Research Article

PIoT Malicious Traffic Detection Method Based on GAN Sample Enhancement

Botao Hou, Ke Zhang, Xiaojun Zuo, Jianli Zhao, and Bo Xi

1State Grid Hebei Electric Power Research Institute, Shijiazhuang 050021, China
2School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050021, China

Correspondence should be addressed to Xiaojun Zuo; ids2020114037@163.com

Received 31 December 2021; Accepted 26 February 2022; Published 23 March 2022

Academic Editor: Jingyu Feng

Copyright © 2022 Botao Hou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To solve the problem of network traffic data imbalance under the background of power Internet of things and improve the poor generalization ability of the model, a PIoT malicious traffic detection method based on GAN sample enhancement is developed. Firstly, network traffic samples are preprocessed. Aiming at the imbalance of network traffic, malicious samples generation based on GAN is adopted, which uses the advantages of confrontation training in GAN to generate a small amount of malicious traffic to balance the PIoT malicious traffic. Secondly, 33 features are selected serially to construct a malicious traffic feature set by using analysis of variance and correlation analysis. Finally, the PIoT malicious traffic detection algorithm is implemented based on CatBoost and grid search. The effectiveness of the proposed method is verified on the public dataset CICIDS2017. The experimental results show that the recall rate of the proposed method on CatBoost reaches 96.60%, which is 21.16% higher than that before unbalancing, and the detection accuracy rate reaches 97.96%, which increases 8% compared to that of the other balanced methods, which significantly improves the detection performance of PIoT malicious traffic.

1. Introduction

With the popularization of the Power Internet of Things (PIoT) and the development of computer network, the PIoT has entered thousands of households. At the same time, increasing network attacks have also targeted the PIoT. In recent years, malicious network activities have become increasingly rampant, which has greatly affected the power generation, transmission, transformation, distribution, and power consumption. Malicious traffic detection as an effective means of defending against network attacks has also received widespread attention.

Compared with the traditional power grid, the current PIoT carries more privacy information. In order to realize the transportation interconnection and information coordination between various power distribution stations and users, the PIoT system has gradually broken the original network closure. Because the PIoT is based on the TCP/IP protocol, some attacks and viruses in the computer network also attack the PIoT. For example, SYN flooding attacks spread rapidly in the PIoT devices, and denial of service attacks (DDoS) can make a large number of client devices become “broilers.” Due to the endless emergence of new types of network attacks and viruses, attackers will use the huge PIoT network to spread viruses, which will also have unpredictable impacts on public electricity safety and even endanger national security.

Malicious traffic detection mainly uses the research and analysis of network traffic data to determine whether there is a problem with the PIoT’s working status and to detect unknown attack behaviors in time [1]. A large number of terminal devices of the PIoT are located at the lowest perception layer of the network. So, large amounts of network traffic data will be generated by information interaction between the PIoT management center and the network. By processing and analyzing the network traffic data, the supervision of the PIoT can be realized. In recent years, malicious traffic detection technology has been gradually put into the PIoT traffic anomaly detection work. By monitoring the network status to capture attack behaviors, some protective measures are conducted.
However, different types of malicious traffic in the PIoT present a large imbalance in the quantity distribution. Normal traffic is more than malicious traffic, especially R2L, U2R, and other small numbers of traffic attacks [2]. It is usually hidden in a large amount of normal traffic but brings a huge challenge to malicious traffic detection. How to detect malicious traffic from a large number of normal traffic is the current key of malicious traffic detection, which needs to be resolved urgently in the field.

Traditional methods use data sampling, data weighting, and one-classification to balance data. Data sampling includes undersampling, oversampling, and mixed sampling [3]. Balanced dataset is obtained by adding a small quantity of malicious samples or reducing a large quantity of normal samples. The dataset can effectively improve the detection performance of malicious samples, but it will change the original data distribution, and it is easy to cause overfitting or lose important information; the data weighting realizes the classification of imbalanced data by increasing attention to minority data without changing the data distribution, but the difficulty of this method is how to set reasonable weight. One-classification method can better handle scenarios with extremely imbalanced data, and it is often used for abnormal detection, but this method can only model for one of the categories and cannot distinguish the specific categories of various malicious traffic.

In response to the above problems, this article introduces the idea of data synthesis and combines machine learning and deep learning methods. A PIoT malicious traffic detection method based on Generative Adversarial Network (GAN) sample enhancement is proposed. First, the network traffic is preprocessed to filter out minority samples; GAN models are built separately for minority samples. By confrontation training, the generation model is trained. Second, the true distribution of original malicious samples is learned; the true samples are generated from the noise. The discriminant model is trained to identify the true samples and generative samples. The generative model and the discriminant model are trained against each other until the Nash Balance is reached. At this time, the generative samples are not much different from the true samples. The malicious samples have been enhanced to alleviate the problem of imbalanced data in the PIoT. Third, the analysis of variance and correlation analysis are applied to conduct feature selection in series to obtain features that make outstanding contributions to the detection of malicious traffic, which can solve the problem of excessive reliance on expert experience to a certain extent. Finally, the CatBoost algorithm is experimentally verified to detect the malicious traffic, and the grid search is used to optimize the model to improve the detection performance.

