Hindawi
Security and Communication Networks
Volume 2018, Article ID 1914980, 11 pages
https://doi.org/10.1155/2018/1914980

Research Article

Network Intrusion Detection Method Based on PCA and Bayes Algorithm

Bing Zhang[1,2] **Zhiyang Liu**[1,2] **Yanguo Jia**[1,2] **Jiadong Ren**[1,2] **and Xiaolin Zhao**[3]

[1]*School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei, China*
[2]*The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao City, 066004, China*
[3]*Beijing Key Laboratory of Software Security Engineering Technique, Beijing Institute of Technology, South Zhongguancun Street, Haidian District, Beijing, 100081, China*

Correspondence should be addressed to Yanguo Jia; jyg@ysu.edu.cn

Received 26 May 2018; Revised 24 September 2018; Accepted 17 October 2018; Published 13 November 2018

Guest Editor: Michał Choraś

Intrusion detection refers to monitoring network data information, quickly detecting intrusion behavior, can avoid the harm caused by intrusion to a certain extent. Traditional intrusion detection methods are mainly focused on rule files and data mining. They have the disadvantage of not being able to detect new types of attacks and have the slow detection speed. To address these issues, an intrusion detection method based on improved PCA combined with Gaussian Naive Bayes was proposed. By weighting the first few feature vectors of the traditional PCA, data pollution can be reduced. The number of final weighted principal components is *2* through sequential selection. The dimensionality reduction of the data is achieved through improved PCA. Finally, the intrusion behaviors were detected by using the Gaussian Naive Bayes classifier. The indexes of detection accuracy, detection time, precision rate, and recall rate were applied to evaluate the results. The experimental results show that, comparing with the traditional Bayes method, the method proposed in this article can reduce the detection time by 60%, shorten it to 0.5s, and increase the detection rate to 91.06%. The mean value of detection accuracy is about 86% by cross-validation.

1. Introduction

While the Internet brings convenience to people, there are also a lot of security problems. Network attacks are happening all the time. Research on Intrusion detection has important practical significance, and it is also a major challenge in the field of network security.

Dorothy Denning [1] defined intrusion detection in 1987. He detected intrusion behavior by monitoring network data, and then the system would give the alerts and responses before invasion. It can be found that an important feature of intrusion detection is instantaneity. The detection method needs to quickly judge the attack information and alarm before the occurrence of the hazard. There are two main types of traditional intrusion detection methods. One is the rule-based intrusion detection. It relies on analyzing the characteristics of specific attack types and then records the attack characteristics to the rule files. Finally, it detects the attacks by matching the rule files. This method is mainly applied to some commercial IDSs or open source IDSs. For example, Snort IDS [2, 3] applied this method because rule-based intrusion detection has the characteristics of fast detection. However, a major problem in this method is that it can not detect new attack types and can only detect the types of attacks that have been discovered. Hacker attacks are constantly changing. New types of attacks often occur, and new types of attacks often cause greater harm. Moreover, the method has higher false alarm rate. With the rise of machine learning and data mining in recent years, data mining methods have been commonly applied to intrusion detection, which is another method. The methods based on data mining establish the model by training through the marked data set. It has a good effect on the detection of unknown attack types, such as SVM [4, 5] and neural network [6]. The application of data mining in intrusion detection requires large collection of data in advance, which limits online intrusion detection [7].

At present, the conventional intrusion detection methods focused on data mining [8–10] and common file analysis [11] are sprang up. An et al. [12] used the method of combining the minimum within class scatter in Fisher discriminant analysis with traditional support vector machine (SVM) in intrusion detection and then proposed a minimum within class scatter support vector machine (WCS-SVM), which is better than the traditional SVM. Kabir et al. [13] proposed an intrusion detection system based on least squares support vector machine (LS-SVM). Mrudula Gudadhe et al. [14] proposed a new method to enhance the decision tree applied in intrusion detection, which allows the formation of a classifier combined with multiple decision trees. Sufyan et al. [15] applied back-propagation artificial neural network models into intrusion detection, which makes IDS more efficiently adapt to new environments and respond to new types of attacks. Because of the large size of the network dataset, manual tagging would consume a lot of time and effort; thus, clustering methods are introduced into the dataset classification [16]. The Y-means clustering algorithm [17] overcomes two disadvantages of dependency and degradation of k-means digital clusters. This method automatically divides the data set into a proper number of clusters. It is feasible and effective to perform intrusion detection using clustering analysis. The k-means [18] algorithm is the simplest segmentation algorithm that solves the well-known clustering problem. Clustering algorithm using SOM and k-means [19] can overcome the shortcomings of traditional SOM, such as not providing accurate clustering results, and can avoid the disadvantages of the traditional k-means, which always relies on the initial value and it is difficult to find the cluster center. The parallel clustering integration algorithm [20] proposed for IDSs can achieve high speed, high detection rate, and low false alarm rate. The ANN classifier [21] also has a good performance in intrusion detection. By using a mixed learning method, the studies in [22–24] have higher detection rates and lower false alarm rates; among them, the combination of clustering and classification can achieve good results. Shah et al. [25] compared the detection performance of the machine learning method directly in the Snort Intrusion detection system.

In addition to data mining-based intrusion detection methods mentioned above, flow-based intrusion detection [26] is an innovative method of detecting high-speed network intrusions. Stream-based intrusion detection only checks the header and does not analyze the payload of the packet. The filtering method [27] applies predefined standard RIA so as to select functions to eliminate extraneous related features from the data set. Vieira et al. [28] proposed a network attack detection and recognition method based on model selection and feature similarity and applied signal processing techniques into intrusion detection.

