Intrusion Detection System Combined Enhanced Random Forest With Smote Algorithm

Wu Tao
Jiangsu University of Technology
Fan Honghui
Jiangsu University of Technology
Zhu HongJin (zhuhongjin@jsut.edu.c)
Jiangsu University of Technology
You CongZhe
Jiangsu University of Technology
Zhou HongYan
Jiangsu University of Technology
Huang XianZhen
Jiangsu University of Technology

Research

Keywords: Network intrusion detection, data imbalance, SMOTE algorithm, Enhanced random forest, similarity, NSL-KDD

Posted Date: March 5th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-270201/v1

Intrusion detection system combined enhanced random forest with SMOTE Algorithm

Tao Wu[1], Honghui Fan[2], HongJin Zhu[2, 3], CongZhe You[2], HongYan Zhou[1] and XianZhen Huang

[3]Correspondence: HongJin Zhu (zhuhongjin@jsut.edu.cn)
[2] School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China

Abstract: Network security is subject to malicious attacks from multiple sources, and intrusion detection systems (IDS) play a key role in maintaining network security. During the training of intrusion detection models, the detection results generally have relatively large false detection rates due to the shortage of training data caused by data imbalance. To address the existing sample imbalance problem, this paper proposed a network intrusion detection algorithm based on enhanced random forest and Synthetic Minority Over-Sampling Technique (SMOTE) algorithm. Firstly, the method used a hybrid algorithm combining the K-means clustering algorithm with the SMOTE sampling algorithm to increase the number of minor samples and thus achieved a balanced data set, by which the sample features of minor samples could be learned more effectively. Secondly, preliminary prediction result was obtained by using enhanced random forest, and then the similarity matrix of network attacks was used to correct the prediction results of voting processing by the analysis of the type of network attacks. In this paper, the performance was tested using the NSL-KDD dataset with a classification accuracy of 99.72% on the training set and 78.47% on the test set. Compared with other related papers, our method has some improvement in the classification accuracy of detection.

Keywords: Network intrusion detection, data imbalance, SMOTE algorithm, Enhanced random forest, similarity, NSL-KDD

1 Introduction

In this era of information explosion, the Internet has occupied a very important position in people's lives, it has enriched people's cultural lifestyle and changed the mode of our production and behavior. However, at some level, it also brings network security problems, network intrusion is more and more frequent, accompanied by the characteristics of large scale, high frequency, and many types. Network security issues are gradually becoming an important topic of concern for researchers, and the main responsibility of network intrusion detection systems is to detect and resolve threat attacks, which is an important way to defend against malicious threats to the network [1]. As a means to effectively circumvent intrusions, network intrusion detection has very strict requirements in terms of detection accuracy. In order to improve the accuracy of detection, many researchers have used optimization tools such as machine learning, feature selection [2]. It also includes the least-squares technique, kernel function methods, neural networks, population optimization algorithms. These optimization tools continue to improve the accuracy of intrusion detection. However, too much research currently overstays at the level of overall accuracy, and there is certain neglect for the detection of smaller-scale data. The imbalance of the data causes the detection model to have a high false alarm rate and a low detection rate for smaller-scale network attacks, so there is still a lot of research significance and room for improvement in the detection effect of minority class samples.

It is notable that using decision forests with poor decision performance, which negatively affects the final voting results and model predictions. In order to solve this problem, this paper proposed an intrusion network detection model based on enhanced random forest and SMOTE algorithm. In the first stage of data preprocessing, the SMOTE technique was employed to analyze the minority class samples and manually synthesize new samples based on the minority class

samples to add to the dataset, and it was further improved by using K-means to make the sample dataset more convergent to the cluster center. The decision tree with good classification performance in the random forest model was calculated and selected for similarity calculation in the second stage. Before generating a new random forest model, we analyzed the types of network attacks and corrected the prediction results of voting processing through reasonable use of the similarity matrix of network attacks. Finally, the enhanced random forest model was trained on the processed NSL-KDD dataset in this paper, and the detection effect achieved the desired results.

The remainder of this paper is organized as follows: In second part presents the related work. The third part presents the framework of the method and the relevant methodological mathematical definitions. The fourth part describes the evaluation criteria for the model and the analysis of the experimental results. The fifth part concludes the whole paper.

2 Related work

An intrusion detection system is an important research area of network security, attracting numerous researchers to improve and optimize the technology, a good detection system needs to have efficient and stable characteristics. At present, many researchers mainly implement detection research by using machine learning algorithms on the public dataset NSL-KDD to improve the detection of malicious network activities by intrusion detection systems in this way.

The development of intrusion detection systems needs to be traced back to 1986 when the research group of Dorothy E. Denning et al. successfully implemented the first intrusion detection model, and subsequent research had focused on feature extraction, classifier optimization, and data pre-processing [3]. Among them, the widely used classification algorithms mainly included Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (K-NN), and other classification algorithms.

Zhao et al. implemented the improvement and parameter optimization of the support vector machine algorithm by analyzing the traditional detection system, where the parameter optimization was achieved through the use of the particle swarm optimization (PSO) algorithm and the combination of the SVM algorithm and the hybrid kernel function [4].

S.J. Horng et al. tried to optimize the feature selection and their proposed detection model by combining hierarchical clustering algorithm with SVM thus achieving the classification detection function [5].

So that the detection model could be further optimized for detection accuracy and false alarm rate, numerous researchers had optimized the feature selection through research. Among them, Sumaiya et al. combined multi-class support vector machines with chi-square feature selection. And through a series of test experiments, the results showed that the bonding method could achieve some improvements compared with other studies [6]. Peng et al. tried to optimize the data type for the net attack by combining the Mini Batch -means combined with Principal Component Analysis (PCA) through the analysis of the clustering algorithm [7].

RM et al. tried to use the random forest and weighted K-means classifiers simultaneously through the analysis of the classification algorithm, and this new hybrid algorithm was tested on the KDDcup99 dataset to evaluate the model performance with good improvement results [8].

The classification optimization algorithms proposed by many scholars prompt us to try to further optimize the random forest classification algorithm to achieve the detection and classification of malicious attacks on the Internet. However, too many studies had focused on the overall detection accuracy and false alarm rate metrics, but they had neglected the imbalance between training data types, and the proportional differences between data samples had been

93 constantly affecting the detection performance, which led to a decrease in detection accuracy and
94 an increase in false alarm rate for fewer types of samples.

95 For this problem of data imbalance, many researchers had tried to solve this kind of problem
96 from the data itself by processing the proportion of training data types, mainly including over-
97 sampling, and under-sampling methods [9].

98 OFek et al. tried to achieve a fast clustering method by combining under-sampling techniques
99 with clustering algorithms by analyzing the original clustering algorithms [10]. Based on this, the
100 training results were weighted, and the algorithm achieved good results with certain applicability
101 and effectiveness in processing the binary classification problem of the dataset.

102 By analyzing the reinforcement learning algorithm and data imbalance, Ma Xiang-Yu et al.
103 used the ability of reinforcement learning auto-learning ability combined with SMOTE algorithm
104 to further optimize the data environment, they finally proposed the anomaly-detection framework
105 [11].

106 Also, Yan et al. tried and implemented a mean SMOTE (M-SMOTE) algorithm through their
107 research on SMOTE algorithm and verified the effectiveness of the algorithm in the classification
108 process of unbalanced network data [12]. In general, although many studies focus on the
109 optimization of outstanding machine learning algorithms and focus too much on the overall training
110 metrics of the dataset, and the optimization methods used mainly include feature selection, data pre-
111 processing, and classifier optimization, we can still make appropriate improvements in this area to
112 obtain improved detection results.

113 **3 The proposed intrusion detection model**

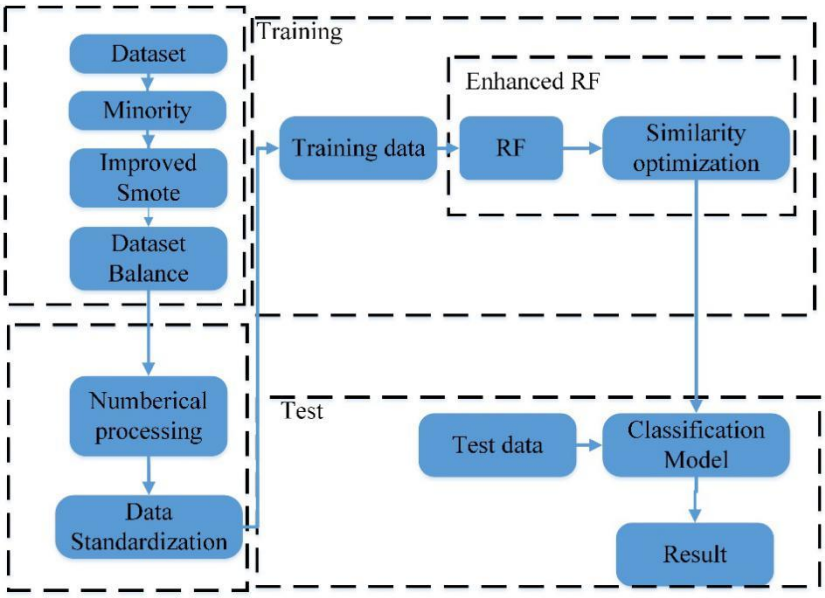


Fig. 1 The Architecture of model

The intrusion detection model involved in this paper selected machine learning algorithms such as random forest, which were commonly used in related studies. The performance of the classifier was improved by optimizing the random forest algorithm for similarity and combining it with data imbalance processing techniques. The overall architecture of the intrusion detection model is shown in Figure 1, which mainly includes the following processes：

(1) The analysis of the NSL-KDD dataset revealed the imbalance in the samples of the network attack type dataset. This imbalance resulted in higher false detection rates and lower accuracy for the detection of smaller-scale samples. Therefore, this paper proposed a combination sampling method by combining the K-means algorithm with the SMOTE algorithm. This approach could reduce the number of outlier samples, enriched the attribute features of the minority samples, and increased the sampling number of the minority samples to build a more balanced sample data of the network environment.

