Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2835023, 15 pages
https://doi.org/10.1155/2020/2835023

Research Article

An Intrusion Detection Method Based on Decision Tree-Recursive Feature Elimination in Ensemble Learning

Wenjuan Lian, Guoqing Nie, Bin Jia, Dandan Shi, Qi Fan, and Yongquan Liang

College of Computer Science & Engineering, Shandong University of Science and Technology, Qingdao, Shandong 266590, China

Correspondence should be addressed to Bin Jia; jiabin@sdust.edu.cn

Received 27 April 2020; Revised 22 September 2020; Accepted 5 October 2020; Published 23 November 2020

Academic Editor: Manjit Kaur

With the rapid development of the Internet, various forms of network attack have emerged, so how to detect abnormal behavior effectively and to recognize their attack categories accurately have become an important research subject in the field of cyberspace security. Recently, many hot machine learning-based approaches are applied in the Intrusion Detection System (IDS) to construct a data-driven model. The methods are beneficial to reduce the time and cost of manual detection. However, the real-time network data contain an ocean of redundant terms and noises, and some existing intrusion detection technologies have lower accuracy and inadequate ability of feature extraction. In order to solve the above problems, this paper proposes an intrusion detection method based on the Decision Tree-Recursive Feature Elimination (DT-RFE) feature in ensemble learning. We firstly propose a data processing method by the Decision Tree-Based Recursive Elimination Algorithm to select features and to reduce the feature dimension. This method eliminates the redundant and uncorrelated data from the dataset to achieve better resource utilization and to reduce time complexity. In this paper, we use the Stacking ensemble learning algorithm by combining Decision Tree (DT) with Recursive Feature Elimination (RFE) methods. Finally, a series of comparison experiments by cross-validation on the KDD CUP 99 and NSL-KDD datasets indicate that the DT-RFE and Stacking-based approach can better improve the performance of the IDS, and the accuracy for all kinds of features is higher than 99%, except in the case of U2R accuracy, which is 98%.

1. Introduction

Cyber-attacks are becoming universal and one type of the most common cyberspace security threats. The attackers exploit the vulnerabilities and security flaws in the computer network and information system to launch attack, which causes the disclosure of system data and the invasion of user privacy and undermines the integrity or availability of data [1]. Cyber-attack is still spreading, targeting information systems, industrial infrastructures, computer networks, and personal end-devices. In addition, it is a typical network intrusion behavior. Intrusion detection is a means to identify the attempted intrusion, ongoing intrusion, or violation.

Since 2014, the National Information Security Vulnerability Sharing Platform (CNVD) [2] in China has witnessed an average annual growth of 15.0% for security vulnerabilities. Among them, the total number of security vulnerabilities recorded in 2018 was 14,201, including 4,898 high-risk vulnerabilities (34.5%). In 2018, Chinese National Internet Emergency Center (CNCERT) sample monitoring found that the number of large-scale distributed denial of service (DDoS) attack with peak traffic exceeding 10 Gbps in China averaged more than 4,000 per month. The denial of service (DoS) attacks usually inject a large number of redundant requests into the target computer or resource. These requests can overload the system to deny sectional or all legitimate requests. Criminals usually attack sites or services located on well-known web servers, such as banks or credit card payment gateways. Because the collapse of public systems will cause great losses, the attacks on above systems (such as on banks) are more terrible.

The CNCERT found that there are 2,108 resource utilization controlled by command and control (C&C) server to initiate DDoS attack. Meanwhile, there are 1.44 million broilers, 1.97 million reflected attack servers, and about 90,000 destination IP addresses. Such distributed attack

sources make it difficult for us to defend against all malicious IP. If we block the request IP on a large scale, it will cause a lot of normal requests to be killed. In addition, it should not be forgotten that the number of broilers was about 1.44 million. This shocking number means that more than 1.44 million computers or mobile phones have been intruded or controlled by attackers. Attackers can control our devices to launch illegal attacks and even read the privacy data of our devices, such as text messages, location, accounts, and passwords.

In order to defend the huge network intrusion, academia and industry have carried out a lot of exploration. An intrusion detection system (IDS) is a network security device that performs real-time monitoring of network transmissions and issues alerts. In addition, it takes the proactive response measures when suspicious transmissions are found. The IDS differs from other network security devices in that it owned the forward-looking security protection technology [3]. But, faced with the explosively severe cyberspace security situation, the traditional intrusion detection method has gradually exposed many drawbacks against the protection of network security. The typical defect is the existences of more serious False Positive (FP) and False Negative (FN).

With the vigorous development of artificial intelligence (AI), many machine learning technologies have been applied to the IDS. Machine learning (ML) improves the above problems to some extent. However, ML has its own shortcomings. When a single machine learning model meets a huge amount of data, its fitting adaptive ability is lower. This leads to poor generalization of ML as facing new data. No matter whether it is supervised learning [4–6] or unsupervised learning clustering [7] and other algorithms, there is no strong generalization ability.

In recent years, the extremely hot deep learning model has not exerted its due advantages in the IDS in the absence of the data like ImageNet. Deep learning needs plentiful of good quality data to support it, so it is less applied in the field of cybersecurity [8, 9].

In order to overcome the shortcomings of existing methods, this paper proposes a novel scheme based on the Recursive Feature Elimination and Stacking model in ensemble learning for the first time and tries to apply it in intrusion detection. Compared with the previous works, our proposed method and model have the following advantages:

(i) The Stacking technology is used, which combines the advantages of traditional machine learning. Stacking is an ensemble learning method that uses a model to perform adaptive voting weighting on the classifier. Stacking can solve the problem of insufficient fitting and generalization ability of traditional machine learning models.

(ii) A novel data processing method based on the Decision Tree-Recursive Feature Elimination (DT-RFE) is used to select features and to reduce the feature dimension. Our method eliminates uncorrelated and redundant data from the dataset to achieve better accuracy and to reduce time complexity.

(iii) We use four distributed models to learn different features so as to predict different types of attacks. In this way, we can further improve the accuracy of the model to a certain extent.

