A Novel Network Anomaly Detection Method based on Data Balancing and Recursive Feature Addition

Xinqian Liu[1,2], Jiadong Ren[1,2*], Haitao He[1,2], Qian Wang[1,2] and Shengting Sun[1,2]
[1] Department of Information Science and Engineering, Yanshan University
[2] Hebei Key Laboratory of Software Engineering
Qinhuangdao, 066001 - China
[e-mail: jdren@ysu.edu.cn]
*Corresponding author: Jiadong Ren

Received February 11, 2020; revised April 11, 2020; revised April 29, 2020; accepted May 12, 2020; published July 31, 2020

Abstract

Network anomaly detection system plays an essential role in detecting network anomaly and ensuring network security. Anomaly detection system based machine learning has become an increasingly popular solution. However, due to the unbalance and high-dimension characteristics of network traffic, the existing methods unable to achieve the excellent performance of high accuracy and low false alarm rate. To address this problem, a new network anomaly detection method based on data balancing and recursive feature addition is proposed. Firstly, data balancing algorithm based on improved KNN outlier detection is designed to select part respective data on each category. Combination optimization about parameters of improved KNN outlier detection is implemented by genetic algorithm. Next, recursive feature addition algorithm based on correlation analysis is proposed to select effective features, in which a cross contingency test is utilized to analyze correlation and obtain a features subset with a strong correlation. Then, random forests model is as the classification model to detection anomaly. Finally, the proposed algorithm is evaluated on benchmark datasets KDD Cup 1999 and UNSW_NB15. The result illustrates the proposed strategies enhance accuracy and recall, and decrease the false alarm rate. Compared with other algorithms, this algorithm still achieves significant effects, especially recall in the small category.

Keywords: Network anomaly detection, data balancing, recursive feature addition, parameters combination optimization, random forests

This work was supported by the National Natural Science Foundation of China (61572420,61772449) and the Natural Science Foundation of Hebei Province (F2019203120).

http://doi.org/10.3837/tiis.2020.07.020 **ISSN : 1976-7277**

1. Introduction

Due to various new users, new devices and new applications constantly connecting to the network, network service has been widely applied in all fields [1]. With the rapid development of the network, different anomalies and attacks occur frequently, which produces great damages to network performance and security [2]. Facing this situation, multiple security mechanisms, such as antivirus software, firewall technology, user authentication and access control [3], have been designed and applied to prevent abnormal behaviors and detect potential risks [4]. However, because of the increasing frequency and intensity of cyber-attacks, the existing security mechanisms are unable to effectively protect cyberspace security [5]. Hence, intrusion detection systems (IDSs) as the second defense line have been researched and utilized.

In general, IDSs are divided into two categories: misuse-based and anomaly-based [6]. The misuse-based IDSs can well detect existing attack behaviors with good accuracy and false alarm rate by building a rule library of normal behaviors and attack behaviors. However, the disadvantage of this method is the unavailability of detecting new attack behaviors [7]. The anomaly-based IDSs establish a detection model based on the normal behaviors in the historical data and try to detect the deviations. When the deviation exceeds a certain threshold, an abnormal behavior is defined. This method can detect new and unknown abnormal behaviors, but its drawback lies in low accuracy and high false alarm rate [8]. Due to the increasing complexity of the network environment and the sustained growth of zero-day attacks, the researches of the current IDSs are focused on the anomaly-based IDSs. Therefore, this paper also commits to studying the anomaly-based IDSs to detect new attacks.

In the anomaly-based IDSs, many data mining and machine learning methods are implemented to improve the IDS performance. For example, Fuzzy logic (FL), Bayesian network (NB), Artificial neural networks (ANN), Random forests (RF), Support vector machine (SVM) and so on [9-14]. In addition, Abdulla et al. [15] constructed an intrusion detection classifier by using WOAR-SVM (weighted one-against-rest SVM), and compared with many SVM classifiers (such as, OAO-SVM (one-against-one SVM), and DAG-SVM (directed acyclic graph SVM)). The results demonstrate that WOAR-SVM model can better offset errors and obtain better detection performance. Ugo et al. [16] built normal flow model by using restricted Boltzmann machine and energy model to detect abnormal network behaviors. Practical experiences show that the application of machine learning algorithms improves the performance of IDSs largely, and different classifiers have a huge impact on anomaly detection results. Therefore, it is necessary to construct an effective anomaly detection classifier. Based on a large number of experiments and the ability of the RF classifier in processing complex and multi-category data [17], the random forests algorithm is as the classification model in this paper.

Although machine learning methods have many advantages, the performance of these methods is greatly reduced due to the inherent characteristics of network traffic: unbalancing and high-dimension. In order to address this problem, many techniques are applied to anomaly detection. Facing the unbalancing characteristic of network traffic, sampling technology, clustering method and outlier detection method are proposed. Hamed et al. [1] applied the SMOTE technique to handle the minority class in the unbalanced network data. In SMOTE, the samples of minority class are produced by certain strategies to achieve the goal of balancing network dataset. The application of sampling technology improves the performance

of anomaly detection to a large extent. However, because of the randomness of the sampling technique, it is impossible to evaluate the sampling data objectively. Wathiq et al. [18-20] proposed an improved K-means method to extract each majority category and select partial representative data. Ren et al. [21] adopted KNN outlier detection algorithm to select data with locating in the central region. Based on the selected data, a better intrusion detection model was obtained. The main problem of cluster methods and outlier detection methods is high time complexity. Therefore, this paper proposes an improved KNN outlier detection algorithm to reduce the time complexity and further improve the performance of anomaly detection.

Considering the high-dimension character of network traffic, feature selection technique is used in the field of anomaly detection [22]. Feature selection technology can greatly reduce time consumption, and improve or at least maintain the anomaly detection effect [23-25]. Feature selection technology generally consists of two categories: classifier independent method (filtering) and classifier dependent method (packaged and embedded) [23]. Filtering method mainly adopts statistics and information theory methods, such as principal component analysis, information gain and mutual information [26]. Aljawarneh et al. [27] adopted information gain as an evaluation index to select 8 features from 41 features of the NSL-KDD dataset as the final features subset. The defect of this method is that it overestimates the importance of some features with more different values so that to select redundant and irrelevant features [28]. The classifier dependent method compensates for this defect, which adopts machine learning algorithms to evaluate the importance of features [18]. Chaouki et al. [29] utilized logistic regression as the evaluation model, and genetic algorithm as the search strategy to select the most relevant features subset with the smallest size. Tarfa et al. [30] proposed a recursive feature selection method based on SVM, in which the variation of cost function was used as the evaluation criterion of feature importance. The disadvantage of this method lies in the high computational cost. Hence, combining the advantages of the above two methods, this paper proposes a feature selection algorithm based on correlation analysis and recursive features addition strategy to select the most relevant features.

Based on the above analysis, this paper proposes a novel network anomaly detection method based on data balancing and recursive feature addition. Data balancing algorithm is designed based on an improved KNN outlier detection. Instead of the traditional K and M value of the KNN outlier detection algorithm, this algorithm proposes the distance and selection threshold to judge outliers and select items. Further, the genetic algorithm is utilized to optimize the threshold parameters of majority categories. In addition, the improved KNN outlier detection algorithm introduces the perspective of mini batch to improve the time efficiency of outlier detection. Then, recursive feature addition method based on correlation analysis is proposed. In correlation analysis, considering the difference between discrete features and continuous features, the cross contingency test including two coefficients is carried out to obtain the correlations of features and the features subset with a strong correlation. Based on the features subset, a recursive feature addition algorithm is proposed to select the optimal features subset with a high average recall rate. On the basis of the above analysis, random forests classifier is as anomaly detection classifier to detect anomalies. On the popular benchmark datasets, the accuracy, false alarm rate and recall of the proposed algorithm are analyzed and verified by comparing with other algorithms.

The structure of this paper is as follows. Section 2 briefly introduces the background technologies. Section 3 describes the network anomaly detection method proposed in this paper. Section 4 analyzes datasets and experimental results in detail. Section 5 summarizes the work of this paper.

2. Background Technology

2.1 Random Forests Model

Random forests (RF) were firstly proposed by Leo [24] as a classification algorithm, and are widely used in intrusion detection, computational biology, and other aspects. RF algorithm is a combinatorial classifier with the decision tree as the basic classifier. The basic idea of this model is: a forest contains multiple decision trees, and each decision tree is constructed from a random sample with putting back, that is to say, in the overall training dataset of one tree, some samples may appear many times, and also may never appear. Every decision tree is fully grown without any pruning, and the final output is the result of the majority voting of all the decision trees. The algorithm mainly consists of three steps:

(1) Select the training dataset. Bootstrap sampling method is adopted in random forests. Given dataset D containing m samples, a sample is randomly selected from D, and then the sample is put back. This sample may be acquired again in the next sampling. Though m times of sampling with replacement, and a dataset which is the same volume with the original dataset is achieved. In this dataset, some samples may be repeated, and some samples may never be collected. If constructing T basic classifiers, T new datasets are required.