Traditional file analysis methods may be effective for conventional types of attacks but not for new attack techniques [29]. Although data mining method has good adaptability to new attack types, it is often higher in time consumption. Principal component analysis (PCA) is a commonly used dimensionality reduction technique. It uses an orthogonal transformation to convert a set of related variables into a set of linear uncorrelated variables, where the first principal component has the largest variance. And PCA has been used for attack detection [30]. Second, the Bayesian method [31] in the data mining method, to a certain extent, is faster than other classifiers because it is a classifier based on conditional probability. Based on these, this paper proposed a novel intrusion detection method combining the improved principal component analysis with Gauss naive Bayes. The proposed method would decrease the training time of Gauss Bayes classifier according to training on the dataset simplified by the improved PCA algorithm and then improved the detection accuracy. Before applying the Bayesian algorithm, the improved PCA was used to reduce the dimension, and the first few eigenvectors of the solution of the principal component analysis were multiplied by a weight coefficient. Then the Bayes classifier was used to compute the probability of each network data that was divided into normal and abnormal. According to the application of PCA, the detection time would be greatly reduced, and the detection rate would decrease slightly. But by exploiting the weight coefficient to improve the traditional PCA, the detection effect has also been improved significantly.

The other sections of this paper are organized as follows. Section 2 introduces the characteristic attributes of network data. In Section 3, the improved principal component analysis combined with Gaussian Naive Bayesian intrusion detection model is described. In Section 4, the KDD99 data set is analyzed, and the experimental results are listed, and cross-validation is used to verify the results. Section 5 summarizes the effect of the model and illustrates the direction of the method improvement and future work.

2. Data Model

2.1. Characteristic Attribute Description of Network Data

2.1.1. Basic Features of TCP Connections. The basic connection feature contains 9 basic attributes of some connections, which are shown in Table 1.

2.1.2. Content Features of TCP Connections. Attacks such as U2R and R2L are generally embedded because they do not have frequent sequential patterns in data records like DoS attacks. In the data payload of a packet, there is no difference between a single packet and a normal connection. In order to detect such attacks, some content features that may reflect the intrusion behavior can be extracted from the data content. There are 13 kinds of content features as shown in Table 2.

2.1.3. Statistical Characteristics of Network Traffic. Since the network attack event has a strong correlation in time, some connections exist between the current connection record and the connection record in the previous period of time, which can better reflect the relationship between the connections. Time interval takes two seconds. There are 9 kinds of network traffic features. As shown in Table 3.

2.2. Feature Definition of Network Data. The basic features of the TCP connection are expressed as B, and there are 9 kinds

Table 1: The basic characteristics of TCP connection.

Features	Descriptions
duration	The connection duration
protocol_type	Protocol types, including TCP, UDP, ICMP
service	The type of network service for the target host
flag	A state that connects normal or wrong
src_bytes	The number of bytes from the source host to the target host
dst_bytes	The number of bytes from the target host to the source host
land	Whether the connection is from the same host or port
wrong_fragment	The number of erroneous segments
urgent	The number of emergency packages

Table 2: Content features of TCP connections.

Features	Descriptions
hot	Number of times to access system sensitive files and directories
num_failed_logins	The number of failed login attempts
logged_in	The successful login is 1, otherwise 0
num_compromised	The number of times the compromised condition appears
root_shell	1 if the root shell was obtained, 0 otherwise
su_attempted	If the "su root" command appears, it is 1, otherwise it is 0
num_root	The number of root user access
num_file_creations	The number of times the file is created
num_shells	The number of times the shell command is used
num_access_files	The number of access control files
num_outbound_cmds	The number of outbound connections in an FTP session
is_hot_login	Whether the login belongs to the "hot" list
is_guest_login	1 if guest login, 0 otherwise

Table 3: Statistical characteristics of network traffic.

Features	Descriptions
count	The number of connections with the same target host as the current connection
srv_count	The number of connections with the same service as the current connection
serror_rate	Percentage of connections with "SYN" errors in connections with the same target host as the current connection
srv_serror_rate	Percentage of connections with "SYN" errors in connections with the same service as the current connection
rerror_rate	Percentage of connections with "REJ" errors in connections with the same target host as the current connection
srv_rerror_rate	Percentage of connections with "REJ" errors in connections with the same service as the current connection
same_srv_rate	Percentage of connections with the same destination as the current connection in the connection with the same target host as the current connection
diff_srv_rate	Percentage of connections with different services from the current connection in connections with the same target host as the current connection
srv_diff_host_rate	Percentage of connections with different target hosts for the current connection in the connection with the same service as the current connection

of connection features, so $B = \{b1, b2, \ldots, b9\}$. The content features of TCP connections are represented as C, and there are 13 kinds of content features, so $C = \{c1, c2, \ldots, c13\}$. The statistical characteristics of network traffic are denoted as F, and there are 9 kinds of traffic characteristics, so $F = \{f1, f2, \ldots, f9\}$. The training network data set is defined as D, and the test network data set is defined as T. A network connection record for the data set is D_i and T_i.