(2) To further reduce the computational overhead time and increase the performance of the detection, it was necessary to convert the non-numerical features in the original dataset digitally, and then convert the values to a specific range by normalization. This allowed normalization of the dataset and feature selection by information gain to filter out unnecessary features.

(3) The classification model was trained by feeding the normalized processed dataset into the enhanced random forest algorithm. The whole process was explained in detail as follows, the traditional random forest model was initially constructed, and then the constructed model was evaluated based on the area under curve (AUC) index for the performance of the decision tree. The decision tree with the more outstanding performance was selected by the above-mentioned approach. Then, the decision trees with high similarity were filtered by calculating the similarity between them, and finally, the decision trees with low similarity and high performance were formed into an enhanced random forest model, and the activity similarity matrix was generated in order to determine the results of subsequent activities.

(4) In the next section, correction and optimization of detection results were performed by calculating the similarity relationship between sample types of network attack data. The process started with a preliminary determination of the cybersecurity attack dataset by enhanced random forest and determined the type of attack by majority voting. In the next step, this paper

made accuracy judgments based on the key features, and if the judgment results indicated that the activity type was not reasonable, the results would be corrected based on the attack type similarity matrix.

(5) A classification model with relatively good performance would be obtained after the enhanced random forest training, and the results would be evaluated by conducting performance tests on the NSL-KDD dataset.

3.1 Construction of balanced data set based on k-means clustering algorithm and smote sampling technique

There are a large number of normal type samples in the network attack data set, while the number of abnormal samples is relatively small, which will interfere with the classification performance in the process of detection model training. Such problems will result in a classification model that performs well in terms of accuracy for majority classes of samples, but the accuracy of the minority classes may not be satisfactory, and the generalization ability of the overall classification model is relatively weak. Among the many sampling algorithms, in order to maintain the diversity of the training sample and preserve the inherent characteristics of the sampled samples. Therefore, the SMOTE algorithm technique is used for the over-sampling of minority class samples in this paper. By analyzing the minority samples, multiple minority samples are manually processed to generate new samples and added to the original dataset. This approach allows optimizing the network environment sample and minimizing the overfitting problem of the model. The main idea of the algorithm can be explained in detail as follows:

(1) For every sample x from the minority class sample, based on the Euclidean distance as the reference standard, the distance from this sample to other samples of the same type is calculated. The k nearest neighbors

of this sample are obtained by the above operation.

(2) By analyzing the number of samples in advance, a more reasonable over-sampling rate N is determined. Based on the determined parameter N, the random selection operation of the number of samples from the k nearest neighbors obtained above is denoted as x_n.

(3) For the random nearest neighbor sample x_n obtained by operation (2), the new sample points are constructed by performing the operation shown in equation (1) with the initial sample points in turn.

$$x_{new} = x + rand(0,1) \times \left| x - x_n \right| \quad \textbf{(1)}$$

The SMOTE algorithm technique achieved some effect and has some improvement on the overfitting problem. Based on the traditional SMOTE algorithm, a series of improved algorithms had been proposed and achieved better performance, including AE-SMOTE, SMOTE-ENN [13], and so on. However, the analysis of the network security dataset revealed that the SMOTE algorithm still had certain problems in dealing with imbalance problems, such as the handling of outlier values. For this type of problem, related studies dealt with it by excluding such values a priori or do not engaging in consideration of outlier values, for example, this type of value was not handled in Borderline-SMOTE [14]. Therefore, in this paper, the interference of outlier points to the sample generation process would be reduced by combining the K-means clustering algorithm with SMOTE, and the detailed steps of the improved SMOTE algorithm are shown as follows:

(1) The data of the minority samples are analyzed and the number of clustered sample centers T is determined, and then the target samples are selected by K-means clustering based on this value.

(2) For the target sample determined in the above step, a random sample is selected and the k nearest neighbors of this sample in the target sample are calculated.

(3) The samples are also selected from the k nearest neighbors based on the pre-analyzed data and set over-sampling rate N . The mean value between all samples is calculated and then a new sample is generated between that value and the neighboring samples following the steps shown in equation (2).

$$\begin{aligned} x_{\text{mean}} &= \frac{1}{k}\sum_{i=1}^{k} x_i \\ x_{\text{new}} &= x_i + rand(0,1)\times\left(x_{\text{mean}} - x_i\right) \end{aligned} \tag{2}$$

The overall flow of the improved sampling processing algorithm is shown in Figure 2, and the figure is demonstrated using the binary classification problem. First, the number of outlier samples was minimized as much as possible by clustering through K-means, and then the obtained outlier samples were used in an optimized way for the subsequent new sample production process. And then, the mean sample of the neighboring points was calculated and an attempt was made to use it as the center of the later sample clustering with the nearest neighbor sample to generate the newborn sample. The attribute characteristics of the new samples obtained in this way would be richer and the number of outliers would be relatively reduced compared with the traditional way, which was more beneficial for the training of the random forest classifier later. The overall process of this sampling is shown in Figure 2.

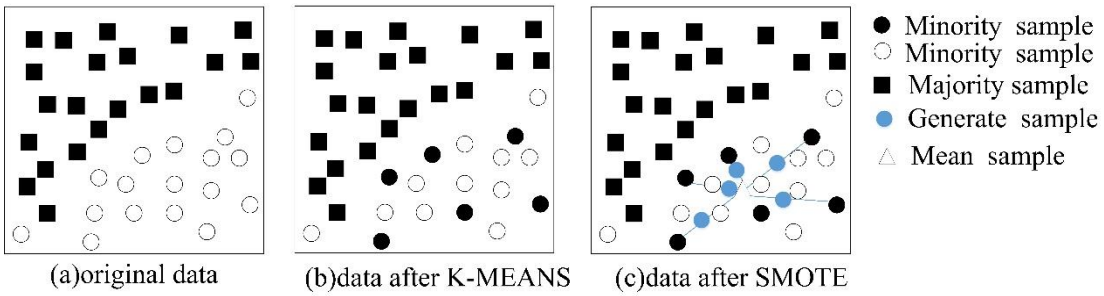


Fig. 2 Sampling processing

3.2 IDS based on enhanced random forest

The random forest algorithm has relatively high detection accuracy compared to other classification algorithms and the algorithm is more tolerant of noisy samples, which results in numerous theoretical and experimental studies focusing on the use of these algorithms. As a combinatorial classifier algorithm, by learning the basic idea of the Bagging algorithm to obtain N Bootstrap training samples with a put-back sampling of the original data set, the disguised augmentation of the samples used for training can be achieved. This approach effectively reduces the probability of overfitting. The data set obtained by the above operation is fed into a decision tree model for training and the final model is combined to generate a forest classification model. The model is predicting the results mainly by majority voting. The overall flow of the random forest is shown in Figure 3.

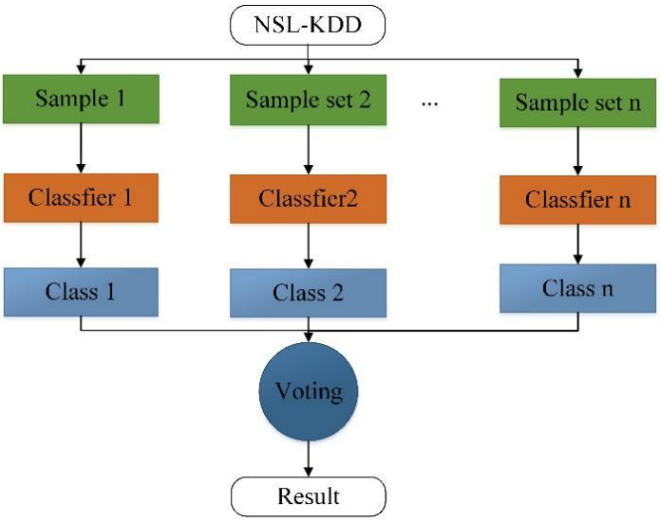


Fig. 3 Random forest classifier

However, there is still much room for improvement in the traditional random forest model. It includes improvements in the classification ability of each decision tree in the forest, further optimization of the correlation between decision trees in the combined forest model, and optimization of the voting method adopted in the process of conducting the result determination. A relatively good combinatorial model needs to have the following characteristics, good decision-making ability within classifiers, and a small correlation between classifiers. This paper would like to optimize the random forest from the following aspects. First, the classifiers with excellent decision performance were selected by the area under curve (AUC) index, then the inter-tree optimization was performed by calculating the similarity between the decision trees, and finally, the result correction process was performed by determining the similarity between the network attack results.

The horizontal and vertical axes of the Receiver operating characteristic (ROC) curve depict the proportion of predicted types that are consistent with positive samples and actual types, respectively. The value of the area under the curve (AUC) is the area between the ROC curve and the coordinate axis. It serves as the corresponding probability value, and it can indicate the superiority of the classifier performance by comparing the high or low value of this value, which is the reason that it is a criterion for internal performance optimization. Besides, the structural similarity between classifiers could be further calculated based on the calculation of classifier nodes and branches. Based on this, the structural similarity between classifiers was further calculated, which could be roughly classified as similar (more than 80%) or dissimilar (less than 40%) by setting a certain threshold value as the judgment criterion. After the similarity comparison by the above steps, the values were transformed into matrix form and the secondary optimization process was

carried out according to the difference in classification and the level of AUC, and the classifiers with strong individual classification ability and low similarity were selected to form a classification model by combining them.

The details of the enhanced random forest model involved in this paper were explained as shown below:

(1) Analyzing and using the Bagging algorithm, the original network security samples were selected and grouped randomly, the number of in of bag (IOB) samples for each group was W, and the obtained sample order set was as follows:

$$\left\{IOB_1, IOB_2, \ldots, IOB_W\right\} \tag{3}$$

(2) Based on the above-obtained training sample set (3), the corresponding optimal splitting attributes and candidate attributes were selected by random attribute selection and Gini index. After N rounds of model training, the corresponding classification model was finally obtained. By calculating and ranking the AUC values, the set of classification models with excellent classification performance was selected.