KDD CUP 99 is the most widely used dataset in the field of intrusion detection, and NSL-KDD is an improved version of KDD CUP 99. A series of comparison experiments on the KDD CUP 99 and NSL-KDD datasets show that our method can preferably improve the performance of the IDS. The proposed method in this paper improves and optimizes the existing IDS, which would reduce the feature dimension of network flow. Our method uses the idea of Stacking-based ensemble learning, and it can improve the generalization and adaptive ability of the model and have the higher accuracy.

The rest of this paper is arranged as following: Section 2 mainly presents the related work to intrusion detection research. Section 3 introduces the NSL-KDD and KDD CUP 99 datasets and analyses related processing procedure. Section 4 gives the dimension reduction method of the dataset. Section 5 raises the proposed RFE-Stacking algorithm. Section 6 carries out the experiments to verify our method and model, and Section 7 concludes the work.

2. Related Work

The IDS includes hardware and software which can actively or passively control hosts or network to detect some intrusions [3]. It embeds intrusion detection technology into a deployable system to identify and handle the violations of security policies in the computer network and system. In addition, industrial Internet security systems usually use the IDS to make up for the deficiencies of traditional network defense strategies [10]. According to the IDS input data source (undetected data), the IDS is usually divided into hybrid intrusion detection system (hybrid IDS), network-based intrusion detection system (NIDS), and host-based intrusion detection system (HIDS).

Although intrusion detection technology has been developed for many years, there are still serious problems such as higher FP and FN. Recently, with the booming development of machine learning, many artificial intelligence techniques have increasingly used in the intrusion detection field. Intrusion detection based on the classification method can extract the features of network flow and host session from an ocean of online data and audit data. In addition, it learns the classification model to discover the classification rules of hidden intrusion behavior in data. Some typical machine learning methods applied into intrusion detection are as follows: Decision tree [4], Naive Bayesian [5], k-nearest neighbor (k-NN) [6], semisupervised machine learning [11], and unsupervised machine learning and deep learning [12].

Shojafar et al. [7] proposed an unsupervised machine learning method. The method uses an automatic clustering algorithm to find the clusters with the maximum similarity between the proposed cluster elements and the smallest similarity with other clusters. The supervised learning

method requires a lot of label data [13]. But, the unsupervised learning method can automatically cluster without a large amount of labeled data. It can improve the situation where there is little labeling data in the field of cybersecurity. Meanwhile, it can improve the situation in which there is few label data in the field of network security. However, the accuracy of the above method is not high, and the detection ability is not enough in the face of unknown attacks.

The traditional IDS mostly uses individual classification techniques, which do not provide the best attack detection rate. The single-model approach is more difficult to accurately predict every type of invasion. Moreover, the generalization ability of the single model is insufficient, and the detection ability is not enough as facing unknown attacks.

A new two-stage hybrid classification method is proposed, in which support vector classification (SVM) is used as the first stage of anomaly detection [14] and artificial neural network (ANN) is used as the second stage of misuse detection. Its core method is to improve accuracy by integrating the respective advantages of the two models. The two-stage model further improves the detection capability. However, the types of two models are insufficient, i.e., the advantages of multiple models cannot be maximized.

Pierre-Francois Marteau proposed covering similarity and a new similarity measure [15]. The dormant attack sequences in the normal sequences within the scope of HIDS are separated by the above similarity. Two well-known coverage similarity and three similarity measures were compared and analyzed. It shows that the covering similarity is an important index for anomaly detection in system calls sequence.

The raise of next-generation information and communication technology has led to significant growth in the number of attack and intrusion. The IDS has some dimensional flaws that tend to add time complexity and reduce resource utilization. The intelligent IDS should analyze the important characteristics of the data to reduce dimensions. The feature extraction can solve the problem to find the most informative and compact feature set. Aiming at performing each single algorithm of feature extraction to the maximum extent and developing a novel intelligent IDS, Hussain et al. [16] realized a set of linear discriminant analysis (LDA) and principal component analysis (PCA) feature extraction algorithms. The overall PCA-LDA method generates the better results and shows a higher precision ratio than a single feature extraction method. The eigenvalue decomposition of the PCA algorithm has some limitations. The principal components obtained by the PCA method may not be optimal in the case of non-Gaussian distribution.

Aburomman and Reaz [17] designed a feature sorting model based on information correlation and gain. Then, the useful or useless features are identified by combining the levels obtained from information correlation and information gain, thereby to complete feature reduction. Next, it feeds the simplified features into the feedforward neural network. In Ref. [18], a deep learning method is introduced to learn the optimal characteristics of network connection and then to choose the memetic algorithm as the final classifier in order to detect abnormal traffic. The results of NSL-KDD and KDD CUP 99 datasets both show the detection rate of 98.11%, except for the detection rate of 92.72% of the R2L attack group in the NSL-KDD dataset. In order to solve the problem of distinguishing between attack traffic and normal data flow in big data, Jia et al. [19] proposed a new real-time DDoS attack detection mechanism. First, the multidimensional characteristics of network traffic are reduced by the PCA algorithm. Next, the correlation of lower dimensional variables is analyzed. In addition, Musafer et al. [20] also proposed a feature extraction method based on trigonometric simplexes for the IDS. Similarly, Taheri et al. [21] used Hamming distance of static binary features. Andresini et al. [8] proposed a multichannel deep feature learning method, and Jiang et al. [9] also use a hierarchical deep learning method. However, the deep learning relies heavily on data volume and data quality, and its model has a sea of parameters and strong adaptive ability. It is easy to cause overfitting on the small dataset, which results in lower performance than expected.

Nowadays, to improve the performance of intrusion detection systems, various machine learning methods have been widely used. As a hot method, the ensemble learning is paid growing attentions [22]. A classifier combination tactic is generally preferred to substitute a single classifier. Mohammadi and Namadchian [23] proposed a new integrated construction method, which used the weights generated by the particle swarm optimization (PSO) to create a classifier set for higher intrusion detection accuracy. Gu et al. [24] applied a logarithmic marginal density ratio transformation on the original features in order to obtain the newest and better-quality transformed training data and then to use SVM integration to establish an intrusion detection framework. Although there are many collection methods, it is still difficult to find a suitable collection configuration for a specific dataset.