Definition 1. A record D_i in the training set and a record T_i in the test set are as follows:

$$D_i = \{B, C, F\}, \quad i = 1, 2, \ldots n, (\text{Training data set includes n records}). \tag{1}$$

$$T_i = \{B, C, F\}, \quad i = 1, 2, \ldots, m, (\text{Test data set includes m records}) \tag{2}$$

Definition 2. As both the training data and the test data are applied to the principal component analysis, the data matrix is defined as X, then X=D or T, and one of the connection records is as follows:

$$x = D_i \text{ or } T_i. \tag{3}$$

Input: *Training Network Data Set* $D = \{b1, b2, \ldots, b9, c1, c2, \ldots, c13, f1, f2, \ldots, f9\}$
Output: *New training data matrix* $D' = \{v1, v2, \ldots, vd'\}$
1. $X' = D - mean$ // *Remove the average value*
2. *Find the covariance matrix* $X'X'^T$ *of the data matrix* X'
3. *Finding the eigenvalues* λ *and eigenvectors of the covariance matrix* $X'X'^T$
4. *Arranging feature values* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$
5. $\omega = (\omega_1, \omega_2, \ldots, \omega_{d'})$
6. $\Omega' = (\kappa\omega_1, \kappa\omega_2, \kappa\omega_3, \ldots, \kappa\omega_n, \ldots, \omega_{d'})$ $(n \leq d')$
7. $D' = \{v1, v2, \ldots, vd'\} = X * \omega'$

Algorithm 1: IPCA.

3. Improved Principal Component Analysis and Gauss Naive Bayes

3.1. Traditional Principal Component Analysis (PCA). Principal component analysis has the advantage of reducing data complexity and identifying the most important features. On the contrary, it has the disadvantage that it may lose useful information.

From the perspective of maximum separability, principal component analysis can be explained. The projection of a connection record x on the hyperplane in the new space is $\omega^T x$. If the sample points are projected as separate as possible, correspondingly, the variance of the sample points after the projection should be as maximized as possible. The variance of the sample after projection is as follows:

$$S = \sum_i \omega^T X X^T \omega. \tag{4}$$

To optimize it, it can be simplified by using the Lagrange multiplier method. The detailed replacement is below.

$$XX^T\omega = \lambda\omega. \tag{5}$$

The covariance matrix XX^T is decomposed by eigenvalue, and the obtained eigenvalues are from large to small: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. Then the feature vector corresponding to the first d' eigenvalues was used to construct $\omega = (\omega_1, \omega_2, \ldots, \omega_{d'})$, which is the solution to the principal component analysis.

3.2. Improved Principal Component Analysis (IPCA). In the field of image processing, the first three eigenvectors of the classical PCA method reflect the overall information of the image [32]. When the lighting conditions have a significant effect, the first three principal components of the PCA method may be polluted seriously. But reducing their weight can improve the accuracy. Inspired by this, the image may be significantly affected by light, and the network data may also be affected by some factors. But the best principal number may not be always 3 in different environments; this value should be determined by trials. In this paper, we set this value as n, and then the improved PCA algorithm weights the first n feature vectors as shown in

$$\Omega' = \left(\kappa\omega_1, \kappa\omega_2, \kappa\omega_3, \ldots, \kappa\omega_n, \ldots, \omega_{d'}\right) \quad \left(n \leq d'\right). \tag{6}$$

In (6), k is the weight coefficient, which is a number between 0 and 1. The purpose of k is to reduce the weight of the first n principal components and decrease the influence of those components. Then, the IPCA algorithm is used to reduce the dimension of the data. The pseudo-code for improved principal component analysis is shown in Algorithm 1.

In Algorithm 1, the mean of the data matrix was removed in Line 1 and Line 2 searched the covariance matrix of the data matrix. Lines 3-4 found the eigenvalues and eigenvalue vectors of the covariance matrix and arranged the eigenvalues from the largest to the smallest. Line 5 selected the feature vector corresponding to the first d′ eigenvalues as the solution of the traditional PCA. Line 6 gave a new solution to the weighting of the traditional PCA solution. Line 7 multiplied the new solution with the data matrix to reduce the data to d'.

To weight the first two principal components in IPCA is the best by sequential selection. But this is limited to this experiment, and this value may change with time. In Section 4.2.2, it is analyzed in detail.

3.3. Gaussian Naive Bayesian Classifier (GNB). With all relevant probabilities known, Bayesian decision theory considers how to choose the best class labels based on these probabilities and misclassified losses. For intrusion detection tasks, we should determine whether the network traffic is normal or abnormal. Assume that there are two possible class labels in $\gamma = \{c_1, c_2\}$, where c_1 stands for normal category mark and c_2 represents an anomaly category tag. For each connection record x, a category flag that maximizes the posterior probability $P(c \mid x)$ is selected. Based on Bayes' theorem, $P(c \mid x)$ can be written as follows:

$$P(c \mid x) = \frac{P(c)\,P(x \mid c)}{P(x)}. \tag{7}$$

In (7), $P(c)$ is a kind of prior probability, $P(x \mid c)$ is the conditional probability of a connection record x relative to the class label c, and $P(x)$ is the evidence factor used for normalization. For a given connection record x, the evidence factor $P(x)$ has no relationships with class labels, so $P(c \mid x)$ is only related to $P(c)$ and $P(x \mid c)$.

The naive Bayes classifier uses the "attribute conditional independence assumption." For known classes, it is assumed

that all attributes are independent of each other. In other words, each attribute can affect the classification result independently,

$$\mathrm{P}(c \mid x) = \frac{P(c)\,P(x \mid c)}{P(x)} = \frac{P(c)}{P(x)}\prod_{i=1}^{d} P(x_i \mid c), \quad (8)$$

where d is the number of attributes for each connection record, and If IPCA is not used, and d is equal to 31. x_i is the value of the connection record x on the i-th attribute. Since $\mathrm{P}(x)$ is the same for all categories, the naive Bayes classifier has the following expression as $\mathrm{h_{nb}}(\mathrm{x})$:

$$\mathrm{h_{nb}}(\mathrm{x}) = \arg\max_{c\epsilon\gamma} P(c) \prod_{i=1}^{d} P(x_i \mid c). \quad (9)$$

For the continuity property, the probability density function is considered. Assume that there exists $\mathrm{P}(x_i \mid c) \sim \mathrm{N}(\mu_{c,i}, \sigma^2_{c,i})$, where $\mu_{\mathrm{c,i}}$ and $\sigma^2_{c,i}$ are the mean and variance of the value of the c-th sample on the i-th attribute. And the $\mathrm{P}(x_i \mid c)$ is shown as follows:

$$\mathrm{P}(\mathrm{x_i}c) = \frac{1}{\sqrt{2\pi\sigma_{c,i}}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma^2_{c,i}}\right). \quad (10)$$

For each connection record, the first is to calculate the posterior probability of the normal and abnormal categories, and the larger one would be selected as marker for the result of the category of the record.