(3) The acquired classification models were optimized based on the similarity, and the classifiers with high similarity and poor classification performance were emptied. The computational approach taken in this paper focused on the calculation and application of structural similarity, and this class of methods learned and borrowed from Bakirli's [15] multi-tree inter-similarity optimization method. By analyzing and utilizing the decision tree

storage structure form, the classifier was transformed and decomposed into the corresponding rule set and candidate rule set. The similarity $similarity_{c1,c2}$ between multiple trees was derived by comparing split nodes among them and generating the similarity matrix $Matrix$. The set of classifiers (4) with high similarity and poor overall performance was selected by setting the corresponding thresholds:

$$\left\{classifier_1, classifier_2, \ldots, classifier_Q\right\} \tag{4}$$

(4) Based on the classification combinations obtained by the above operation, the results were determined. The traditional rules for determining to vote in the classification model were based on the majority voting principle as shown in (5):

$$F(x) = \arg\max_{\forall T} \sum_{i=1}^{Q} \left(classifier_Q(x) = T \right) \tag{5}$$

In (5) $F\left(X\right)$ denotes the combined classification model after the optimal selection shown above. $ClassifierQ\ \left(x\right)$ denotes the Q single classifiers in the combined classification model, and T denotes the label classification result.

In this paper, the results of the voting session were processed with corrections. Certain criteria needed to be established because the detection accuracy of the classification model was not completely accurate, and if there was a misclassification during the detection process, the method could be used to correct the detection results. Therefore, some activity rules needed to be pre-defined. The voting result $F\left(X\right)$ was obtained after the majority voting of the result by the combined classifier, the result of this determination was within a reasonable range if the attribute characteristics of the type were within the predefined activity rules. If the attribute characteristics

297 of the activity did not match the set activity rules, it was necessary to calculate the similarity
298 relationship between the decision results to generate the $SimMatrix$. Finally, a relatively more
299 reasonable decision could be chosen based on the probability value $SimMatrix$ obtained. This
300 operation focused on the following components:

301 i. The setting of activity rules.

302 ii. Generation of the activity similarity matrix.

303

304 **3.2.1 Setting of activity rules**

305 Through the analysis of the NSL-KDD dataset samples, more critical features and candidate
306 features were found, which could be used as the basis for setting the corresponding activity rules.
307 The number of attribute features of NSL-KDD was relatively large to 41, and the attack type labels
308 were roughly divided into two categories, including normal and anomaly. The anomaly data types
309 could be divided into 4 major categories, including Denial of Service attacks (DOS), Probe, Users
310 to Root attacks (U2R), and Remote to Local attacks (R2L).

311 In order to reduce the computational overhead time and reduce the false detection rate when
312 analyzing and processing data of larger size and dimensionality, similar studies included
313 optimization methods commonly used in Mohammadi [16], Selvakumar [17], and Staudemeyer [18],
314 such as feature compression. After extensive experimental research by analyzing and processing the
315 entire KDDcup99 dataset, Staudemeyer massively compressed the attribute features to 11, including
316 duration, service, and other types. And based on this, by combining the decision tree classification
317 method with correlation, the extracted features compressed the scale more efficiently compared to
318 the previous ones.

319 On this basis, the inherent features of the dataset were selected through information gain. The

attributes selected for the active rule setting included service, src_bytes, dst_bytes, hot, num_file_creations, dst_host_srv_count, dst_host _same_src_port_rate and so on.

After experimental comparison, it could be seen that the effect of service was relatively better, specifically, the set of service attributes of U2R included tftp_u, ftp_data, gopher, pm_dump. The set of attributes of R2L included telnet, ftp_data, ftp, other, http, imap4, login. The set of service attributes for the rest of the data samples contained R2L.

Therefore, the following rules were established for the event. If the service attribute of the detection target was a subset of the set corresponding to R2L and did not belong to the data set corresponding to U2R, the set T_{ser} of result types was determined to be R2L, probe, DOS, Normal. If the service attribute of the detection target was not a subset to which R2L belongs, the set T_{ser} of network attack types was distributed in the probe, DOS, Normal. The specific explanation of the activity set where the process judgment was located is shown in (6):

$$T = \begin{cases} F(x) & F(x) \in T_{ser} \\ \max\limits_{j} AcCorr_{[classifer(x)][j]} & F(x) \notin T_{ser} \end{cases} \quad \textbf{(6)}$$

where $F(x)$ is obtained by majority voting, x is the intrusion behavior characteristic, and T_{ser} is the set of types obtained by the activity rule.

If the result $F(x)$ obtained by the combined classifier is in T_{ser}, the final invasion type is determined as $F(x)$. In contrast, if the results do not match, the set of attack types in T_{ser} and $F(x)$ are calculated by similarity, and the one with the largest probability value in the set T_{ser} of types is selected as the final classification result.

3.2.2 Generation of the activity similarity matrix

For the analysis of the intrusion detection data samples, it was clear that the network malicious attack types had certain similar property operations between them, such as DOS, U2R, PROBE,

R2L, and other attack types, which could lead to the reduction of Src_byte, dst_bytes byte values. U2R and R2L type attacks could be detected by hot, num_failed_logins feature behavior and malicious interactions had a strong correlation in time. Therefore, the analysis of the correlation between attack types could be performed in advance, and in this way, the correction operation of the random forest determination results was achieved. A large number of experiments for intrusion type detection were conducted in this paper, and on this basis, comparisons were made with the real type to generate the corresponding activity matrix in Figure 4.

Fig. 4 Matrix of attack type similarity

The corresponding target relationship probability values were obtained by the operations shown below:

$$sim_{a,b} = \frac{time[a][b]}{\sum_{r=1}^{n} time[a][r]} \tag{7}$$

In (7), $\sum_{r=1}^{n} time[a][r]$ denotes the total number of attack types determined as a after pre-experimental processing, and $time[a][b]$ denotes the proportion of attack types determined as b among the attack types determined as a obtained above.

4 Results and discussion

This section provided a detailed description of the experimentally relevant data set and the preprocessing process, followed by a brief description of the evaluation metrics used in this paper, and finally, some comparative analysis of the experimental results was described. This paper used Python for code implementation. The experimental environment was configured as follows: processor Intel Core i5-10400, 16G memory device, and operating system Win10 Professional 64-bit.

4.1 Data set description

Through the analysis of the experimental datasets used in similar studies, and the one that is better known in the field of intrusion detection and has obvious disadvantages is the KDDcup99 dataset. The main problem in this dataset is that the training data of the system, which consists of up to 75% redundant data, tends to be somewhat misleading to the classifier. This allows the classifier to focus on records with more frequent occurrences and to learn relatively less from minority classes of data including R2L, U2R, which has a great impact on the model detection effect and making the classification results have a more obvious bias. Considering the above factors, this paper tries to adopt a NSL-KDD dataset optimized for frequent records in the KDD dataset. As shown in the table, the overall data types of the NSL-KDD dataset can be broadly classified into two categories, including normal and anomaly. The exception types in the data set can be subdivided into four major categories and many subtypes, including DOS, Probe, U2R, and R2L. Table 1 shows the distribution of the specific type composition of the dataset.

Table 1 Attack Type Distribution

Attack Type	Class	Subclass
Normal	normal	normal
anomaly	Probe	ipsweep, mscan, nmap, portsweep, saint, satan
	DOS	apache2,back,land,mailbomb,neptune,pod,processtable,smurf, teardrop, udpstorm
	U2R	buffer_overflow,httptunnel,loadmodule,perl,ps,rootkit,sqlattack, xterm
	R2L	ftp_write,guess_passwd,imap,multihop,named,phf,sendmail,snmpgetattack,snmpguess,spy,warezclient,warezmaster,worm,xlock,xsnoop

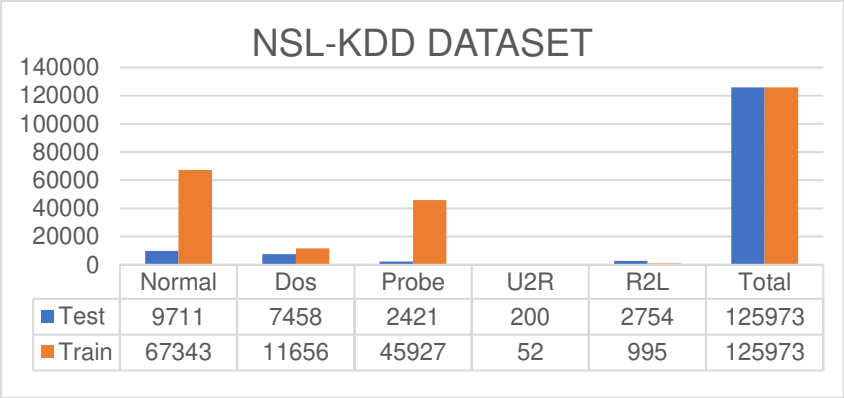


Fig. 5 Overall NSL-KDD data statistics

The data imbalance problem that exists in the NSL-KDD dataset is shown in Figure 5. The proportion of normal type samples in the overall data sample dominates, but those types of sample data such as Probe, R2L, and U2R, which are more frequent in real attack activities, are slightly under-represented in the overall data set.

4.2 Data preprocessing

In order to further reduce the computational overhead time and ensure the processing of important attribute information, it was necessary to numerically transform the attributes that were not directly available in the original dataset and perform data normalization operations on the key data. The number of attribute features in the original dataset was up to 41, mainly including basic features such as duration, content features such as hot, and time and host-based network traffic statistics such as count, dst_host_count. Among them, preprocessing operations focused on the

processing of character-based attributes, mainly including protocol_type, service, and flag. First, numeric values were assigned to the tail column data types of each data sample, mainly including the following five types: 0 for normal type, 1 for Probe type attack, 2 for DOS type attack, 3 for U2R type attack, and 4 for R2L type attack. Then the character-based values were transformed into binary code features for easy identification and processing by one-hot encoding, for example, the protocol_type field was preprocessed to represent the TCP protocol using [1,0,0]. Finally, to prevent the performance of the model from being affected by data processing overflow problems during training due to overly large data values, a normalized processing operation for the original data was necessary and mapped it to the [0,1] interval range.

$$r_n = \frac{r - r_{\min}}{r_{\max} - r_{\min}} \tag{8}$$

In (8), where r_{min} represents the minimum value of the current attribute feature, r_{max} represents the maximum value of the current attribute feature, and r_n represents the value after normalization.

