Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection

Hongwei Ding [a,b], Leiyang Chen [a], Liang Dong [a], Zhongwang Fu [a], Xiaohui Cui [a,b,*]

[a] School of Cyber Science and Engineering, Wuhan University, Wuhan, China
[b] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan University, Wuhan, China

ARTICLE INFO

Article history:
Received 28 August 2021
Received in revised form 26 January 2022
Accepted 29 January 2022
Available online 4 February 2022

Keywords:
Intrusion detection
Class imbalance
K-nearest neighbor
Generative adversarial network
TACGAN

ABSTRACT

With the continuous emergence of various network attacks, it is becoming more and more important to ensure the security of the network. Intrusion detection, as one of the important technologies to ensure network security, has been widely studied. However, class imbalance leads to a challenging problem, that is, the normal data is much more than the attack data. Class imbalance will lead to the deviation of decision boundary, which makes higher value attack data classification error. In the face of imbalanced data, how to make the classification model classify more effectively is called imbalanced learning problem. In this study, we propose a tabular data sampling method to solve the imbalanced learning problem, which aims to balance the normal samples and attack samples. Firstly, for normal samples, on the premise of minimizing the loss of sample information, the K-nearest neighbor method is used for effective undersampling. Then, we design a tabular auxiliary classifier generative adversarial networks model (TACGAN) for attack sample oversampling. TACGAN model is an extension of ACGAN model. We add two loss functions in the generator to measure the information loss between real data and generated data, which makes TACGAN more suitable for the generation of tabular data. Finally, the normal data after undersampling and the attack data after oversampling are mixed to balance the data. We have carried out verification experiments on three real intrusion detection data sets. Experimental results show that the proposed method achieves excellent results in Accuracy, F1, AUC and Recall.

1. Introduction

With the rapid updating and development of network technology, network security issues have become particularly prominent. Network security has become an important factor hindering the development of network technology. As one of the important technologies to ensure network security, intrusion detection system (IDS) [1] has been given more and more attention by researchers.

As an active defense technology, intrusion detection can improve the security of the network. Previous intrusion detection methods are mainly based on traditional machine learning methods, such as random forest, support vector machine, naive bayesian and decision tree [2–4]. The data in the current network environment is more massive, complex and multidimensional than ever before. Traditional machine learning methods are usually difficult to effectively classify complex high-dimensional data. Since the theory of deep learning was put forward, as an important branch of machine learning, deep learning has attracted more and more attention of researchers. Some scholars also began to introduce deep learning methods into the field of intrusion detection. Common intrusion detection methods based on deep learning include deep autoencoder networks [5,6], convolutional neural networks [7,8] and recurrent neural networks [9,10]. These deep learning algorithms can effectively improve the performance of intrusion detection systems.

Although scholars have done extensive research on intrusion detection and made good progress, the class imbalance is still an important factor limiting the performance of intrusion detection. Imbalanced data sets represent skewed distributions, i.e., one class has fewer instances than another. The ratio of the majority class to the minority class is called the imbalance rate (IR, as shown in Eq. (1), where n_{maj} and n_{min} represent the number of majority classes and minority classes respectively). When the attack data is obviously less than the normal data, the class imbalance problem will appear [11]. Class imbalance is a common problem in the field of intrusion detection. For example, in the standard network intrusion detection data set KDDCUP99, there are 97,278 normal samples in the training set, while there are only 52 R2L attack data. The imbalance rate between normal

[*] Corresponding author at: School of Cyber Science and Engineering, Wuhan University, Wuhan, China.

E-mail addresses: hwding@whu.edu.cn (H. Ding), cly_edu@whu.edu.cn (L. Chen), dongliang0607@whu.edu.cn (L. Dong), fuzhongwangcs@whu.edu.cn (Z. Fu), xcui@whu.edu.cn (X. Cui).

https://doi.org/10.1016/j.future.2022.01.026
0167-739X/© 2022 Elsevier B.V. All rights reserved.

samples and attack samples was 1870.73. When this extreme imbalance occurs, the classifier may predict all the results as normal samples, which leads to extremely low prediction accuracy of attack data. In the real network environment, although the number of attack samples is small, the value represented is more important. Compared with the misclassification of normal samples, the misclassification of attack samples is obviously more disadvantageous.

$$IR = \frac{n_{maj}}{n_{min}} \tag{1}$$

Based on this, in order to solve the class imbalance problem in intrusion detection data, we design a new data sampling method. Before data sampling, in order to eliminate the influence of data sparsity, a feature dimension reduction method based on deep autoencoder network is proposed. Feature dimensionality reduction can not only eliminate the influence of sparse data, but also reduce the complexity of data sampling. The proposed method includes two kinds of data sampling methods, namely, undersampling of normal samples and oversampling of attack samples. First of all, we use K-nearest neighbor method to divide the data into outliers, boundary data and trusted data. Then, on the premise of minimizing the loss of normal sample information, the undersampling of samples is carried out. Specifically, we undersampled the outliers and the normal data close to the attack sample. Then, a generative adversarial network model, TACGAN, is designed to oversample attack samples. TACGAN model is an extension of ACGAN model. We add two loss functions in the generator to measure the information loss between the real data and the generated data, so that TACGAN is more suitable for the generation of tabular data. In addition, in order to eliminate the noise that may exist in the generated data, a data filtering module is designed in TACGAN.

To sum up, the main contributions of this study can be summarized as follows:

- We propose a new hybrid sampling method to solve the class imbalance problem in intrusion detection data set. On the one hand, this method can effectively undersampling normal samples. On the other hand, TACGAN can learn the distribution of attack samples to generate almost real sample data, so as to rebalance the data set.
- In this study, the deep generative model is used to replace the traditional oversampling method, so as to solve the problem that the traditional oversampling method cannot effectively learn the distribution of samples, which leads to the generation of unreal samples.
- We design a new generative adversarial network model, TACGAN, which is more suitable for the generation of tabular data.

The rest of the paper is organized as follows. The second section introduces the advanced technology of intrusion detection, and summarizes the related methods of dealing with imbalanced data. The third section explains our method in detail. In the fourth section, the comparison and verification experiments of related algorithms are carried out, and the results are analyzed and discussed. The fifth section is the conclusion part, which summarizes the methods proposed in this study and discusses the possible research directions in the future.

2. Related works

In this section, we summarize the algorithms and research work related to this study.

2.1. Intrusion detection system

Machine learning and deep learning have been widely used in intrusion detection models in the past ten years. These methods can achieve good prediction results by learning the effective features in the data. Most of the traditional machine learning methods are based on supervised learning model [2,3,12]. Liang et al. [13] Proposed an industrial network intrusion detection algorithm based on multi feature data clustering optimization model. The algorithm classifies the weighted distance and safety factor of data according to the priority threshold of each node's data attribute characteristics. Chang et al. [14] Discussed the feasibility of random forest algorithm in feature selection of important data, and combined with support vector machine to classify the effective features finally selected. Bhattacharya et al. [15] Proposed a model based on principal component analysis (PCA) and firefly algorithm to classify intrusion detection data sets. The model first uses the hybrid PCA-firefly algorithm to reduce the dimension of data, and then uses the XGBoost algorithm to classify the reduced data. After the deep learning theory was put forward, it has been widely concerned by researchers because of its good feature learning ability. Some scholars who study intrusion detection begin to introduce deep learning method into the field of intrusion detection [16–18]. Shone et al. [19] Proposed an nonsymmetric deep autoencoder (NDAE) for unsupervised feature learning, and constructed a new classification model by combining NDAE with RF classification algorithm. The model achieves good results on the standard intrusion detection data. In order to realize network intrusion detection of small sample data, Xu et al. [20] Designed a deep neural network (DNN) detection framework named FC-Net, which mainly includes two parts: feature extraction network and comparison network. FC-Net learns the feature map for classification from a pair of network traffic samples, then compares the input feature map, and finally discriminate whether the pair of samples belong to the same type. Rehman et al. [21] Proposed a combined model based on convolutional neural network (CNN) and attention-based gated recurrent unit (GRU) to detect monomer and hybrid attacks. Experimental results show that the combination of CNN and GRU effectively improves the performance of attack detection.

2.2. Imbalanced data processing methods

Imbalanced learning is a long-term challenging problem in the field of machine learning. In the application of many practical problems, the imbalance rate IR will reach a very high value. When this extreme imbalance occurs, the classifier may predict all the results as majority classes, resulting in a very low classification accuracy of minority classes. In practical application, although the number of minority classes is small, it can provide more important information and has higher value than majority classes. Therefore, the cost of classification error for minority classes is often much higher than that for majority classes. Based on this, minority classes in imbalanced data are often the focus of data mining.

2.2.1. Traditional methods

Traditional methods for dealing with imbalanced data include data level, algorithm level and ensemble learning [22–24](As shown in Fig. 1). The data level is mainly based on the oversampling and undersampling of samples, which aims to balance the samples before data classification [25,26]. The common data level methods include random oversampling, SMOTE algorithm [27] and random undersampling. To solve the imbalanced learning problem from the algorithm level, it is mainly to improve the

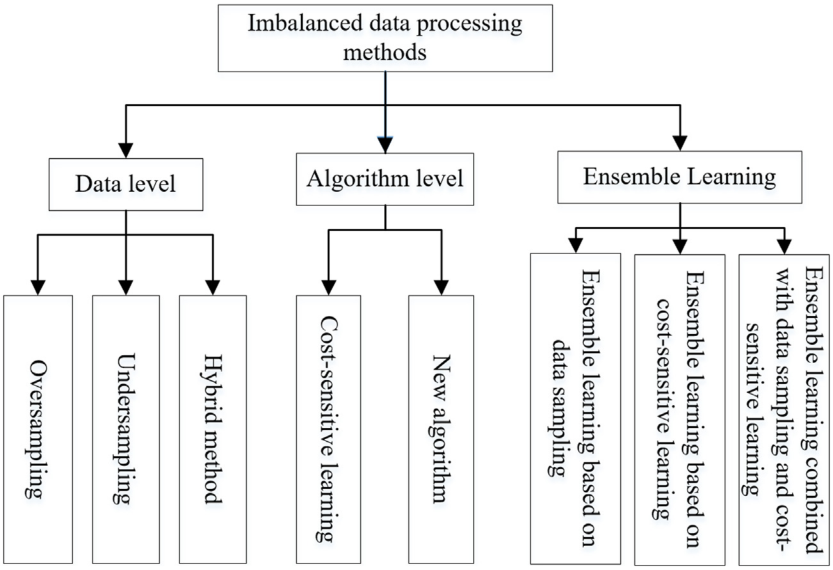

Imbalanced data processing methods
Data level
Algorithm level
Ensemble Learning
Oversampling
Undersampling
Hybrid method
Cost-sensitive learning
New algorithm
Ensemble learning based on data sampling
Ensemble learning based on cost-sensitive learning
Ensemble learning combined with data sampling and cost-sensitive learning

Fig. 1. Imbalanced data processing methods.

existing algorithm or design a new algorithm to train the imbalanced data, so as to improve the classification accuracy of the minority class [28]. The most frequently used method at the algorithm level is the cost-sensitive classification method [29]. Different from the traditional single classification model, ensemble learning integrates the basic model into a unified model by constructing different basic models. Ensemble learning can make up for the shortcomings of a single classification model, so that the classification results have better robustness [30].