3.4. The Detection Process of the Model. For the detection process based on improved PCA and Bayes intrusion detection model, we first normalize the training data set D and test data set T. The normalization of data is mainly to facilitate the selection of weight coefficients. Normalization has no influence on the detection rate. The normalized new value is calculated by the following equation. The mapping range of the new value is 0 to 1:

$$X_{\mathrm{norm}} = \frac{X - X_{\mathrm{min}}}{X_{max} - X_{min}}. \quad (11)$$

After the data sets are normalized, the dimension of training data set D and the test data set T are reduced by the improved PCA, and a new training set D' and a test set T' are obtained.

Assume that the variables obey the Gaussian distribution, the posterior probability is calculated, and the category flag that maximizes the posterior probability $\mathrm{P}(c \mid x)$ is selected as the record's detection result. The detailed detection process of the model is shown in Figure 1. And the detailed steps are described as follows.

(1) IPCA process. The first step is to remove the average value of the data, the second step is to calculate the covariance matrix of the data matrix, the third step is to calculate the eigenvalues and eigenvectors of the covariance matrix, and the fourth step is to sort the eigenvalues from the largest to the smallest, and the first few eigenvectors are weighted in step 5. In the last step, the weighted eigenvectors are multiplied by the data matrix to obtain a reduced-dimensional training data set D' and a test data set T'.

(2) GNB process. A Gaussian Bayes classifier is applied to the dimensionality-reduced test data set T' to classify the category of each record. First, the conditional probability $\mathrm{P}(x_i \mid c)$ of each attribute is calculated according to (10), and the prior class probability $\mathrm{P}(c)$ of records belonging to normal and anomaly are calculated separately. Finally, the prior probability of recording with normal and anomaly are computed, and the category of the record with large prior probability is selected as the detection result of the record.

The model needs to continuously adjust the weight coefficient of the improved PCA so as to find the most optimal weight coefficient. The sum of the training and testing time of the model is regarded as the detection time. The detection rate is the ratio of the number of correct records divided by the total number of records in the test set.

4. Experimental Results and Analysis

4.1. Experimental Data. Intrusion detection requires a large amount of effective experimental data. The experiment is conducted on the KDDCup99 data set in this paper. The KDD99 data set is a reference data set in the domain of network intrusion detection and lays the foundation for the research of network intrusion detection based on computational intelligence. Besides the KDD99 data set, DARPA98 and NSL-KDD are also two verification data sets commonly used. The KDD99 data set is obtained after data mining and preprocessing on the DARPA98 data set. The NSL-KDD data set is a refined version of KDD99, after removing redundant data. This paper selects the classic KDD99 data set, namely, collecting network connection data from a simulated US Air Force LAN in nine weeks.

The data contain the data with identification and no identification. We use the data with identification. Test data and training data have different probability distributions. The test data has some types of attacks that are not present in the training data. The training data set includes a normal marked type and 22 training attack types. In addition, 14 attack types only appear in the test data set. All these attack types can be classified into four exception types which are *denial of service attacks, unauthorized access from remote hosts, unauthorized local superuser privileged access*, and *port scanning*. The four attack types are uniformly marked as abnormal and the rest are marked as normal. The KDD99 data set consists of a total of 5 million records, 10% percentage of which are chosen randomly as target training data set that contains a total of 494,021 records, and the test set includes a total of 311,029 records. The data set includes a total of 41 attributes. Through the analysis on 41 fixed feature attributes, the first 31 feature attributes including 9 discrete types and 22 continuous types can reflect the state changes. The new data set owns 31 feature values.