The contributions of this article are summarized as follows:

(1) Aiming at the problem of the imbalance of the PIoT traffic data, a malicious samples generation based on GAN method is proposed. Based on the analysis of the imbalance of traffic data, the GAN is used to learn the distribution of malicious samples by inputting improved multipeak noise. RPMSProp Gradient optimization algorithm changes the traditional GAN to prevent mode crash. Through adversarial training, the GAN can generate malicious traffic samples to balance data.

(2) Aiming at the problem that feature selection is overly dependent on expert experience, a method of serial features selection, which are, respectively, analysis of variance and correlation analysis, is designed. This method combines their respective advantages to select features and build a set of malicious traffic features.

(3) Realizing the low malicious traffic detection performance, this article performs experimental verification based on the CatBoost algorithm. To obtain the best detection performance, grid search is adopted to find appropriate parameters and achieves model optimization. And the effectiveness of the method proposed in this paper is verified simultaneously.

2. Related Work

With regard to malicious traffic research, researchers have proposed different solutions. The traditional port-based method [4] assumes that network protocols use fixed port numbers, such as HTTP protocol port 80 and e-mail protocol SMTP port 25. But many applications usually use dynamic ports to avoid intrusion detection systems and firewalls in computer. Even legal applications like Skype also use dynamic ports to overcome firewall limitations. The method is simple and easy to implement, but the overall detection efficiency is not high. Madhukar et al. [5] used port-based detection technology to reach 30% false detection rate. Due to the widespread use of SSL/TLS and other related encryption technologies in recent years, the method of deep packet inspection [6] was also restricted. Deep Packet Inspection (DPI) usually detects the payload content in packets and uses pattern matching to match the flow data character by character. This method is computationally complex. And the payload content is encrypted, so the use of the method is limited. Therefore, Yoon et al. [7] proposed a way to decrypt SSL/TLS in the corporate network environment. This way of decryption was to obtain the communication key through ARP fraud in the communication process and used the unencrypted feature of the SSL handshake phase to obtain the encryption algorithm and encryption secret. This method of decryption is easy to understand, but it is also used by attackers to intercept the communication key to decrypt sensitive information and steal the privacy of PIoT.

Machine learning and deep learning methods are also widely used. Maseer et al. [8] used 10 popular supervised learning and unsupervised learning methods to detect four types of network traffic including BENIGN, Brute Force, XSS, and SQL Injection. The types of malicious traffic that can be detected are limited, and the proportions of the four
types of network traffic are imbalanced. The author cannot solve the problem of data imbalance, which incurs that the accuracy rate on benign traffic reached 100%, and the accuracy rate of XSS and SQL Injection malicious traffic is less than 1%. Zeng et al. [9] proposed a lightweight deep learning framework DFR, which achieved the high-quality classification of traffic without involving user privacy. However, according to the statistics of the categories in the dataset in the paper, it can be seen that the data is seriously out of balance, among which benign traffic accounts for 95.32%, while malicious traffic only accounts for 4.68%. This may cause model training to have insufficient ability to characterize a small amount of malicious traffic, and it is difficult to effectively classify malicious traffic. Cheng et al. [10] used the Word2Vec model to convert the traffic load into sentence vectors and realized malicious encrypted C2&C traffic recognition through multicores one-dimensional convolution. CNN is good at capturing the local pattern features of sequence data, but Goodfellow et al. [11] found that the existing deep learning algorithm itself has defects in counteracting attacks. Doshi et al. [12] extracted traffic in the environment of the home IoT and performed feature selection to perform anomaly detection. The extracted features must be lightweight and only target the DDoS attack method in the Mirai botnet malware, which lacks generality.

In summary, traditional methods like port-based and DPI-based methods are not ideal for encrypted traffic. The decryption method weakens user privacy and needs a large amount of calculation, which makes its applicability limited. Although machine learning and deep learning methods are widely used, most of the data imbalance processing is not performed in the detection process, and the detection results focus on benign traffic samples. Considering the above issues, this paper comes up a PIoT malicious traffic detection method based on GAN sample enhancement. This method uses the advantages of counterattack to generate data to balance the PIoT traffic data, and then the generative data and true data are input to the serial feature selection module, which uses the serial analysis of variance and correlation analysis to capture the malicious traffic features, regardless of whether the features are lightweight. Lastly, the CatBoost algorithm is applied to accurately classify a variety of malicious traffic and reduces the risk of overfitting by optimizing the model.

The remainder of this paper is arranged as follows. Section 3 discusses the proposed malicious traffic generation based on GAN and malicious traffic detection based on CatBoost technique. Section 4 gives the system environment and parameters configuration and compares them with other methods. Section 5 describes the conclusion and next work.