In the research field of intrusion detection, class imbalance has also attracted extensive attention of researchers [31–33]. In order to solve the class imbalance problem in intrusion detection data set, oversampling and undersampling methods are introduced into intrusion detection system [34,35]. For example, aiming at the problem of imbalanced data in network intrusion detection, Jiang et al. [36] Proposed a detection framework combining hybrid sampling and deep hierarchical network. In this framework, one-side selection (OSS) algorithm and SMOTE technology are used for undersampling and oversampling, respectively, in order to balance the data set. Zhang et al. [32] Proposed a new imbalanced data set processing technique called SGM, which combines SMOTE oversampling technique and clustering based undersampling Gaussian mixture model (GMM). The intrusion detection framework based on SGM can effectively solve the class imbalance problem and improve the detection accuracy. In addition, algorithm level and ensemble learning can also be used to solve the imbalanced learning problem in intrusion detection [37–39]. For example, in order to alleviate the inconsistency between dimensionality reduction and feature retention in imbalanced data, Zhou et al. [40] Proposed a variational long short-term memory (VLSTM) learning model based on reconstructed feature representation. VLSTM model can deal with imbalanced data and high-dimensional features effectively. Bdi et al. [37] Designed an improved algorithm called I-siamidIDS to solve the imbalanced learning problem. I-siamidIDS can detect majority and minority classes at the algorithm level without using any data level balancing techniques. I-siamidIDS uses a two-layer integration mechanism. The first layer is used to identify normal and attack samples, and the second layer is used to classify attack samples. Zhou et al. [38] Developed a new integrated system based on the improved area under curve adaptive enhancement algorithm to achieve more effective detection. The system combines multiple classifiers based on M-AdaBoost-A into a whole by adopting different strategies.

Although the traditional imbalanced data processing methods have been widely studied, they have inherent defects. Random oversampling will lead to overfitting due to the duplication of samples. SMOTE oversampling method is to learn from the local neighborhood of sample points, without considering the overall distribution of minority classes. Therefore, the data generated by this method cannot effectively fit the distribution of minority classes, which makes the authenticity of the generated samples lack. Random undersampling is easy to delete more useful information, resulting in changes in the distribution of the original data. The cost-sensitive classification method is usually difficult to determine the appropriate cost factor matrix in practical application. In addition, the setting of error classification cost needs to be given by experts in related fields, so this kind of method has poor scalability. In ensemble learning, the available data of a single basic model is usually less, which is easy to cause overfitting, thus making it less practical.

2.2.2. Generative adversarial network

Generative Adversarial Network (GAN) is a framework to learn from unknown data distribution and generate similar samples. GAN is mainly composed of generator and discriminator. The generator is mainly used to generate samples whose distribution is as close to the real data as possible. The discriminator is used to receive the mixed data of the original sample and the generated sample, and discriminate the real and fake data from them. Generation model G and discrimination model D are completely independent, and their optimization process is independent alternating iterative training. For the whole network, the loss function can be written as:

$$\min_{G} \max_{D} L(D, G) = E_{x \sim P_r}[log(D(x))] + E_{x \sim P_g}[log(D(x'))] \tag{2}$$

where x represents real data, x' represents fake sample data generated by generator G. P_r and P_g represent real data distribution and generated data distribution respectively. When P_r and P_g have the same distribution, it is difficult for D to discriminate whether the samples are real or fake, that is, the probability is 0.5, and the generator can generate enough realistic samples.

GAN has been widely concerned by industry and academia because of its excellent data synthesis ability, and has been successfully applied to the fields of computer vision, natural language processing, network security and so on [41–43]. Due to the good data generation ability of GAN, it is also used to solve the problem of imbalanced learning [44,45]. For example, Xu et al. [46] Proposed a generative adversarial network model that can generate tabular data. By adding noise and KL divergence to the loss function, the discrete characteristic data is effectively generated. Engelmann et al. [47] Adopted a method based on conditional Wasserstein GAN (CWGAN), which can effectively model tabular data sets with numerical variables and category variables. Experimental results on seven credit scoring data sets show that the over sampling method based on CWGAN is better than the random over sampling and smote style methods. Chen et al. [48] Proposed an acoustic scene classification data enhancement scheme based on generative adversarial neural networks with auxiliary classifiers (ACGANs). Combined with the designed long-term scalogram extracted by wavelet scale, the framework can achieve high classification accuracy and generalization ability for test samples. Andresini et al. [49] Described a deep learning method for network traffic classification. The basic idea is to represent the network traffic as a 2D image, and use the image to train GAN and convolutional neural network. GAN is used to expand the malicious network traffic, and CNN is used as the classifier of intrusion detection model. Merino [43] and shahriar [50] use GAN to generate malicious traffic data, so as to balance normal traffic and abnormal traffic. Experimental results show that the data generated by this method is very close to the distribution of various attack data.

Although relevant scholars consider the impact of class imbalance on classification effect and apply GAN to the generation of minority class samples, they do not consider the impact of majority classes. majority class samples are the dominant classes in the data set. Only by reducing the advantages of negative classes and enhancing the separability of the overlapping regions of classes can the classification performance be further improved [51,52]. Therefore, based on the above methods cannot effectively solve the problems of class overlap and generating real minority class samples, this study proposes a combination algorithm based on KNN and generative adversarial network to solve the unbalanced learning problem in intrusion detection. KNN is used for oversampling the majority class samples in the class overlap region, while TACGAN is used to generate more real minority class samples. The final experimental results show that the proposed hybrid method has better classification effect than the traditional method and the general deep generation model.

3. Methodology

In this section, we propose an intrusion detection system framework, TACGAN-IDS, which is used to deal with imbalanced data. The framework effectively combines undersampling and oversampling methods to achieve class balance.

3.1. TACGAN- IDS framework

The framework of IDS system proposed in this paper is shown in Fig. 2, which mainly includes four parts: data set preprocessing, TACGAN model, data undersampling, and deep classification network. The specific detection process of the IDS framework is as follows.

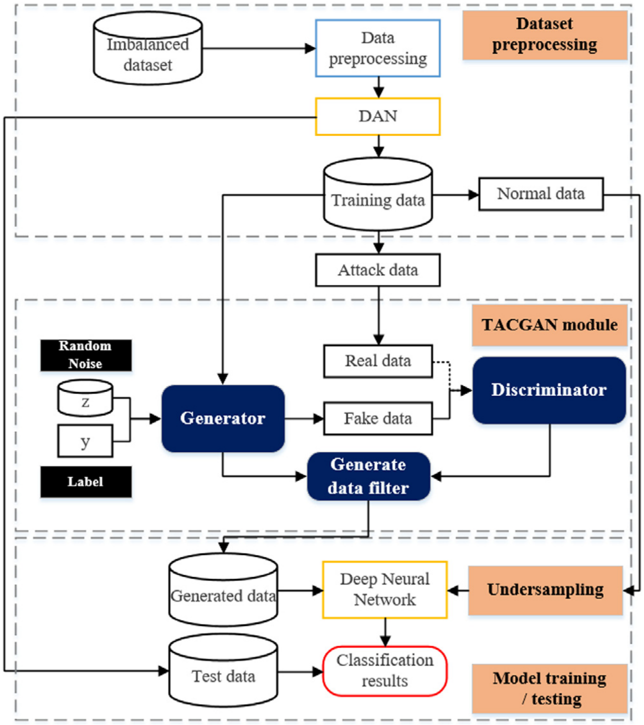


Fig. 2. TACGAN-IDS model framework.

- Data set preprocessing. The process includes three steps: data preprocessing, feature extraction and data partition. *The data preprocessing process includes:* (1) Attribute mapping, which transforms character network data features into numerical data. (2) Data normalization, due to the large difference between the data of the same attribute features, which affects the training effect, so the data should be normalized to the [0,1] interval.

  The feature extraction network adopts deep autoencoder network (DAN). In the process of data preprocessing, the data after attribute mapping has the characteristics of high-dimensional sparsity. Therefore, we use DAN model to extract effective features and remove redundant features.

  Data set division. We divide the original data set into training set and test set according to the ratio of 4:1.
- Attack data generation based on TACGAN. Compared with normal data, attack type data in network intrusion detection data set is usually difficult to obtain. Therefore, we design a TACGAN model, which is mainly used to generate attack type data. The filter is used to further filter the synthetic data generated by the generator, so that the distribution of the generated sample data is more consistent with the attack data.
- Undersampling of normal data based on K-nearest neighbor. Although the normal samples are much more than the attack samples, they also contain noise samples and redundant samples. Therefore, we design a normal sample undersampling method based on K-nearest neighbor.
- Training and testing of deep classification model. The balanced data set is used to train the deep neural network model, and then the final classification results are tested through the test set.

3.2. Data set preprocessing

3.2.1. Data preprocessing

This paper uses KDDCUP99 data set, UNSW-NB15 data set and CICIDS2017 data set as intrusion detection experimental data. KDDCUP99 data set and UNSW-NB15 data set contain character data, so they need to be preprocessed. We take KDDCUP99 data set as an example to introduce the preprocessing process.

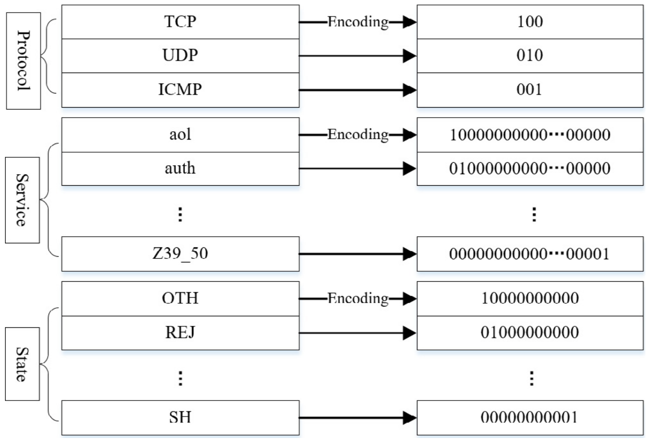


Fig. 3. Numeralization.