4.2. Experimental Results

4.2.1. The Effect of Classical Classifiers in Intrusion Detection. The experiment is conducted in a PC equipped with an

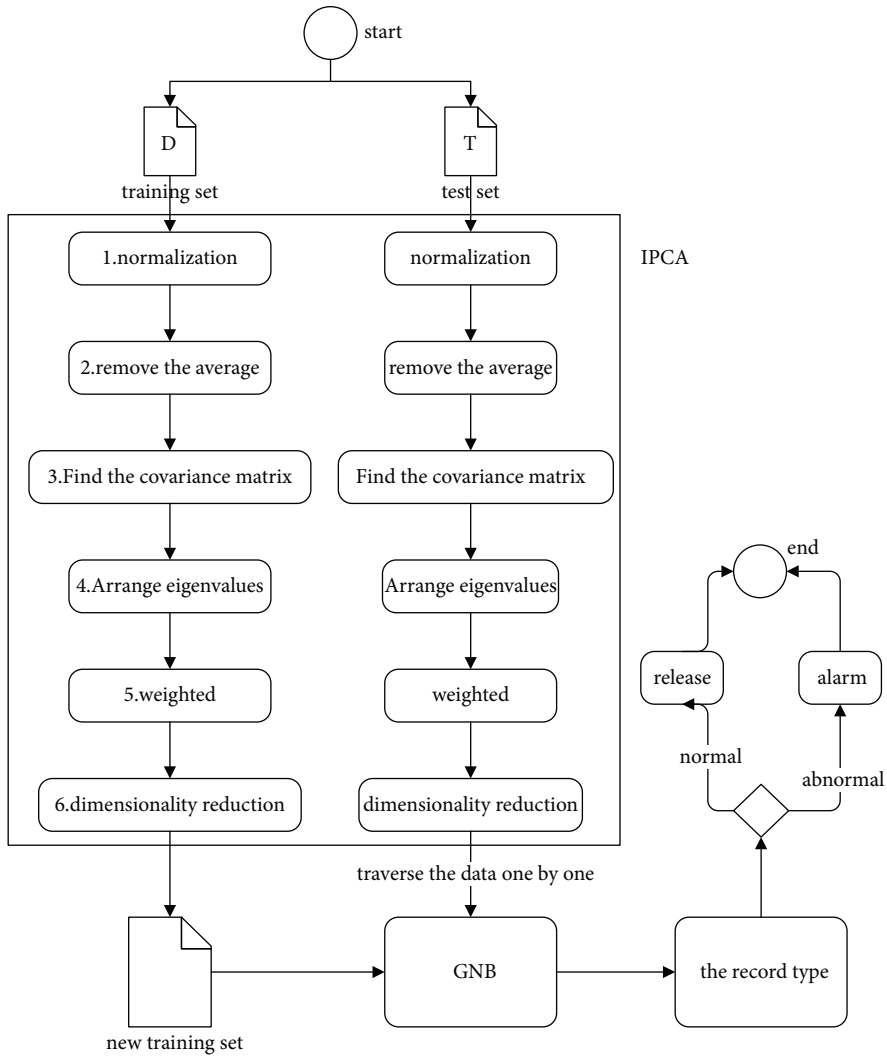

Figure 1: Intrusion detection process of IPCA and GNB model.

Intel G2020 CPU, 8GB RAM, and a Windows 7 operating system. The algorithms commonly used in the domain of intrusion detection are K-Nearest Neighbor (KNN) [33], support vector machine (SVM) [34], Gradient Boosting Decision Tree (GBDT) [35], etc. Their execution effects based on the KDD99 dataset in intrusion detection are shown in Table 4.

By comparison among those classic classifiers, although GNB has the lowest detection accuracy, it can train and test the model within 1.42s. Other classifiers have higher detection rate, but they can not meet the requirements of intrusion detection with a longer detection time. SVM can even take up to 10 hours, and GBDT also takes more than 2 minutes. Considering the time consumption, therefore, the GNB classifier was selected as an intrusion detection classifier in this paper, but there is a need for some improvements or optimization. Although the detection rate for GNB classifier is not as good as other classifiers, it has shorter detection time. Moreover, after the improvement on PCA which is to preprocess the input data for GBN classifier, the detection rate of the model has a big improvement and is close to that of other classifiers, and the detection time would be greatly shortened.

4.2.2. GNB Combined with IPCA. When the data dimension is reduced, some of the original data information will be lost, so the detection effect may be decreased. In order to show the time index in intrusion detection more clearly, the detection time of the model combining PCA and Gaussian Naive Bayes was recorded according to different number of principal components. The relationship between the principal component number and time is plotted as a line graph shown in Figure 2.

From Figure 2, it can be seen that as the number of principal components increases, the training and testing time of the model will be longer. It is well understood that the more data features the model input, the greater the amount of data were, and the longer training and testing time the model would spend. At the same time, it can be seen that when the PCA algorithm was not combined, the detection time of the

Table 4: Comparison of effects of classifiers commonly used.

Classifier	KNN	GNB	SVM	GBDT
Detection time (s)	2888.12	1.42	36939.30	171.95
Detection rate	0.9332	0.8328	0.8992	0.9355

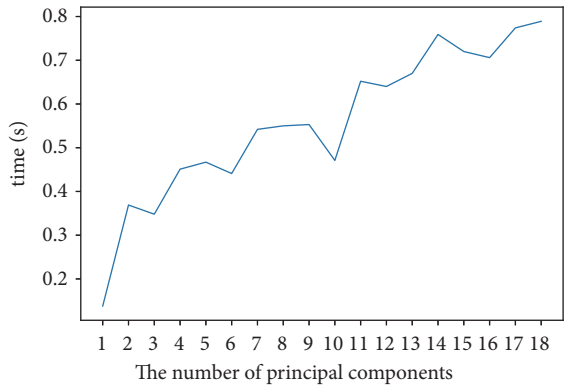

Figure 2: The relationship of principal component number and detection time.

Gauss-Bayes method was 1.42s, and the detection time was greatly decreased under 1s after the combination. The shortest time required is only 0.138s.

In order to select the appropriate number of weighted principal components, we compared the effects of different principal components from 1 to 18 on experimental results. And we found that the trend with 6-18 principal components has similar decreasing trend to that of 2-6, so we have not put them in Figure 3. The results of the relationship between detection rate and weight coefficient with different principal components from 1 to 6 are shown in Figure 3.

In Figure 3, the abscissa represents a weight coefficient from 10^{-1} to 10^{-6}. When the coefficient is 10^{-4}, the detection rate has been significantly improved. After that, it has small change and tends to be steady. Thus, we set 10^{-4} as the optimal weight coefficient. To compare the influence of weighted principal number on detection rate more clearly, after fixing the optimal weight coefficient, we discovered the best detection rate corresponding to the principal component number. The statistical data are shown in Figure 4.

According to Figure 4, we can get that when the number of principal components is 1, the improvement on detection rate is not obvious. When the number of principal components is 6, the detection rate dropped. The main reason is that when the number of principal components is too small, it is difficult to eliminate data pollution completely; when it is too large, the valid information in the data will be lost. Then the optimal number of principal components is 2. At the same time, from Figure 3, it can be seen that the detection rate is obviously improved when the weight coefficient is 10^{-4} and the principal component number is 2. When the principal number is 2, the detailed detection accuracy with different weight coefficient is shown in Table 5.