With the rapid development of 5G, Big Data, Blockchain, and Industrial Internet, the active defense and endogenous security against network intrusion has become increasingly important. Jia et al. [25] proposed a new deep neural network model to apply to the IDS. The model with four hidden layers improves the detection rate and detection accuracy on the KDD and NSL dataset. However, the experiment showed that the accuracy of U2R is only 90.91%. Jiang and Zhou [26] invented an intrusion detection method based on asymmetric deep belief network (ADBN). The ADBN model can extract features that are more conducive to classification and save more test time in the model initialization stage. It would achieve better detection accuracy for small class samples. However, the overall detection rate of the dataset is relatively low. Lu et al. [27] proposed a method by using the deep self-encoder with unsupervised learning for migration learning. It is a positive exploration because there are still some inherent shortcomings of unsupervised learning.

In conclusion, the previous work mainly used simple dimensionality reduction algorithms, such as PCA to perform dimensionality reduction for all features only once. In addition, they use the single model to directly learn and classify features. The previous work focused on optimization and improvement of the single model, and the most of them

only focused on overall classification accuracy, while ignoring the accuracy on small samples. Similarly, the deep learning methods used in previous works often fail to obtain particularly ideal results due to insufficient dataset quantity and quality. In order to improve the above methods, this paper proposes an intrusion detection method based on DT-RFE in ensemble learning. Compared with the PCA algorithm, the DT-RFE has simpler application conditions and pays more attention to the actual effect. In addition, compared with other single-model methods and simple ensemble learning methods, our multilayer and multimodel Stacking method has stronger integration ability. The adaptive ensemble learning method based on machine learning models can better reflect the advantages of heteroid models.

3. Preliminaries

The process of sorting data in the data source into the data warehouse according to certain rules is called data preprocessing. In this paper, the original NSL-KDD and KDD CUP 99 datasets need be preprocessed to verify our method and model. On the one hand, the data in the original sample should be normalized, and the sample data are fabricated into the format that is suitable for calculation. On the other hand, some important features that affect the prediction result are selected by the feature selection algorithms, with the purpose of reducing data redundancy and computation complexity.

3.1. Data Preprocessing. There are four types of intrusions in the original dataset, and each intrusion record is made up of the 41-dimensional feature vector. The example of a raw record in the intrusion detection dataset is as follows:

$$\begin{aligned} X_i = \{&0, \text{tcp}, \text{http}, \text{SF}, 215, 45076, 0, 0, 0, 0, 0, 1, 0,\\ &0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00,\\ &1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,\\ &0.00, 0.00, 0.00, 0.00, 0.00, \text{normal}\}. \end{aligned} \tag{1}$$

X_i has 42 dimensions, including 41 attributes and one label. Here, "normal" is the label that records X_i. In addition, i in X_i means that X_i is a row of the dataset. The first step in data processing is data filtering. Many intrusion records are the same in actual captured data, so we remove duplicate data to eliminate information redundancy. Furthermore, the 3 features in them are character-type features, and they are "protocol_type," "service," and "flag." Therefore, we use LabelEncoder() to convert all the data captured into digital types from different IDS input sources to simplistically process the data. The target labels with values between 0 and n_classes-1 can be transformed into a continuous numeric variable by LabelEncoder(). As shown in Table 1, symbol features are mapped to digital features.

The difference of value range and metrics varies greatly among different features. For avoiding the disappearance of the small-valued attribute and to reduce the repetitive calculation of the iteration amount, the numerical data need be encoded by one-hot. The one-hot coding is used to represent the protocol_type, service, and flag attributes in X_i, and the 84-dimensional data are obtained. Here, the example obtained is shown in Table 2.

Table 1: Example of data numeralization.

	Protocol _type	Service	Flag
0	1	20	9
1	2	44	9
2	1	49	5
3	1	24	9
4	1	24	9

Because the NSL-KDD and KDD CUP 99 datasets are divided into five categories, each of which has a different amount of data, for example, denial of service (DoS) attacks and normal data have hundreds of thousands of data. With only thousands of data, this situation is called the uneven distribution of data categories. For the distribution, the common accuracy rate cannot be used as an indicator of the evaluation model. We rename each attack tag, i.e., normal = 0, DoS = 1, Probe = 2, R2L = 3, and U2R = 4. The rules are as follows:

(i) 'normal': 0

(ii) 'back': 1, 'worm': 1, 'land': 1, 'pod': 1, 'smurf': 1,'teardrop': 1, 'mailbomb': 1, 'apache2': 1, 'processtable': 1, 'Neptune': 1, and 'udpstorm': 1

(iii) 'portsweep': 2, 'satan': 2, 'ipsweep': 2, 'mscan': 2, 'saint': 2', and nmap': 2

(iv) 'guess_passwd': 3, 'multihop': 3, 'phf': 3, 'warezclient': 3, 'httptunnel': 3, 'warezmaster': 3, 'sendmail': 3, 'named': 3, 'snmpgetattack': 3, 'snmpguess': 3, 'xlock': 3, 'ftp_write': 3, 'imap': 3, 'spy': 3, and 'xsnoop': 3

(v) 'loadmodule': 4, 'buffer_overflow': 4, 'ps': 4, 'perl': 4, 'rootkit': 4, 'sqlattack': 4, and 'xterm': 4

After encoding the 41-dimensional feature vector by one-hot, X_i becomes 122-dimensional feature vectors. Next, we normalize the dataset and turn all data into the same value interval. Normalization can prevent some large-value features from erroneously affecting the results. The data are converted from X_{ij} to X_{ij}', where i and j represent the rows and columns of the data in the dataset. The function is shown as follows:

$$\begin{aligned} X_{ij}' &= \frac{X_{ij-\text{AVG}_j}}{\text{STAD}_j},\\ \text{AVG}_j &= \frac{1}{n}\left(X_{1j} + X_{2j} + \cdots + X_{nj}\right),\\ \text{STAD}_j &= \frac{1}{n}\left(\left|X_{1j} - \text{AVG}_j\right| + \left|X_{2j} - \text{AVG}_j\right| + \cdots + \left|X_{nj} - \text{AVG}_j\right|\right), \end{aligned} \tag{2}$$

where the average value is AVG_j and STAD_j is the average absolute deviation.