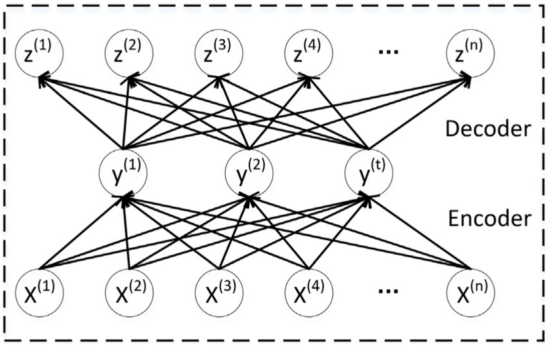


Fig. 4. Structure of AE. The structure of AE consists of encoder and decoder.

Numerization. The original KDDCUP99 data set contains 41 feature attributes, three of which are character attributes, namely Protocol_ Type, Service and Flag. Protocol_ Type includes three protocol types, Service includes 70 service types, and Flag includes 11 states. The encoding process of three types of character features is as follows (Fig. 3).

Normalization. After the numeralization processing, the character data in the data set is converted to the numerical data, but the numerical values in the numerical data are quite different, for example, the value range of the feature attribute "duration" is [0,58329]. The large difference of numerical value is easy to cause slow convergence of network and saturation of neuron output, so it is necessary to normalize the original data. In this study, the maximum minimum normalization method is used to normalize the data in the data set to [0,1] interval. The Eq. (3) is as follows:

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3}$$

Where x^* is the normalized data, x is the data to be processed, x_{min} is the minimum data value in the current attribute, and x_{max} is the maximum data value in the current attribute.

3.2.2. Feature extraction

Through data preprocessing, we can see that the dimension of intrusion detection data after one-hot encoding has become sparse high-dimensional feature data. The sparsity of data is widely considered as one of the reasons for the poor classification accuracy. In this study, we use deep autoencoder network to extract effective features from the original intrusion data.

AutoEncoder (AE) is an unsupervised learning algorithm, which does not need to use label information of data. AE consists of an encoder and a decoder. The encoder is used to extract the features of the original data, and the decoder is used to reconstruct the original data. The learning process of AE is to reduce the error between reconstructed data and input data through training, so as to learn the implicit feature representation of data.

As shown in Fig. 4, it is a traditional AutoEncoder structure. Let the original spatial data be $R^{m\times n}$, m be the number of data instances in the original space, and n be the dimension of each instance data, $x^{(i)} \in R^n, (i = 1, 2, \ldots, m)$. Each training data $x^{(i)}$ is operated by the encoder (Eq. (5)), and the feature representation $y^{(i)}$ of the hidden layer can be obtained.

$$y^{(i)} = f_\theta(x^{(i)}) = \sigma(Wx^{(i)} + b) \tag{4}$$

Where $\theta = (W, b)$ is the network parameter, W is the weight matrix from the input layer to the hidden layer, and b is the bias vector, σ Is the activation function. Then, the feature representation of the hidden layer is decoded (Eq. (6)) to obtain the reconstruction vector $z^{(i)}$.

$$z^{(i)} = g_{\theta'}(y^{(i)}) = \sigma(W'y^{(i)} + b') \tag{5}$$

where $\theta = (W', b')$ is the decoder parameter, W' is the weight matrix from the hidden layer to the output layer, usually $W' = W^T$. The model parameters can be optimized by minimizing the reconstruction error (Eq. (7)).

$$\theta^*, \theta'^* = \arg\min_{\theta,\theta'} \sum_{i=1}^{m} L(x^{(i)}, z^{(i)}) = \arg\min_{\theta,\theta'} \sum_{i=1}^{m} L(x^{(i)}, g_{\theta'}(f_\theta(x^{(i)}))) \tag{6}$$

where L is the cost function. In this study, cross-entropy loss is used as the cost function, and the expression is Eq. (8).

$$L(x^{(i)}, z^{(i)}) = -\sum_{j=1}^{n} (x_j^{(i)} \lg z_j^{(i)} + (1 - x_j^{(i)}) \lg(1 - z_j^{(i)})) \tag{7}$$

Then the cost function of the whole data set is Eq. (9).

$$J = -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} (x_j^{(i)} \lg z_j^{(i)} + (1 - x_j^{(i)}) \lg(1 - z_j^{(i)})) \tag{8}$$

In order to prevent overfitting, we add an L2 regularization weight attenuation term. is the corresponding penalty factor, which controls the attenuation degree of the weight in the penalty term. Then the improved cost function is Eq. (10).

$$J' = J + \lambda \|W\|_2 \tag{9}$$

The deep autoencoder network used in this study is a deep neural network model structure composed of multi-layer autoencoder networks. The hidden layer of the encoder is designed as a 100 × 64 fully connected dense structure, and the feature dimension of the output layer is 20. The hidden layer uses Relu as the activation function. The hidden layer of the decoder is designed as a 64 × 100 fully connected dense structure, using Relu as the activation function, and the output layer uses sigmoid function as the activation function.

3.3. Undersampling of normal data

3.3.1. Class overlap

In the study of imbalanced learning problems, the imbalanced rate between classes is not the only factor that makes model learning difficult. In fact, even when there is an imbalance between classes, some data sets can still get good classification results through appropriate algorithms (As shown in Fig. 5a).

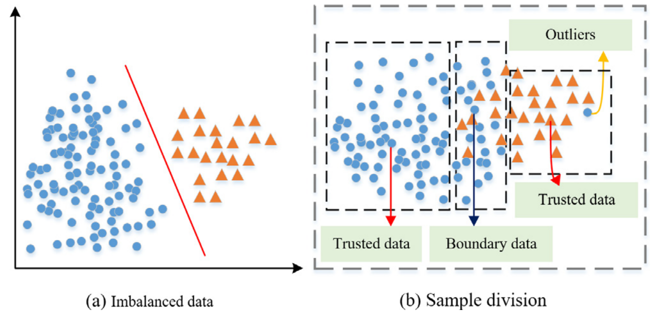


Fig. 5. Sample example. Figure (a) shows the imbalanced data set without class overlap, Figure (b) shows the imbalanced data set with class overlap, and divides the data.

However, when there is overlap between classes, even balanced data has the problem of learning difficulty. For imbalanced data, when there is overlap between classes, the classification accuracy of the positive class tends to drop sharply. Most of the traditional imbalanced learning research is only to obtain more balanced training data, and does not consider the problem of overlapping between classes.

Class overlap can be defined as: when two or more class instances share a common area in the data space, the problem of class overlap will occur. In the shared area with overlapping classes, even if the instances belong to different classes, their eigenvalue attributes are similar. Due to the similarity of attributes between features, instances of different classes in the overlapping area will cause them to become extremely complicated in the task of imbalanced data classification. An example of imbalanced data with overlapping classes is shown in Fig. 5b. In imbalanced data sets with overlapping regions, the majority class of overlapping regions is usually the dominant class. In the overlapping region, the majority class will be more frequently and clearly input to the classification model for training than the minority class. Therefore, the decision boundary of the trained classification model is usually more inclined to the majority samples, which makes the minority samples near the decision boundary more prone to misclassification.

3.3.2. Data undersampling based on KNN

K-nearest neighbor algorithm is a commonly used method in classification algorithms based on instance learning [53]. Let $X = \{x_1, x_2, \ldots, x_n\}$, the label of each sample x_i is known. For the test sample point x, in the set X, the k points nearest to it are denoted as $X' = \{x'_1, x'_2, \ldots, x'_k\}$. Then, the classification method of nearest neighbor rule is to classify point x as the class of the data with the most samples in X'.

In order to solve the overlapping problem of boundary region, we design a boundary sample sampling method based on K-nearest neighbor. In this study, based on the neighborhood features of the samples, we divide the samples in the data set into three types: Outliers, boundary data and Trusted data (as shown in Fig. 5b). The specific division method is as follows:

- Each sample is defined as the sample to be tested in turn, and other samples except the selected sample to be tested are defined as known samples;
- The distance between each sample to be tested and the known sample is calculated;
- Sort the distance between each sample to be tested and the known sample in an increasing relationship;
- Five sample points with the smallest distance are selected;
- The number of minority sample points and majority sample points was calculated;
- We record the number of similar samples in 5 samples as s. The definitions of the three types of data are as follows:
  Outliers: When s = 0, it means that the 5 samples around the sample data belong to different classes from this sample. Therefore, we mark such samples as outliers.
  Boundary data: When s = 1, 2, 3 or 4, it means that the sample contains both the same and different samples. Therefore, we divide these samples into boundary samples.
  Trusted data: When s = 5, it means that the five samples around the sample are all samples with the same attributes. Therefore, we classify this sample as trusted data.

Based on the division of the above samples, the specific process of undersampling normal data samples in this study is as follows:

Outliers undersampling: Since outliers can be regarded as noise samples, we choose to delete outliers in normal data samples (Outliers in normal samples can be regarded as noise in attack samples). According to the above definition of outliers, s represents the number of samples around a sample that belongs to the same class as it. We select the normal sample when s = 0 as the deletion object, that is, we delete the outliers in the normal sample.

Boundary data undersampling: From the above boundary sample definition, when s = 1 or 2, only 1 or 2 of the five nearest samples representing this normal class sample are of the same class. It can be judged that this sample is closer to the area where the attack sample is located. Therefore, in order to reduce the classification difficulty of attack samples in class overlapping areas, we delete such normal traffic data.

3.4. Design of TACGAN model

3.4.1. TACGAN

ACGAN is a generative adversarial network method for image synthesis, which greatly improves the performance of image generation. ACGAN combines the advantages of conditional generative adversarial network (CGAN), semi-supervised generative adversarial network (SGAN) and information maximizing generative adversarial network (infoGAN). In ACGAN framework, every generated sample has corresponding class label. The category label of generated samples is represented by one-hot encoding to distinguish different generated samples. Generator G uses noise z and class label c to generate sample $X_{fake} = G(z, c)$, and discriminator D outputs the probability of real and fake samples and the probability on class label.