4.3 Evaluation metrics

In order to get a comprehensive understanding of the overall classification of the model and the performance effect of classification of fewer classes of samples. The performance evaluation metrics selected in this paper include accuracy, false positive rate (FPR), recall, which were more commonly used in similar studies, in addition to the commonly used accuracy rate. All of the above indicators were derived from the analysis and application of the basic attributes of the confusion matrix including True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), and the specific explanation of each indicator is shown below:

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$FPR = \frac{FP}{FP + TN} \quad (10)$$

$$DR = TPR = \text{Recall} = \frac{TP}{TP + FN} \quad (11)$$

4.4 Experimental results on NSL-KDD

The experimental process and the evaluation of experimental results were mainly divided into training and testing. The original NSL-KDD Train+ dataset was allocated in the ratio of 80:20 for training and testing validation of the model, respectively. Finally, the performance of various classifiers was tested and evaluated on the NSL-KDD Test+ dataset. The test classifiers used in this paper mainly included classical machine learning classification algorithms such as SVM, RF, and KNN. The detection effect of each classifier on the categories could be clearly shown in the following figure, in which the detection accuracy of the proposed algorithm on the validation set is as high as 99.72%, which is about 2% better than other classifiers. The details of the experiments of the proposed method for binary classification on the NSL-KDD dataset are shown in detail in the following Figure 6.

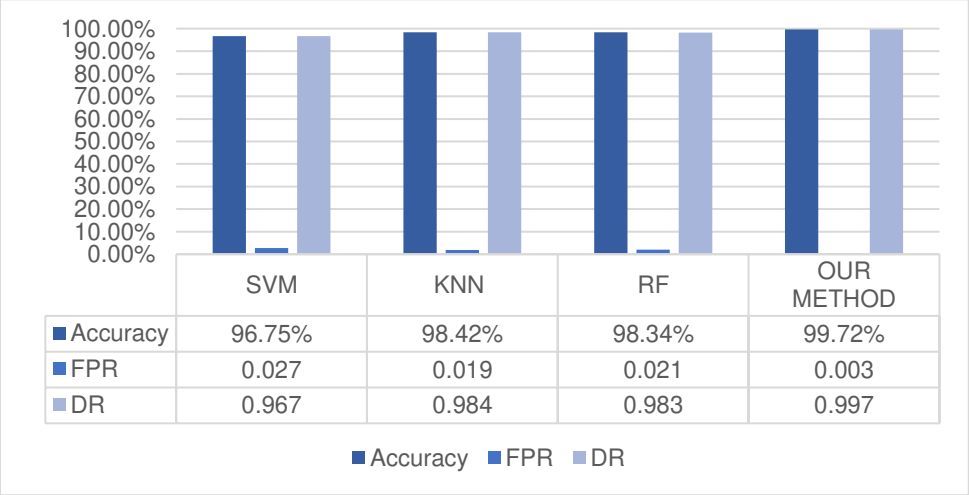


Fig. 6 Compression of algorithms on training set（2-class）

After fine-tuning the operation on the basis of the above generated classifier, the detection effect of the model in multi-classification is shown in Table 2. Compared with other classifiers, the

proposed model in this paper shows multiple improvements in the detection rates of different attack types. In particular, the detection rate on Probe is about 1.43% higher than the SVM classification model and 1% better compared to the random forest algorithm.

Table 2 Compression of algorithms on training set（5-class）

Attack Type	**Detection Rate**			
	SVM	KNN	RF	OUR METHOD
Probe	98.31	99.68	98.74	99.74
Dos	98.32	99.57	99.59	99.99
U2R	99.96	99.88	99.91	99.92
R2L	96.08	99.45	93.37	99.89

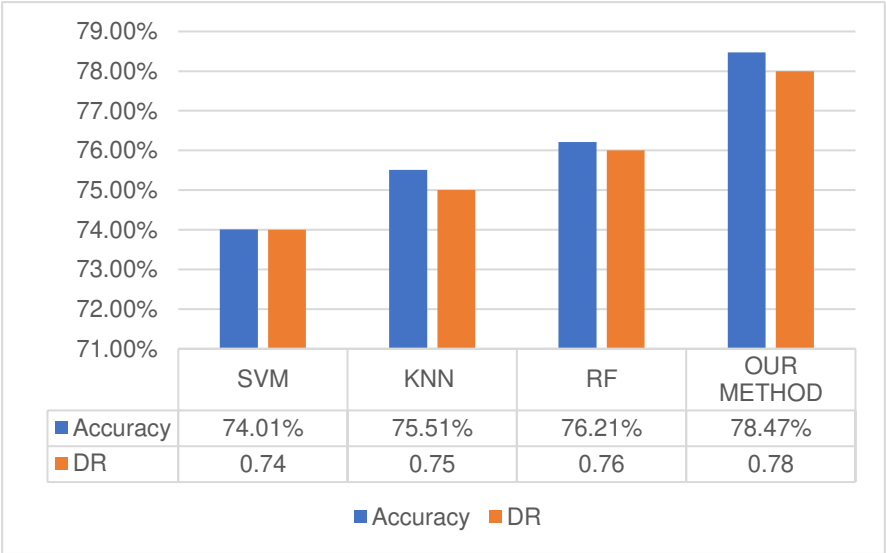


Fig. 7 Compression of algorithms on KDDTest+

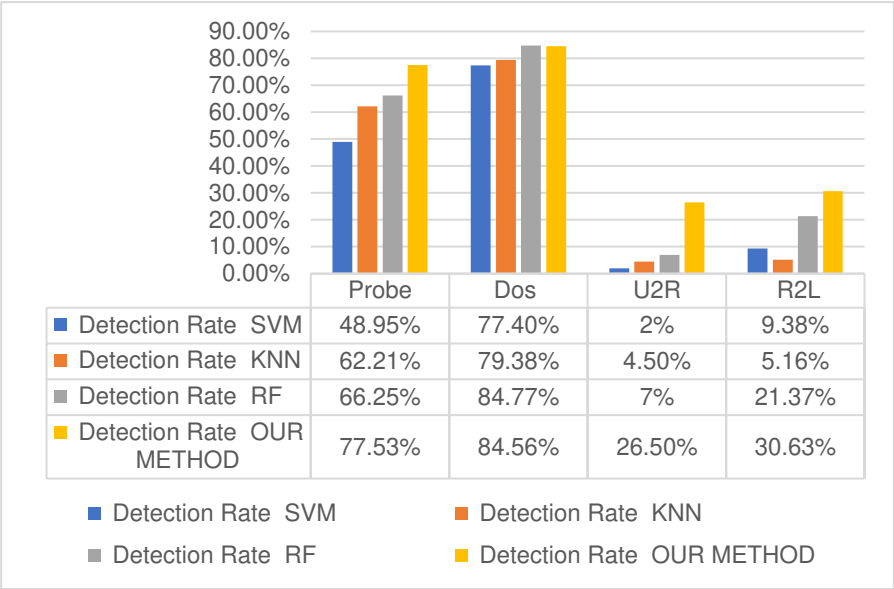


Fig. 8 Multi-class Compression of algorithms on Original dataset

From the information in the above Figure 7 and Figure 8, the highest accuracy of our classification model was 78.4 when it was evaluated in the KDDTest+ dataset. Besides, it was notable that the detection effect of all types of classification models for minority samples such as U2R and R2L was slightly lower compared with other majority classes, which was due to the imbalance of data between samples during the training process leading to the training model focusing too much on the detection of majority class samples.

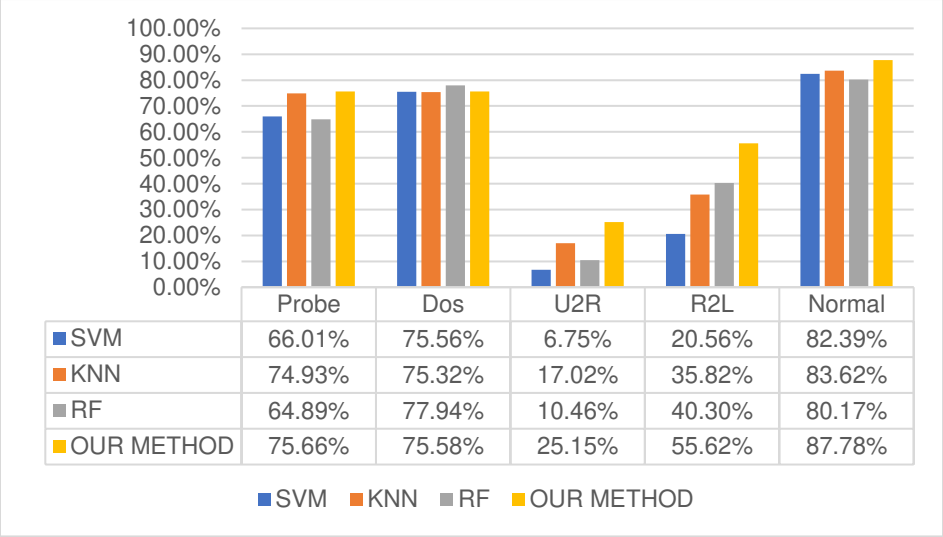


Fig. 9 Multi-class Compression of algorithms on Sampling dataset

The classification algorithms were evaluated in the NSL-KDD dataset after being processed by the sampling method proposed in this paper, and the classification results could be clearly shown in Figure 8 and Figure 9. The proposed hybrid method combining the K-means clustering method with SMOTE sampling technique, which improves the detection effect of each classifier for

minority class samples and effectively alleviates the problem of data imbalance. Therefore, the proposed method in this paper optimizes the intrusion detection dataset to a certain extent and the sampled dataset has some practical significance compared with the original dataset.