Table 2: Example of data one-hot encoding result.

	Protocol_type_icmp	Protocol_type_tcp	Protocol_type_udp	Service_IRC	Service_X11	...	Flag_RSTR	Flag_S0	Flag_SF	Flag_SH
0	0	1	0	0	0	...	0	0	1	0
1	0	0	1	0	0	...	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...
125968	0	1	0	0	0	...	0	1	0	0
125969	0	1	0	0	0	...	0	0	1	0

In the above calculations, the following judgments are required:

$$X_{ij}' = \begin{cases} \dfrac{X_{ij} - \mathrm{AVG}_j}{\mathrm{STAD}_j}, & \text{if } \mathrm{AVG}_j > 0, \\ 0, & \text{if } \mathrm{AVG}_j = 0. \end{cases} \tag{3}$$

After this function, the normalization of the dataset and the preliminary work of the dataset have been completed.

3.2. Feature Extraction. After data preprocessing is completed, it is usually impossible to directly input the data into a learner due to the high dimensions of data, so it is necessary to select some fewer valuable features to train by machine learning. Good feature extraction can find the most useful and important features.

In order to speed up the subsequent training algorithm, in the above obtained 122-dimensional feature data, the main characteristics of DoS, Probe, R2L, and U2R were found by reducing the dimensions or extracting features. In this paper, a novel DT-RFE is used. Here, RFE is to iteratively build the model and then pick the best (or worst) features that are selected according to the coefficients. Next, the selected features are put aside, and it repeats on the remaining features. The process continues until all the features are traversed. The eliminated sequences in this process are the ordering of features. The character of the RFE itself allows us to better perform manual feature selection. The stability of the RFE depends largely on which the model is used at the bottom during iteration. If the relationship between a feature and a response variable is nonlinear, a tree-based method or an extended linear model can be used. Usually, some tree-based methods are easier to use; this is because they model nonlinear relationships and do not require much debugging. In feature extraction, the underlying model is selected as a simple Decision Tree algorithm.

In addition, information entropy is an important index of feature selection in Decision Tree. There are many types of sample in the training dataset that need to be classified. Decision Tree calculates the information entropy of the dataset and divides the dataset layer by layer. Finally, each type of sample is divided separately. Entropy is the measure of the uncertainty of a random variable. Suppose that X is a random variable with a finite number of values, and its probability distribution is denoted as follows:

$$P\left(X = x_i\right) = p_i. \tag{4}$$

Among them, x_i corresponds to p_i one by one. The entropy of the random variable X is defined as follows:

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i. \tag{5}$$

It is not difficult to find that the uncertainty of the random variable is greater, while the entropy is greater. Because the probability value must be less than 1, the logarithm of this probability must be less than 0. So, there is a minus sign in the formula to counteract the negative number produced by log. When the difference of p_i corresponding to different x_i is greater, the entropy $(H(X))$ is greater.

Similarly, suppose that the joint probability distribution of random variable (X, Y) is expressed as follows:

$$P\left(X = x_i, Y = y_j\right) = p_{ij}. \tag{6}$$

Conditional entropy $H(Y|X)$ represents the uncertainty of the random variable Y under the given condition X, and it is calculated as follows:

$$H(Y|X) = \sum_{i=1}^{n} p_i H\left(Y|X = x_i\right). \tag{7}$$

The information gain of feature A to training dataset D is $G(D|A)$. The formula is shown as follows:

$$G(D|A) = H(D) - H(D|A). \tag{8}$$

The information gain represents the degree to which the inaccuracy of Y information is reduced after the information of feature X is learned. Using $G(D|A)$ as a feature to divide the dataset may cause the problem of preferring to select features with more values. So, information gain ratio is another criterion of feature selection, which can correct the above problem:

$$G_R = \frac{G(D|A)}{H(D)}. \tag{9}$$

Suppose that the number of leaf nodes in Decision Tree T is $|T|$, and t is one of the leaf nodes. There are N_t samples in this node, and the number of sample points of k is N_k. H_t is the entropy on the leaf node, and $\alpha(\geq 0)$ is an optional parameter related to the penalty term. So, the loss function of Decision Tree T is defined as follows:

$$L_\alpha(T) = \sum_{t=1}^{|T|} N_t H_t(T) + \alpha|T|, \tag{10}$$

where the calculation of entropy is as follows:

$$H_t(T) = -\sum_{k} \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t}. \tag{11}$$

The loss function (it is also called as objective function) is used to evaluate the difference degree between the true value and the predicted value. The goal of model learning is to reduce the loss function.

The first term on the right side of the equal sign in the loss function (10) can be defined as follows:

$$C(T) = \sum_{t=1}^{|T|} N_t H_t(T) = -\sum_{t=1}^{|T|} \sum_{k=1}^{K} N_{tk} \log \frac{N_{tk}}{N_t}. \tag{12}$$

In the case, the loss function can be simplified as follows:

$$C_\alpha(T) = C(T) + \alpha|T|, \tag{13}$$

where the $C(T)$ represents the prediction error of the model to the training data. In addition, the complexity of the model is also important. $|T|$ represents the complexity of the model,

which can be regarded as a penalty term in the loss function. In addition, α can determine the degree of penalty and can balance the model complexity and prediction error.

In our proposed method, the Recursive Feature Elimination Cross-Validation (RFECV) uses cross-validation based on RFE, and it is to preserve the most representative characteristics. The cross-validation based on RFE is performed on different feature combinations. By calculating the sum of its decision coefficients, the score with importance of different features is finally got, and then the best feature combination is retained.

As shown in Algorithm 1, the REF method uses Decision Tree as the training of multiple rounds. According to the weight coefficients generated by training, better features are retained for the next round of training. For prediction models with feature weights, RFE recursively reduces the size of feature set under review to select features. Firstly, based on the original features, the prediction models are trained to assign a weight to each feature. And then, the feature set can be simplified through deleting the features whose weight has the smallest absolute value. Such recursion will continue until the number of remaining features reaches the required number. Compared with RFE, RFECV adds a cross-validation process to better select the optimal number of features. For a feature set with d, the number of all its subsets is 2^{d-1} (including the empty set). The Decision Tree calculates the validation error of all subsets and selects the subset with the smallest error as the selected feature.