$$D(X) = P(S|X), P(C|X) \tag{10}$$

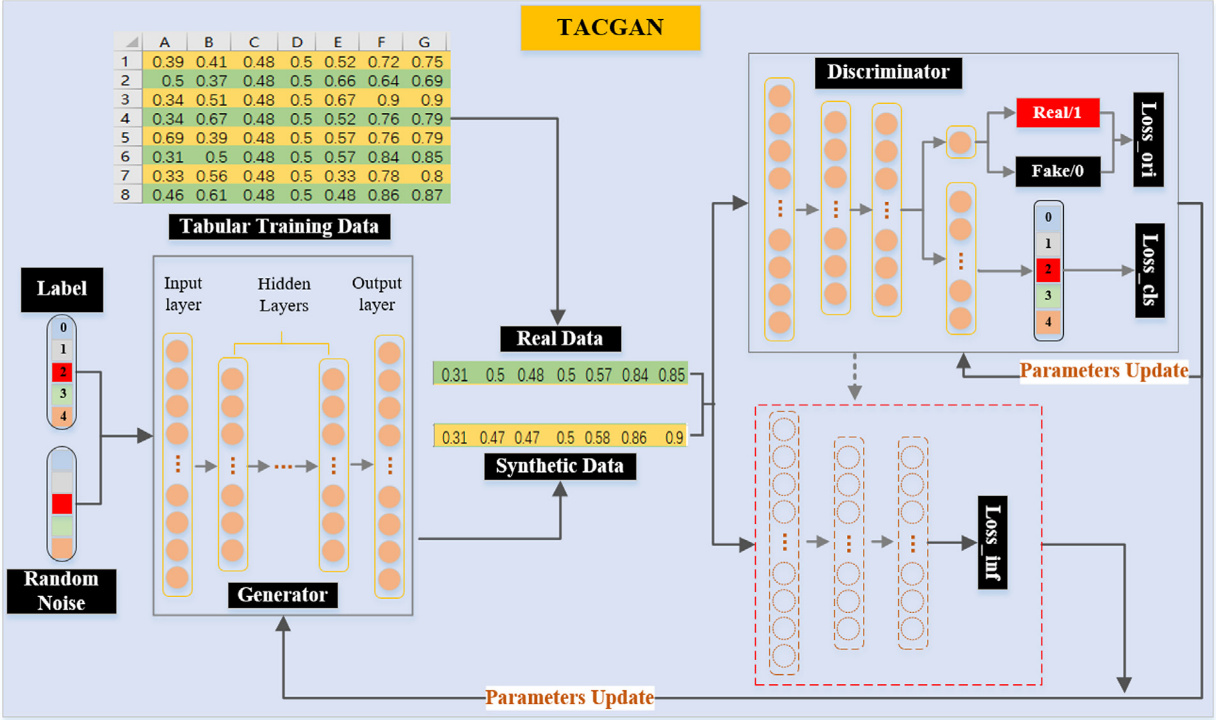
Where $P(S|X)$ is the probability of D discriminate whether the sample is real data; $P(C|X)$ denotes the probability of D discriminating the class label to which the sample belongs.

Compared with CGAN, the discriminator of ACGAN can not only discriminate the "real and fake" of samples, but also discriminate the category of samples. However, the strategy behind the ACGAN is to instead of feeding the class information to the discriminator, one can task the discriminator with reconstructing the label information. Based on this, the objective function of ACGAN discriminator is divided into two parts: log likelihood L_s of correct source and log likelihood L_c of correct class.

$$L_S = E[\log p(s = real|X_{real})] + E[\log p(s = fake|X_{fake})] \tag{11}$$

$$L_C = E[\log p(C = c|X_{real})] + E[\log p(C = c|X_{fake})] \tag{12}$$

In the training process, the training goal of discriminator D is to maximize $L_S + L_C$, and the training goal of generator G is to maximize $L_S - L_C$.



	A	B	C	D	E	F	G
1	0.39	0.41	0.48	0.5	0.52	0.72	0.75
2	0.5	0.37	0.48	0.5	0.66	0.64	0.69
3	0.34	0.51	0.48	0.5	0.67	0.9	0.9
4	0.34	0.67	0.48	0.5	0.52	0.76	0.79
5	0.69	0.39	0.48	0.5	0.57	0.76	0.79
6	0.31	0.5	0.48	0.5	0.57	0.84	0.85
7	0.33	0.56	0.48	0.5	0.33	0.78	0.8
8	0.46	0.61	0.48	0.5	0.48	0.86	0.87

Fig. 6. Model structure diagram of TACGAN.

The structure of our proposed TACGAN model is shown in Fig. 6. TACGAN is used to generate pseudo data samples similar to the real data distribution. The generation process of attack samples based on TACGAN is as follows:

- The original sample is divided into training set T_{train} and test set T_{test}. T_{train} is used for training TACGAN model and classifier (MLP), T_{test} maintains the original distribution for final effect evaluation.
- After preprocessing the data in training set T_{train}, the dimension reduced data set can be obtained through DAN. The data set after dimensionality reduction is divided into attack sample T_{attack} and normal sample T_{normal}. T_{attack} is used for oversampling of attack data and T_{normal} is used for undersampling of normal data
- The TACGAN model is trained by iterative training generator G and discriminator D:
  Discriminator D training: firstly, the random noise z is input into the generator to generate a set of pseudo sample data. Then, the pseudo sample data generated by generator G and the real attack sample are input to discriminator D at the same time, and the parameters of discriminator D are updated through error, so as to train the discriminator.
  Generator G training: when the discriminator training is completed, fix the parameters of D. after the generator generates data again, input it to the discriminator, back propagate the error to the generator, and update the generator parameters to train the generator.
  Iterative training: The generator and discriminator are trained alternately until Nash peace is reached, that is, the discrimination probability of discriminator D for pseudo samples and real samples is 0.5.
- The trained TACGAN model is used to generate attack data similar to real samples.
- Generated sample filtering: when the TACGAN training is completed, the discriminator D can better distinguish between true and false samples. Therefore, we use D to filter the inferior samples in the generated samples, so as to obtain more real attack sample data.
- Finally, we mix the expanded attack samples with normal samples to make the data set reach class balance, i.e. IR = 1.

3.4.2. Improvement of loss function

The design of ACGAN improves the quality of the generated image. In order to better generate effective tabular data, we add a new loss function for the generator, namely information loss. As shown in Fig. 6, the neural network in the red box represents the part before the classification layer in the discriminator, which is a virtual network structure. Our main purpose is to extract the data features before the classification layer. Using these features, the output layer can not only discriminate "real or fake" data, but also discriminate the class label. Therefore, these features extracted by the hidden layer contain the key feature information of the input sample. Based on this, we construct a new loss function by measuring these key features.

Information loss is to design a new regularization penalty function to force the distribution of generated data close to the original data. In order to define the information loss, we measure the information deviation between the original data and the generated data from the distance and similarity. The distance loss function is defined as follows:

$$L_{\text{dis}} = \left\| E(f_x)_{x \sim p_r(x)} - E(f_{x'})_{x' \sim p_g(x')} \right\|_2 \tag{13}$$

where f_x and $f_{x'}$ represent the high-dimensional features of the original samples and the samples generated by the generator, respectively, and $E(\cdot)$ represents the average features of all samples from a batch. The feature distance between the two types of data is measured by L-2 norm, that is, Euclidean norm. Therefore, L_{dis} means a measure of the first-order statistics between the original data and the generated data features.

In addition, we also define the similarity measure function between two kinds of data. In this study, cosine similarity is used to measure the similarity between the original data and the generated data. Cosine similarity is widely used to measure the similarity between features in few-shot learning [54], and it is proved to be an effective similarity measurement method. The specific definitions are as follows:

$$L_{sim} = \frac{E(f_x)_{x\sim p_r(x)} \cdot E(f_{x'})_{x'\sim p_r(x')}}{|E(f_x)_{x\sim p_r(x)}||E(f_{x'})_{x'\sim p_r(x')}|} \tag{14}$$

the value range of L_{sim} is (0,1). The closer the value of L_{sim} is to 1, the more similar the two types of data are. Based on this, we define information loss as:

$$L^G_{dev} = L_{dis} - L_{sim}+1 \tag{15}$$

Finally, the generator loss function in the TACGAN model is as follows:

$$L_G = L^G_{ori} + \eta L^G_{dev} \tag{16}$$

where L^G_{ori} is the loss function of the original TACGAN and η is the weight coefficient of the control information loss.

3.4.3. Generate data filter

In order to generate more effective attack sample data, we add a generation data filter in IDS (as shown in Fig. 2). Although the trained generator can generate effective minority class data, it may also generate a small number of noise data. The noise data defined here means that there is a certain deviation between the distribution of the generated data and the original data. Based on this, we use the trained discriminator D to design a filter. In TACGAN, D is a classifier used to discriminate whether the data is real or fake. When TACGAN training is completed, D will reach Nash equilibrium state, that is, the discrimination probability given for the input data is 0.5. Therefore, the discrimination probability of the generated data in D is used as the filtering condition. When the discrimination probability value deviates from the predetermined threshold, we regard the data as noise data. In order to express quantitatively, we represent the generated data as $X = \{x_1, x_2, \ldots, x_n\}$ and the output value in discriminator is $D(x_i)$. In addition, data similarity difference (SD_i) is defined as the filtering condition. The specific definition of SD_i is as follows:

$$SD_i = 2 * |0.5 - D(x_i)| \tag{17}$$

It can be seen from the definition of SD_i that the more the value of $D(x_i)$ deviates from 0.5, the greater the value of SD_i. The value range of SD_i is [0,1], that is, the larger the value is, the greater the similarity difference is.

4. Experiment

4.1. Benchmark data set

We use three data sets commonly used in intrusion detection field for experimental verification, namely KDDCUP99 [55], UNSW-NB15 [56] and CICIDS2017 [57].

KDDCUP99 data set: KDDCUP99 data set is a standard data set commonly used in intrusion detection. Each instance data in the data set contains 41 feature attributes and one label attribute. There are five types of data in the data set: Normal, DOS, Probe, R2L, U2R. Generally, the 10% KDDCUP99 data set is used as the training set, and the Corrected data set is used as the test set. The data details are shown in Table 1. For DOS data, in order to construct imbalanced data, we randomly selected 45,927 training data and 12,4287 test data from the original data set, and the data of other attack types remained unchanged.

UNSW-NB15: UNSW-NB15 data set is created by IXIA PerfectStorm tool, which contains real normal traffic and synthetic attack traffic. This data set has nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The data contains 47 feature attributes and two label attributes. Data details are shown in Table 1.

Table 1
Distribution of the data sets.

Data set	Class	Samples(Tr)	IR(Tr)	Attributes
KDDCUP99	Total	148,490	–	41
	Normal	97,278	–	
	DoS	45,927	2.12	
	Probe	4,107	23.69	
	R2L	1,126	86.39	
	U2R	52	1,870.73	
UNSW-NB15	Total	182,331	–	47
	Normal	136,999	–	
	Fuzzers	6,062	22.60	
	Analysis	677	202.36	
	Backdoor	583	234.99	
	DoS	4,089	33.50	
	Exploits	11,132	12.31	
	Generic	18,871	7.26	
	Reconnae	3,496	39.19	
	Shellcode	378	362.43	
	Worms	44	3,113.61	
CICIDS2017	Total	145,306	–	78
	Benign	105,326	–	
	Web Attack	1,476	71.36	
	DoS	21,586	4.88	
	Brute Force	5,236	20.12	
	Bot	865	121.76	
	Port Scan	10,817	9.74	

CICIDS2017: The CICIDS2017 data set contains data of normal types and common attack types, similar to the data in the real world. The data set collected five days of data, the first day only contains normal type data, the other four days contain attack type data. The implemented attacks include Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS. The training data contains 78 feature attributes and one label attribute. Data details are shown in Table 1.