(2) Construct random forests. Assuming that there are currently d features, $k<d$ features are randomly selected as the classification features set. By CART method, construct a single decision tree, and find the optimal features in the k features for splitting. Each decision tree is generated without limiting and any pruning. Parameter k controls the randomness of features. When $k=d$, it is the same as the ordinary decision tree. When $k=1$, a feature is randomly selected, and $k=\log_2 d$ is recommended.

(3) Vote. The random forests classification utilizes the majority voting method to make decisions. Assuming a single decision tree h_i makes prediction on multiple categories $\{c_1,c_2,\ldots,c_N\}$. The prediction of sample x in h_i is an N-dimensional vector $(h_i^1(x); h_i^2(x); \ldots; h_i^N(x))$, where $h_i^j(x)$ represents the predicted value of h_i in category c_i, and the voting method is expressed as $H(x)=c_{\arg\max_j \sum_{i=1}^{T} h_i^j(x)}$.

The advantage of this algorithm is that it is an integrated learning method. For any type of data, the random forests algorithm can construct more complex classification rules and products a better classification effect than most other algorithms [31]. It can also effectively process high-dimensional data. In addition, this algorithm is insensitive to parameter settings and can be easily adjusted to an appropriate model. Therefore, random forests classifier is more suitable for anomaly detection datasets with large volume and multiple categories.

2.2 Genetic Algorithm

Genetic Algorithm is a method to search for the optimal solutions by simulating the natural evolution process, which simulates the reproduction, hybridization and mutation in natural selection, and natural genetic process [29]. This algorithm has been widely used in combinatorial optimization, machine learning, signal processing, adaptive control and artificial life and other fields, mainly including two aspects: coding and genetic operation.

Coding mainly refers to the coding of the variables of the solution, which can be divided into binary coding and real coding. In the genetic algorithm, a solution is represented by the numeric string, and the genetic operator also directly works on the numeric string. Genetic operation includes three basic genetic operators: selection, crossover, and mutation. Selection and crossover basically complete the search function of genetic algorithm. Mutation increases

the ability to find the optimal solution. The basic operation process of genetic algorithm includes the following aspects:

(1) Initialization. Set the evolution counter t=0 and the maximum evolution t. Randomly generate M individuals as the initial population $P(0)$.

(2) Individual evaluation. Fitness of each individual in population $P(t)$ is calculated.

(3) Selection operation. The purpose of selection is to pass on optimized individuals directly to the next generation or to pass on new individuals to the next generation through the crossover. Selection operations are based on the fitness assessment of individuals in a population.

(4) Crossover operation. Crossover refers to the operation of replacing and recombining parts of two parents to generate new individuals. This operation is carried out by randomly selecting two individuals in the matching library according to a certain crossover probability generally ranging from 0.6 to 0.9, and the crossover position is also random.

(5) Mutation operation. Mutation means to randomly change the values of certain genes of individuals in a population with a small probability P_m<0.5 generally. The basic process of mutation operation is to generate a random number *rand* between [0,1]. If *rand*<P_m, mutation operation is carried out.

(6) Termination condition judgment. If t=T, the individual with the maximum fitness will be output as the optimal solution and the calculation will be terminated.

The advantages of the genetic algorithm are as follows. The algorithm has a good global search ability, and can search all solutions in solution space quickly without falling into the trap of local optimal solution. However, the genetic algorithm is prone to premature convergence. It is always a difficult problem in the genetic algorithm to preserve the good individuals and maintain the diversity of the population.

3. Network Anomaly Detection Method

In the high-speed network, the traffic data generated by different users, applications and devices is massive. Meanwhile, in the mass network traffic, most traffic data are produced by normal network behaviors, and only a few are caused by attack or anomaly. In these slight abnormal data, the volume of different attacks is very different due to diversity of attack modes. For example, DoS attack generates a large amount of traffic, while R2L attack just produces a small amount of traffic. It is worth noting that these small categories of attacks tend to be more harmful. Therefore, the unbalanced of different categories shows great challenges and significance to anomaly detection. In addition, for better reflecting the differences between various traffic categories, high-dimensional features of network traffic are generated. However, previous researches have proved that excessive features or too few features are not conducive to anomaly detection. Therefore, selecting effective features is helpful to improve anomaly detection performance. Based on the above analysis and the characteristics of network traffic, this paper proposes a novel network anomaly detection method. The overall process is shown in **Fig. 1**.

3.1 Data Balancing Algorithm based on Improved KNN Outlier Detection

In this section, data balancing algorithm based on improved KNN outlier detection is proposed to select data. It is worth noting that this method is not applied to the whole traffic data, but to each category of traffic data respectively. Outliers in each category are deleted, and some

representative groups with concentrated distribution and high density are selected to represent this category.

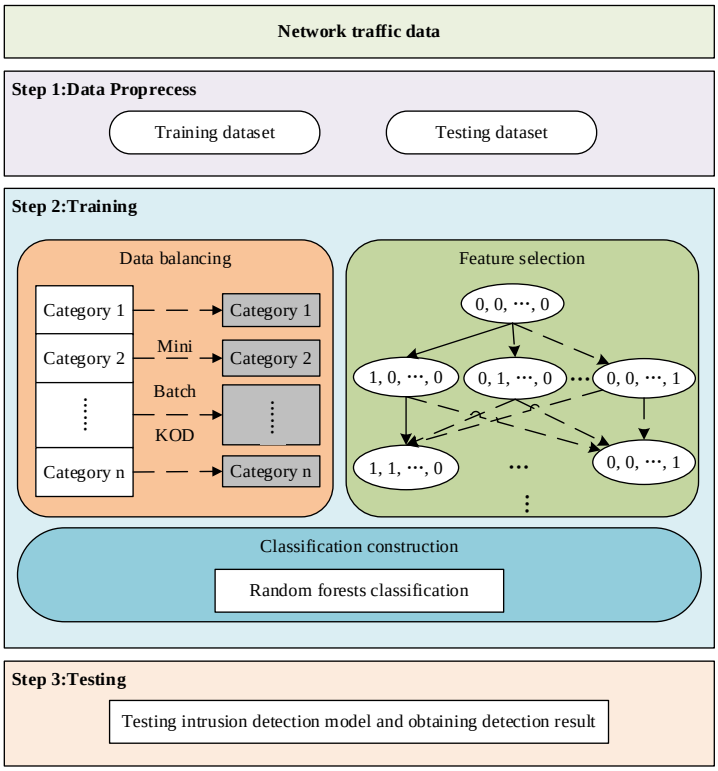


Fig. 1. The overall flow of the proposed anomaly detection method based on data balancing and recursive feature addition

KNN outlier detection algorithm (KOD) is a distance-based outlier detection algorithm, which was firstly proposed by Knorr et al [21] in 1998. This method is relatively simple and easy-to-use on the basis of the KNN algorithm. The core concept of KOD is to calculate the average k-nearest distance between each item and other items in dataset *D*, and sort the average k-nearest distance in descending order. The top *N* items with the largest distance are considered as outliers. Outliers are identified as items with sparse distribution and far away from high-density groups. Similar to the KNN algorithm, this algorithm has a large dependence on the *K* value, and too large or too small *K* value unable to judge outliers. For this problem, the distance threshold *Thred* is designed instead of *K* value, which referred to the literature [21], and *Thred* is set to 0.5.

Another improvement of the KNN outlier detection algorithm is the introduction of the mini batch. KNN outlier detection algorithm is similar to other outlier detection algorithms (for example, distance-based and density-based outlier detection algorithms), which calculate the distance or density between each item and all other items. In massive traffic, this causes a mass of time consume. Therefore, the mini batch method is introduced into the process of distance calculation. The advantage of mini batch is that instead of all samples, a part of samples are extracted to calculate distance. The running time of this algorithm will be greatly reduced due to the reduction of calculation samples. In addition, in the subsequent data selection processing, the selection threshold *THRED* is proposed to select representative data.

In short, the Data balancing algorithm based on improved KNN outlier detection is improved in the following aspects: (1) distance threshold and selection threshold are designed; (2) the viewpoint of mini batch is introduced into the outlier detection algorithm. The detailed process of this algorithm is shown in algorithm 1.