4. Model Building

The Stacking fusion algorithm by combining DT-RFE is creatively proposed, and its implementation consists of three stages. Firstly, the dataset should be prepared and normalized. Next, a feature extraction for dimension reduction is built.

After the feature extraction, a machine learning algorithm is used to classify and verify the dataset, and finally the ensemble learning is used to generate the generic function classifier. Here, the ensemble learning will test a series of classifiers to integrate the learning results through some rules so that it can obtain better generalization performance than a single learner.

However, there are still two main problems of the integrated algorithm: one is how to select some individual learners and the other is how to choose the strategies to integrate these individual learners into a powerful learner. A good integrated algorithm is to ensure the diversity of individual learners (excellent and different), and the integration of unstable algorithms can also get a significant performance improvement. Common kinds of integration learning are as follows: (1) bagging for reducing variance, (2) boosting for reducing bias, and (3) stacking for improving prediction results.

In this paper, Stacking is used as a powerful ensemble learning model to apply to the fusion algorithm. Stacking was proposed by Wolpert [28] in 1992. Its basic idea is to use a model to fuse the prediction results of several single modules in order to reduce the generalization error of the single individual. Unlike the voting and weighting methods used by the bagging and boosting algorithms, in order to obtain the weight value of each basic classifier, the Stacking algorithm will train another classification model that can learn the weight value of each classifier. Therefore, these single modules are called primary classifiers, and the Stacking fusion model is called the secondary (or meta [29]) classifier. As shown in Figure 1, Stacking first trains several single classifiers from the initial training set, then integrates the output of the single module as sample features, and uses the original sample labels as new data sample labels in order to generate a new training set. Subsequently, a new model for the new training set is trained, and finally the new model is adopted to predict the samples. The model of the Stacking fusion algorithm is essential to design a hierarchical structure, and each layer contains a sea of classification models. All single classification models are generated using different learning algorithms (some heterogeneous models). Algorithm 2 shows the operation flow of the Stacking fusion model.

Here, we randomly divide the initial training set D into k similarly-sized sets $D_1, D_2, \ldots, D_k$, D_j. In addition, $\overline{D_j}$ represents the j-th testing set and training set, respectively. When the T primary learning algorithms are given, the primary learner $h_{jt}$is obtained by using the t-th learning algorithm on $\overline{D_j}$. For each sample x_i in $\overline{D_j}$, we define the prediction result of the t-th model as $Z_{jt} = h_{jt}(x_i)$. Then, the secondary model training sample feature generated by sample x_i is $Z_i = Z_{i1}, Z_{i2}, \ldots, Z_{iT}$. The sample label is still the original sample, and it is labeled as y_i. Therefore, after k-round T model training and prediction, the secondary training set $D' = \{(Z_i, y_i)\}_{i=1}^{m}$ is obtained, and then $D'$is used to train the secondary model. The secondary model h' is a function of $(Z_{i1}, Z_{i2}, \ldots, Z_{iT})$ for y.

In the training phase of Stacking, the training set of the secondary model is generated by using the primary model. If the first-level model is used to directly predict the initial training set samples to generate a second-level training set, there will be a great risk of overfitting. Therefore, the unused samples that are used to train the first-level model will generate the training set of the second-level model. In general, the cross-validation is a more common method. Next, we use k-fold cross-validation as an example to show how the training set of the secondary model is acquired.

Figure 1 shows the process that constructs a fused model training set for a single module through 5-fold cross-validation. It is seen that the whole of 5 times are trained, and predictions are made on the five 1-fold verification sets and the test set, respectively. By concatenating the prediction results of five 1-fold verification sets and averaging the prediction results of five test sets, a list of features on the new training set and the new test set can be obtained.

Here, we first try to choose some simpler models, including Logistic Regression, Random Forest [30], SVM, and Decision tree. After analysis, it is found that these models do not have outstanding utilization of features. Therefore, some mainstream models such as AdaBoost and Gradient Boosted Decision Tree (GBDT) have been tried to enhance the nonlinear expression ability of the models. Aiming at

Input: Training sample set
(1) Initialize original feature set $S = \{1, 2, \ldots, D\}$ and feature ordering set $R = []$.
(2) **for** $d = 1, 2, \ldots, D$ **do**
(3) The Decision Tree classifier is trained, and the feature selection of single variable by F-test (ANOVA) is obtained.
(4) Calculate ranking criterion score
(5) Find the feature with the lowest ranking score
(6) Update feature set $R = [p, R)$
(7) Remove other features in S: $S = S/p$
(8) **until s** = [ ]
(9) **end for**
output: Feature sort set R

Algorithm 1: The DT-RFE algorithm.

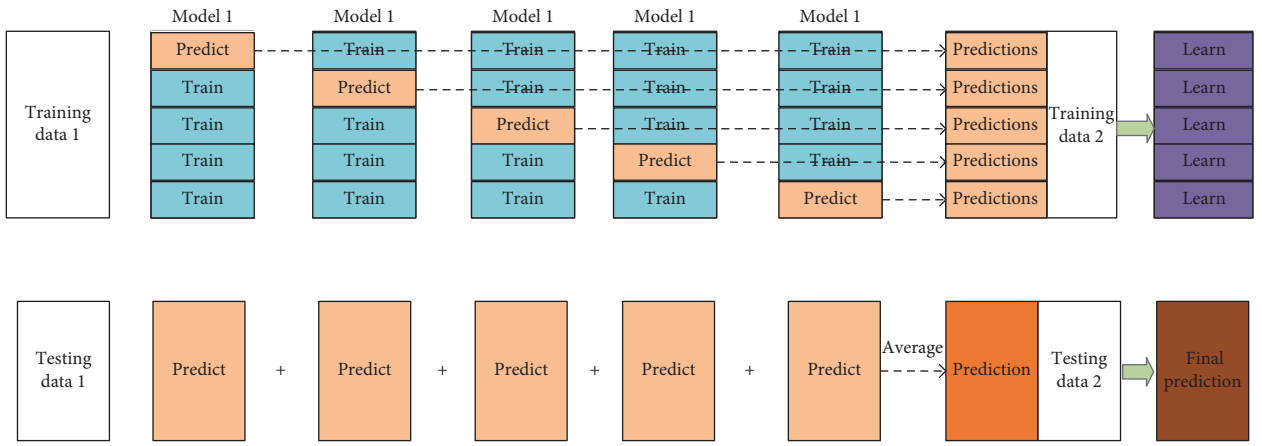

Figure 1: 5-fold cross-validation process for building a fusion model training set.