4.2. Evaluation metrics

In order to quantitatively evaluate the intrusion detection system proposed in this study, we use Accuracy, Precision, F-measure, Recall and AUC as evaluation metrics. Accuracy is the ratio at which positive and negative samples are correctly classified. Recall represents the ratio of positive examples correctly classified to samples divided into positive examples. Precision is defined as the ratio at which positive examples are predicted correctly. F-measure is a relatively comprehensive evaluation index, which is the weighted harmonic average of Precision and Recall. ROC curves can be created by drawing TPR (true positive rate) on the Y axis and FPR (false positive rate) on the X axis. AUC (area under curve) is the area under the ROC curve. As a numerical value, AUC can intuitively evaluate the quality of the classifier. The larger the value, the better the classification performance.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \tag{18}$$

$$Precision = \frac{TP}{TP+FP} \tag{19}$$

$$F\text{-}measure = \frac{(1+\beta^2)\,\mathrm{Pr}\,ecision \times \mathrm{Recall}}{\beta^2(\mathrm{Pr}\,ecision \times \mathrm{Recall})} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{20}$$

$$Recall = TPR = \frac{TP}{TP+FN} \tag{21}$$

$$FPR = \frac{FP}{FP + TN} \tag{22}$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively. β is a coefficient describing the relative importance of precision and recall, and is usually set to 1. The standard is determined based on the harmonic mean between precision and recall. In this experiment, we also set β to 1, that is, our final evaluation index is F1.

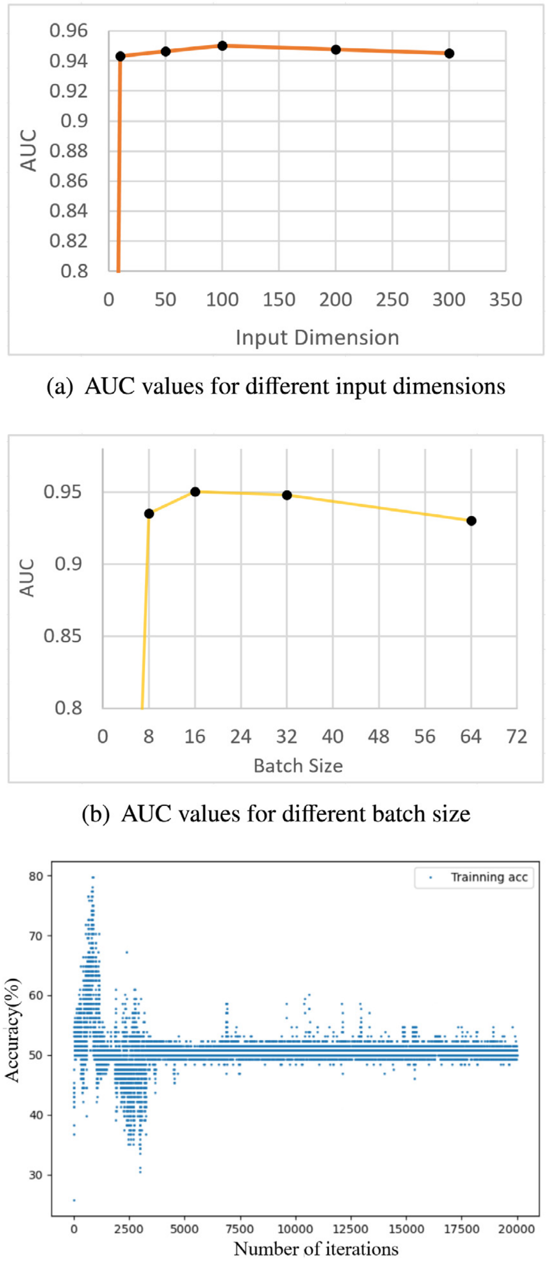
4.3. Experiment procedure

This study uses Keras and TensorFlow framework to achieve model construction and experimental testing. The parameter setting and processing process of each module are as follows.

As described in Section 3.1, the original network feature data is preprocessed first, and then enters the feature extraction module. KDDCUP99 and UNSW-NB15 data sets need one-hot encoding for character data, which makes the data set sparse. Therefore, we use DAN for feature extraction to eliminate the influence of sparse data. There is no character data in the CICIDS2017 data set, and the sparseness is low. Therefore, this research did not perform feature reduction on the data. We determined the structure of DAN model by testing the final classification effect of MLP. Since the KDDCUP99 data set is preprocessed to obtain 122 dimensional data with high dimensions, the hidden layer of the DAN structure is set to 100 × 64×20 × 64×100 and the final dimension reduction result is 20 dimensions. The dimension of data in UNSW-NB15 data set is 47, so the hidden layer of DAN structure is set to 32 × 20×32, and the final dimension reduction result is 20 dimensions. The dimension reduced data is input into the TACGAN module.

In KNN based undersampling, we set the K value to 5. Several experiments show that the number of neighborhoods with k = 5 is a simple and effective method, and is sufficient to analyze the distribution of three types of samples in the data set [51]. Based on this, we take k = 5 as the neighborhood value for dividing three types of samples.

The setting of super parameters in TACGAN is completed by the combination of random grid search and manual selection. Specifically, we determine the super parameters by verifying the classification performance of MLP. The input of the generator is a class label and a set of random noise z, and z conforms to Gaussian distribution. In many experiments, it is found that the experimental results are not sensitive to the dimensional change of random noise z, so we set it as the standard value of 100 (As shown in Fig. 7a). The generator and discriminator alternate training round is 1, because no significant improvement is observed for setting higher values. In this study, we need to generate discrete tabular data, so we design the hidden layer of the generator as 128 × 256×512 fully connected dense layer. Except for the output layer, each layer in the generator uses batch standardization with momentum of 0.8. In addition, we use LeakyRelu activation function in all hidden layers and tanh activation function in output layer. The hidden layer of the discriminator is 100 × 64 fully connected dense layer. Considering the insufficient training data of attack samples, a dropout of 0.2 is added to each layer to prevent overfitting. Like the generator, all hidden layers use LeakyRelu as the activation function. The last layer of the discriminator uses sigmoid as the "real or fake" classifier and softmax as the classifier of category label. The training epochs of TACGAN is set to 20,000. Through the test, we found that the best effect can be achieved when the batch size is set to 16 (As shown in Fig. 7b). Fig. 7c shows the accuracy training curve of UNSW-NB15 data. We can see that the final discriminator D will fit to 50%, thus losing the ability to discriminate real and fake samples. It can be judged that the training of TACGAN model is completed. The final training results of the other two data sets will also get similar training curves. Finally, for the generated data filtering, we only select the data whose similarity difference SD is within the range of [0,0.1] as the final experimental data.



(a) AUC values for different input dimensions

(b) AUC values for different batch size

(c) UNSW-NB15 accuracy training curve

Fig. 7. Training results of different parameters.

4.4. Baseline methods

We compared the classification effect of ten kinds of data including the original data. The introduction of each algorithm is as follows:

- *Original:* Raw data without any processing.
- *ROS(Random Oversampling):* Without synthesizing new data, the data is expanded by randomly copying minority class

Table 2
Experimental results of binary classification. Among them, we compare nine algorithms including our method.

Methods	KDDCUP99			UNSW-NB15			CICIDS2017		
	F1	Recall	ACC	F1	Recall	ACC	F1	Recall	ACC
Original	0.9164	0.8481	0.8908	0.8875	0.8040	0.8612	0.9037	0.8922	0.9049
DAN	0.9281	0.8718	0.9048	0.9065	0.8403	0.8820	–	–	–
ROS	0.9287	0.8736	0.9054	0.8980	0.8214	0.8731	0.9160	0.9129	0.9163
SMOTE	0.9235	0.8665	0.8989	0.8980	0.8270	0.8722	0.8936	0.9176	0.8908
SMOTE+ENN	0.9291	0.8743	0.9059	0.9030	0.8329	0.8782	0.9182	0.8952	0.9204
ADASYN	0.9270	0.8719	0.9032	0.9265	0.8831	0.9047	0.9152	0.9103	0.9157
CGAN	0.9311	0.8785	0.9083	0.9125	0.8489	0.8892	0.9321	0.9185	0.9331
WGAN	0.9359	0.8862	0.9145	0.9114	0.8451	0.8882	0.9230	0.9082	0.9243
ACGAN	0.9297	0.8748	0.9066	0.9192	0.8707	0.8958	0.9334	0.9216	0.9343
MAGNETO	0.9435	0.9004	0.9239	0.9430	**0.9656**	0.9206	**0.9583**	**0.9542**	0.9585
IGAN-IDS	0.9377	0.8850	0.9171	0.9271	0.9114	0.9045	0.9548	0.9351	0.9557
Ours	**0.9522**	**0.9138**	**0.9353**	**0.9439**	0.9403	**0.9239**	0.9581	0.9479	**0.9586**

samples. Finally, the number of minority and majority classes is the same, so as to obtain a balanced data set.

- *SMOTE:* By randomly generating new samples on the connecting line between the samples of adjacent minority classes, the expanded samples are obtained.
- *SMOTE+ENN:* SMOTE+ENN is a hybrid method based on the combination of nearest neighbor rule undersampling and SMOTE.
- *ADASYN:* Adasyn method can adaptively synthesize new samples according to the distribution of positive samples. Fewer minority class samples are generated in easy classification areas, and more minority class samples are synthesized in difficult classification areas.
- *CGAN:* Additional condition information is added to the generator and discriminator of the original GAN to realize the condition generation model.
- *WGAN:* WGAN uses Wasserstein distance instead of JS divergence as the optimization objective, and introduces gradient penalty term, which can effectively solve the problems of gradient disappearance and pattern collapse in the original GAN training.
- *ACGAN:* The algorithm described in Section 3.4.1 above.
- *Ours:* The method proposed in this paper

4.5. Discussion and analysis of experimental results

4.5.1. Results of binary classification

In this part, we compare the binary classification results of normal data and attack data. Table 2 shows the comparison results between TACGAN-IDS and other methods. The results show that the performance of TACGAN-IDS on KDDCUP99, UNSW-NB15 and CICIDS2017 data sets is better than other methods. As shown in Table 2, the 12 methods include four traditional imbalanced data processing methods and five deep generative models. The other three kinds of data are raw data, DAN dimension reduction data and TACGAN-IDS processed data. For the final classifier, we use multi-layer perceptron (MLP) classification model. As a deep learning model, MLP has proved its excellent classification ability.

For the classification effect of the original data, MLP attains a considerable performance, especially on F1 (0.9164 on KDDCUP99 and 0.8875 on UNSW-NB15) and Accuracy (0.8908 on KDDCUP99 and 0.8612 on UNSW-NB15). As a deep learning dimension reduction method, deep autoencoder network can effectively reduce the data dimension and reduce the computational complexity. The DAN model designed in this paper can effectively deal with data with high sparsity. It can be seen from Table 2 that F1, Recall and Accuracy of data processed by DAN have been improved. Among them, F1 and Accuracy increased by about 1% ∼ 2%, Recall increased by 3% ∼ 4%.