Algorithm 1. Data balancing algorithm based on improved KNN outlier detection
Input： dataset *D*, *Thred*, *THRED*
Output： new Dataset *D'*
(1) Standardize dataset *D* to obtain *ND*
(2) *L=D.length*; // the length of dataset *D*
(3) *Index* = *Ø*; // Index stores the index value of the selected data
(4) For (*i*=0; *i*<*L*; *i*++)
(5) *RD* = *RandomSample*(*ND*);
(6) *m* = *RD.length*;
(7) *d* = 0;
(8) *sum_d* = 0;
(9) For (*j*=0; *j*<*m*; *j*++)
(10) Calculate Euclidean distance *d* between *ND*(*i*) and *ND*(*j*);
(11) If (*d*<=*Thred*)
(12) *sum_d* = *sum_d* + *d*;
(13) End For
(14) if (*sum_d*>=*THRED*)
(15) *Index.add*(*i*);
(16) End For
(17) According to *Index*, obtain new dataset *D'*=*D*(*Index*);

In algorithm 1, it's worth noting that before calculating distance, the data need to be standardized. But the output dataset of this algorithm is a part of the raw dataset, not the standardized dataset as line (17). Line (1) performs 0-1 normalization. Lines (4)-(13) calculate the Euclidean distance *d* between each item and other items, and judges whether the distance *d* is less than the threshold *Thred*. Add the distance *d* with less than *Thred* to get *sum_d*. In lines (14)-(15), compare the threshold *THRED* with *sum_d*. If *sum_d* > *THRED*, *D*(*i*) is selected. This indicates that this item is close to other data and lies in a concentrated and high density area. That means this item is not an outlier. Otherwise, the item is considered to be an outlier. To better illustrate this problem, **Fig. 2** shows the differences between outliers and non-outliers in detail.

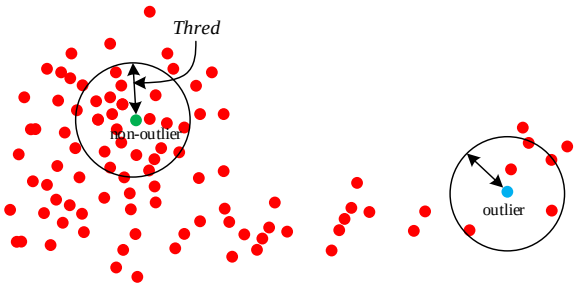


Fig. 2. The comparison between outliers and non-outliers

Since the data volume of different categories is greatly different in network traffic, the distance sum *sum_d* with satisfying the condition $d<Thred$=0.5 is also greatly diverse. Therefore, the *THRED* value of different categories is also discrepant. In order to select the appropriate *THRED*, this paper adopts the genetic algorithm to implement a combination optimization of parameters *THRED*. The specific optimization process is shown in algorithm 2.

Algorithm 2. Parameters combination optimization algorithm based on genetic algorithm.

Input: training dataset *Train*， {sum_d_1, sum_d_2,…, sum_d_l}, testing dataset *Test*, iteration number G， category number l

Output: Set of optimal parameters

(1) Initialize Up ={$max(sum\_d_1)$, $max(sum\_d_2)$,…, $max(sum\_d_l)$}, Low = {$min(sum\_d_1)$, $min(sum\_d_2)$,…, $min(sum\_d_l)$}

(2) P = Random_Generate_Initial_Population(N, Up, Low)

(3) For i = 1:G do

(4) Calculate Fitness of P

(5) Save_Best_Chromosome(P)

(6) though the roulette wheel method to select the new generation P_{new} in P

(7) P_{new} = $Crossover(P_{new})$

(8) P_{new} = Mutation(P_{new})

(9) $P = P_{new}$

(10) End For

Produce 1. Crossover(P, l)

(1) For i=1:2:$length(P)$

(2) $r = rand(1,l)$

(3) exchange $P[i, r:end]$ with $P[i+1, r:end]$ to obtain $P[i]$ with $P[i+1]$

(4) End For

Produce 2. Mutation(P, l)

(1) For i=1:$length(P)$

(2) $r = rand(1,l)$

(3) $r\_prob = rand(0,1)$

(4) If $r\_prob>0.5$ $P[i, r] = P[i, r]+rand$

(5) Else $P[i, r] = P[i, r]-rand$

(6) End If

(7) End For

In algorithm 2, before the selection and optimization of parameters *THRED*, the distance vector sum_d_i of all categories is calculated, which avoids repeated calculation of sum_d_i and reduces calculation time. Chromosome coding adopts floating coding. Although binary coding is the more conventional coding, the real coding in this method is more practical, which avoids frequent conversions between binary and decimal. Each chromosome $X=\{r_1, r_2,..., r_l\}$ represents a solution, l represents the number of categories, r_i represents the threshold *THRED* of each category, $r_i \in [min(sum\_d_i), max(sum\_d_i)]$. After randomly generating N chromosomes as the initial population, the fitness value of each chromosome is calculated. According to the

fitness value, selection, crossover and mutation operations are carried out until the maximum iteration number. In this method, in order to enhance the recall of each category, the fitness value refers to the sum of the recall rate of each category, and the formula is as follows.

$$sum\,\mathrm{R} = \sum_{i=1}^{l} \mathrm{Re}\,call_i, \tag{1}$$

$$\mathrm{Re}\,call_i = h_{ii} / \sum_{j=1}^{l} h_{ij}, \tag{2}$$

Where, h_{ii} refers to the number of class i correctly identified as class i, and h_{ij} refers to the number of class i wrongly identified as class j.

The result of the fitness value will be used in the following selection operation. The selection operation utilizes the roulette wheel method, and the higher the fitness of chromosomes, the greater the chance to be selected into the next generation. In the new generation population, crossover operation is executed as Produce 1. Make a pair to be crossed, and an intersection point is randomly selected. The chromosomes of this pair from the intersection point to the end point are exchanged. Carry out the mutation operation as Produce 2. Select a mutation point, and this gene value has a 0.5 probability to increase a random value or decrease a random value.

3.2 Recursive feature addition based on correlation analysis

This section demonstrates the recursive feature addition algorithm based on correlation analysis in detail. The algorithm consists of two phases: feature correlation analysis and recursive feature addition algorithm. In the phase of feature correlation analysis, a cross contingency test is applied to describe the correlation between feature and category. After calculating the correlation coefficient, the features with correlation coefficient less than 0.3 are deleted to obtain a features subset with a strong correlation. This feature subset is utilized to the following recursive feature addition algorithm, which adds one feature at one time to make the selected features subset achieve the optimal classification effect. The combination between feature correlation analysis and recursive feature adding algorithm could guarantee the detection effect and reduce the computational burden of the single recursive feature addition algorithm.

In the phase of correlation analysis, the cross contingency test applies different correlation indexes for different feature types (the symbolic feature and numerical feature). The correlation between symbolic features and categories is evaluated by *V* coefficient. *T*he calculation formula of *V* coefficient is shown in (3).

$$V = \sqrt{\frac{\chi^2}{N(K-1)}}, \tag{3}$$

Where, χ^2 is chi-square statistic, N denotes the number of samples, and K denotes the smaller value of the actual number of rows and columns in contingency table.

The correlation between numerical features and categories is evaluated by *Eta* coefficient, and the specific calculation formula is shown as follows.

$$Eta^2 = \frac{SS_b}{SS_t} = \frac{SS_t - SS_w}{SS_t} \tag{4}$$

Where, SS_b represents the difference between groups, $SS_t = \sum (y - \overline{y})^2$ represents the overall difference, $SS_w = \sum (y - \overline{y_i})^2$ represents the difference within the group, $\overline{y}$

represents the mean of all the numerical variables, $\overline{y_i}$ represents the mean of all the numerical variables corresponding to the nominal variable *i*.

To sum up, the equation of *CC* (correlation coefficient) between features and category is expressed as follows:

$$CC = \begin{cases} V & \text{f} \in symbolic\ features \\ Eta & \text{f} \in numerical\ features \end{cases}. \tag{5}$$

When *CC*<0.3, the correlation is weak; When *CC*>0.6, the correlation was strong. Therefore, the features subset with strong correlation is obtained according to the correlation coefficient. This features subset is used into the following recursive feature addition algorithm.

Recursive feature addition algorithm is a kind of packaged feature selection algorithm, which combines with classification model and evaluation indexes. Recursive feature addition is a greedy search strategy for feature selection. In this method, an empty feature set is initialized and new features are added to it step by step. If one feature can improve the evaluation effect to the greatest extent, it is selected. Otherwise, thrown away. In this paper, the random forest is applied as a classifier and the average recall rate is as an evaluation index. With the increase of features subset, the average recall rate of selected features is analyzed. When the average recall rate does not increase, it indicates that the selected features subset has the optimal classification effect, and this features subset is the final selected features subset. The detailed algorithm process is shown in algorithm 3.

