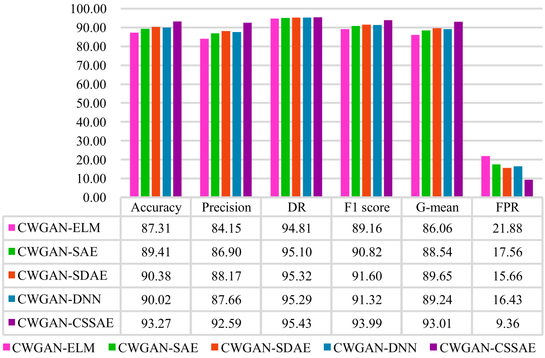
	Accuracy	Precision	DR	F1 score	G-mean	FPR
■CWGAN-ELM	84.20	92.86	78.26	84.94	84.88	7.95
■CWGAN-SAE	87.71	95.19	82.58	88.44	88.33	5.51
■CWGAN-SDAE	89.25	96.16	84.49	89.95	89.85	4.46
■CWGAN-DNN	89.02	95.77	84.44	89.75	89.60	4.93
■CWGAN-CSSAE	90.33	96.74	85.92	91.01	90.90	3.83

FIGURE 18. Comparison of detection performance of different classification models on the KDDTest$^+$ dataset (%).



	Accuracy	Precision	DR	F1 score	G-mean	FPR
■CWGAN-ELM	75.11	95.02	73.44	82.85	77.92	17.33
■CWGAN-SAE	77.69	95.09	76.70	84.91	79.38	17.84
■CWGAN-SDAE	78.97	95.63	77.86	85.84	80.86	16.03
■CWGAN-DNN	78.79	95.91	77.39	85.66	81.17	14.87
■CWGAN-CSSAE	80.78	95.98	79.86	87.18	82.36	15.06

FIGURE 19. Comparison of detection performance of different classification models on the KDDTest-21 dataset (%).



	Accuracy	Precision	DR	F1 score	G-mean	FPR
■CWGAN-ELM	87.31	84.15	94.81	89.16	86.06	21.88
■CWGAN-SAE	89.41	86.90	95.10	90.82	88.54	17.56
■CWGAN-SDAE	90.38	88.17	95.32	91.60	89.65	15.66
■CWGAN-DNN	90.02	87.66	95.29	91.32	89.24	16.43
■CWGAN-CSSAE	93.27	92.59	95.43	93.99	93.01	9.36

FIGURE 20. Comparison of detection performance of different classification models on the UNSW-NB15 dataset (%).

class imbalance. The misclassification cost of each category is the same, which is why they have a low recognition rate for the minority samples in the imbalanced dataset. CSSAE introduces a cost-sensitive loss function, which gives a larger misclassification cost to minority samples during the training process, thereby improving the detection accuracy of minority attacks.

The data in Figures 18, 19 and 20 shows that on the KDDTest$^+$, KDDTest-21 and UNSW-NB15 datasets, the accuracy, precision, DR, F1 score and G-mean of CWGAN-CSSAE are higher than other classification models. It can be seen that the CSSAE proposed from the perspective of the algorithm is very effective for the classification of the imbalanced datasets.

TABLE 8. Comparison results (%) of different detection models on the NSL-KDD (KDDTest$^+$) dataset.

Models	Acc	Precision	DR	F1	G-mean	FPR
SCDNN[21]	72.64	/	57.48	/	/	/
RBM[43]	73.23	95.09	75.30	75.30	73.48	/
GoogLeNet [17]	77.04	91.66	65.64	76.50	/	/
ResNet50[17]	79.14	91.97	69.41	79.12	/	/
RNN-IDS[19]	83.28	73.06	73.12	83.22	84.09	**3.44***
IGAN-IDS[49]	84.45	/	/	84.17	/	/
GAR-Forest [42]	85.06	87.5	85.1	/	/	12.2
TSE-IDS[18]	85.79	88.0	86.80	/	/	11.7
SWSNM[48]	86.4	/	/	87.6	/	18.5
AE[20]	88.28	91.23	87.68**	89.51	/	/
DAE[20]	88.65	96.48**	83.08	89.28	/	/
SAVAER-DNN[11]	89.36**	/	**95.98***	90.08**	/	4.70
CWGAN-CSSAE	**90.34***	**96.74***	85.92	**91.01***	**90.90***	3.83**