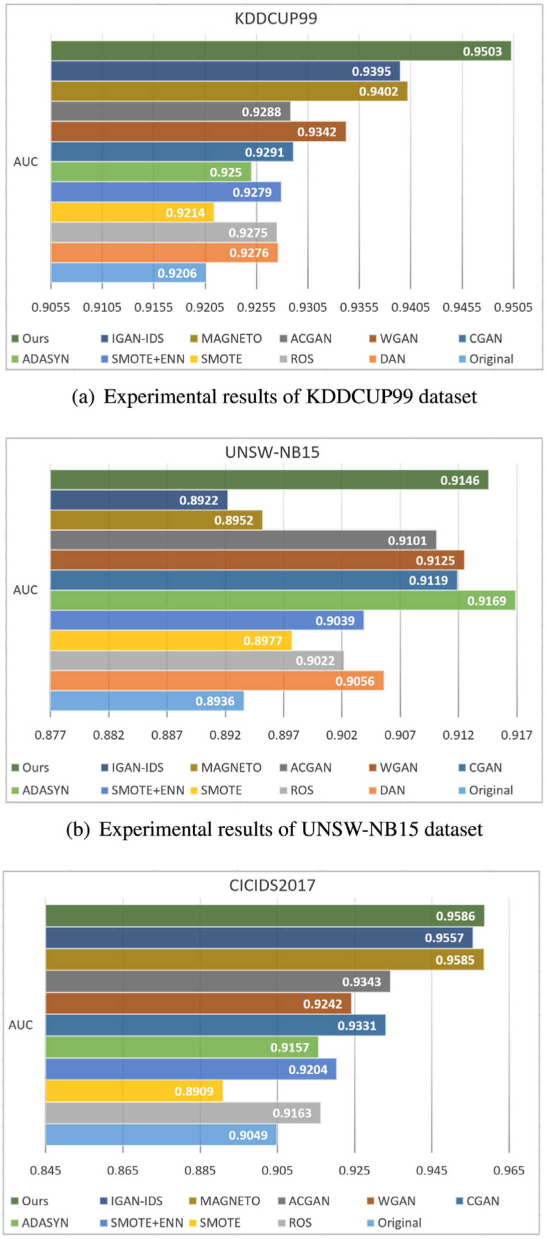
ROS, SMOTE, SMOTE+ENN and ADASYN are four common methods to deal with imbalanced data. The four methods have certain effect on the expansion of tabular data. SMOTE method can generate random samples between the connecting lines of adjacent minority samples, so as to expand the samples. SMOTE+ENN effectively undersampling the boundary data on the basis of SMOTE, so as to further improve the classification effect. ADASYN and SMOTE are similar algorithms, which can also improve the classification effect. However, the traditional oversampling methods start from the local neighborhood of the sample points, and do not consider the minority class overall distribution. Therefore, the data generated by these methods cannot effectively fit the distribution of minority classes, which makes the authenticity of the generated samples lack.

CGAN, WGAN and ACGAN are three deep generative models. Deep generative model is to learn the true distribution of minority class from the global perspective, so as to generate more realistic attack sample data. It can be seen from Table 2 that the detection results of the three deep generative models are generally better than the traditional methods. MAGNETO [49] and IGAN-IDS [58] are two effective intrusion detection methods based on GAN. As can be seen from the Table 2, compared with IGAN-IDS, our method shows excellent performance on three data sets. Compared with MAGNETO, our method has better performance on KDDCUP99 and similar performance on UNSW-NB15 and CICIDS2017 data sets. Compared with the detection results of the original unprocessed data, the performance of our proposed TACGAN-IDS method has been greatly improved. On the KDDCUP99 data set, F1, Recall and Accuracy are improved by about 4%, 7% and 5% respectively. On UNSW-NB15 data set, F1, Recall and Accuracy increased by about 6%, 14% and 6%, respectively. The detection indexes on the CICIDS2017 data set were improved by about 5%. In addition, we also give the comparison results of AUC, as shown in Fig. 8. Compared with other methods, the AUC value of TACGAN-IDS model is also improved effectively. AUC comparison results show that our method is more competitive for improving the overall classification effect of the data.

4.5.2. Results of multi-class classification

In order to make a more objective and accurate evaluation of the method proposed in this paper, we use the Precision, AUC and Accuracy to make a more objective evaluation of TACGAN-IDS method. In addition, in order to get more accurate comparison results, we use Macro-F1 and Macro Recall to compare the multi-class classification results.

Table 3 shows the comparison of experimental results of multi-class classification on KDDCUP99 data set. Compared with binary classification, multi-class classification data is more complex, and there may be class overlap between each two classes of data. It can be seen from Table 3 that ROS, SMOTE, SMOTE + ENN and ADASYN still have some improvement effect. CGAN, ACGAN, MAGNETO and IGAN-IDS are better than the four traditional methods. Due to the unlabeled learning, WGAN is easy



(a) Experimental results of KDDCUP99 dataset

(b) Experimental results of UNSW-NB15 dataset

(c) Experimental results of CICIDS2017 dataset

Fig. 8. AUC comparison results of three data sets.

to introduce noise data in the boundary area, so its Precision is lower than other methods. Finally, the Accuracy and Precision of TACGAN-IDS method have reached the best effect, and the AUC value is slightly lower than that of MAGNETO and IGAN-IDS.

Macro-F1 and Macro-Recall can evaluate the multi-class classification results more effectively. Macro-F1 needs to calculate the F1 score of each class first, and then get the F1 score of the whole sample by calculating the average value. Because the sample size of U2R and R2L attack data in the original data is very low and cannot be effectively trained, the calculated Macro-F1 value is low. The results of multi-class data classification for five kinds of data are shown in Fig. 9. The Macro-F1 and Macro-Recall of four traditional methods and five deep generative models have been improved to varying degrees. Because our method generates data more in line with the real sample distribution, so that the data set achieves class balance, so we get a better classification effect.

Table 3
Comparison of experimental results of multi-class classification on KDDCUP99 data set.

Methods	KDDCUP99		
	Precision	AUC	ACC
Original	0.8093	0.8354	0.9115
ROS	0.8201	0.8663	0.9265
SMOTE	0.8417	0.8554	0.9173
SMOTE+ENN	0.8418	0.8777	0.9280
ADASYN	0.8382	0.8695	0.9271
CGAN	0.8450	0.9007	0.9283
WGAN	0.8165	0.8629	0.9261
ACGAN	0.8526	0.8744	0.9245
MAGNETO	0.8503	0.9362	0.9234
IGAN-IDS	0.8478	**0.9403**	0.9248
Ours	**0.8556**	0.9234	**0.9297**

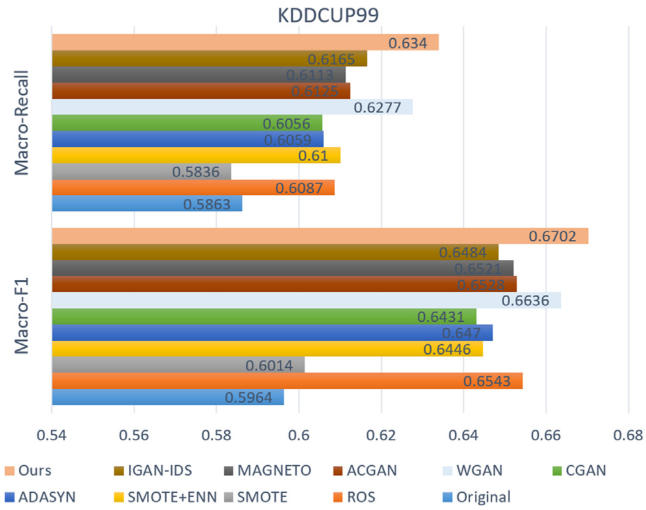


Fig. 9. Comparison of results between Macro-F1 and Macro-Recall.

Compared with the original data, Macro-F1 and Macro-Recall on TACGAN-IDS are improved by about 5% and 8% respectively.

The detection results of binary classification and multi-class classification reported above show that the performance of TACGAN-IDS is better than other competitors. We can conclude that TACGAN-IDS can effectively deal with imbalanced data by combining feature dimension reduction, undersampling and oversampling.

4.5.3. Experimental results of single class imbalanced data

In order to further demonstrate the effectiveness of TACGAN-IDS method in dealing with data with different imbalanced rates, we conducted experiments on 9 types of attack data in UNSW-NB15 data set. As can be seen from Table 4, in general, the smaller the IR value, the better the detection result, and the larger the IR value, the worse the detection effect. Among the 9 kinds of attacks, the detection effect of Generic data is the best, and F1, recall and AUC reach 0.9944, 0.9945 and 0.9952 respectively. Compared with the detection results of the original data, the proposed method has an obvious improvement effect, especially for the poorly detected attack data. For example, the F1, Recall and AUC of Fuzzers data are 0.7732, 0.9943 and 0.9034 respectively, which has been greatly improved compared with the original data. Since the sample size of worms data is extremely small, TACGAN cannot learn more effectively. Although the worms data does not get the ideal detection effect, the improvement effect is still considerable compared with the original data. In addition, in order to more clearly show the improvement effect of TACGAN-IDS on the three indicators, we compared the mean value of each

Table 4
Comparison of detection results of single attack category in UNSW-NB15.

Attack	IR	Original			Ours		
		F1	Recall	AUC	F1	Recall	AUC
Analysis	202.36	0.7771	0.7045	0.8503	**0.8317**	**0.7140**	**0.8569**
Backdoor	234.99	0.9053	0.8270	0.9135	**0.9367**	**0.8917**	**0.9445**
DoS	33.50	0.8677	0.8717	0.9337	**0.9318**	**0.9676**	**0.9719**
Exploits	12.31	0.9253	0.8661	0.9313	**0.9441**	**0.9692**	**0.9596**
Fuzzers	22.60	0.3914	0.2572	0.6193	**0.7732**	**0.9943**	**0.9034**
Generic	7.26	**0.9944**	0.9945	0.9952	0.9865	**0.9964**	**0.9897**
Reconnaissance	39.19	0.8310	0.7161	0.8573	**0.8553**	**0.8973**	**0.9298**
Shellcode	362.43	0.3886	0.2427	0.6212	**0.4155**	**0.8164**	**0.8868**
Worms	3113.61	0.2666	0.1538	0.5769	**0.4926**	**0.3846**	**0.6921**
*Mean	–	0.7053	0.6260	0.8110	**0.7964**	**0.8479**	**0.9039**

Notes: "Mean" is the mean value of the results of each test index.