Algorithm 3. Feature selection algorithm based on correlation analysis
Input: *Training dataset*, *Testing dataset*, feature set *F*, the size of feature set *N*
Output: The selected features set *SF*
(1) Calculate *CC* between features and categories
(2) Get a new features sub-set *new_F* by deleting features of *CC*≤0.3
(3) Set an empty selected features set *F'*
(4) Set an empty max *Avg_Recall* set *R*
(5) For *i* = 1:*N* do
(6) For each $f \in new\_F$ do
(7) Calculate *Avg_Recall* of $\{F' \cup f\}$ with Random Forests Algorithm
(8) End For
(9) Get *f* of *max Avg_Recall*
(10) *F'.add(f)*
(11) *R.add(max Avg_Recall)*
(12) *new_F.remove(f)*
(13) End For
(14) While $R_i > R_{i-1}$ do
(15) *SF. add(F'$_i$)*
(16) End While

In algorithm 3, the average recall index (*Avg_Recall)* refers to the average value of recall rate of all categories. The average recall rate index is put forward to effectively evaluate the detection rate of each category, especially the detection rate of small categories. The average recall rate is calculated as shown in (6).

$$Avg\_\text{Re}call = \sum_{i=1}^{l} \text{Re}call_i / l. \tag{6}$$

The final result of this method is to obtain a features subset with the highest average recall rate. In this method, the classification effect of individual feature is not the focus, but the classification performance of features set or associated features. The main goal of this method is to eliminate feature redundancy, ensure the relevance between features and categories, and achieve the optimal classification effect of features subset.

In order to better illustrate the recursion features addition algorithm, an example is given in **Fig. 3**. Assume that the data has three features $\{f_1, f_2, f_3\}$, and each feature has two states 0 and 1, in which 0 indicates that the feature has not been selected, and 1 indicates that the feature has been selected. At the initial state, all three features are not selected, and it is shown as {0, 0, 0}. Next, the *Avg_Recall* of each feature is calculated. The *Avg_Recall* of feature f_2 is highest, and feature f_2 is selected. Then, on the basis of feature f_2, the *Avg_Recall* of features subset $\{f_1, f_2\}$ and $\{f_2, f_3\}$ is calculated respectively, and the *Avg_Recall* of $\{f_1, f_2\}$ is the highest, and feature f_1 is selected. Finally, all features are selected. In figure 3, the solid lines show the process of feature selection, and the result of feature selection is $\{f_2, f_1, f_3\}$. Notice that with the increasing of selected features, the *Avg_Recall* is not continuously improved, that is to say, the *Avg_Recall* of features subset $\{f_2\}$ is not necessarily lower than that of features subset $\{f_1, f_2\}$.

3.3 Algorithm Description

This section will describe the overall process of the proposed algorithm in detail, which mainly includes two stages: offline training stage and online detecting stage. In the training stage, anomaly detection performance is the main evaluation index. However, in the testing stage, time and computation efficiency are very important. Therefore, this paper improves the detection precision through data balancing and feature selection in the training stage, and adopts a simple RF model in the detecting stage to ensure the time efficiency. The detailed process is shown in algorithm 4.

Algorithm 4. Network anomaly detection algorithm based on data balancing and feature selection

Input: Training dataset *Train*, Testing dataset *Test*

Output: detection result *R*

(1) [*H*, *K*] = size(*Train*); // *H* is the row number of training dataset, and *K* is the column size of training dataset.

(2) Preprocessing *Train*, *Test*

(3) *l* = *unique*(*Train*(:,*K*)); // *l* is the category number of training dataset.

(4) Classify *Train* to $Train_1, Train_2, \ldots, Train_l$;

(5) According to Algorithm 1, calculate $\{sum\_d_1, sum\_d_2, \ldots, sum\_d_l\}$;

(6) According to Algorithm 2, get $[THRED_1, THRED_2, \ldots, THRED_l]$;

(7) For i = 1:*l*

(8) $newTrain_i = Train_i(sum\_d_i > THRED_i)$;

(9) End For

(10) $newTrain = [newTrain_1; newTrain_2; \ldots; newTrain_l]$;

(11) According to Algorithm 3, select features subset *SF*

(12) *model* = *classRF_train*(*newTrain*(:,*SF*),*newTrain*(:,*K*)); //train detection model

(13) *R* = *classRF_predict*(*Test*(:,*SF*), *model*); //test detection model

In algorithm 4, lines (1)-(4) is a process of data preprocessing. Lines (5)-(11) are the process of data balancing and feature selection. In line (12-13), random forest classifier is trained by the new training dataset, and tested by the testing dataset. Detection results are obtained.

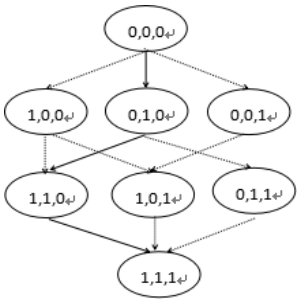


Fig. 3. The example of recursive feature addition algorithm

4. Experimental Results and Analysis

This section will evaluate the performance of the proposed method. All experiments are conducted in Windows 7 PC，Intel(R) Xeon(R) CPU E5-2603 0 @1.80GHz 1.80GHz and 8.00GB RAM. Matlab2017b is adopted to implement the algorithm in this paper.

4.1 Network Traffic Datasets

(1) KDD Cup 1999 dataset [32] is the benchmark dataset of the intrusion detection field, which is widely used in the research of the intrusion detection system. The dataset includes a total of 41 features and a category label (normal or 4 attack categories). And these features can be divided into 4 types: TCP connection basic features, TCP connection content features, time-based traffic statistical features, and host-based traffic statistical features, as shown in **Table 1**. The four attack types refer to DoS, Probe, U2R, and R2L [33]. KDD Cup 1999 dataset provides training and testing datasets.

(2) UNSW-NB15 dataset [34] was generated by Australian security laboratory using IXIA PerfectStrom tool, which combines real normal traffic and artificial attack traffic in modern networks. The dataset consists of 42 features and 1 category label (normal behavior and 9 attack categories) [35]. The features of this dataset are divided into 4 categories: basic features, content features, time features, and additional generated features. The detailed feature description is shown in **Table 2**. Attack categories conclude Fuzzers, Analysis, Backdoor, DoS, Exploit, Generic, Reconnaissance, Shellcode, and Worm.

Table 1. Features of KDD Cup 1999 dataset

Class	Attribute name
Basic features	duration(1), protocol_type(2), service(3), flag(4), src_bytes(5), dst_bytes(6), land(7), wrong_fragment(8), urgent(9)
Content features	hot(10), num_failed_logins(11), logged_in(12), num_compromised(13), root_shell(14), su_attempted(15), num_root(16), num_file_creations(17), num_shells(18), num_access_files(19), num_outbound_cmds(20), is_hot_login(21), is_guest_login(22)
Time based traffic features	count(23), srv_count(24), serror_rate(25), srv_serror_rate(26), rerror_rate(27), srv_rerror_rate(28), same_srv_rate(29), diff_srv_rate(30), srv_diff_host_rate(31)
Time based traffic features	dst_host_count(32), dst_host_srv_count(33), dst_host_same_srv_rate(34), dst_host_diff_srv_rate(35), dst_host_same_src_port_rate(36), dst_host_srv_diff_host_rate(37), dst_host_serror_rate(38), dst_host_srv_serror_rate(39), dst_host_rerror_rate(40), dst_host_srv_rerror_rate(41)

Table 2. Features of UNSW-NB15 dataset

Class	Attribute name
Basic features	dur(1), proto(2), service(3),state(4), spkts(5), dpkts(6), sbytes(7), dbytes(8), rate(9), sttl(10), dttl(11), sload(12), dload(13), sloss(14), dloss(15)
Content features	swin(16), dwin(17), stcpb(18), dtcpb(19), smean(20), dmean(21), trans_depth(22), res_bdy_len(23)
Time features	sintpkt(24), dintpkt(25), sjit(26), djit(27), tcprtt(28), synack(29), ackdat(30)
Addition features	ct_srv_src(31), ct_state_ttl(32), ct_dst_ltm(33), ct_src_dport_ltm(34), ct_dst_sport_ltm(35), ct_dst_src_ltm(36), is_ftp_login(37), ct_ftp_cmd(38), ct_flw_http_mthd(39), ct_src_ltm(40),ct_srv_dst(41),is_sm_ips_ports(42)

4.2 Dataset Preprocessing

Training and testing datasets are preprocessed before applied to anomaly detection methods. Data preprocessing consists of three steps: (1) the repeated instances of the training dataset are deleted; (2) the symbolic features are converted into numerical representations, such as these features of protocol, service, and state in the UNSW-NB15 dataset. Take the feature of state as an example, this feature consists of 11 variables, which are represented by values 1~11. (3) category labels are converted into numerical representations. For example, in the UNSW-NB15 dataset, 1 represents Normal category, 2 represents Backdoor category, and so on.