Input: D = $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_\mathbf{m}, \mathbf{y}_\mathbf{m})\}$; the single model learning algorithm $\delta_1, \delta_2, \ldots, \delta_\mathbf{T}$; and the fusion learning algorithm model δ
Steps:
(1) **for** $t = 1, 2, \ldots, T$ **do**
(2) $h_t = \delta_t(D)$;
(3) **end for**
(4) $D' = \varnothing$
(5) **for** $i = 1, 2, \ldots,$ **m do**
(6) **for** $t = 1, 2, \ldots,$ **T do**
(7) $Z_{it} = h_t(x_i)$;
(8) **end for**
(9) $D' = D' \cup ((Z_{i1}, Z_{i2}, \ldots, Z_{iT}), y_i)$;
(10) **end for**
(11) $h' = \delta(D')$;
Output: $H(x) = h'(h_1(x), h_2(x), \ldots, h_T(x))$

Algorithm 2: Stacking fusion model.

improving the accuracy and recall of the model, different types of machine learning methods are used for stacking in the end, and it will give full play to the greatest advantages of each model to improve the stability of the model and to avoid the bias of a single model. Among the attacks, DoS attack and Probe use three types of machine learning methods, i.e., Random Forest, AdaBoost, and GBDT as first-level models. R2L uses Decision Tree, Random Forest, and GBDT as first-level models. U2R uses Decision Tree, AdaBoost, and GBDT as first-level models. The Decision Tree is used as the whole secondary model. Compared with other types of the integrated model, the Stacking algorithm has a stronger ability of nonlinear

expression. When the meta-model with learning weights is increased, it will help to reduce the generalization error.

Stacking is a hierarchical structure which contains at least two layers. The last layer contains only one model classifier, which is an ensemble of the above models. The computational complexity of Stacking depends on the selected basic classifiers, such as Decision Tree, SVM, or deep learning model. The models must be serial between layers but can be parallel within layers. So, the Stacking's model complexity is the sum of complexity of the most complex model in each layer:

$$o(\text{Stacking}) = \max[o(M_{11}), \ldots, o(M_{1m})] + \cdots + o(M_n). \tag{14}$$

The overall framework of the model proposed in Figure 2 consists of three steps.

5. Experiment and Analysis

5.1. Attack Description in Dataset. This paper uses the KDD CUP 99 [31] and NSL-KDD [32] datasets to demonstrate the superiority of Stacking with the DT-REF method. KDD CUP 99 was created in the United States of America by the Defense Advanced Research Projects Agency (DARPA) of the US Department of Defenses' MIT Lincoln Laboratory [33].

NSL-KDD improves and optimizes the existing defects in KDD CUP 99 which are mentioned in [32]. Training set and test set of the NSL-KDD has a rational distribution ratio. The redundant records in the original dataset are solved, so it will not cause the classifier to bias records with a large amount of data. Therefore, the achievements in different papers on the NSL-KDD can be reasonably compared. Because KDD CUP 99 is very classic in the field, this paper retains the experiment on it.

The NSL-KDD and KDD CUP 99 datasets divide various attacks into four categories and are described as follows:

(i) DoS: DoS attacks disturb normal access to network services. Its main purpose is to make network resources unable to serve normal network requests by completely utilizing memory resources, thereby it denies users to access to services. In addition, DoS includes such attacks as smurf, neptune, ping of death, back, and so on.

(ii) Probe: Before launching an attack, hackers first need to scan the target website to collect and analyze target information. In addition, the information is searched and used in order to find the known vulnerabilities in target objects. Hackers will use the information to launch accurate and efficient attacks. These scanning and analysis tasks are called Probe, for example, satan, portsweep, nmap, and so on.

(iii) Remote to local (R2L): An attack that illegally accesses local resources from an external network is called R2L. Through the Internet, users can send the packets to remote terminals. But, they have no right to disclose vulnerabilities and take advantage of the permissions that local users possess on the computer. In addition, R2L includes ftp_write, phf, multihop, and so on.

(iv) User to root (U2R): This type of attack is generally referred to as privilege escalating and an attack that illegally promote common users' permissions to administrator privileges. Attackers use the vulnerabilities to increase their ordinary permissions to root permissions. U2R comprises perl, rootkit, and so on.

5.2. Evaluation Indicators. The key to measuring the quality of an intrusion detection system is whether it has a high detection rate and a low FP rate. Therefore, in order to evaluate the performance of the intrusion detection model proposed, in this paper, accuracy (ACC), precision (PRE), detection rate (DR), and F1-Score (f_1) are selected. The accuracy rate is used to measure the accuracy of the classification. The detection rate is used to measure the percentage of abnormal behaviors that are detected. The accuracy is used to measure how many use cases in the test results are abnormal behaviors. In addition, the f_1 is used to comprehensively judge DR and ACC. The corresponding calculation formulas are shown as follows:

$$\begin{aligned} \text{ACC} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\ \text{PRE} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{DR} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ f_1 &= \frac{2 * \text{DR} * \text{PRE}}{\text{DR} + \text{PRE}}, \end{aligned} \tag{15}$$

where True Positive (TP) is the number of positive samples correctly identified, True Negative (TN) is the number of positive samples correctly identified, False Positive (FP) is the number of positive samples identified by mistake, and False Negative (FN) is misidentified that the number of negative samples.

Due to the influence of noise, there will be a certain bias between the training set and the test set, which often causes the model to obviously perform very well on the training set, but the performance on the test set is greatly reduced. For cross-validation, this paper uses 2-, 5-, 10-, 30-, and 50-fold cross-validation for comparison.