3. PIoT Malicious Traffic Detection Based on GAN Sample Enhancement

3.1. PIoT Malicious Traffic Detection Framework. This paper puts forward a PIoT malicious traffic detection method based on GAN sample enhancement, which is mainly composed of four parts including the data preprocessing module, malicious sample generation based on GAN module, the serial feature selection module, and the malicious traffic detection based on CatBoost algorithm module. The overall structure is shown in Figure 1. Firstly, the data preprocessing module mainly includes data consolidation, missing value processing, and data normalization. On this basis, malicious sample generation based on GAN module is called. GAN is used to repeatedly generate data and discriminate data for the input noise. The discriminant loss, which can measure the distance between generative malicious samples and original malicious samples, is backpropagated to optimize the generation model and discriminant model parameters. The backpropagation is stopped when the discriminant loss reaches the set threshold. At this point, the generative malicious samples are consistent with the original malicious samples, which raises the number of minority samples and solves the problem of the imbalance of PIoT traffic data. Then, from the view of the divergence of features and the Pearson correlation coefficient between features and the category, variance analysis and correlation analysis are serially used to construct a malicious traffic feature set to solve the problem of excessive reliance on expert experience in feature selection. Lastly, the CatBoost classifier is trained to identify various malicious traffic in malicious traffic detection based on CatBoost algorithm module. To verify the efficiency and effectiveness of our presented method and improve the detection efficiency, there is a grid search method used to optimize the algorithm. After that, the training model is saved to use conveniently next time.

3.2. Data Preprocessing. For the purpose of reducing the noise redundancy of the original dataset and improving the multiclass detection accuracy of this method, this paper firstly merges the PIoT traffic files by row and then processes the missing value in different ways according to the degree of missing data and data type. This paper conducts experiments on CICIDS2017 dataset [13]. To prevent different magnitude from affecting the performance of the method, data normalization is implemented, which effectively avoids the impact caused by outliers and extreme values. The specific strategy is as follows:

(1) Merge 8 dataset files from the beginning of collection on Monday morning on 2017.7.3 to the end of collection on Friday afternoon on 2017.7.7.

(2) Delete features with the missing rate greater than 60%. According to experience, the feature is of little significance for malicious traffic detection when the missing rate is greater than 60%. For features with a missing rate less than the threshold 60%, use the average (quantitative attribute) or mode (qualitative attribute) of the feature to fill in the missing.

(3) Standardize all variables that are not in the same interval. Based on the mean and standard deviation of the original data, the original value $x$ is standardized to $x^*$ using $z = \text{score}$. The specific formula is as follows:
After data preprocessing, the imbalance of malicious samples in the dataset and the problem of data dimensional redundancy are also discovered when splitting the dataset by category. The details will be explained in sections 3.3 and 3.4.

### 3.3. Malicious Sample Generation Based on GAN

The CICIDS2017 dataset is a network security intrusion detection dataset presented by the Canadian Institute of Network Security in 2017, which resembles the true real-world data and comprises benign and the up-to-data common attacks including FTP-Patator, Brute Force, SSH-Patator, and DoS. The attacks are implemented from Monday morning of 2017.7.3 to Friday afternoon of 2017.7.7. The specific attack time and types are shown in Table 1.

Although the dataset provides a wide variety of attacks and increases the types of malicious traffic detection, the dataset has a serious category imbalance problem [14], in which the percentage of benign traffic and malicious traffic reaches 4:1, as shown in Figure 2. Severe data imbalance will mislead the classifier to make wrong judgments. It is more inclined to train the benign traffic samples, which account for a large proportion, and the training of a handful of malicious traffic samples is insufficient [15]. The result is that the model will always predict samples as benign traffic and cannot be recognized when identifying a minority category. Given the above situation, the common methods that solve the problem of data imbalance include data sampling, weighting, one-classification, and data synthesis. The data sampling method is easy to cause the risk of overfitting and underfitting. The difficulty of the weighting method is to set reasonable weights. One-classification method is only suitable for the case of binary classification, and it fails to detect a variety of malicious traffic. The idea of data synthesis is to use the existing feature similarity to generate more samples. This method has many successful cases, such as medical image analysis. Among them, GAN-based image generation is widely used and has achieved good results [16–18]. It can generate superresolution images and expand natural image data sets. Text generation also applies GAN. SeqGAN [19] is based on GAN’s text generation model.