It can be seen that the weight of the first two eigenvectors can play a significant improvement effect; when the coefficient is 10^{-4}, the accuracy rate has been significantly improved, which reaches to 91.06%. Therefore, the detection rate of the model combining GNB and IPCA is close to that of other classifiers, and the detection time is much more fewer than the time consumption of other classifiers.

Table 5: Weights and accuracy.

Coefficient k	Correct rate
10^{-1}	0.8052
10^{-2}	0.8052
10^{-3}	0.8029
10^{-4}	0.9106
10^{-5}	0.9109
10^{-6}	0.9109

4.2.3. Evaluation of Models. In this section, the results on three models *GNB*, *PCA + GNB*, and *IPCA + GNB* are compared from the perspective of time consumption and accuracy rate. And the detailed information is shown in Table 6.

From Table 6, we can see that the IPCA combined with Gauss naive Bayes model has good effect. Comparing with result from GNB, the time is shortened by 0.858s, and the accuracy rate is increased by 9%. Training data set *D* contains about 500,000 records, and test data set T contains more than 300,000 records. On such amount of data, it just takes about 0.562 seconds to train the model and test the data. The accuracy rate reached to 91.06%. In addition, other indicators such as precision, recall, and f1-score were also used to evaluate the model. The statistics of these three indicators on evaluating classical data mining methods mentioned in previous section are shown in Table 7.

The effects of the model presented in this paper are greatly improved compared to the traditional GNB, which ultimately are close to or even better than the effects of KNN and SVM in all kinds of evaluation indicators. At the same time, it can

Table 6: Comparison of three models.

Model	GNB	PCA + GNB	IPCA + GNB
time (s)	1.42	0.557	0.562
Accuracy rate	0.8328	0.8052	0.9106

Table 7: Precision, recall, and f1-score.

model	category	precision	recall	f1-score
KNN	normal	0.75	0.99	0.85
	abnormal	1.00	0.92	0.96
	avg / total	0.95	0.93	0.94
SVM	normal	0.66	1.00	0.79
	abnormal	1.00	0.88	0.93
	avg / total	0.93	0.90	0.91
GDB	normal	0.76	0.99	0.86
	abnormal	1.00	0.92	0.96
	avg / total	0.95	0.94	0.94
GNB	normal	0.54	0.98	0.70
	abnormal	0.99	0.80	0.88
	avg / total	0.90	0.83	0.85
PCA+GNB	normal	0.00	0.00	0.00
	abnormal	0.81	1.00	0.89
	avg / total	0.65	0.81	0.72
IPCA+GNB	normal	0.80	0.73	0.76
	abnormal	0.94	0.96	0.95
	avg / total	0.91	0.91	0.91

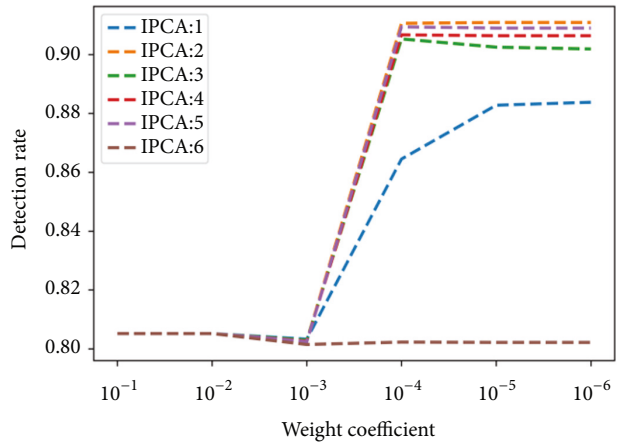

Figure 3: The relationship of detection rate and weight coefficient.

also be seen that the values of three evaluation indicators have decreased after the introduction of PCA. After improving PCA, the indicators have increased significantly. The intrusion detection method proposed in this paper has the highest detection accuracy compared with the previous two methods (GNB, PCA+GNB). After introducing PCA, the detection rate is slightly decreased, which is obviously improved by IPCA. The time involved in the experiment is the execution time of an experiment. Due to different performance and stability for the computer, each experiment result will be slightly different, but it would not be big difference. In order to make a clearer comparison of the differences between the three methods in the aspect of time consumption of intrusion detection, the time is recorded by doing the experiment ten times for each method. They are shown in Figure 5.

In Figure 5, it is clear to see the time contrast among the three methods. The average detection time of GNB method is 1.259s, and the average detection time of PCA and GNB is 0.558s. The average detection time of IPCA and GNB is 0.494s. It is proved that the intrusion detection method proposed in this paper has the highest detection accuracy and the shortest detection time in the three methods.

Table 8: Cross-validation results.

	1	2	3	4	5	mean
10^{-1}	0.4308	0.8483	1.0000	0.9313	0.8050	0.8031
10^{-2}	0.4308	0.8483	1.0000	0.9313	0.8050	0.8031
10^{-3}	0.4422	0.8474	1.0000	0.9316	0.8053	0.8053
10^{-4}	0.5931	0.9124	1.0000	0.9505	0.8609	0.8634
10^{-5}	0.6170	0.9438	1.0000	0.9542	0.9021	0.8834
10^{-6}	0.6172	0.9439	1.0000	0.9542	0.9021	0.8834

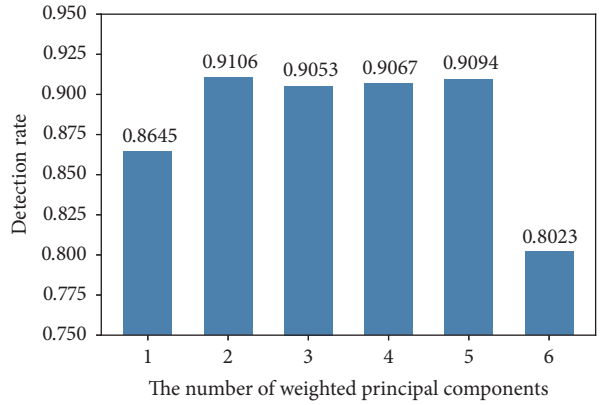

Figure 4: Detection rate with different principal component number.