4.3 Experimental Evaluation Index

In addition to recall rate mentioned above, *Acc*(Accuracy), *DR*(Detection rate) and *FAR*(False alarm rate) are also adopted as evaluation indexes of anomaly detection methods [36].

$$Acc = (TP + TN) / (TP + TN + FP + FN), \tag{7}$$

$$DR = TP / (TP + FN), \tag{8}$$

$$FAR = FP / (TN + FP). \tag{9}$$

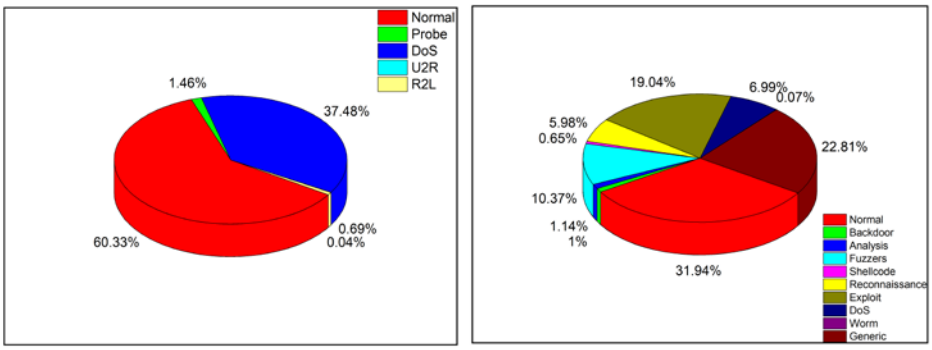
Where *TP*(True positive) and *TN*(true negative) refer that the attacks and normal behaviors are correctly classified, respectively. *FP*(false positive) represents that normal behaviors are incorrectly predicted attacks. *FN*(false negative) refers that attacks are incorrectly predicted normal behaviors.

4.4 Experiment Analysis

This section describes the experiment process and results in detail. Firstly, it shows the overall distribution of network traffic. Next, the time validity of the improved KNN outlier detection algorithm is analyzed, and the detection validity of the data balancing algorithm based on the improved KNN outlier detection algorithm is verified. Then, the impact of correlation analysis and the validity of the recursive feature addition algorithm based on correlation analysis is proved. Finally, we verify the extensibility of the proposed method and compare with other algorithms to further illustrate the effectiveness of our method.

A. Distribution analysis of network traffic

Experiment 1 visualized the data distribution of KDD Cup 1999 and UNSW-NB15 dataset through the pie chart as shown in **Fig. 4**. In **Fig. 4(a)**, the proportion of U2R and R2L types in KDD Cup 1999 dataset is only 0.03% and 0.6%. In **Fig. 4 (b)**, the proportion of Normal type is the largest, and Generic and Exploits types also occupy a large proportion. But the Worm, Shellcode, Backdoor and Analysis types account only about 0.1%, 0.6%, 1.1%, and 1.0%, respectively. Therefore, network traffic data is an extremely unbalanced dataset.



(a) (b)

Fig. 4. The overall distribution of network traffic (a) KDD Cup 1999 dataset (b) UNSW-NB15 dataset

B. Time validity analysis of improved KNN outlier detection algorithm

In this algorithm, mini batch was introduced to reduce the time complexity of the KNN outlier detection algorithm. In this paper, the proportion of random sampling was set to 0.1. Take KDD Cup 1999 dataset as an example as shown in **Fig. 5**. As can be seen from **Fig. 5**, when the amount of dataset is small, the effect of mini batch is not remarkable. With the increase of data volume, the time efficiency becomes more obvious. The amount of Normal category is 87832. After using the mini batch, the calculation time is saved about 3/4. Therefore, the improved KNN outlier detection algorithm can effectively reduce calculation time in the outlier detection process.

C. Validity analysis of data balancing algorithm based on improved KNN outlier detection

This experiment is used to illustrate the detection effectiveness of the data balancing algorithm based on the improved KNN outlier detection. It is worth noting that for many categories of small number, these categories are at a disadvantage in anomaly detection. Hence, outlier detection was not performed on these categories. In the KDD Cup 1999 dataset, the Normal, DoS and Probe categories are defined to detect outliers and select part respective data. The *THRED* parameters were set as {77.8148, 0.009, 12.5} by the combination optimization of genetic algorithm. The data volume before and after data balancing is shown in **Table 3**. From **Table 3**, the number of Normal category decreases more, while the number of DoS category decreases less. This is because the data distribution of DoS category is relatively tight and the outliers are less. **Fig. 6** shows the comparison of anomaly detection performance before and after data balancing. **Fig. 6** shows that the accuracy and detection rate increase slightly after data balancing, and the false alarm rate also increases slightly, about 2%, which is still a very good detection result. The recall rates of Normal and DoS categories remained basically unchanged. However, those of Probe, U2R and R2L categories increase significantly. This is because through the KNN outlier detection method, some data which locate the edge of the corresponding category and make confuse to classify are deleted. From Table 3, we can speculate that the Normal category is more scattered or closer to other categories. Therefore, the detection results of KDD Cup 1999 dataset show that this algorithm is effective in improving the anomaly detection performance.

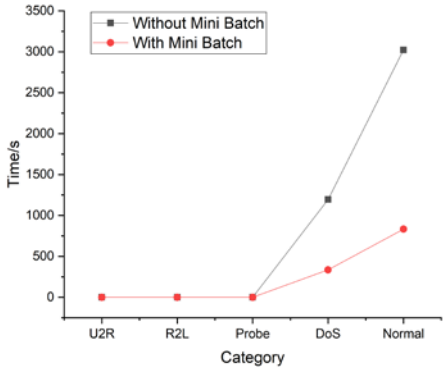


Fig. 5. Time analysis of improved the KNN outlier detection algorithm on KDD Cup 1999 dataset

Table 3. The number of each category before and after data balancing on KDD Cup 1999 dataset

Category	Number / Before balancing	Number / After balancing
Normal	87832	63378
DoS	54572	51649
Probe	2131	1983

In UNSW_NB15 dataset, data balancing algorithm is executed on the Normal, Fuzzers, Reconnaissance, Exploits, DoS and Generic categories, and *THRED* parameters were set as {7.41, 206.31, 49.07, 52.98, 50.66, 497.61}. The amount of various categories before and after the data balancing as shown in **Table 4**. From **Table 4**, the data becomes is more balanced. **Fig. 7** shows the comparison results of anomaly detection performance. The results showed that after data balancing, the false alarm rate decreased significantly. For the recall rate of each category, except for Fuzzers and DoS, the recall rates of other categories significantly increase or remain basically unchanged. Similarly, the detection results of UNSW_NB15 dataset also demonstrate the effectiveness of this algorithm.

In order to illustrate the effect of the proposed data balancing algorithm, compared with the random sample, taking UNSW_NB15 dataset as an example, the result is as shown **Fig. 8**. It can be seen the random sample algorithm also can enhance the classification effect. But our algorithm still has a better detection effect, especially in *FAR* and recall of Normal. It is worth to note that due to the randomness of the random sample, it is hard to obtain the distribution law and execute the further improvement.