5) PERFORMANCE COMPARISON WITH EXISTING INSTRUSION DETECTION MODELS

To verify the superiority of the proposed CWGAN-CSSAE network intrusion detection model, the experiment in this section compares CWGAN-CSSAE with the latest models in the existing literature. The performance indicators for comparison include accuracy, Precision, DR, F1 score, G-mean and FPR. The intrusion detection models selected as experimental comparison objects include SAVAER-DNN [11], ResNet50 [17], GoogLeNet [17], TSE-IDS(Two-Stage Classifier Ensemble for IDS) [18], RNN-IDS (recurrent neural network) [19], DAE [20], AE [20], SCDNN [21], GAR-Forest [42], Gaussian–Bernoulli RBM [43], CGANs-DNN [44], GFBLS [45], LSTM$_4$ [45], DAE-DFFNN [46], DT[47], SWSNM[48], IGAN-IDS[49]. To enhance the persuasiveness of the experimental results, all intrusion detection models use the same test set. The experimental results of different network intrusion detection models on the KDDTest$^+$ test set are shown in Table 8. Table 9 shows the experimental results on the KDDTest-21 test set. The experimental results on the UNSW-NB15 test set are shown in Table 10. (* randed first, ** randed second).

According to Table 8, for the NSL-KDD (KDDTest$^+$) test set, the accuracy of the CWGAN-CSSAE network intrusion detection model reached 90.34%, the precision reached 96.74%, F1 score reached 91.01%, and G-mean reached 90.90 %, which are higher than other intrusion detection models. The FPR value is 3.83%, which is slightly higher than the RNN-IDS model in the literature [19]. Overall, the CWGAN-CSSAE network intrusion detection model has a better classification performance on the NSL-KDD (KDDTest$^+$) test set.

Table 9 shows that the accuracy of the CWGAN-CSSAE network intrusion detection model on the NSL-KDD

TABLE 9. **Comparison results (%) of different detection models on the NSL-KDD (KDDTest-21) dataset.**

Models	Acc	Precision	DR	F1	G-mean	FPR
LSTM$_4$[45]	66.74	/	/	76.21	/	/
GFBLS[45]	67.47	/	/	76.29	/	/
RNN-IDS[19]	68.55	/	/	/	/	/
TSE-IDS[18]	72.52	85.00	72.50	/	/	18.00
CGANs-DNN[44]	73.14	**97.20***	76.86	84.92	/	**12.66***
SAVAER-DNN[11]	80.30**	/	**95.19***	86.92**	/	18.22
CWGAN-CSSAE	**80.78***	95.98**	79.86**	**87.18***	**82.36***	15.06**

TABLE 10. **Comparison results (%) of different detection models on the UNSW-NB15 dataset.**

Models	Acc	Precision	DR	F1	G-mean	FPR
IGAN-IDS[49]	82.53	/	/	82.86	/	/
DT[47]	85.56	/	/	/	/	15.78
TSDL[22]	89.13	/	/	/	/	**0.75***
TSE-IDS[18]	91.27	/	91.30	/	/	8.90
DAE-DFFNN[46]	92.40	/	93.00	/	/	8.20
SAVAER-DNN[11]	93.01	/	91.94	93.54	/	5.67**
CWGAN-CSSAE	**93.27***	**92.59***	**95.43***	**93.99***	**93.01***	9.36

(KDDTest-21) test set reached 80.78%, F1 score reached 87.18%, and G-mean reached 82.36%, which are higher than other intrusion detection models. The DR value is 79.86%, the Precision is 95.98%, and the FPR is 15.06%, both ranking second. CWGAN-CSSAE has a good classification effect on the NSL-KDD (KDDTest-21) test set.

Table 10 shows that on the UNSW-NB15 dataset, the accuracy of the CWGAN-CSSAE network intrusion detection model reached 93.27%, Precision reached 92.59%, DR reached 95.43%, F1 score reached 93.99%, and G-mean reached 93.01%, which are higher than other intrusion detection models. The FPR value is 9.36%, which needs to be further improved. In general, CWGAN-CSSAE has a good classification effect on the UNSW-NB15 dataset.

Generally, the above comparative experimental results can fully prove that the proposed CWGAN-CSSAE algorithm is effective for network intrusion detection. In particular, the Accuracy, F1 score and G-mean achieved the best results, indicating that CWGAN-CSSAE can better deal with imbalanced datasets and improve the detection accuracy of minority attacks and unknown attacks.

VI. CONCLUSION

In order to improve the detection accuracy of minority attacks and unknown attacks in intrusion detection, as well as overcome the defects of imbalance of data categories, this article proposes a novel network intrusion detection model CWGAN-CSSAE from the perspective of data and algorithm. In terms of data, an improved CWGAN is proposed to generate new attack samples of specified classes, so as to reduce the imbalance of training dataset and increase the diversity of training samples. Improved CWGAN fully combines the advantages of CGAN and WGAN, and introduces gradient penalty and L2 regularization to enhance the stability of network training. In terms of algorithm, the CSSAE network is proposed, which sets different misclassification costs for different types of samples based on the number of various types of samples, in order to improve the model's detection accuracy of minority attacks. The detection performance of CWGAN-CSSAE was evaluated on the benchmark NSL-KDD and UNSW-NB15 datasets, which achieved the expected results. The experimental results show that the proposed CWGAN-CSSAE not only has a good detection effect on the minority types of attacks in the imbalanced dataset, but also can detect unknown types of attacks well.