Besides, we compare the proposed method with excellent research methods in order to further show the superiority of the proposed method. The comparison results obtained are described in detail in Table 3, where the data set used, the classification model, the accuracy rate, and other factors are used as aspects of the comparison. As described in the table, in terms of the overall accuracy, the enhanced random forest method used in this paper and the random forest method used in the literature [28] have higher accuracy of 99.72 and 99.4, respectively. Second, compared with the classification methods such as SVM used in the literature [33], the detection accuracy of this paper is both improved.

Table 3 The comparison of proposed model with the state-of-art on the NSL-KDD

Study	Data Set	Classifier	ACC（%）
Golrang et al. [19]	NSL-KDD	Random Forest	99.4
Gao et al. [20]		Incremental extreme learning machine (I-ELM) and Adaptive principal component (A-PCA)	81.22
Belouch et al. [21]		RepTree	89.85
Salo F et al. [22]		Ensemble (SVM, IBK and MLP)	98.24
Our method		Enhanced Random Forest	99.72

5 Conclusion

In this paper, we analyzed the attack types and similarities of malicious intrusion attacks in NSL-KDD dataset, and then an intrusion detection system model was proposed and discussed based on enhanced random forest and SMOTE algorithm. Firstly, the equalization of training samples was achieved by combining the K-means algorithm with the SMOTE algorithm to some extent to compensate for the under-training of smaller scale samples. Then the optimization of the similarity

between decision trees was used to further enhance the detection performance of random forest. Initial detection was obtained by enhancing random forest, and finally the results were further corrected by the similarity of intrusion attacks. This paper evaluated the enhanced random forest algorithm on the NSL-KDD data set and achieved a relatively ideal effect. In the future, our study will further optimize the model accuracy and computational overhead time through feature extraction and classifier selection. Intrusion detection systems have great research significance as an important way to defend against malicious activities, and the use of ensemble learning methods can further improve the accuracy and robustness of detection, so machine learning technology has an important role in advancing research in the field of network security.

Abbreviations

IDS: intrusion detection systems; SMOTE: Synthetic Minority Over-Sampling Technique; SVM: Support Vector Machine; RF: Random Forest; K-NN: K-Nearest Neighbors; PSO: particle swarm optimization; PCA: Principal Component Analysis; M-SMOTE: mean SMOTE; AUC: area under curve; ROC: Receiver operating characteristic; IOB: in of bag; DOS: Denial of Service attacks; U2R: Users to Root attacks; R2L: Remote to Local attacks; FPR: false positive rate; TP: True Positive; TN: True Negative; FP: False Positive; FN: False Negative;

Availability of data and materials

The dataset supporting the conclusions of this article is available in the [https://www.unb.ca/cic/datasets/nsl.html].

Competing interests

The author declared that there is no conflict of interest.

Funding

This work was supported by National Science Fund of China (Nos.61806088, 61902160), and Qin gLan Project of Jiangsu Province.

Author's contributions

All authors contributed to the conception and design of the experiments and the interpretation of simulation results. Wu performed the experiments and data analysis, and Fan wrote the first draft of the manuscript. Zhu and You substantially revised the manuscript and Zhou and Huang contributed additional revisions of the text. All authors read and approved the final manuscript.

Acknowledgements

Not applicable.

Author details

[1] School of Mechanical Engineering, Jiangsu University of Technology, Changzhou 213001, China

[2] School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China

References

1. G Fernandes, J J P C Rodrigues, L F Carvalho, A comprehensive survey on network anomaly detection. Telecommunication Systems. 70(3), 447-489 (2019). https://doi.org/10.1007/s11235-018-0475-8
2. D Ramotsoela, A Abu-Mahfouz, G Hancke, A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study. Sensors. 18(8), 2491 (2018). https://doi.org/10.3390/s18082491
3. D E Denning, An intrusion-detection model. IEEE Transactions on software engineering, (2), 222-232 (1987). https://doi.org/10.1109/TSE.1987.232894
4. F Zhao, Detection method of LSSVM network intrusion based on hybrid kernel function. Modern Electronics Technique. 21, 027 (2015).
5. S J Horng, M Y Su, Y H Chen, A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert systems with Applications. 38(1), 306-313 (2011). https://doi.org/10.1016/j.eswa.2010.06.066
6. P Tao, Z Sun, Z Sun, An improved intrusion detection algorithm based on GA and SVM. IEEE

Access. 6, 13624-13631 (2018). https://doi.org/10.1109/ACCESS.2018.2810198

7. K Peng, V C M Leung, Q Huang, Clustering approach based on mini batch kmeans for intrusion detection system over big data. IEEE Access. 6, 11897-11906 (2018). https://doi.org/10.1109/ACCESS.2018.2810267

8. R M Elbasiony, E A Sallam, T E Eltobely, A hybrid network intrusion detection framework based on random forests and weighted K-means. Ain Shams Engineering Journal. 4(4), 753-762 (2013). https://doi.org/10.1016/j.asej.2013.01.003

9. J L Leevy, T M Khoshgoftaar, R A Bauder, A survey on addressing high-class imbalance in big data. Journal of Big Data. 5(1), 1-30 (2018). https://doi.org/10.1186/s40537-018-0151-6

10. N Ofek, L Rokach, R Stern, Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem. Neurocomputing. 243, 88-102 (2017). https://doi.org/10.1016/j.neucom.2017.03.011

11. X Ma, W Shi, AESMOTE: Adversarial Reinforcement Learning with SMOTE for Anomaly Detection. IEEE Transactions on Network Science and Engineering (2020). https://doi.org/10.1109/TNSE.2020.3004312

12. B Yan, G Han, Y Huang, New traffic classification method for imbalanced network data. J. Comput. Appl. 38(1), 20-25 (2018).

13. G E Batista, R C Prati, M C Monard, A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter. 6(1), 20-29 (2004). https://doi.org/10.1145/1007730.1007735

14. H Han, W Wang, B Mao, in International conference on intelligent computing. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. vol. 3644 (Springer,

Berlin, Heidelberg, 2015), pp. 878-887. https://doi.org/10.1007/11538059_9

15. G BAKIRLI, D Birant, DTreeSim: A new approach to compute decision tree similarity using re-mining. Turkish Journal of Electrical Engineering & Computer Sciences. 25(1), 108-125 (2017). https://doi.org/10.3906/elk-1504-234
16. S Mohammadi, H Mirvaziri, M Ghazizadeh-Ahsaee, Cyber intrusion detection by combined feature selection algorithm. Journal of information security and applications. 44, 80-88 (2019). https://doi.org/10.1016/j.jisa.2018.11.007
17. B Selvakumar, K Muneeswaran, Firefly algorithm based feature selection for network intrusion detection. Computers & Security. 81, 148-155 (2019). https://doi.org/10.1016/j.cose.2018.11.005
18. R C Staudemeyer, C W Omlin, Extracting salient features for network intrusion detection using machine learning methods. South African computer journal. 52(1), 82-96 (2014). https://doi.org/10.18489/sacj.v52i0.200
19. A Golrang, A M Golrang, S Y Yayilgan, A Novel Hybrid IDS Based on Modified NSGAII-ANN and Random Forest. Electronics. 9(4), 577 (2020). https://doi.org/10.3390/electronics9040577
20. J Gao, S Chai, B Zhang, Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis. Energies. 12(7), 1223 (2019). https://doi.org/10.3390/en12071223
21. M Belouch, S El Hadaj, M Idhammad, A two-stage classifier approach using reptree algorithm for network intrusion detection. International Journal of Advanced Computer Science and Applications. 8(6), 389-394 (2017). https://doi.org/10.14569/IJACSA.2017.080651

602 22. F Salo, A B Nassif, A Essex, Dimensionality reduction with IG-PCA and ensemble classifier
603 for network intrusion detection. Computer Networks. 148, 164-175 (2019).
604 https://doi.org/10.1016/j.comnet.2018.11.010