Table 4. The number of each category before and after data sampling on UNSW_NB15 dataset

Category	Number / Before balancing	Number / After balancing
Normal	56000	52078
Fuzzers	18184	679
Reconnaissance	10491	8476
Exploits	33393	23331
DoS	12264	8814
Generic	40000	29255

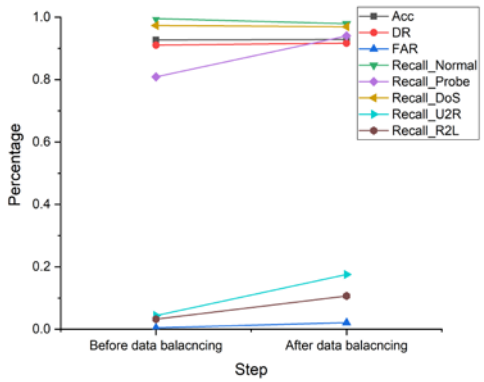


Fig. 6. Comparison of anomaly detection results before and after data balancing on KDD Cup 1999 dataset

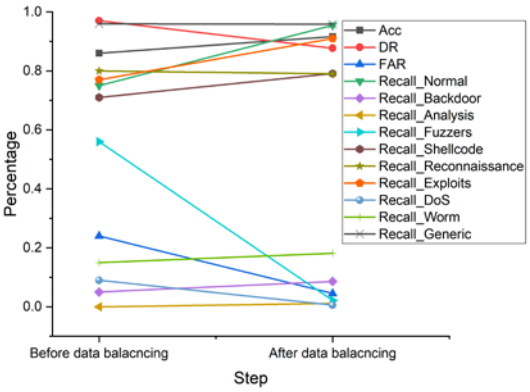


Fig. 7. Comparison of anomaly detection results before and after data balancing on UNSW_NB15 dataset

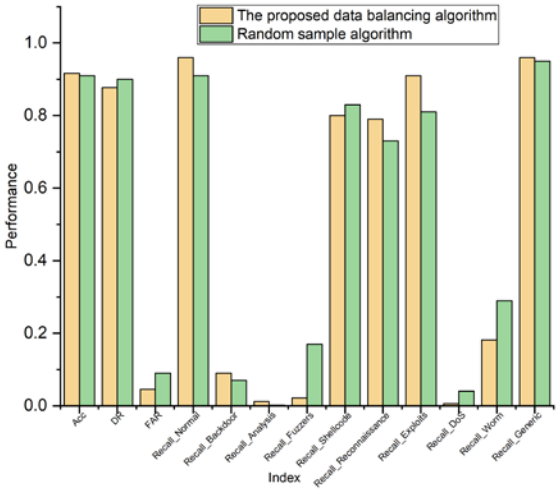


Fig. 8. Comparison between the proposed data balancing algorithm with random sample algorithm on UNSW_NB15 dataset

D. Validity analysis of recursive feature addition algorithm based on correlation analysis

In this experiment, the influence of correlation analysis is firstly illustrated by comparing anomaly detection results of the recursive feature addition algorithm before and after deleting the features subset with a small correlation, taking KDD Cup 1999 dataset as an example. In the KDD Cup 1999 dataset, correlation coefficients are shown in **Table 5**. Delete the features with $CC \leqslant 0.3$ to obtain a features subset with the strong correlation coefficient. The deleted features subset is {1,7,8,9,11,13,15,16,17,18,19,20,21,24}, and the remaining 27 features were utilized into the next recursive feature addition algorithm.

The result of recursive feature addition algorithm is shown in **Table 6**. In the original features set, and the number of selected features is 16. After correlation analysis, the number of selected features is 19. In **Fig. 9**, with the processing of recursive feature addition, the average recall rate gradually increases with the enhancing of selected features and then decreases after reaching the maximum value. In the original features set, and the maximum of the average recall rate was 0.696. After correlation analysis, the maximum value of the average recall rate was 0.691. The average recall rate is basically consistent. Therefore, deleting features with small correlation coefficient does not affect the overall detection

performance. Except that, we further analyzed the influence of correlation analysis on time efficiency. Recursive feature addition method is carried out on the original features set, and the time consumption was about 12000s. However, it took about 6000s on the features subset with a strong correlation, and the negligible time of correlation analysis is ignored. It can be seen that the recursive feature addition algorithm based on correlation analysis greatly improves time efficiency.

Table 5. The correlation coefficient (*CC*) on KDD Cup 1999 dataset

Feature	*CC*	Feature	*CC*	Feature	*CC*	Feature	*CC*	Feature	*CC*
29	0.984	3	0.728	4	0.536	18	**0.237**	15	**0.032**
34	0.965	35	0.725	35	0.517	17	**0.219**	8	**0.013**
33	0.959	30	0.666	6	0.51	11	**0.202**	7	**0**
12	0.932	36	0.653	2	0.422	13	**0.179**	20	**0**
23	0.837	5	0.649	27	0.374	1	**0.147**	21	**0**
25	0.837	14	0.638	40	0.367	24	**0.123**		
38	0.837	32	0.58	28	0.366	16	**0.075**		
26	0.835	22	0.559	41	0.365	9	**0.068**		
39	0.835	10	0.555	31	0.333	19	**0.05**		

Table 6. Features subset selected by recursive feature addition algorithm without and with correlation analysis

Item	Selected features subset
Without correlation analysis	{6,27,23,3,12,30,8,17,37,40,13,31,2,32,18,5}
With correlation analysis	{6,27,23,3,12,30,25,29,26,40,38,39,2,32,14,28,4,5,34}

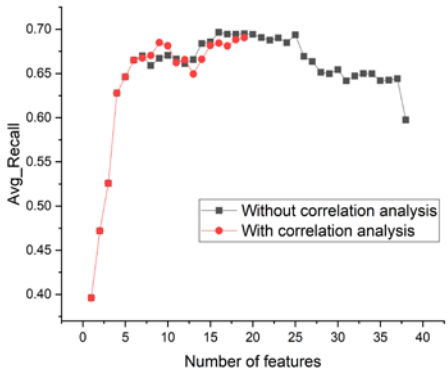


Fig. 9. Comparison of ave_recall of different features subset without and with correlation analysis

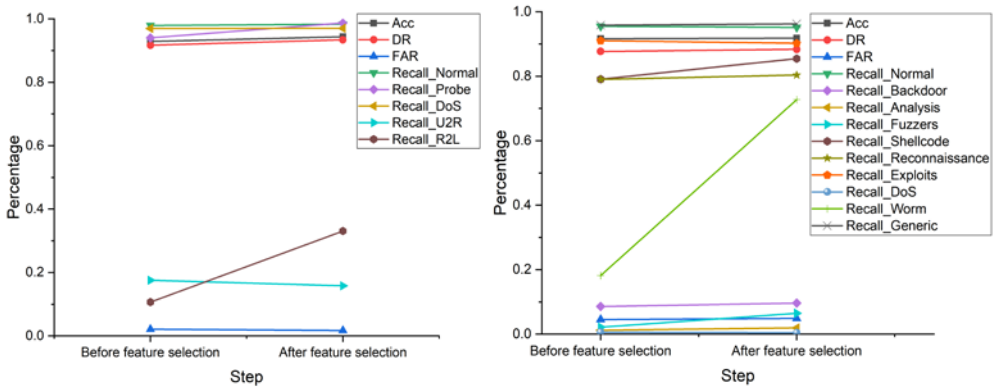
We conducted a deep analysis of the features subset selected eventually. The similarity analysis of feature subsets selected by the recursive feature addition algorithm before and after correlation analysis was carried out by using Jaccard coefficient and J=0.4. The feature subsets selected before and after correlation analysis have a great similarity, which indicates that some features have a large correlation and influence for the abnormal detection effect. In addition, it can be found that the features with a great effect on the detection effect do not have a significant correlation. On the contrary, some features have a strong correlation but do not show effective detection effects. Therefore, the initial screening of features is still a meaningful task and needs further improvement and analysis.

This part will analyze the detection performance of the recursive feature addition algorithm based on correlation analysis. On the KDD Cup 1999 dataset, the detection performance

comparison before and after feature selection is shown in **Fig. 10**. After feature selection, the overall performance of anomaly detection is slightly improved, but the recall of R2L category is significantly improved. In UNSW_NB15 dataset, the optimal features subset {3,5,7,9,10,12,15,16,27,28,34,36} was obtained. The comparison of the overall performance of anomaly detection before and after feature selection is shown in **Fig. 10**. The results show that the overall detection results remain unchanged before and after feature selection, but the recall rates of Shellcode and Worm increases significantly. Therefore, according to the detection results of KDD Cup 1999 and UNSW_NB15 datasets, the recursive feature addition algorithm based on correlation analysis is effective.

E. The overall performance analysis of the proposed algorithm

This section demonstrates the final detection effect of the proposed algorithm in detail. **Fig. 11** shows the detection results of KDD Cup 1999 dataset and UNSW-NB15 dataset. The results showed that the overall effect of anomaly detection is slightly improved after two stages of data balancing and recursive feature addition, but the recall rates of Probe, U2R and R2L categories significantly increase. For UNSW-NB15 dataset, the change of accuracy and detection rate is not very significant, but the false alarm rate significantly reduces, which is caused by the significant increase of recall rates of Normal and other attack categories. Except for Fuzzers category, the recall rates of all categories have significantly increased. Therefore, for the overall performance of anomaly detection, the anomaly detection method proposed in this paper is very excellent.