This research uses the idea of data synthesis to generate small quantities of malicious traffic samples based on GAN [20] malicious traffic generation technology. For the CICIDS2017 dataset, this paper organizes the data into a two-dimensional table. Each row of the table represents a piece of traffic data, and each column represents one-dimensional flow features. The table contains n rows and m columns. Each column can be regarded as a random variable; all random variables follow a joint distribution, and any row in the table can be expressed as formula (2):

\[ r_j = [c_{j,1}, c_{j,2}, \ldots, c_{j,m}], \quad j \in \{1 \ldots n\}, \]

where \( r_j \) is an observation sample from the joint distribution. The samples from the original dataset are defined as \( s = (r, y) \), and the filtered malicious traffic samples are \( s^* = (r^*, y^*) \). This paper changes the label from \( y^* \) to \( y^*_2 \), that is, \( s^* = (r^*, y^*_2) \), which denotes the malicious sample from the true dataset. At the same time, this paper uses \( s^*_G = (G(z), y^*_2) \) to represent the generative sample, where \( y^*_2 \) is the original class label, \( y^*_1 \) is the true malicious sample label, \( y^*_2 \) is the generative malicious sample label, \( z \) is the input multiplex noise, and \( G(z) \) is a malicious sample from the generative model G. The generative model G adopts a fully connected network and takes multiplex noise \( z \) as input. Under the guidance of the discriminant model D, it learns the features of malicious traffic and fits \( z \) to the true distribution of the malicious samples as much as possible, which can generate malicious samples \( s^*_G \) similar to true malicious samples \( s^* \). The discriminant model D uses a fully connected multilayer perceptron. During the training process, its input is \( r^* \) and \( G(z) \), and its output is the function loss that denotes the probability of the current malicious traffic sample label being \( y^*_1 \) instead of \( y^*_2 \). The output feeds back to G to adjust the parameters of G for the purpose that the generative model G can generate samples that mix the false with the genuine. The two models are trained alternately and improved simultaneously until the output of the discriminant model D is \( D(G(z)) = 0.5 \); that is, the possibility that D judges \( G(z) \) whether is a true malicious sample or a generative malicious sample is 0.5. The training process of GAN is shown in Figure 3.
For the discriminant model $D$, $x$ is the input of $D$, $p_{\text{data}}(r^*)$ is the distribution of the true malicious sample $r^*$, and $p_z(z)$ is the distribution of noise $z$. The goal is to discern whether the input data is a true sample or a generative sample. When the data input into $D$ is $x = r^*$, due to $x \sim p_{\text{data}}(r^*)$, $D(x) = 1$, $E_{x \sim p_{\text{data}}(r^*)} \left[ \log D(x) \right]$ is the maximum. When the data input into $D$ is $x = G(z)$, due to $x \sim p_z(z)$, $D(G(z)) = 0$, $E_{x \sim p_z} \left[ \log (1 - D(G(z))) \right]$ is the maximum. In order to make the discriminant model better and better, define the following objective function:

$$\arg \max_D V(D, G) = E_{x \sim p_{\text{data}}(r^*)} \left[ \log D(r^*) \right] + E_{x \sim p_z} \left[ \log (1 - D(G(z))) \right].$$

(3)

It is obvious that the best discriminant model is $D^* = \arg\max_D V(G, D)$. When $D = D^*$, the $D$ is fixed, and the best $G$ is $G^*_D = \arg\min_G (\arg\max_D V(G, D))$, which is the minimum. Because $E_{x \sim p_{\text{data}}(r^*)} \left[ \log D(x) \right]$ do not include $G$, we directly ignore it. The objective function (4) is defined.
arg min \( V(D_G^*, G) = E_{x\sim p_z(z)}[\log (1 - D(G(z)))] \). \hspace{1cm} (4)

The problem can be converted to a max-min problem.

The loss function of GAN is shown in

\[
\min_G \max_D V(D, G) = E_{x\sim p_{data}}(r^*)[\log D(r^*)] + E_{x\sim p_z(z)}[\log (1 - D(G(z)))] . \hspace{1cm} (5)
\]

When \( z \) is input into GAN, \( G \) generates samples by the noise \( z \). Generative samples and true samples mix with each other; after that, these samples are input \( D \). \( D \) distinguishes between generative samples and true samples, which will give a function loss that is a probability of true samples. Then, the function loss will be fed back to \( D \) to optimize parameters in order to make \( D(G(z)) = 0.5 \) as much as possible. Similarly, it feeds back to \( G \) to optimize parameters for the purpose of letting \( p_z(z) \) fit \( p_{data}(r^*) \), which can generate fake samples \( G(z) \), which is difficult for \( D \) to judge whether the samples are from true samples or generative samples. The parameters of \( D \) and \( G \) update iteratively until the game process reaches a balance.

Due to the complexity of the distribution of true malicious traffic samples, it is not just a single-peak normal distribution. Therefore, the method of this article uses multipeak noise, which makes the synthesized data closer to the true sample distribution \( p_{data}(r^*) \), and the gradient optimization method \( RMSProp \) is used to find the optimal value, which alleviates the problem of vanishing gradient. The training process is shown in Algorithm 1. Among them, \( \theta_d \) is the parameter of the discriminant model \( D \), \( \theta_g \) is the parameter of the generative model \( G \), \( \eta \) is learning rate, \( d \) is clipping parameter, \( m \) is the batch size, \( Epochs \) is the number of iterations, and \( n_d \) is discriminant model iteration times when the generative model is iterated once.