Figures

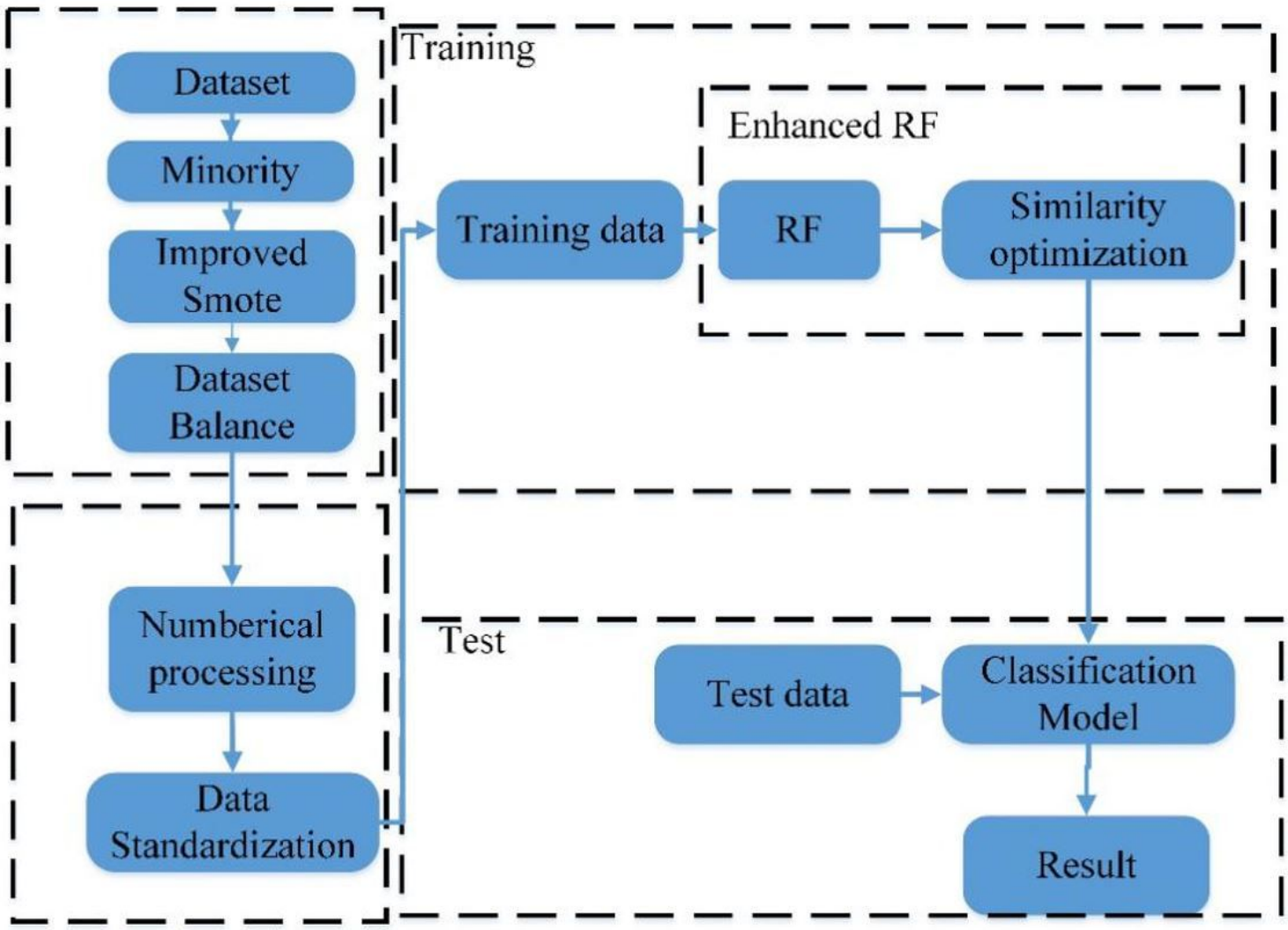


Figure 1

The Architecture of model

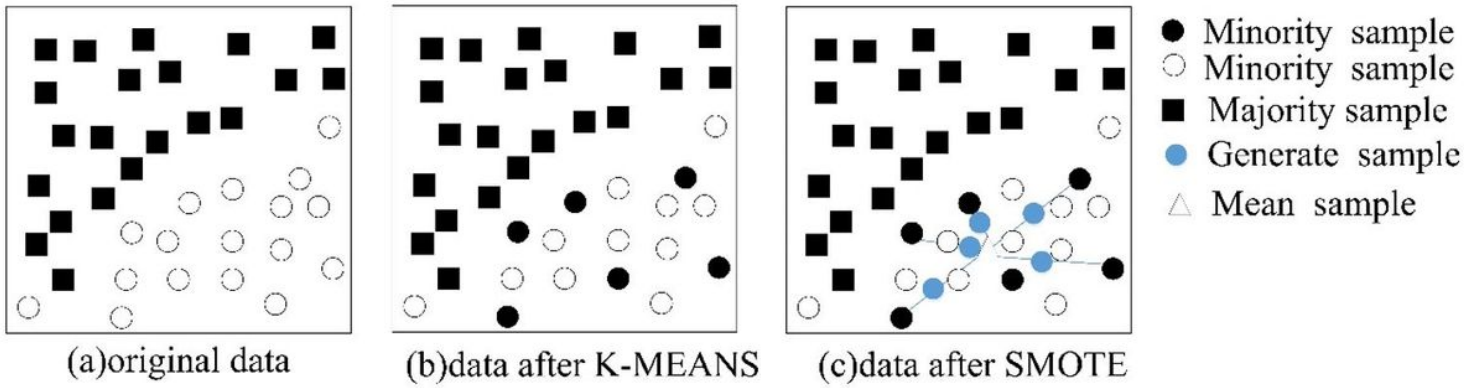


Figure 2

Sampling processing

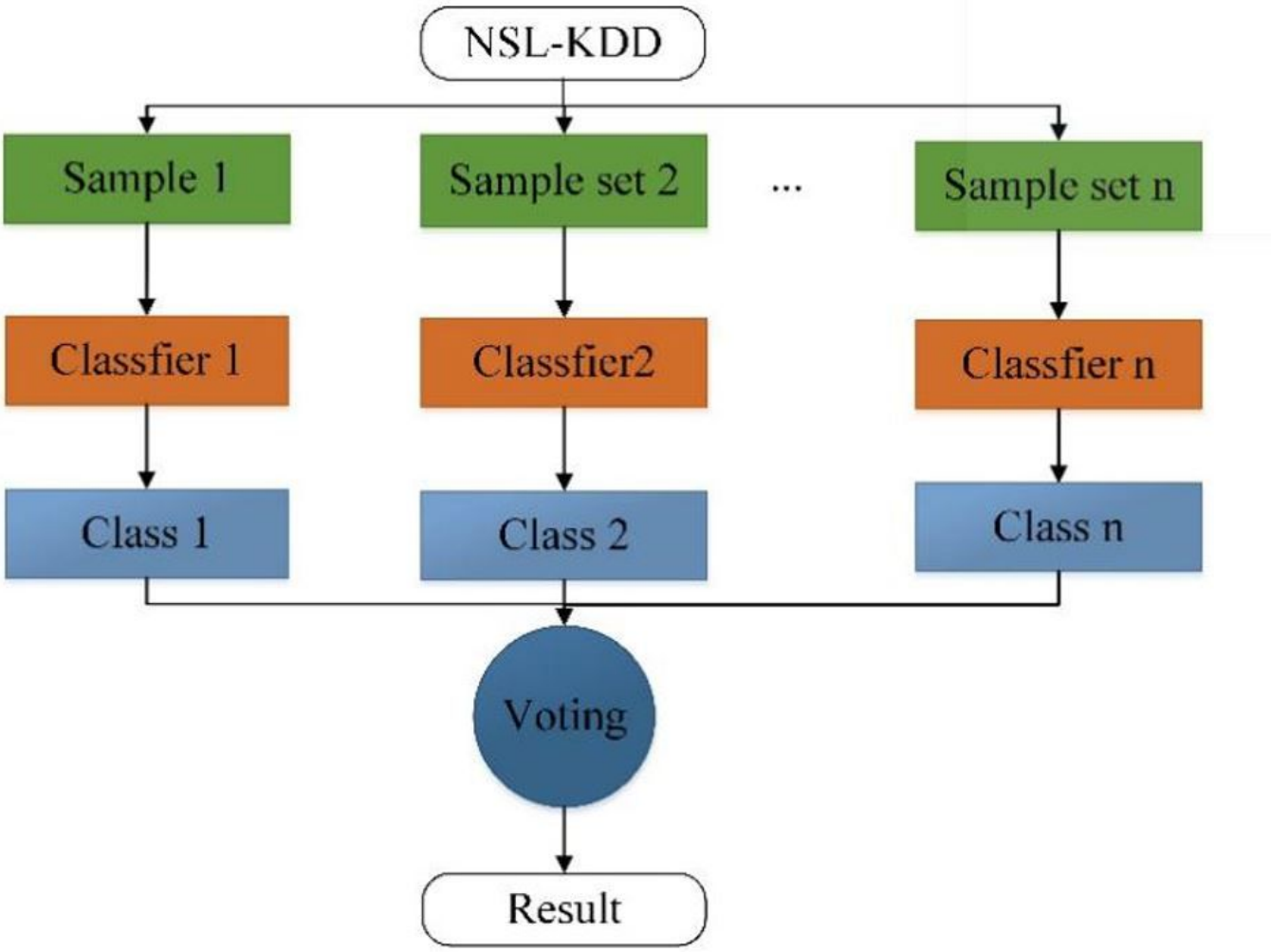


Figure 3

Random forest classifier

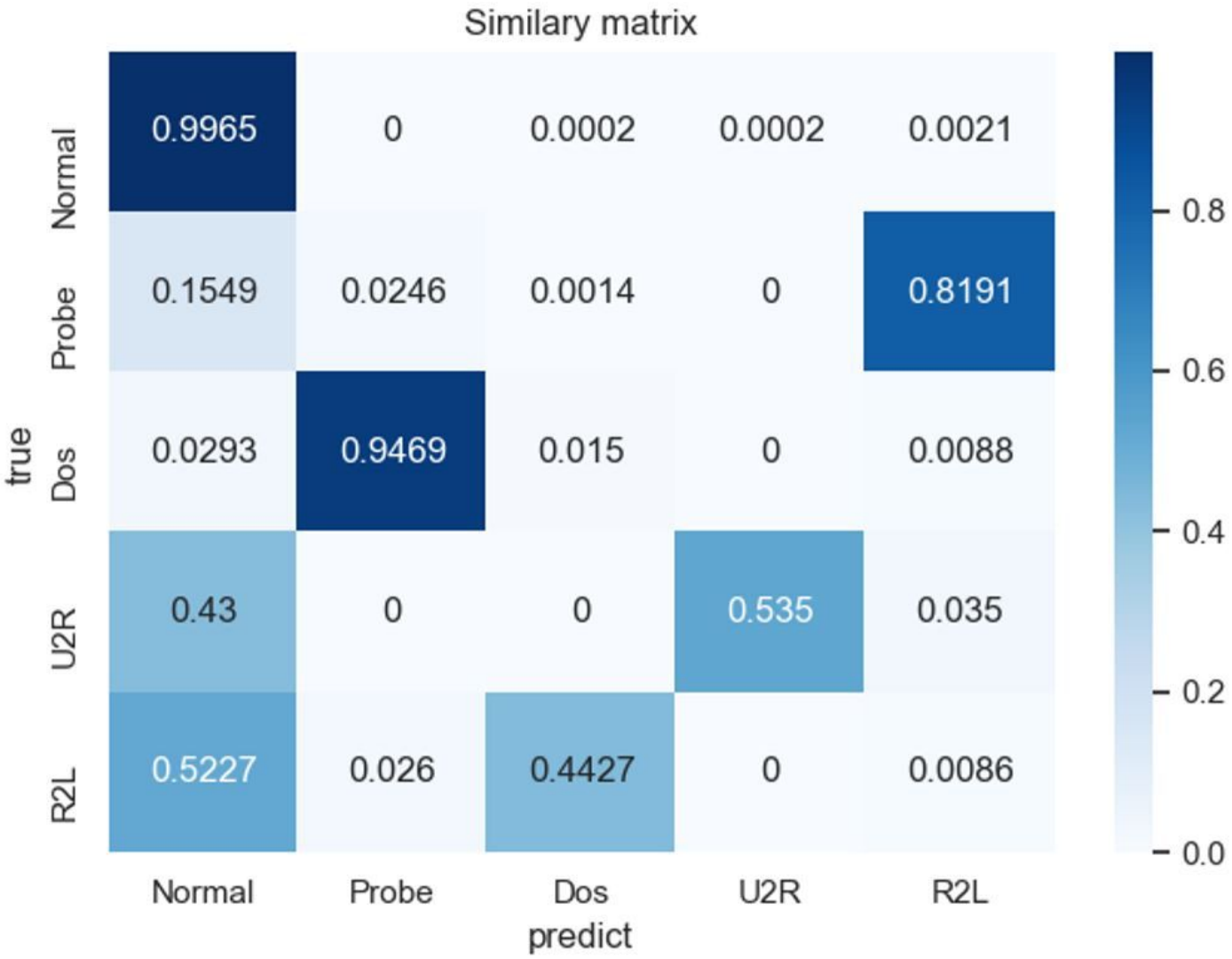


Figure 4

Matrix of attack type similarity