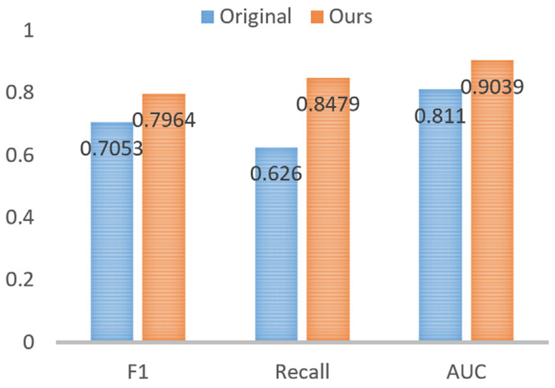
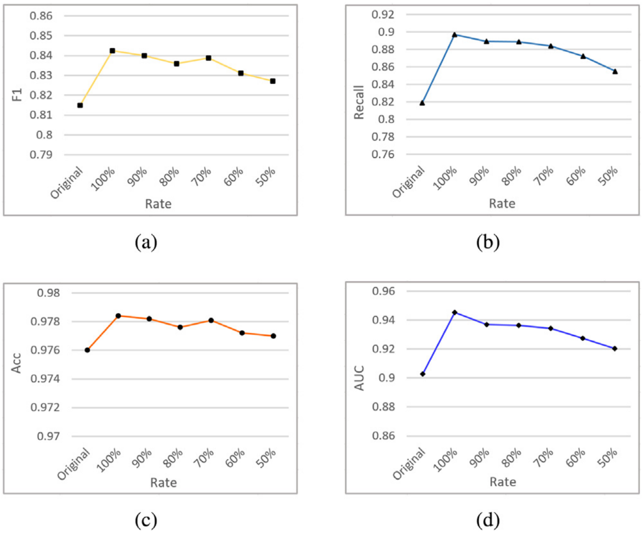


Fig. 10. Comparison of mean results.



(a) (b)

(c) (d)

Fig. 11. Comparison results under different IR. (a), (b), (c) and (d) respectively represent the comparison results of F1, recall, AUC and accuracy, where the abscissa is the random sampling ratio and the ordinate is the test results of each index.

indicator. As shown in Fig. 10, compared with the original data, F1, Recall and AUC of TACGAN-IDS increased by about 9%, 22% and 9% respectively.

4.5.4. TACGAN-IDS with different imbalance rates

In this section, we use different imbalance rates between majority and minority classes to evaluate the sample sampling performance of TACGAN-IDS. Different imbalance rates are obtained by random undersampling of minority classes. We take the Probe attack samples and normal samples in KDDCUP99 data set as examples for verification experiments. There were 4,107 samples in the original Probe data, and only 2054 samples were retained after 50% undersampling. The imbalance rate of Probe samples increased from 23.69 to 47.36. We only changed the imbalance rate of the training set, and the test set remained intact.

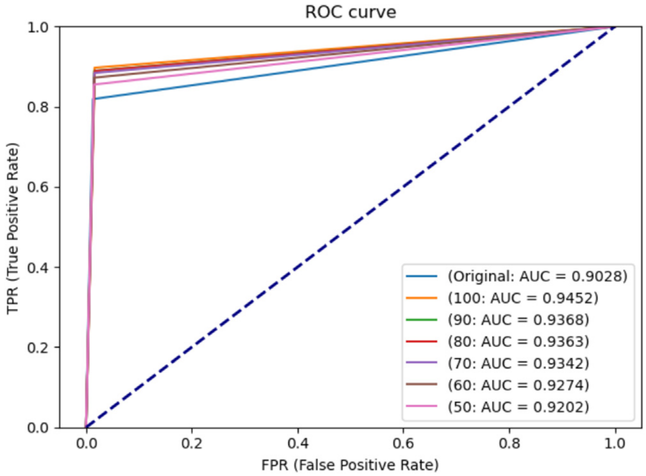


Fig. 12. ROC curves under different IR, where Original represents the original data, and 100 and other values represent the value of sampling rate.

Fig. 11 shows the detection results of TACGAN-IDS with different IR, where Original represents the experimental results of unexpanded original data. It can be seen from the four detection indexes that the detection performance of TACGAN-IDS is relatively stable under different imbalance rates. Because TACGAN can generate minority class samples, the detection system can reduce the impact of the increase of imbalance rate. However, with the increase of imbalance rate, the four detection indexes decreased slightly. As our expectations, when only a part of samples are used for TACGAN training, the complete distribution of Probe data cannot be learned effectively. Therefore, all indicators have a certain downward trend. It is worth noting that from the four detection indexes and ROC curve (Fig. 12), TACGAN-IDS still shows good performance even when the non-equilibrium rate becomes twice the original. In general, TACGAN-IDS still has high robustness and good detection performance when the class imbalance is serious.

4.5.5. Ablation study

In order to make a more thorough analysis of the TACGAN model, we conducted an ablation study to analyze the effectiveness of each module. Taking UNSW-NB15 as the experimental data, the details of ablation study are as follows.

Table 5
Results of the ablation study.

Methods	Module				F1	Recall	AUC	ACC
	DAN	US	TACGAN	DF				
(1) MLP	–	–	–	–	0.8875	0.8040	0.8936	0.8612
(2) DAN(O)	√	–	–	–	0.9065	0.8403	0.9056	0.8820
(3) TACGAN(O)	–	–	√	–	0.9230	0.9215	0.8805	0.8953
(4) DF(N)	√	√	√	–	0.9369	0.9305	0.9058	0.9147
(5) US(N)	√	–	√	√	0.9289	0.9352	0.8841	0.9026
(6) TACGAN-IDS	√	√	√	√	**0.9439**	**0.9403**	**0.9146**	**0.9239**

Notes: MLP means to classify the original data using multi-layer perceptron; DAN (O) indicates that only the deep autoencoder network is used for dimensionality reduction, and then MLP is used for classification; TACGAN (O) means that only the TACGAN module is used to expand the original data, and then MLP is used for classification; DF(N) indicates that the data filtering module in TACGAN-IDS is removed, and the other modules remain unchanged; US(N) means that the undersampling module in TACGAN-IDS is removed, and the other modules remain unchanged; TACGAN-IDS indicates that a complete system module is used.

The results of ablation experiment are shown in Table 5. Model (2) represents the experimental results using only DAN modules. Comparing model (2) with model (1), we can conclude that DAN module is helpful to improve the performance of IDS. Through the comparison between module (3) and module (1), it is proved that TACGAN can generate effective minority samples. The detection result of module (4) shows that the detection performance of IDS will be reduced when the DF module is removed from TACGAN-IDS. This is because the DF module can remove the noise in the generated data, and the experimental results also show the effectiveness of the DF module. Similarly, the experimental results after removing the undersampling module are shown in module (5). The results show that removing DF module will affect the detection performance of IDS. The US module can undersampling the majority class in the overlapping area to balance the number of majority class and minority class samples, so as to avoid the occurrence of decision boundary offset. From the above analysis, it can be seen that the modules in TACGAN-IDS can effectively improve the detection performance of the system.

5. Conclusions and future work

Intrusion detection is one of the key technologies to protect network security. However, the imbalanced learning problem will seriously affect the performance of intrusion detection system. Based on this, we propose a new intrusion detection system framework, namely TACGAN-IDS. Firstly, a feature extraction module is introduced into TACGAN-IDS to reduce the influence of sparse data on detection performance. Secondly, we design an undersampling mechanism based on K-nearest neighbor to balance the samples in the overlapping region. Finally, we design an effective TACGAN model to generate minority class data. Information loss is introduced into the generator of TACGAN model, which makes the generated minority class data more consistent with the sample distribution of the original data. In addition, at the end of the TACGAN model, a data filtering module is introduced to filter the noise data in the generated samples. The experimental results show that our method is better than the other six imbalanced data processing methods. Through experiments on attack data with different imbalanced rates, the results show that TACGAN-IDS still has a good effect on data with high IR. The research on ablation also shows the powerful ability of each module in TACGAN-IDS against imbalanced intrusion detection.

Although this work verifies the effectiveness of TACGAN-IDS for imbalanced data processing, it does not design a more effective classification model. Therefore, in the future work, we consider designing an effective deep learning classification model, combined with the method in this paper, so as to further improve the detection performance (For example, attention mechanism is added to the deep learning model to enhance the overall semantic understanding of network attributes). Deep reinforcement learning effectively combines the perception ability of deep learning with the decision-making ability of reinforcement learning, and can also be considered to be added to the research of intrusion detection. In addition, although this study considers the impact of class overlap and designs the corresponding solutions, we think it needs to be improved. Therefore, in future research, we will consider designing relevant clustering algorithms to achieve more effective under sampling.

CRediT authorship contribution statement

Hongwei Ding: Conceptualization, Methodology, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Leiyang Chen:** Conceptualization, Methodology, Writing – review & editing. **Liang Dong:** Validation, Data curation. **Zhongwang Fu:** Investigation, Validation. **Xiaohui Cui:** Conceptualization, Writing – original draft, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University. This research was supported by the National Key R&D Program of China (No.2018YFC1604000).

References

[1] C. Wu, W. Li, Enhancing intrusion detection with feature selection and neural network, Int. J. Intell. Syst. 36 (7) (2021) 3087–3105.

[2] D. Papamartzivanos, F.G. Mármol, G. Kambourakis, Dendron: Genetic trees driven rule induction for network intrusion detection systems, Futur. Gener. Comp. Syst. 79 (2018) 558–574.

[3] S. Roshan, Y. Miche, A. Akusok, A. Lendasse, Adaptive and online network intrusion detection system using clustering and extreme learning machines, J. Frankl. Inst.-Eng. Appl. Math. 355 (4) (2018) 1752–1779.

[4] J.A. Sukumar, I. Pranav, M. Neetish, J. Narayanan, Network intrusion detection using improved genetic k-means algorithm, in: 2018 International Conference on Advances In Computing, Communications and Informatics, ICACCI, IEEE, 2018, pp. 2441–2446.

[5] M. Altaha, J.-M. Lee, M. Aslam, S. Hong, An autoencoder-based network intrusion detection system for the SCADA system, J. Commun. 16 (6) (2021).

[6] P.R. Kannari, N.C. Shariff, R.L. Biradar, Network intrusion detection using sparse autoencoder with swish-PReLU activation model, J. Ambient Intell. Humaniz. Comput. (2021) 1–13.

[7] S. Zheng, Network intrusion detection model based on convolutional neural network, in: 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference, IAEAC, 5, IEEE, 2021, pp. 634–637.

[8] R.V. Mendonça, A.A. Teodoro, R.L. Rosa, M. Saadi, D.C. Melgarejo, P.H. Nardelli, D.Z. Rodríguez, Intrusion detection system based on fast hierarchical deep convolutional neural network, IEEE Access 9 (2021) 61024–61034.