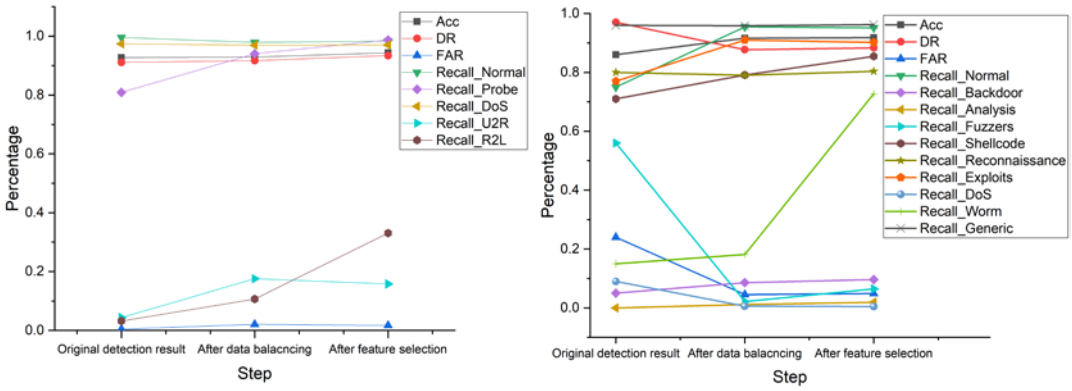
(a) (b)

Fig. 10. Comparison of anomaly detection results before and after feature selection (a) KDD Cup 1999 dataset (b) UNSW-NB15 dataset

F. The multi-aspect analysis of this algorithm

This part mainly contains three aspects: the scalability analysis, the time effectiveness of the test stage, and the computation effectiveness of the test stage. Firstly, in order to illustrate the scalability of the proposed algorithm, this algorithm was checked on testing datasets of different sizes to judge whether the detection performance remains stable. Through randomly sampling without putting back, three testing datasets of different sizes (25% testing dataset, 50% testing dataset, and 100% testing dataset) were produced. Take KDD Cup 1999 dataset as an example, **Fig. 12** shows the detection effect of different testing datasets. As the testing datasets increases, the overall detection results remain changeless. And the recall rates remain stable or increased slightly in different categories. It shows that the algorithm proposed in this paper can adapt testing datasets of different sizes and has good scalability. Then, the time efficiency of the network anomaly detection method is shown in **Fig. 13**. The detection time is

only about 2.5s for 100% testing dataset. Hence, our method can efficiently detect network abnormal behaviors and meet real-time requirements. The memory consuming of the test stage in 25% testing dataset, 50% testing dataset, and 100% testing dataset are 0.98M, 1,25M, and 2.70M, respectively. We can see that the computation cast is very small for the online testing and helpful to detection intrusion on time. Hence, based on the above analysis, our algorithm is excellent in the scalability, the time effectiveness, and the computation effectiveness.



(a) (b)

Fig. 11. Anomaly detection results of the proposed algorithm (a) KDD Cup 1999 dataset (b) UNSW-NB15 dataset

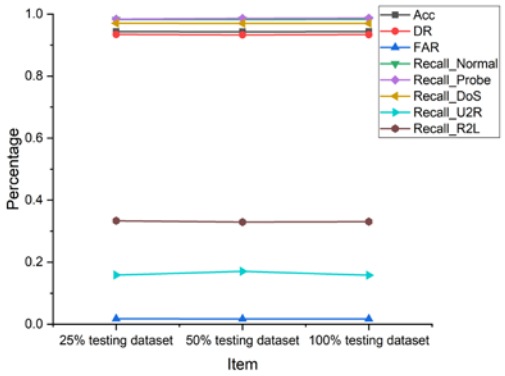


Fig. 12. Anomaly detection effects of the proposed algorithm on testing dataset of different sizes

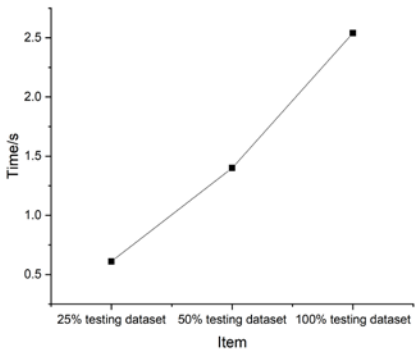


Fig. 13. Detection efficiency of the proposed algprithm

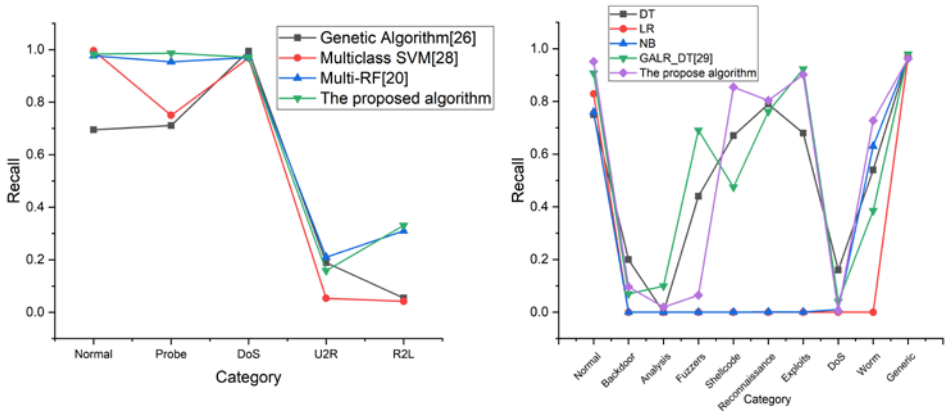
G. Comparative analysis with other algorithms

In order to better verify the proposed algorithm in this paper, the algorithm was compared with other algorithms on various evaluation indexes. **Table 7** and **Fig. 14** show the comparison results on the KDD Cup 1999 dataset. The results show that the proposed algorithm has the optimal accuracy and false alarm rate with only 19 features. Besides, our method has perfect recall rates, especially in Probe and R2L categories. For the UNSW_NB15 dataset, the overall detection effect of this algorithm and other algorithms are compared, as shown in **Table 8** and **Fig. 14**. The proposed algorithm has the best accuracy and false alarm rate, and obtains effective recall rates, especially in the types of Normal, Shellcode and Worm. However, the recall rate of Fuzzers has decreased significantly, which still needs further improvement. Based on the result of different classifiers, we find the tree classifier has a better effect on multi-class data. Compared with deep learning method (ICVAE-DNN[38]), our method is still better, which further illustrate the proposed strategies is effective in improving

the detection effects. In short, compared with other algorithms, this algorithm still has a preferable detection performance.

Table 7. Comparison of anomaly detection effect between the proposed algorithm and other algorithms on KDD Cup 1999 dataset

Method	Number of features	Classifier	*Acc*	*FAR*
Genetic Algorithm [26]	41	DT	0.8142	0.0639
Multiclass SVM [28]	41	RF	0.906	0.072
Multi-RF [21]	41	RF	0.94	0.0234
GAODNN[37]	41	NDNN	0.9323	0.0275
The proposed algorithm	19	RF	**0.9433**	**0.0171**



(a) (b)

Fig. 14. Comparison of Recallbetween the proposed algorithm and other algorithms(a) KDD Cup 1999 dataset (b) UNSW-NB15 dataset

Table 8. Comparison of anomaly detection effect between the proposed algorithm and other algorithms on UNSW-NB15 dataset

Method	Number of features	Classifier	*Acc*	*FAR*
Moustafa and Slay[35]	42	DT	0.8455	0.2491
		LR	0.6933	0.1708
		NB	0.6956	0.2332
		ANN	0.8134	0.2113
		EM	0.7847	0.2379
GALR_DT [29]	20	DT	0.8142	0.0639
ICVAE-DNN[38]	42	DNN	0.8908	0.1801
The proposed algorithm	12	RF	**0.9178**	**0.0487**

5. Conclusion

In order to detect network anomaly behaviors more effectively, this paper designs data balancing algorithm based on improved KNN outlier detection and recursive feature addition algorithm based on correlation analysis to achieve network anomaly detection. The improvement of the data balancing algorithm is mainly reflected in the following two aspects:(1) the distance threshold and selection threshold are designed, and the threshold parameters are set by combination optimization of genetic algorithm. (2) mini batch is introduced to the process of outliers detection. These two improvements contribute to select more respective data with less time complexity. Through KDD Cup 1999 and UNSW-NB15 datasets, experimental results have illustrated that the data balancing algorithm based on

improved KNN outlier detection can reduce the degree of unbalance within data categories, so as to significantly improve the accuracy of anomaly detection, reduce the false alarm rate, and improve the recall rates of different data categories to some extent. The recursive feature addition method based on correlation analysis combines classifier independent feature selection method and classifier dependent feature selection method to select the optimal features subset. Experiment results show that this algorithm can further improve the time consuming of features selection and the performance of anomaly detection, especially in the recall of some categories. Furthermore, the extensibility of the proposed method is analyzed on multiple datasets of different sizes. The result indicates that the proposed algorithm has excellent extensibility. Compared with other algorithms, this method has verified with high accuracy, low false alarm rate, and good recall rates. Future research work will further solve the problem of unbalanced intrusion detection dataset to improve the recall rates of different categories more effectively.