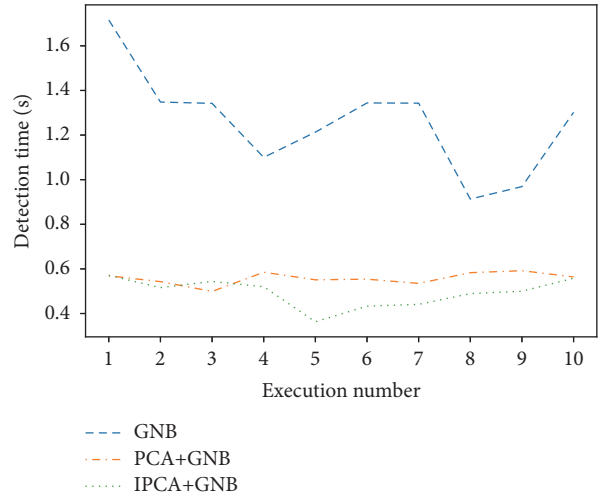

Figure 5: The relationships between the detection time and the execution number.

4.3. Cross-Validation. The accuracy obtained from the above experiments is based on just one experiment for each method. Training data and test data come from the training data and test data that have been divided in the KDD99 dataset, so it may not have universal significance. To get a convincing accuracy rate of intrusion detection, cross-validation is applied here. Instead of using the test data set prepared by KDD99, training data set D is divided into L subsets with the same size, then select one of them as the verification set at each time, and the remaining L-1 subsets are regarded as training data sets. The cross-validation results are shown in Table 8. In this paper, we set L=5.

The results of cross-validation prove that the optimization parameter is 10^{-6}, which is different from the optimal parameters of the above experiment, but the results of these several parameters are not much different, and different data set may result in different optimal parameters. Therefore, the optimal weight coefficient still takes 10^{-4}. The average detection rate is 86.34%. Although the detection rate by the experiment on the test set in KDD99 is higher, the detection accuracy is still up to 86%, which proves the efficiency of the method.

5. Conclusion and Future Work

This paper proposed an intrusion detection method based on improved PCA and Bayes. Comparing with different classifiers, it shows that Bayes classifier is more suitable for

intrusion detection because of its fast speed for classification. The intervention of principal component analysis can greatly reduce the detection time, and then the weight coefficient was defined to improve the PCA, so as to simplify the input data. By comparing the detection rate and detection time with the classical Bayesian intrusion detection method, it proves that the method presented in this paper works best in network intrusion detection. This method has high accuracy, and it can also solve the high requirement of intrusion detection timely.

What is more, some works need to be further improved in our future research; for instance, this paper only focuses on the overall detection rate with normal and abnormal. It does not pay attention to the detection effect on different types of attack. And the proposed model may not work well for a particular attack. There is also no scientific selection method for the selection of weight coefficients for the improved PCA method. The future work would mainly focus on the selection of coefficient and explore the relationships between the weight coefficient and the characteristics of the data itself.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Key R&D Program of China under Grant No. 2016YFB0800700, the National Natural Science Foundation of China under Grant Nos. 61802332, 61772449, 61772451, 61572420, 61807028, and 61472341, and the Natural Science Foundation of Hebei Province, China, under Grant No. F2016203330 and the doctoral Foundation Program of Yanshan University under Grant No. BL18012. The authors are grateful to the valuable comments and suggestions of the reviewers.

References

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

[2] W. Park and S. Ahn, "Performance Comparison and Detection Analysis in Snort and Suricata Environment," *Wireless Personal Communications*, vol. 94, no. 2, pp. 241–252, 2016.

[3] R. T. Gaddam and M. Nandhini, "An analysis of various snort based techniques to detect and prevent intrusions in networks: Proposal with code refactoring snort tool in Kali Linux environment," in *Proceedings of the 2017 International Conference on Inventive Communication and Computational Technologies, ICICCT 2017*, pp. 10–15, India, March 2017.

[4] E. A. Shams and A. Rizaner, "A novel support vector machine based intrusion detection system for mobile ad hoc networks," *Wireless Networks*, pp. 1–9, 2017.

[5] W. Shang, L. Li, M. Wan, and P. Zeng, "Industrial communication intrusion detection algorithm based on improved one-class SVM," in *Proceedings of the World Congress on Industrial Control Systems Security, WCICSS 2015*, pp. 21–25, UK, December 2015.

[6] T. Jan, "Ada-Boosted Locally Enhanced Probabilistic Neural Network for IoT Intrusion Detection," in *Proceedings of the Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 583–589, Springer, 2018.

[7] C.-T. Huang, R. K. C. Chang, and P. Huang, "Signal Processing Applications in Network Intrusion Detection Systems," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, Article ID 527689, 2 pages, 2009.

[8] U. Adhikari, T. H. Morris, and S. Pan, "Applying Non-Nested Generalized Exemplars Classification for Cyber-Power Event and Intrusion Detection," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 3928–3941, 2018.