Further improving the data enhancement algorithm and improving the detection performance of the model are problems that need to be explored and solved in future work. In addition, the current standard intrusion detection dataset is obtained through manual processing in advance. In future research, we will try to directly use the original network traffic as the training dataset to improve the applicability of the network intrusion detection model.

REFERENCES

[1] H. G. Zhang and Y. Mu, "Cyberspace security," *China Commun.*, vol. 13, no. 11, pp. 68–69, Nov. 2016.

[2] *Energy Giant EDP Hit With RagnarLocker Ransomware*. Accessed: Apr. 15, 2020. [Online]. Available: https://www.techradar.com/news/energy-giant-edp-hit-with-ragnarlocker-ransomware

[3] *Honda Hit By Possible Ransomware Attack*. Accessed: Jun. 8, 2020. [Online]. Available: https://adamlevin.com/2020/06/08/honda-hit-by-possible-ransomware-attack/

[4] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.

[5] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, Oct. 2017.

[6] J. Lee and K. Park, "GAN-based imbalanced data intrusion detection system," *Pers. Ubiquitous Comput.*, 2019, doi: 10.1007/s00779-019-01332-y.

[7] S. N. Mighan and M. Kahani, "A novel scalable intrusion detection system based on deep learning," *Int. J. Inf. Secur.*, 2020, doi: 10.1007/s10207-020-00508-5.

[8] Q. Tian, D. Han, K.-C. Li, X. Liu, L. Duan, and A. Castiglione, "An intrusion detection approach based on improved deep belief network," *Appl. Intell.*, vol. 50, pp. 3162–3178, May 2020, doi: 10.1007/s10489-020-01694-4.

[9] E. Hodo, X. Bellekens, A. Hamilton, P. L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," *Tetrahedron Lett.*, vol. 42, no. 39, pp. 6865–6867, May 2018.

[10] Y. D. Zhang, S. Y. Chen, Y. H. Peng, and J. Yang, "Survey of deep learning-based network intrusion detection," *J. Guangzhou Univ. (Natural Sci. Ed.)*, vol. 18, no. 3, pp. 17–26, Jun. 2019.

[11] Y. Yang, K. Zheng, B. Wu, Y. Yang, and X. Wang, "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization," *IEEE Access*, vol. 8, pp. 42169–42184, 2020, doi: 10.1109/ACCESS.2020.2977007.

[12] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 559–563, 2017.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[14] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1322–1328.

[15] G. Li, Y. Shen, P. Zhao, X. Lu, J. Liu, Y. Liu, and S. C. H. Hoid, "Detecting cyberattacks in industrial control systems using online learning algorithms," *Neurocomputing*, vol. 364, pp. 338–348, Oct. 2019. [Online]. Available: https://arxiv.org/abs/1912.03589

[16] J. Al-Sawwa and S. A. Ludwig, "Performance evaluation of a cost-sensitive differential evolution classifier using spark–imbalanced binary classification," *J. Comput. Sci.*, vol. 40, Feb. 2020, Art. no. 101065, doi: 10.1016/j.jocs.2019.101065.

[17] Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, "Intrusion detection using convolutional neural networks for representation learning," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2017, pp. 858–866.

[18] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94497–94507, 2019.

[19] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[20] R. C. Aygun and A. G. Yavuz, "Network anomaly detection with stochastically improved autoencoder based models," in *Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, New York, NY, USA, Jun. 2017, pp. 193–198.

[21] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," *Sensors*, vol. 16, no. 10, p. 1701, Oct. 2016.

[22] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: A two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[23] J. Y. Kim, S. J. Bu, and S. B. Cho, "Malware detection using deep transferred generative adversarial networks," in *Proc. 24th Int. Conf. Neural Inf. Process. (ICONIP)*, Guangzhou, China, 2017, pp. 539–547.

[24] J. Long, J.-P. Yin, E. Zhu, and W.-T. Zhao, "A novel active cost-sensitive learning method for intrusion detection," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Kunming, China, Jul. 2008, pp. 1099–1104.

[25] Y. Zhang, X. Chen, D. Guo, M. Song, Y. Teng, and X. Wang, "PCCN: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows," *IEEE Access*, vol. 7, pp. 119904–119916, 2019, doi: 10.1109/ACCESS.2019.2933165.