References

[1] H. Pajouh, G. Dastghaibyfard and S. Hashemi, "Two-tier network anomaly detection model: a machine learning approach," *Journal of Intelligent Information Systems*, vol. 48, no. 1, pp.61-74, 2017. Article (CrossRef Link)

[2] A. Amaral, L. Mendes, B. Zarpelao and M. Junior, "Deep IP flow inspection to detect beyond network anomalies," *Computer Communications*, vol. 98, pp.80-96, January, 2017. Article (CrossRef Link)

[3] CF. Tsai, YF. Hsu, CY. Lin and WY. Lin, "Intrusion detection by machine learning: A review," *Expert Systems with Application*, vol. 36, no. 10, pp.11994-12000, December, 2009. Article (CrossRef Link)

[4] YY. Chung and N. Wahid, "A hybrid network intrusion detection system using simplified swarm optimization(SSO)," *Applied Soft Computing Journal*, vol. 12, no. 9, pp.3014-3022, September, 2012. Article (CrossRef Link)

[5] B. Jain, "Intrusion prevention and vulnerability assessment in BCEFHP intrusion detection system [Master.dissertation]," *Indian Institute of Technology, Kanpur*, 2005.

[6] D. Kwon, H. Kim, J. Kim, S. Suh and K. Jim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, vol. 22, pp. 949-961, 2019. Article (CrossRef Link)

[7] 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, October, 2005. Article (CrossRef Link)

[8] P. Sangmee, N. Thanon and N. Elz, "Anomaly detection using new MIB traffic parameters based on profile," in *Proc. of Computing Technology and Information Management*, pp. 648-653, April 24-26, 2012.

[9] M. Ahmed, AN. Mahmood and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network & Computer Applications*, vol. 60, pp.19-31, January, 2016. Article (CrossRef Link)

[10] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández and E. Vázquez, "Anomaly-based network intrusion detection: techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1, pp.18-28, Febuary, 2009. Article (CrossRef Link)

[11] AA. Ghorbani, W. Lu and M. Tavallaee, "Network intrusion detection and prevention: concepts and techniques," *Advances in Information Security*, vol. 28, no. 3, pp.42-48, 2012.

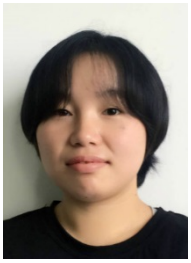
[12] C. Modi, D. Patel and et al., "Review: A survey of intrusion detection techniques in Cloud," *Journal of Network & Computer Applications*, vol. 36, no. 1, pp. 42-57, January, 2013. Article (CrossRef Link)

[13] W. N. Lin, M. Z. Chen, Y. Q. Zhan, and C.B. Liu, "Research on an intrusion detection algorithm based on PCA and Random-forest classification," *Netinfo Security*, vol. 11, pp.50-54, 2017. Article (CrossRef Link)

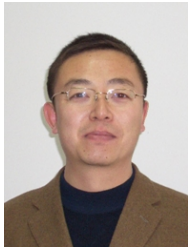
[14] CF. Tsai, YF. Hsu, CY. Lin and WY. Lin, "Intrusion detection by machine learning: a review," *Expert Systems with Applications*, vol. 36, no. 10, pp. 11994-12000, December, 2009. Article (CrossRef Link)

[15] AA. Aburomman and M.B. Reaz, "A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems," *Information Sciences*, vol. 414, pp.225-246, 2017. Article (CrossRef Link)
[16] U. Fiore, F. Palmieri, A. Castiglione and AD. Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp.13-23, December, 2013. Article (CrossRef Link)
[17] J.D. Ren, J.W. Guo and et al., "Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithm," *Security and Communication Networks*, vol. 2019, pp. 1-11, June, 2019. Article (CrossRef Link)
[18] W. Al-Yaseen, Z. Othman and M. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp.296-303, January, 2017. Article (CrossRef Link)
[19] W. Al-Yaseen, Z. Othman and M. Nazri, "Intrusion detection system based on modified k-means and multi-level support vector machines," in *Proc. of International Conference on Soft Computing in Data Science*, pp. 265-274, 2015. Article (CrossRef Link)
[20] W. Al-Yaseen, Z. Othman and M. Nazri, "Hybrid modified k-means with C4.5 for intrusion detection systems in mul-tiagent systems," *Scientific world journal*, vol. 2015, no. 2, pp.294761, July, 2015. Article (CrossRef Link)
[21] J.D. Ren, X.Q. Liu, Q. Wang and et al., "Multi-layer intrusion detection method based on KNN outlier detection and random forest," *Journal of Computer Research and Development*, vol. 56, no. 03, pp. 566-575, 2019. Article (CrossRef Link)
[22] AS. Eesa, Z. Orman and AMA. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2670-2679, November, 2015. Article (CrossRef Link)
[23] M. Bennasar, Y. Hicks and R. Setchi, "Feature selection using joint mutual information maximisation," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8520-8532, December, 2015. Article (CrossRef Link)
[24] W. Gao, L. Hu, P. Zhang and F. Wang, "Feature selection by integrating two groups of feature evaluation criteria," *Expert Systems with Applications*, vol. 110, pp. 11-19, November, 2018. Article (CrossRef Link)
[25] R. Alshboul, F. Thabtah, N. Abdelhamid and M. Al-Diabat, "A visualization cybersecurity method based on features' dissimilarity," *Computers & Security*, vol. 77, pp. 289-303, Augest, 2018. Article (CrossRef Link)
[26] EDL. Hoz, EDL. Hoz, A. Ortiz, J. Ortega and AM. Alvarez, "Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps," *Knowledge-Based Systems*, vol. 71, pp. 322-338, November, 2014. Article (CrossRef Link)
[27] S. Aljawarneh, M. Aldwairi and MB. Yassein, "Anoma-ly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152-160, March, 2018. Article (CrossRef Link)
[28] G. Herman, B. Zhang, Y. Wang, G. Ye and F. Chen, "Mutual information-based method for selecting informative feature sets," *Pattern Recognition*, vol. 46, no. 12, pp. 3315-3327, December, 2013. Article (CrossRef Link)
[29] C. Khammassi and S. Krichen, "A GA-LR wrapper Approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255-277, 2017. Article (CrossRef Link)
[30] T. Hamed, R. Dara and SC. Kremer, "Network intrusion detection system based on recursive feature addition and bigram technique," *Computers & Security*, vol. 73, pp. 137-155, 2018. Article (CrossRef Link)
[31] B Leo, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001. Article (CrossRef Link)
[32] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999.
[33] J. Wu, W. Zhang and H. Ma. "Data analysis of KDDCUP99 dataset," *Computer Application and Software*, vol. 11, pp. 321-325, 2014. Article (CrossRef Link)
[34] https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets, 2015.
[35] N. Moustafa N and J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Systems Security*, vol. 25, no. 1, pp. 18-31, January, 2016. Article (CrossRef Link)

[36] MRG. Raman, N. Somu, K. Kirthivasan and et al., "An efficient intrusion detection system based on hypergraph genetic algorithm for parameter optimization and feature selection in support vector machine," *Knowledge-Based Systems*, vol. 134, pp. 1-12, October, 2017. Article (CrossRef Link)
[37] M. Tan; M. Peng and et al., "NDNN structure and parameter optimization based on GA and its application in intrusion detection," *Automation & Instrumentation*, vol. 10, pp. 14-18, 2019. Article (CrossRef Link)
[38] YQ. Yang, KF. Zheng and et al., "Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network," *Sensors*, vol. 19, pp. 2528, June, 2019. Article (CrossRef Link)



Xinqian Liu, born in 1992. She is currently pursuing the Ph.D. degree in computer science and technology of Yanshan University from 2015. Her research interests include social network analysis, network security, instruction detection and DDoS attack detection.



Jiadong Ren, born in 1967, PhD. He is a professor in the School of Information Science and Engineering, Yanshan University. His current research interests include data mining and software security.

Haitao He, born in 1968. PhD. He is a professor in the School of Information Science and Engineering, Yanshan University. Her current research interests include data mining and software security.

Qian Wang, born in 1977. PhD, lecture. She is currently a lecture in the School of Information Science and Engineering, Yanshan University. Her current research interests include data mining, network security and software security.

Shengting Sun, born in 1993. He is currently pursuing the master degree in software engineering of Yanshan University from 2018. His research interests include data mining, software security, network security, instruction detection and anomaly detection.