[9] W. Elmasry, A. Akbulut, A.H. Zaim, Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic, Comput. Netw. 168 (2020) 107042.

[10] R. Atefinia, M. Ahmadi, Network intrusion detection using multi-architectural modular deep neural network, J. Supercomput. 77 (4) (2021) 3571–3593.

[11] V.A. Fajardo, D. Findlay, C. Jaiswal, X. Yin, R. Houmanfar, H. Xie, J. Liang, X. She, D. Emerson, On oversampling imbalanced data with deep conditional generative models, Expert Syst. Appl. 169 (2021) 114463.

[12] T. Hamed, R. Dara, S.C. Kremer, Network intrusion detection system based on recursive feature addition and bigram technique, Comput. Secur. 73 (2018) 137–155.

[13] W. Liang, K.-C. Li, J. Long, X. Kui, A.Y. Zomaya, An industrial network intrusion detection algorithm based on multifeature data clustering optimization model, IEEE Trans. Ind. Inf. 16 (3) (2019) 2063–2071.

[14] Y. Chang, W. Li, Z. Yang, Network intrusion detection based on random forest and support vector machine, in: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded And Ubiquitous Computing, Vol. 1, EUC, IEEE, 2017, pp. 635–638.

[15] S. Bhattacharya, P.K.R. Maddikunta, R. Kaluri, S. Singh, T.R. Gadekallu, M. Alazab, U. Tariq, et al., A novel PCA-firefly based xgboost classification model for intrusion detection in networks using GPU, Electronics 9 (2) (2020) 219.

[16] H. Wang, Z. Cao, B. Hong, A network intrusion detection system based on convolutional neural network, J. Intell. Fuzzy. Syst. 38 (6) (2020) 7623–7637.

[17] H. Choi, M. Kim, G. Lee, W. Kim, Unsupervised learning approach for network intrusion detection system using autoencoders, J. Supercomput. 75 (9) (2019) 5597–5621.

[18] P. Devan, N. Khare, An efficient XGBoost–DNN-based classification model for network intrusion detection system, Neural Comput. Appl. (2020) 1–16.

[19] N. Shone, T.N. Ngoc, V.D. Phai, Q. Shi, A deep learning approach to network intrusion detection, IEEE Trans. Emerg. Top. Comput. Intell. 2 (1) (2018) 41–50.

[20] C. Xu, J. Shen, X. Du, A method of few-shot network intrusion detection based on meta-learning framework, IEEE Trans. Inf. Forensic Secur. 15 (2020) 3540–3552.

[21] A. Rehman, S.U. Rehman, M. Khan, M. Alazab, T. Reddy, CANintelliIDS: detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU, IEEE Trans. Netw. Sci. Eng. (2021).

[22] T. Maciejewski, J. Stefanowski, Local neighbourhood extension of SMOTE for mining imbalanced data, in: 2011 IEEE Symposium on Computational Intelligence and Data Mining, CIDM, IEEE, 2011, pp. 104–111.

[23] L. Zhang, D. Zhang, Evolutionary cost-sensitive extreme learning machine, IEEE Trans. Neural Netw. Learn. Syst. 28 (12) (2016) 3045–3060.

[24] K.-B. Lin, W. Weng, R.K. Lai, P. Lu, Imbalance data classification algorithm based on SVM and clustering function, in: 2014 9th International Conference on Computer Science & Education, IEEE, 2014, pp. 544–548.

[25] Y. Zhu, Y. Yan, Y. Zhang, Y. Zhang, EHSO: Evolutionary hybrid sampling in overlapping scenarios for imbalanced learning, Neurocomputing 417 (2020) 333–346.

[26] A. Xw, X.A. Jian, B. Tz, A. Lj, Local distribution-based adaptive minority oversampling for imbalanced data classification - ScienceDirect, Neurocomputing 422 (2021) 200–213.

[27] D. Georgios, B. Fernando, L. Felix, Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE, Inf. Ences 465 (2018) 1–20.

[28] S. Ebenuwa, M.S. Sharif, M. Alazab, A. Al-Nemrat, Variance ranking attributes selection techniques for binary classification problem in imbalance data, IEEE Access (2019) 24649–24666.

[29] S.H. Khan, M. Hayat, M. Bennamoun, F.A. Sohel, R. Togneri, Cost-sensitive learning of deep feature representations from imbalanced data, IEEE Trans. Neural Netw. Learn. Syst. 29 (8) (2017) 3573–3587.

[30] Y. Liu, A. An, X. Huang, Boosting prediction accuracy on imbalanced datasets with svm ensembles, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2006, pp. 107–118.

[31] S. Bagui, K. Li, Resampling imbalanced data for network intrusion detection datasets, J. Big Data 8 (1) (2021) 1–41.

[32] H. Zhang, L. Huang, C.Q. Wu, Z. Li, An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset, Comput. Netw. 177 (2020) 107315.

[33] Y. Zhang, X. Chen, D. Guo, M. Song, Y. Teng, X. Wang, PCCN: parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows, IEEE Access 7 (2019) 119904–119916.

[34] X. Tan, S. Su, Z. Huang, X. Guo, Z. Zuo, X. Sun, L. Li, Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm, Sensors 19 (1) (2019) 203.

[35] A.A. Alfrhan, R.H. Alhusain, R.U. Khan, SMOTE: Class imbalance problem in intrusion detection system, in: 2020 International Conference on Computing and Information Technology, ICCIT-1441, IEEE, 2020, pp. 1–5.

[36] K. Jiang, W. Wang, A. Wang, H. Wu, Network intrusion detection combined hybrid sampling with deep hierarchical network, IEEE Access 8 (2020) 32464–32476.

[37] P. Bedi, N. Gupta, V. Jindal, I-SiamIDS: an improved siam-IDS for handling class imbalance in network-based intrusion detection systems, Appl. Intell. 51 (2) (2021) 1133–1151.

[38] Y. Zhou, T.A. Mazzuchi, S. Sarkani, M-adaboost-a based ensemble system for network intrusion detection, Expert Syst. Appl. 162 (2020) 113864.

[39] M.O. Miah, S.S. Khan, S. Shatabda, D.M. Farid, Improving detection accuracy for imbalanced network intrusion classification using cluster-based under-sampling with random forests, in: 2019 1st International Conference on Advances In Science, Engineering and Robotics Technology, ICASERT, IEEE, 2019, pp. 1–5.

[40] X. Zhou, Y. Hu, W. Liang, J. Ma, Q. Jin, Variational LSTM enhanced anomaly detection for industrial big data, IEEE Trans. Ind. Informat 17 (5) (2020) 3469–3477.

[41] W. Li, W. Ding, R. Sadasivam, X. Cui, P. Chen, His-GAN: A histogram-based GAN model to improve data generation quality, Neural Netw. 119 (2019) 31–45.

[42] W. Li, L. Fan, Z. Wang, C. Ma, X. Cui, Tackling mode collapse in multi-generator GANs with orthogonal vectors, Pattern Recognit. 110 (2021) 107646.

[43] T. Merino, M. Stillwell, M. Steele, M. Coplan, J. Patton, A. Stoyanov, L. Deng, Expansion of cyber attack data from unbalanced datasets using generative adversarial networks, in: International Conference on Software Engineering Research, Management and Applications, Springer, 2019, pp. 131–145.

[44] K. Lei, Y. Xie, S. Zhong, J. Dai, M. Yang, Y. Shen, Generative adversarial fusion network for class imbalance credit scoring, Neural Comput. Appl. 32 (12) (2020) 8451–8462.

[45] W. Li, L. Xu, Z. Liang, S. Wang, J. Cao, T.C. Lam, X. Cui, JDGAN: Enhancing generator on extremely limited data via joint distribution, Neurocomputing 431 (2021) 148–162.

[46] L. Xu, K. Veeramachaneni, Synthesizing tabular data using generative adversarial networks, 2018, arXiv preprint arXiv:1811.11264.

[47] J. Engelmann, S. Lessmann, Conditional wasserstein GAN-based oversampling of tabular data for imbalanced learning, Expert Syst. Appl. 174 (2021) 114582.

[48] H. Chen, Z. Liu, Z. Liu, P. Zhang, ACGAN-based data augmentation integrated with long-term scalogram for acoustic scene classification, 2020, arXiv e-prints.

[49] G. Andresini, A. Appice, L. De Rose, D. Malerba, GAN augmentation to deal with imbalance in imaging-based intrusion detection, Futur. Gener. Comp. Syst. 123 (2021) 108–127.

[50] M.H. Shahriar, N.I. Haque, M.A. Rahman, M. Alonso, G-ids: Generative adversarial networks assisted intrusion detection system, in: 2020 IEEE 44th Annual Computers, Software, and Applications Conference, COMPSAC, IEEE, 2020, pp. 376–385.

[51] K. Napierala, J. Stefanowski, Types of minority class examples and their influence on learning classifiers from imbalanced data, J. Intell. Inf. Syst. 46 (3) (2016) 563–597.

[52] P. Vuttipittayamongkol, E. Elyan, A. Petrovski, On the class overlap problem in imbalanced data classification, Knowl.-Based Syst. (2021) 106631.

[53] L.E. Peterson, K-nearest neighbor, Scholarpedia 4 (2) (2009) 1883.

[54] Y. Chen, X. Wang, Z. Liu, H. Xu, T. Darrell, A new meta-baseline for few-shot learning, 2020, arXiv preprint arXiv:2003.04390.

[55] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD cup 99 data set, in: 2009 IEEE Symposium on Computational Intelligence For Security and Defense Applications, 2009, pp. 1–6.

[56] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: 2015 Military Communications and Information Systems Conference, MilCIS, IEEE, 2015, pp. 1–6.

[57] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, ICISSp 1 (2018) 108–116.

[58] S. Huang, K. Lei, Igan-IDS: An imbalanced generative adversarial network towards intrusion detection system in Ad-hoc networks, Ad Hoc Netw. 105 (2020) 102177.

Hongwei Ding is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China. He current research interests include deep learning and image processing.

Leiyang Chen received the master's degrees from the School of Computer and Information Engineering, Henan University, in 2019. He is currently pursuing the Ph.D. degree With the School of Cyber Science and Engineering, Wuhan University. His research interests include edge computing, transfer learning and information security.

Liang Dong received the Master of Engineering degree from Henan Polytechnic University in 2020. He is currently pursuing the Ph.D. degree with Wuhan University, China. His research interests include pattern recognition, deep learning, and explainable AI.

Zhongwang FU received the M.S. degree from Hubei University, in 2018. He is currently pursuing the Ph.D. degree with Wuhan University, China. His research interests include big data, cluster intelligence theory, social computing.

Xiaohui Cui received the Ph.D. degree in computer science and engineering from the University of Louisville, Louisville, KY, USA, in 2004. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. His main research interests include big data, cluster intelligence theory, blockchain technology, food safety, and high-performance computing.