When the discriminant model \( D \) converges to 0.5, the training ends. At this time, the model obtains the best parameter settings, and the “game process” between the generative model and the discriminant model reaches a balance, which can generate malicious traffic samples that the discriminant model cannot distinguish. A spot of malicious samples in the CICIDS2017 dataset can be extended with generative malicious samples, which alleviates the problem of data imbalance.

The malicious sample generation based on GAN proposed in this paper uses GAN to synthesize more high-quality malicious traffic samples. The synthetic samples not only retain the important features of the original sample, but also can solve the problem of classifier failure caused by data imbalance to some extent.

### 3.4. Serial Feature Selection

Selecting important and small amounts of network traffic features plays a key role in training machine learning models. Existing feature selection algorithms require certain professional domain knowledge as support, but some practical personnel often do not have this ability. How to set the traffic features reasonably will greatly affect the efficiency of the entire malicious traffic detection model. Through in-depth analysis of various types of malicious traffic and benign traffic, this paper uses the analysis of variance and the correlation analysis to perform feature selection in series. Use the divergence of features and the Pearson correlation coefficient between features and labels to select features by setting thresholds. This method can reduce data dimensional redundancy and calculation under the premise of ensuring detection performance overhead.

Analysis of variance method: Analysis of Variance [21] calculates the variance of each feature, sets the variance threshold, and performs feature screening. The smaller the variance, the closer the value of the feature, which is of little value for distinguishing benign traffic from each category of malicious traffic. The larger the variance, the more useful the feature to distinguish between benign traffic and malicious traffic. The formula of the analysis of variance method is shown in the following formula:

\[
s_j^2 = \frac{\sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2}{n}, \hspace{1cm} (6)
\]

where \( s_j^2 \) represents the variance of the \( j \)th feature, \( \bar{x}_j \) represents the mean value of the \( j \)th feature, \( x_{ij} \) represents the value of the \( j \)th feature of the \( i \)th sample, and \( n \) represents the number of samples.

Correlation analysis method: use Pearson correlation coefficient [22] to quantify the correlation between each feature and label, and the correlation between any two features. The absolute value range is \([0,1] \), 0–0.09 means that the two values are irrelevant, 0.1–0.3 means weak correlation, 0.3–0.5 means medium correlation, and 0.5–1.0 means strong correlation. The larger the value, the stronger the correlation. The positive or negative correlation coefficient means that the two values have positive correlation or negative correlation. This method calculates the correlation coefficients between 78 feature variables and labels, removes irrelevant and weakly related features, and removes strongly related features between features. The formula of the correlation analysis method is shown in the following formula:

\[
r_{x\times y} = \frac{\sum_{i=1}^{n} (x_{ip} - \bar{x}_p)(x_{iq} - \bar{x}_q)}{\sqrt{\sum_{i=1}^{n} (x_{ip} - \bar{x}_p)^2} \sqrt{\sum_{i=1}^{n} (x_{iq} - \bar{x}_q)^2}}. \hspace{1cm} (7)
\]

Among them, \( r_{x\times y} \) represents the correlation coefficient between feature \( x_p \) and feature \( x_q \), \( x_{ip} \) and \( x_{iq} \) respectively represent the value of the \( i \)th sample on feature \( x_p \) and feature \( x_q \), \( \bar{x}_p \) and \( \bar{x}_q \) represent the sample mean of the feature \( x_p \) and feature \( x_q \) respectively.

Use the variance analysis method to calculate the variance of the CICIDS2017 dataset in turn by formula (6), set the variance threshold to 0, and filter out 8 features with zero variance. These 8 features have the same value on the entire sample, which does not affect distinguishing between benign traffic and malicious traffic. Formula (7) is used to calculate the correlation between all feature variables and labels, and the correlation between each feature variable. The features that are moderately related and strongly related to the label are reserved, and the weakly related and irrelevant features are abandoned. The trade-off between feature variables is the
opposite. Finally, the 33 most representative network traffic features are screened out to constitute a malicious traffic feature set. Training machine learning model to detect benign traffic and malicious traffic refers to the malicious traffic feature set. Table 2 gives part of the features of CICIDS2017. The malicious traffic feature set solves the problem of excessive dependence on expert experience in feature selection to a certain extent.


In Section 3.2, aiming at the imbalance of the dataset, malicious sample generation based on GAN method is used to enhance the training samples. In Section 3.3, the malicious traffic feature set is constructed using the variance analysis method and the correlation analysis method. On this basis, this article inputs the malicious traffic feature set into the CatBoost [23] model for training.

CatBoost uses a symmetric tree as the base classifier, and its calculated feature importance is closely related to the training of the model. In traditional gradient boosting algorithms (such as Gradient Boosting Decision Tree, GBDT), a lot of decision trees are combined to generate a high-precision model, as shown in the following formula:

$$f(x) = \sum_{k=1}^{K} T(x; \theta_k),$$  \hspace{1cm} (8)

where $x$ is the feature vector, $T(x; \theta_k)$ represents the decision tree, $\theta_k$ is the decision tree parameter, and $K$ is the number of trees.