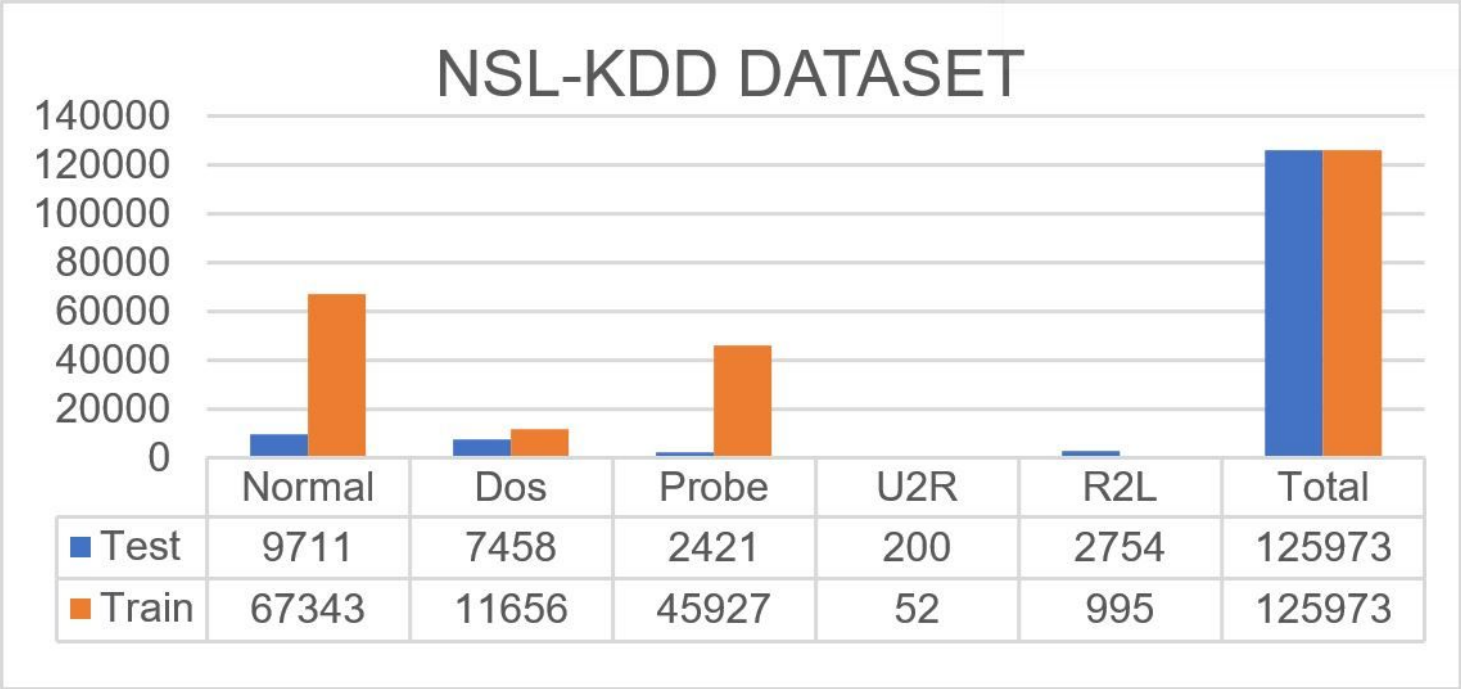


Figure 5

Overall NSL-KDD data statistics

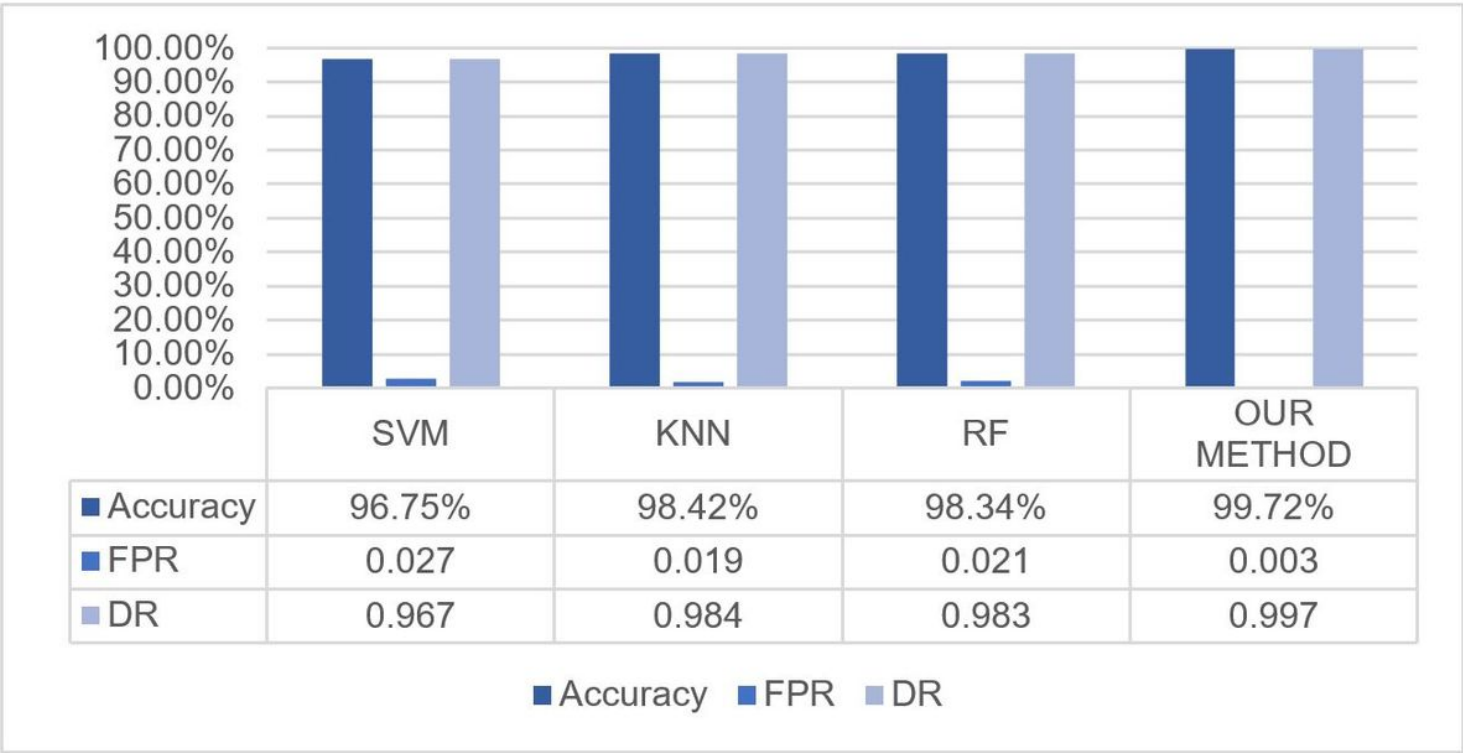


Figure 6

Compression of algorithms on training set (2-class

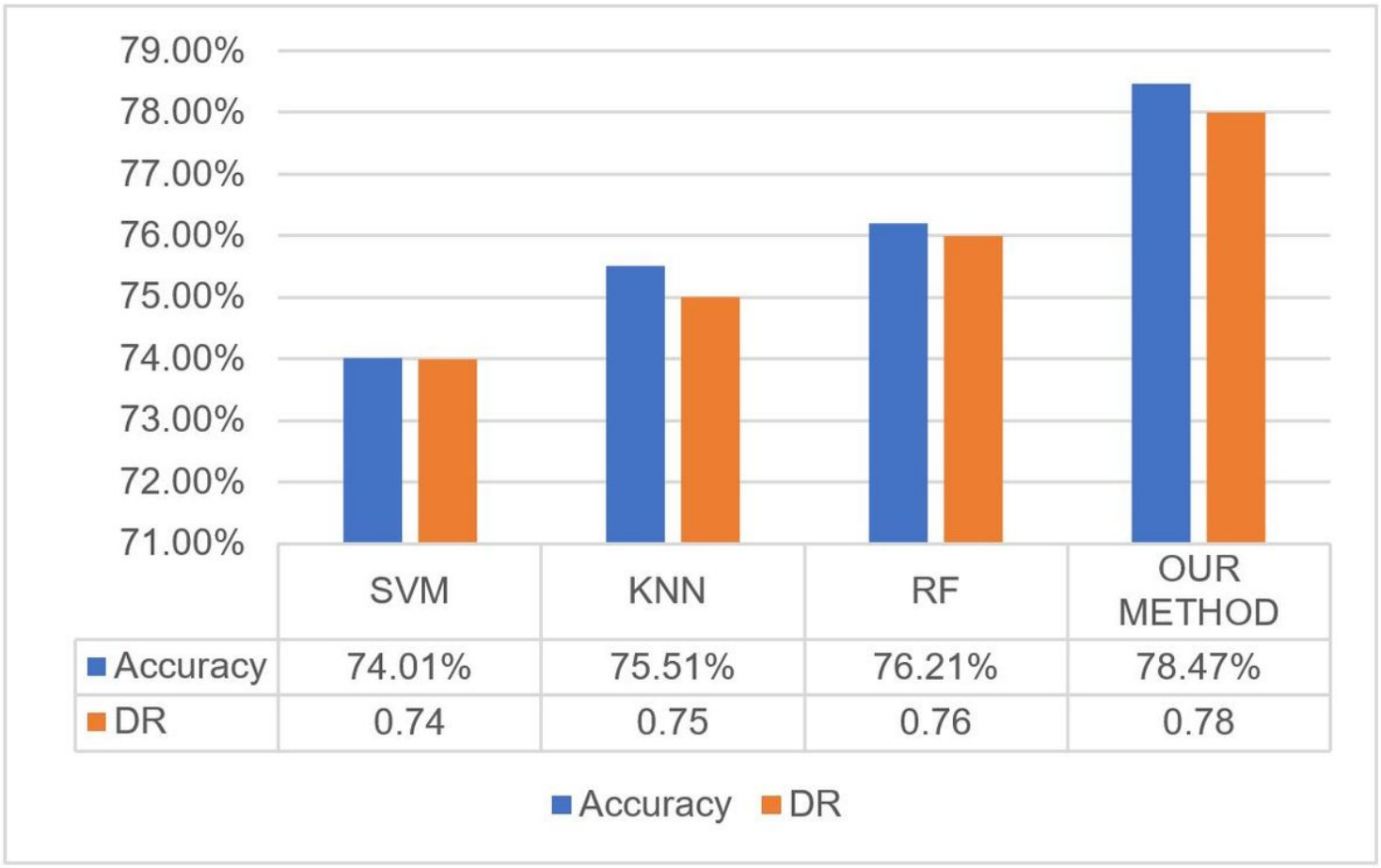


Figure 7

Compression of algorithms on KDDTest+

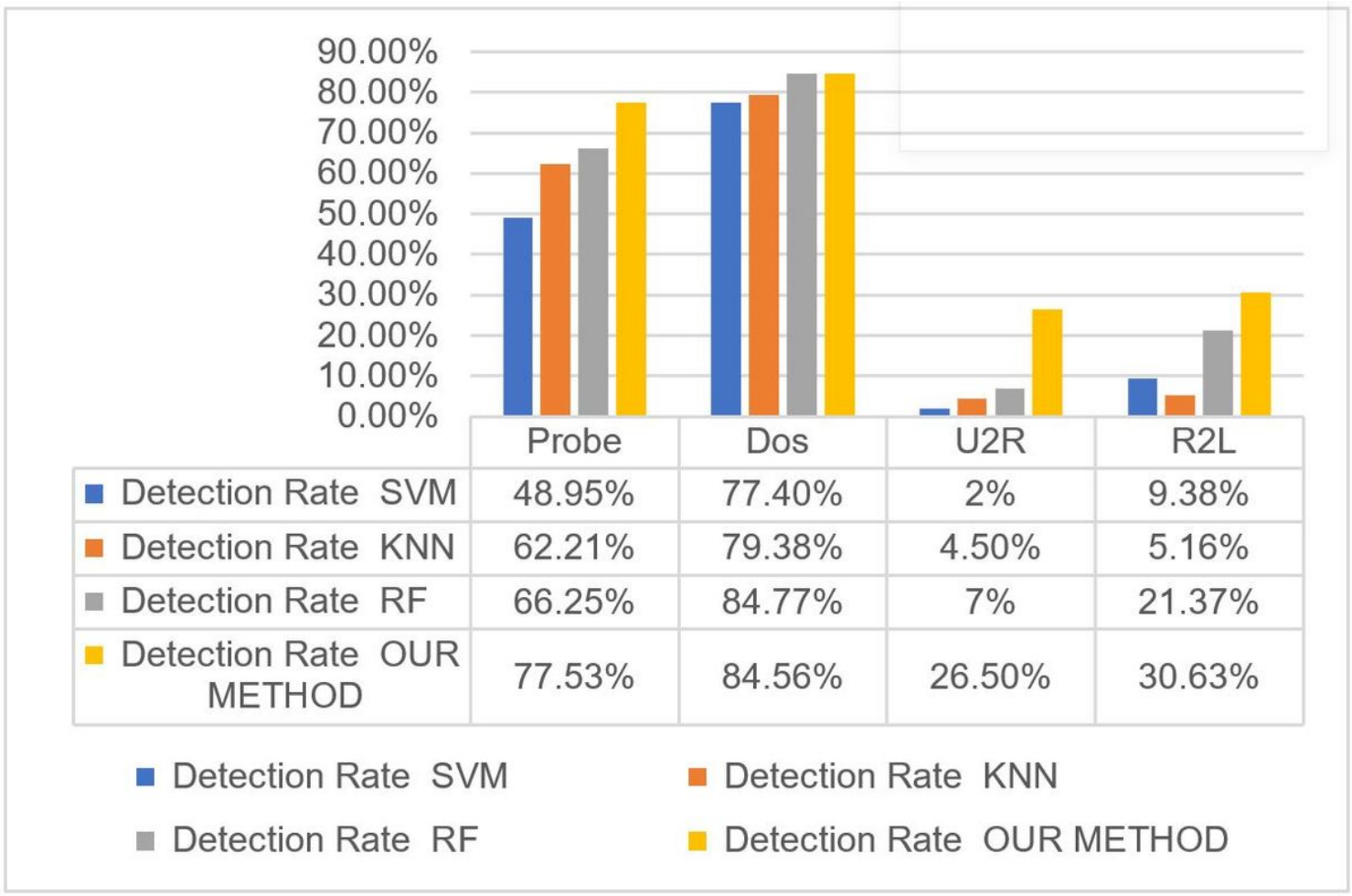
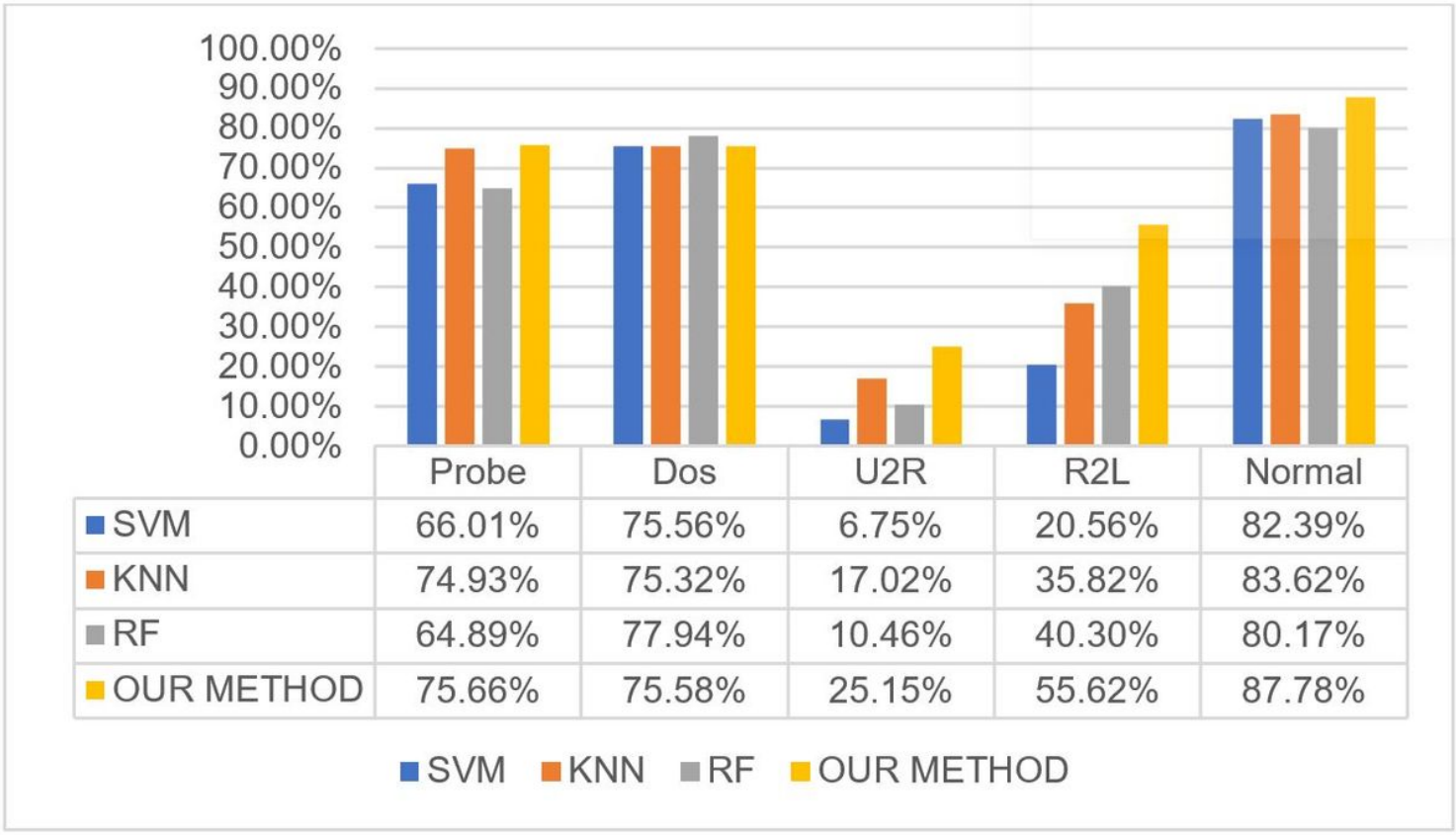


Figure 8

Multi-class Compression of algorithms on Original dataset



	Probe	Dos	U2R	R2L	Normal
SVM	66.01%	75.56%	6.75%	20.56%	82.39%
KNN	74.93%	75.32%	17.02%	35.82%	83.62%
RF	64.89%	77.94%	10.46%	40.30%	80.17%
OUR METHOD	75.66%	75.58%	25.15%	55.62%	87.78%

Figure 9

Multi-class Compression of algorithms on Sampling dataset