5.3. Analysis of Experimental Results. The environment and tool information needed for experiment are shown in Table 3.

Each step of RFECV yields a specific number of characteristics of ACC so that the minimum number of features can be obtained as ACC reaches a maximum. At the same time, RFECV can pick out the selected features. Therefore, RFECV is used to select the optimal number of features, as shown in Figure 3. The features of the final selection of DoS, Probe, R2L, and U2R are shown in Table 4.

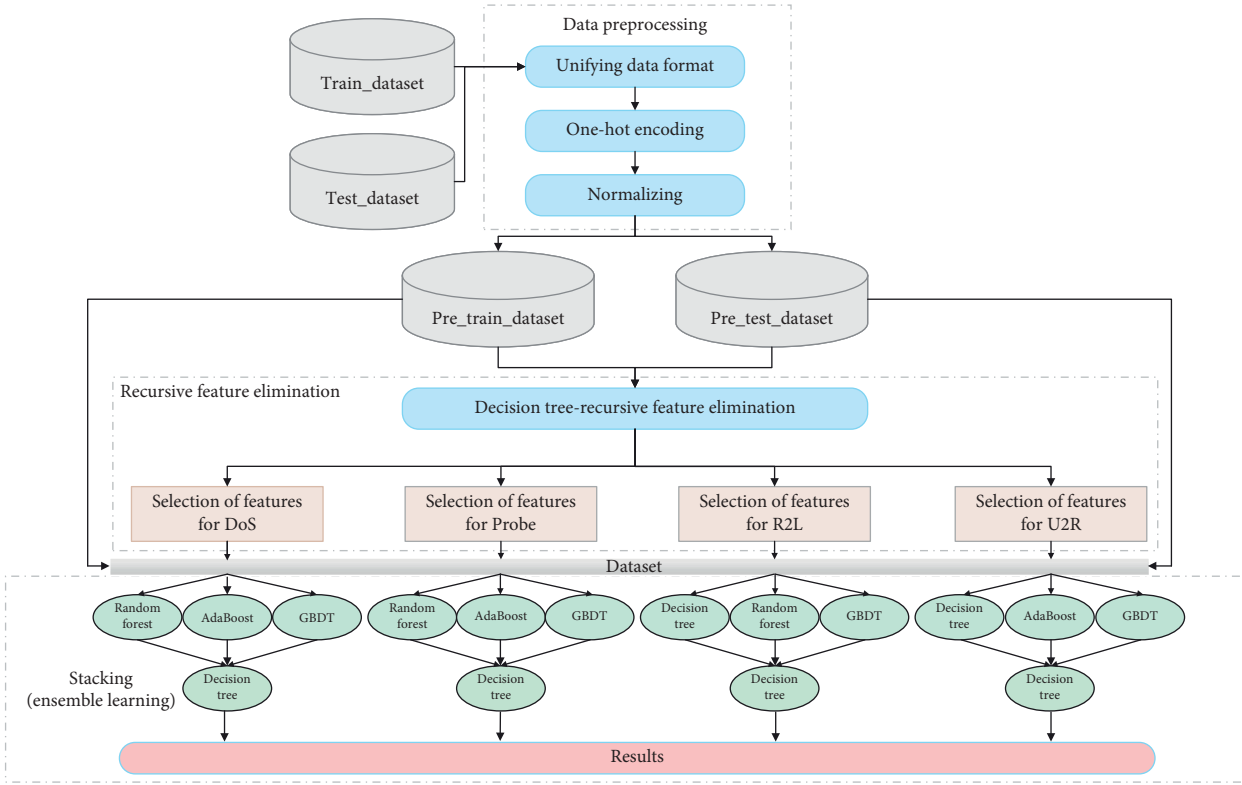

Figure 2: The overall framework of the proposed model.

Table 3: Experimental environment.

Project	Environment
Operating system	Ubuntu 18.04 LTS
CPU	I5-9300H
Memory	16G
Python	3.6.8
Scikit-learn	0.21.2
Anaconda	4.8.3
Mlxtend	0.17.2

The x-axis of Figure 3 represents the number of features currently selected by REFCV in the recursive process, and the y-axis represents the cross-validation score under the number. DT-RFE calculates the ranking of all features' importance of result under the current goal. It can be seen that different optimal numbers of features are required for the prediction of different attack types. In other words, the predicting different types of targets requires the different features to get better results. With the ranking and optimal number of features, we can obtain the required features like Table 4. As shown in Figure 3, we conduct experiments on NSL-KDD and KDD CUP 99, respectively. On the result of feature selection, we take NSL-KDD as an example to show it and as shown in Table 4.

Firstly, the feature learning is performed by using the marked training set as input to the IDS. After several experiments, the optimal parameters were selected and saved, and the IDS model trained was retained. Then, the remaining data form a test dataset to measure the detection accuracy the model preforms on each attack. The experimental results on NSL-KDD and KDD CUP 99 datasets are shown in Figure 4 and Tables 5 and 6.

Tables 5 and 6 show the accuracy of each model for all attack types, and the last column is the accuracy of ensemble learning (Stacking). It can be seen that the traditional Logistic Regression and SVM have low DR for the four attack types, and the classification accuracy is not high. The AdaBoost algorithm is a bit worse than other ensemble learning algorithms. In addition, Stacking has the highest accuracy among all models.

Figure 4 shows a more detailed visualization in Table 6. It not only includes accuracy of each model but also corresponding precision, detection rate, F1-score, and 5-fold accuracy. These indicators can evaluate the model comprehensively. It can be seen that any score of our Stacking method is higher than the other models. In Figure 4, for the R2L attack type, no matter which algorithm is selected, except ACC, other detection indicators are not high. The experimental results in using the stacking algorithm show that our method can maintain high detection accuracy for all kinds of attack. Except for the lower accuracy of R2L, the others reach more than 99%, especially for small samples.