[26] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," *Proc. ICLR*, Red Hook, NY, USA, May 2019, pp. 1–19. [Online]. Available: https://arxiv.org/abs/1902.09843

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[28] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[29] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875

[30] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[31] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5679–5779.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[33] A. Tharwat, "Classification assessment methods," *Appl. Comput. Informat.*, vols. 10, pp. 1–13, Aug. 2020.

[34] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 dataset," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Ottawa, ON, Canada, Jul. 2009, pp. 1–6.

[35] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, 2015.

[36] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Canberra, ACT, Australia, Nov. 2015, pp. 1–6.

[37] Y. Z. Li and Y. Yuan, "Convergence analysis of two-layer neural networks with ReLU activation," in *Proc. NeurIPS*, Red Hook, NY, USA, Dec. 2017, pp. 597–607.

[38] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[39] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[40] Y. Bengio, P. Lamblin, P. Dan, and L. Hugo, "Greedy layer-wise training of deep networks," in *Proc. NeurIPS*, Vancouver, BC, Canada, 2019, pp. 153–160.

[41] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[42] N. K. Kanakarajan and K. Muniasamy, "Improving the accuracy of intrusion detection using GAR-Forest with feature selection," in *Proc. 4th Int. Conf. Frontiers Intell. Comput., Theory Appl. (FICTA)*. New Delhi, India: Springer, 2016, pp. 539–547.

[43] Y. Imamverdiyev and F. Abdullayeva, "Deep learning method for denial of service attack detection based on restricted Boltzmann machine," *Big Data*, vol. 6, no. 2, pp. 159–169, Jun. 2018.

[44] Z. L. Peng, W. Wan, T. Jing, and J. X. Wei, "Research on intrusion detection method based on CGANs," *Netinfo Secur.*, vol. 20, no. 5, pp. 47–56, May 2020.

[45] Z. Li, A. L. G. Rios, G. Xu, and L. Trajkovic, "Machine learning techniques for classifying network anomalies and intrusions," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.

[46] M. AL-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial Internet of Things based on deep learning models," *J. Inf. Secur. Appl.*, vol. 41, pp. 1–11, Aug. 2018.

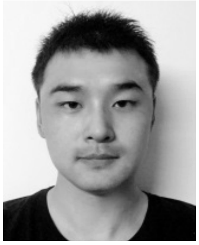
[47] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J., Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, Apr. 2016, doi: 10.1080/19393555.2015.1125974.

[48] R. A. Alshinina and K. M. Elleithy, "A highly accurate deep learning based approach for developing wireless sensor network middleware," *IEEE Access*, vol. 6, pp. 29885–29898, 2018, doi: 10.1109/ACCESS.2018.2844255.

[49] S. Huang and K. Lei, "IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks," *Ad Hoc Netw.*, vol. 105, Aug. 2020, Art. no. 102177, doi: 10.1016/j.adhoc.2020.102177.

GUOLING ZHANG was born in Weifang, Shandong, China, in 1995. He received the B.S. degree from Air Force Engineering University, in 2018, where he is currently pursuing the M.S. degree with the College of Air and Missile Defense. His research interests include pattern recognition, intelligent information processing, intrusion detection, and artificial intelligence.

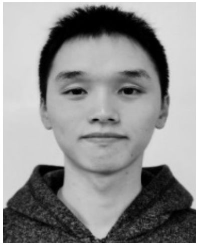
XIAODAN WANG was born in Hanzhong, Shaanxi, China, in 1967. She received the Ph.D. degree in computer science from Northwestern Polytechnical University, China. She is currently a Professor and the Ph.D. Advisor with the College of Air and Missile Defense, Air Force Engineering University. Her research interests include pattern recognition, machine learning, computer vision, and artificial intelligence. As a Professor, she has published more than 100 papers in international conferences and journals.



RUI LI was born in Xinzhou, Shanxi, China, in 1992. He received the B.S. and M.S. degrees from Air Force Engineering University, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the College of Air and Missile Defense. His research interests include pattern recognition, machine learning, and artificial intelligence.

YAFEI SONG (Member, IEEE) was born in Henan, China, in 1988. He received the Ph.D. degree from Air Force Engineering University, in 2015. He is currently a Lecturer with the College of Air and Missile Defense, Air Force Engineering University. He has been working as a Postdoctoral Researcher at Air Force Engineering University, since April 2017. His research interests include pattern recognition, intelligent information processing, and evidential reasoning.

JIAXING HE was born in Xi'an, Shaanxi, China, in 1997. He is currently pursuing the M.S. degree with the College of Air and Missile Defense, Air Force Engineering University. His research interests include intrusion detection and generative adversarial networks.



JIE LAI was born in Chengdu, Sichuan, China, in 1994. He is currently pursuing the Ph.D. degree with the College of Air and Missile Defense, Air Force Engineering University. His research interests include pattern recognition, intelligent information processing, machine learning, and artificial intelligence.