The training dataset $D = \{(x_i, y_i)\}_{i=1,2,..,n}$, where $x_i \in \mathbb{R}^t$ is the t-dimensional feature vector, $y_i \in \mathbb{R}$ is the label value, and $n$ is the number of samples. The goal is to minimize the expected loss function $L(y, F(x))$ through the training function $F: \mathbb{R}^t \rightarrow \mathbb{R}$. In gradient descent, when the $t^{th}$ iteration of $F^t$ is performed, the base classifier $h^t$ is obtained through the function group $H$ to minimize the expected loss. The loss of the $t^{th}$ iteration is

$$h^t = \arg \min_{h \in H} L(y, F^{t-1}(x) + h(x)).$$  \hspace{1cm} (9)

The gradient step $h^t(x)$ uses the negative gradient $-g^t(x, y)$ of the loss function to approximate

$$g^t(x, y) = \frac{\partial L(y, F^{t-1}(x))}{\partial F^{t-1}(x)}.$$  \hspace{1cm} (10)

In actual calculations, formula (11) is usually calculated approximately through the training dataset:

$$h^t = \arg \min_{h \in H} (\mathcal{E}(-g^t(x, y) - h(x)))^2.$$  \hspace{1cm} (11)

The inconsistency between the conditional distribution of $g(x, y)|x_i$ and the conditional distribution on the training set will cause prediction bias. To solve this problem, CatBoost uses ordered boosting to replace the gradient estimation method in the traditional algorithm. CatBoost will train a separate model $M_i$ for each sample $x_i$ and the model $M_i$ is obtained by training using a training set without samples $x_i$. Use $M_i$ to get the gradient estimation about the sample and use the gradient to train the base classifier to get the final model [24]. The CatBoost algorithm changes the gradient estimation method in the traditional gradient boosting algorithm and uses unbiased gradient estimation in each iteration to complete the tree building process, which slows down the prediction offset and enhances the generalization ability of the model.

4. Experiment and Analysis

This research designs a PLoT malicious traffic detection method based on GAN sample enhancement, uses the public dataset CICIDS2017 for experiment and evaluation, and verifies it on the CatBoost model. By comparing the confusion matrix and evaluation indicators between the balanced dataset and the imbalanced dataset, the effectiveness
4.1. System Environment. The experiment is based on python 3.7 programming, combined with NumPy, pandas, matplotlib, keras, TensorFlow, and other python libraries. The data loading, processing, training, and testing are all completed on the Windows 10 operating system. The computer has 8 GB of memory and a processor, which is Intel(R) Core(TM) i5-6300HQ CPU @ 2.30 GHz 2.30 GHz, and the programming software used is Jupyter Notebook.

4.2. Experimental Data Configuration. The dataset uses 70,000 traffic samples for training. The proportion of benign traffic samples to each malicious traffic sample in the training set is 1:1, all are 5000 traffic samples, and the percentage of the training set and testing set is 7:3. When dividing the dataset, it is found that the number of malicious traffic samples such as Heartbleed is less than 5000. Therefore, the malicious sample generation based on GAN algorithm is used to increase the number of malicious traffic samples including Bot, Heartbleed, 3 types of Web Attacks, and Infiltration. The GAN parameters are set. The Epoch is set to 30, the learning rate \( \eta \) is set to 0.0002, the clipping parameter \( d \) is 0.1, the batch size \( m \) is 128, and the number of iterations of the discriminant model \( n_g \) is set to 5 when the generation model is iterated once. In order to obtain the optimal parameter combination, this paper arranges and combines the possible values of each CatBoost model parameter to generate a "grid." Then, each combination is used for CatBoost training, and the performance is evaluated by cross validation. After the fitting function tries all parameter combinations, it returns an appropriate classifier and automatically adjusts to the best parameter combination through clf.best_params_ parameter value. Therefore, the CatBoost model parameters are set to iterations = 100, depth = 5, learning_rate = 0.3.

4.3. Evaluation Indicators. In order to evaluate the classification performance of malicious traffic detection method based on GAN sample enhancement, four evaluation indicators such as accuracy are defined:

\[
\begin{align*}
\text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN}, \\
\text{Recall} &= \frac{TP}{TP + FN}, \\
\text{Precision} &= \frac{TP}{TP + FP}, \\
F1 - \text{Score} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},
\end{align*}
\]

where \( TP \) represents the number of correctly identified positive samples, \( TN \) represents the number of correctly identified negative samples, \( FP \) denotes the number of wrongly identified positive samples, and \( FN \) denotes the number of wrongly identified negative samples.