[9] R. Taormina and S. Galelli, "A Deep Learning approach for the detection and localization of cyber-physical attacks on water distribution systems," *Journal of Water Resources Planning & Management*, vol. 144, no. 10, Article ID 04018065, 2018.

[10] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework," *Journal of Network and Computer Applications*, vol. 67, pp. 147–165, 2016.

[11] F. Raynal, Y. Berthier, P. Biondi, and D. Kaminsky, "Honeypot forensics," in *Proceedings of the Proceedings fron the Fifth Annual IEEE System, Man and Cybernetics Information Assurance Workshop, SMC*, pp. 22–29, USA, June 2004.

[12] W. J. An and M. G. Liang, "A new intrusion detection method based on SVM with minimum within-class scatter," *Security and Communication Networks*, vol. 6, no. 9, pp. 1064–1074, 2013.

[13] E. Kabir, J. Hu, H. Wang, and G. Zhuo, "A novel statistical technique for intrusion detection systems," *Future Generation Computer Systems*, vol. 79, pp. 303–318, 2018.

[14] M. Gudadhe, P. Prasad, and K. Wankhade, "A new data mining based network intrusion detection model," in *Proceedings of the 2010 International Conference on Computer and Communication Technology, ICCCT-2010*, pp. 731–735, India, September 2010.

[15] S. T. Al-Janabi and H. A. Saeed, "A Neural Network Based Anomaly Intrusion Detection System," in *Proceedings of the 2011 Developments in E-systems Engineering (DeSE)*, pp. 221–226, Dubai, United Arab Emirates, December 2011.

[16] K. D. Denatious and A. John, "Survey on data mining techniques to enhance intrusion detection," in *Proceedings of the International Conference on Computer Communication and Informatics*, pp. 1–5, 2012.

[17] Y. Guan, A. A. Ghorbani, and N. Belacel, "Y-means: A clustering method for intrusion detection," in *Proceedings of the CCECE 2003 Canadian Conference on Electrical and Computer Engineering: Toward a Caring and Humane Technology*, pp. 1083–1086, Canada, May 2003.

[18] H. Li, "Research and Implementation of an Anomaly Detection Model Based on Clustering Analysis," *Journal of Beijing Information Science & Technology University*, pp. 458–462, 2010.

[19] H.-B. Wang, H.-L. Yang, Z.-J. Xu, and Z. Yuan, "A clustering algorithm use SOM and K-means in intrusion detection," in *Proceedings of the 1st International Conference on E-Business and E-Government (ICEE '10)*, pp. 1281–1284, May 2010.

[20] H. Gao, D. Zhu, and X. Wang, "A Parallel Clustering Ensemble Algorithm for Intrusion Detection System," in *Proceedings*

of the 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), pp. 450–453, Hong Kong, China, August 2010.

[21] Akashdeep, I. Manzoor, and N. Kumar, "A Feature Reduced Intrusion Detection System Using ANN Classifier," *Expert Systems with Applications*, vol. 88, pp. 249–257, 2017.

[22] Z. Muda, W. Yassin, M. N. Sulaiman, and N. I. Udzir, "Intrusion detection based on K-Means clustering and Naïve Bayes classification," in *Proceedings of the 7th International Conference on Information Technology in Asia (CITA '11)*, pp. 1–6, IEEE, July 2011.

[23] M. Ishida, H. Takakura, and Y. Okabe, "High-performance intrusion detection using OptiGrid clustering and grid-based labelling," in *Proceedings of the 11th IEEE/IPSJ International Symposium on Applications and the Internet, SAINT 2011*, pp. 11–19, Germany, July 2011.

[24] H. Om and A. Kundu, "A hybrid system for reducing the false alarm rate of anomaly intrusion detection system," in *Proceedings of the 2012 1st International Conference on Recent Advances in Information Technology, RAIT-2012*, pp. 131–136, India, March 2012.

[25] S. A. R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to Snort system," *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018.

[26] M. F. Umer, M. Sher, and Y. Bi, "Flow-based intrusion detection: Techniques and challenges," *Computers & Security*, vol. 70, pp. 238–254, 2017.

[27] P. Bermejo, L. De La Ossa, J. A. Gámez, and J. M. Puerta, "Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking," *Knowledge-Based Systems*, vol. 25, no. 1, pp. 35–44, 2012.

[28] T. P. B. Vieira, F. D. Tenrio, J. P. C. Costa et al., "Model order selection and eigen similarity based framework for detection and identification of network attacks," *Journal of Network Computer Applications*, vol. 90, pp. 26–41, 2017.

[29] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," pp. 217–228.

[30] S. Almotairi, A. Clark, G. Mohay, and J. Zimmermann, "A technique for detecting new attacks in low-interaction honeypot traffic," in *Proceedings of the 2009 4th International Conference on Internet Monitoring and Protection, ICIMP 2009*, pp. 7–13, Italy, May 2009.

[31] W. Lee and S. J. Stolfo, *A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems (Computer Security, Network Security)*, Columbia University, 1999.

[32] R. J. Li, Q. L. Han, and X. H. Yang, "New Optimization Method of PCA Face Recognition," *Journal of Dalian Jiaotong University*, vol. 29, no. 4, pp. 48–51, 2008.

[33] Y. Li and L. Guo, "An active learning based TCM-KNN algorithm for supervised network intrusion detection," *Computers & Security*, vol. 26, no. 7-8, pp. 459–467, 2007.

[34] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," *Computers & Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005.

[35] L. Li, Y. Yu, S. Bai, Jianjun Cheng, and Xiaoyun Chen, "Towards Effective Network Intrusion Detection: A Hybrid Model Integrating Gini Index and GBDT with PSO," *Journal of Sensors*, vol. 2018, Article ID 1578314, 9 pages, 2018.