4.4. Experimental Evaluation

4.4.1. Validity and Consistency Detection of Generative Malicious Samples. In order to verify the feature consistency of CICIDS2017 before and after balancing and the effectiveness of the method that malicious sample generation based on GAN, this paper compares the probability density of the features of the dataset before and after balancing, as shown in Figure 4. The article selects two features including Total Fwd Packets and Fwd Packet Length Min. Draw the kernel density estimation diagram (KDE diagram) of the original dataset and the balanced dataset for each feature, and fit the distribution of the data before and after balancing, which can reflect the consistency of the original data malicious traffic samples and the generative malicious traffic samples in each feature.

It can be seen from Figure 4 that the abscissa represents the feature value, and the ordinate represents the probability density corresponding to the value. The red curve is the original data distribution after preprocessing, and the blue curve is the data distribution after balancing. It can be seen from the figure of feature kernel density estimation that the distribution of data features before and after balancing of the dataset is basically the same. The malicious samples generated based on the GAN are the same as the original malicious samples, which verifies the consistency of the features of the method before and after use.

To verify the effectiveness of malicious sample generation based on GAN, the original malicious samples and the generative malicious samples are mixed as a balanced dataset. Data preprocessing and serial feature selection on the original dataset are performed, and a subset of malicious samples containing 33 features is obtained. Then, the article divides the dataset according to 7:3 and inputs the training set into the CatBoost model for training. Finally, a confusion matrix and four evaluation indicators are used to evaluate the effectiveness of the method. As shown in Figure 5, the ordinate represents the true category of the sample, and the abscissa represents the category predicted by the CatBoost model. Figure 5(a) is the confusion matrix obtained from the original dataset. From Figure 5(a), it can be seen that a large amount of malicious traffic is misclassified as benign traffic, including 61% of Bot malicious traffic, 85% of Web Attract–Brute Force malicious traffic, and 97% of Web Attract–XSS malicious traffic. By analyzing the number of above malicious samples, it is found that their number is 1966, 1507, and 652, respectively, which is significantly different from the number of benign traffic samples of 2273097. Due to the lack of sufficient malicious traffic data, CatBoost does not learn enough malicious traffic during training, and it is difficult to effectively deal with this sparse data. The result is that a variety of malicious traffic is misclassified as benign traffic. Figure 5(b) is a confusion matrix obtained from the balanced dataset. Except for a small part of Web Attack-Brute Force and Web Attack-XSS...
malicious traffic, they are misclassified due to the two types of malicious traffic belonging to the Web Attack malicious category, which has similar traffic behaviors and similar characteristics. It can be seen that each category can be classified correctly, which verifies the effectiveness of malicious sample generation based on GAN.

Table 3 shows the performance of the CatBoost algorithm before and after balancing the dataset. Through comparison, it can be seen that although the accuracy of model detection is slightly reduced after balancing, the recall, precision, and F1-score have been greatly improved. Malicious traffic Detection requires a high recall rate. Once malicious traffic is misjudged as benign traffic; the traffic may cause irreversible losses to the system. The balanced dataset makes up for the defect of low recall rate. The malicious traffic in the network traffic is detected to avoid potential risks, and the effectiveness of the GAN-based malicious sample generation method proposed in this paper is verified again, the problem of data imbalance is alleviated, and the efficiency of malicious traffic detection is improved.

4.4.2. Malicious Traffic Detection Algorithm Implementation. After dividing the dataset, the article uses the CatBoost algorithm on the training set to train the data and uses grid search for parameter adjustment. The parameter task_type is set to “GPU” to improve the training efficiency of the model. The parameters including learning_rate, depth, and iterations are adjusted separately and finally updated depth = 5, iterations = 100, and learning_rate = 0.3, and the best training model is obtained at this time. Figure 6 shows the impact of different learning rates on score. The abscissa represents the learning rate, the ordinate represents the score, the shaded part represents the fluctuation range of the

![Figure 4: KDE diagram of two features.](a) ![Figure 4: KDE diagram of two features.](b)

![Figure 5: Comparison of the confusion matrix before and after balancing.](a) ![Figure 5: Comparison of the confusion matrix before and after balancing.](b)
current score value, and the red line and green line represent the results on the training set and the test set, respectively. It is shown that the $r^2$ score value reaches the maximum value when the learning rate is 0.3. The model effect is best to reach 0.98, and the adjusted model is saved.

To verify whether the CatBoost algorithm and other machine learning methods proposed in this article have improved in various identifying malicious traffic, three algorithms of Decision Tree, SVM, and Naive Bayes are performed on the balanced CICIDS2017 dataset. The experimental results are shown in Table 4. Among the three classic machine learning algorithms, Decision Tree performs best. It is slightly lower in terms of detection accuracy, recall, precision, and F1-score compared with the CatBoost algorithm. This is because the CatBoost algorithm has obvious advantages in processing the categorical features and can effectively deal with the problems of gradient deviation and prediction offset, which can raise the performance and generalization ability of the algorithm.

### Table 2: The malicious traffic feature set.

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow duration</td>
<td>The duration of flow</td>
</tr>
<tr>
<td>Total fwd packets</td>
<td>The number of packets from client to server</td>
</tr>
<tr>
<td>Flow Bytes/s</td>
<td>Packet bytes transmitted per second</td>
</tr>
<tr>
<td>Flow IAT min</td>
<td>The minimum interval arrival time of two flow</td>
</tr>
<tr>
<td>Bwd IAT min</td>
<td>The minimum interval arrival time of flow from server to client</td>
</tr>
<tr>
<td>Bwd PSH flags</td>
<td>The number of PSH flags from server to client</td>
</tr>
<tr>
<td>Bwd URG flags</td>
<td>The number of URG flags from server to client</td>
</tr>
<tr>
<td>Fwd Packets/s</td>
<td>The number of packets from client to server per second</td>
</tr>
<tr>
<td>Bwd Packets/s</td>
<td>The number of packets from server to client per second</td>
</tr>
<tr>
<td>Total backward packets</td>
<td>The number of packets from server to client</td>
</tr>
<tr>
<td>Fwd packet length min</td>
<td>The minimum length of the packet from client to server</td>
</tr>
<tr>
<td>FIN flag count</td>
<td>The count of packets with FIN flag</td>
</tr>
<tr>
<td>ACK flag count</td>
<td>The count of packets with ACK flag</td>
</tr>
<tr>
<td>URG flag count</td>
<td>The count of packets with URG flag</td>
</tr>
<tr>
<td>SYN flag count</td>
<td>The count of packets with SYN flag</td>
</tr>
<tr>
<td>CWE flag count</td>
<td>The count of packets with CWE flag</td>
</tr>
<tr>
<td>ECE flag count</td>
<td>The count of packets with ECE flag</td>
</tr>
<tr>
<td>Down/Up ratio</td>
<td>The ratio of down/Up</td>
</tr>
<tr>
<td>Avg fwd segment size</td>
<td>The average size of packets from client to server</td>
</tr>
</tbody>
</table>

4.4.3. **Horizontal Contrast.** In order to further illustrate the effectiveness of the PlIoT malicious traffic detection method based on GAN sample enhancement, the method described in this article and the method in the literature [25, 26] are used to compare the performance on the four indicators of accuracy, recall, precision, and F1-score. The classification performance of the detection model is shown in Table 5. Chen et al. [25] used the keyword avoidance sample enhancement method to automatically segment the marked traffic according to special characters. Then, extracting and establishing a keyword database that contains malicious
traffic features is performed. The noise was randomly added to non-keywords. The noise data and the keyword database were merged to form an enhanced dataset, and then we use a fully supervised learning model based on text classification detected; Yang et al. [26] used a convolution gated recursive unit to build a keyword library for model training. Characters were used as features of text classification for malicious detection. The two methods performed imbalance processing and sample enhancement from the perspective of text classification. The comparison results with the method proposed in this paper are shown in Table 5.

The comparison results show that the method in this paper has obvious advantages in the four evaluation indicators of accuracy, recall, precision, and F1-score. The detection effect is better than that in the literature [25, 26]. The analysis shows that the text-based classification method uses a keyword library. It is difficult to cover all malicious traffic keywords and potential attack payloads to solve the imbalance problem. Although the detection performance for conventional malicious traffic is good, it is hard to detect the diversity of malicious traffic. Aiming at the imbalance of the dataset, the method in this paper first separates a small number of malicious samples, establishes a GAN model according to each type of malicious sample, learns the feature distribution of each type of malicious sample through GAN, and retains the original features of malicious traffic to the maximum extent. The method proposed can improve the model’s learning ability of malicious traffic samples. The results prove the effectiveness of the PloT malicious traffic detection method based on GAN sample enhancement proposed in this paper.

5. Conclusion

This paper proposes a PloT malicious traffic detection method based on GAN sample enhancement, studies the data imbalance in the CICIDS2017 dataset, uses GAN to generate malicious samples to balance the dataset, and combines the analysis of variance and correlation analysis to select features in series. The CatBoost model is trained on the model, and the grid search is used for model tuning. At the end of this article, it is compared with the existing machine learning algorithm Decision Tree, SVM, and Naive Bayes. The results show that the CatBoost algorithm has the best performance. In addition, compared with the current popular method of establishing a keyword database to solve the imbalance problem, the method in this paper has the best detection performance. The method improves the accuracy rate of malicious traffic detection and performance of the other three indicators, which verifies the effectiveness of the method in this paper. In future work, the amount of network traffic data is taken into consideration. Because of a large data volume, Hadoop technology is used to perform distributed processing on a large amount of network traffic data to improve the detection speed. At the same time, scripts are written to realize the automation of malicious traffic detection.

Data Availability

The data used to support this study are available at https://www.unb.ca/cic/datasets/ids-2017.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been supported by the National Key Research and Development Program of China (2018YFB0804701), the National Natural Science Foundation of China (62072239),
References