In Figure 4, we show the results of the 5-fold cross-validation, and the results obtained on the 2-, 10-, 30-, and 50-fold are similar. Therefore, only the most representative 5-fold cross-validation results are shown in the figure. This shows that our method not only achieves better results on the training set but also achieves better results on the unlearned testing set. It reflects that our method is not only

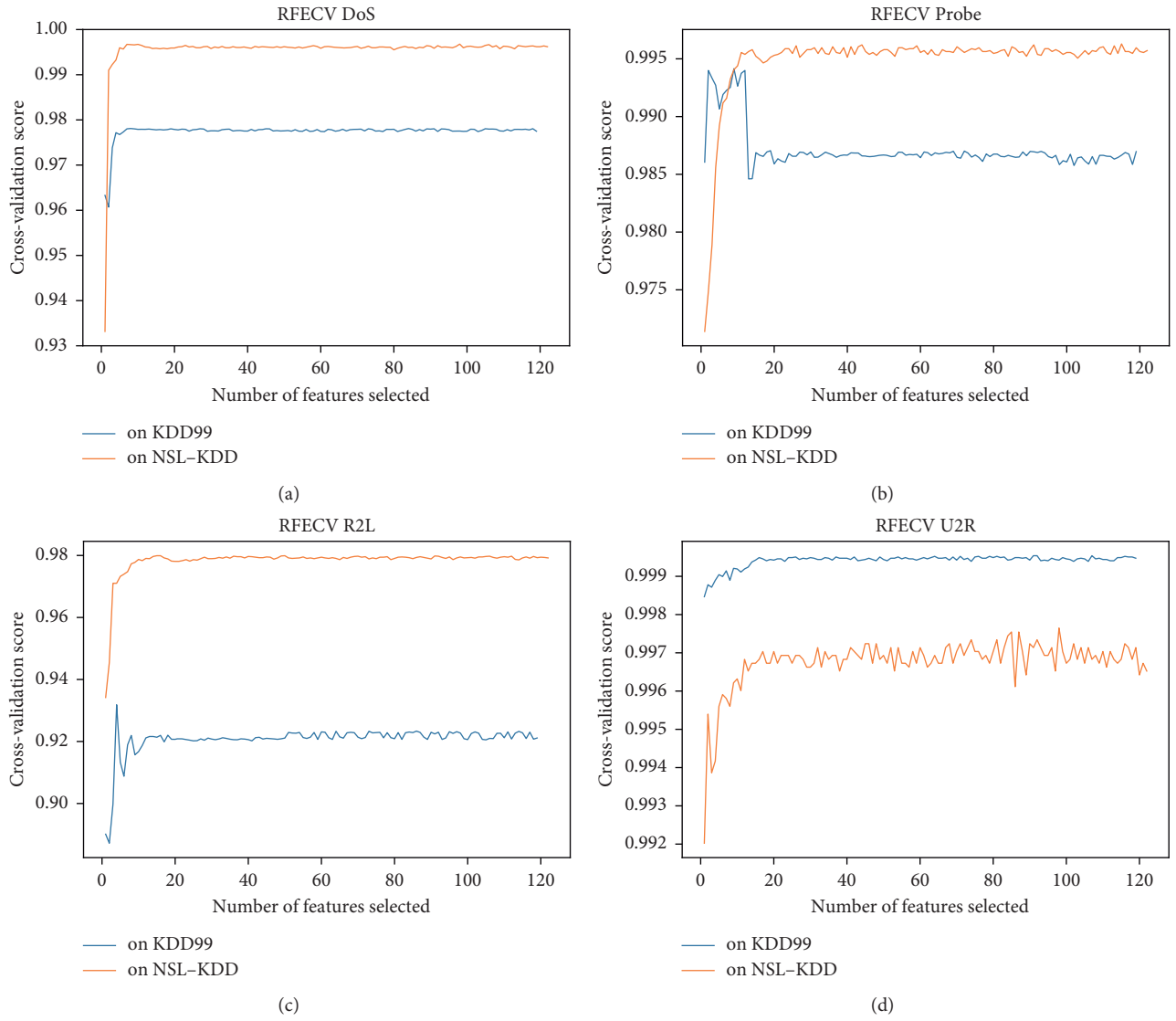

Figure 3: Graphs generated by RFECV based on the ACC of specific quantitative features: (a) RFECV DoS, (b) RFECV Probe, (c) RFECV R2L, and (d) RFECV U2R.

Table 4: Features of DoS, Probe, R2L, and U2R on NSL-KDD

DoS	'dst_host_same_srv_rate'; 'dst_host_serror_rate'; 'num_compromised'; 'same_srv_rate'; 'diff_srv_rate'; 'dst_host_count'; 'dst_host_srv_serror_rate'; 'ecr_i'; 'RSTR'; 'wrong_fragment'; 'dst_bytes'; 'src_bytes.'
Probe	'src_bytes'; 'telnet'; 'smtp'; 'private'; 'http'; 'ftp_data'; 'finger'; 'dst_host_rerror_rate'; 'dst_host_same_src_port_rate'; 'dst_host_diff_srv_rate'; 'dst_host_same_srv_rate'; 'rerror_rate'; 'dst_bytes.'
R2L	'duration'; 'imap4'; 'ftp_data'; 'dst_host_srv_diff_host_rate'; 'dst_host_same_src_port_rate'; 'dst_host_same_srv_rate'; 'dst_host_srv_count'; 'dst_host_count'; 'num_access_files'; 'num_failed_logins'; 'hot'; 'dst_bytes'; 'src_bytes.'
U2R	'duration'; 'dst_host_srv_diff_host_rate'; 'dst_host_count'; 'srv_count'; 'num_shells'; 'num_file_creations'; 'root_shell'; 'dst_bytes'; 'dst_host_same_srv_rate'; 'hot'; 'src_bytes.'

accurate in each category but also has good generalization performance.

Table 7 and Figure 5 show the classification accuracy of our method and these methods in [8, 9, 16, 18, 25, 34]. The latest paper [9] does not provide the accuracy of each category, so their overall accuracy is used as the average. Our proposed Stacking with the DT-RFE method outperforms the other methods in terms of the ACC of DoS, Probe, and U2L, especially U2R. Compared with the previous methods, this paper divides four different subsets on the NSL-KDD and KDD CUP 99 datasets for training and extracts unique feature attributes for each type of attack. The detection result is shown in Figure 5, but the simple machine learning algorithm is as high as 80%. In detection of U2R and R2L

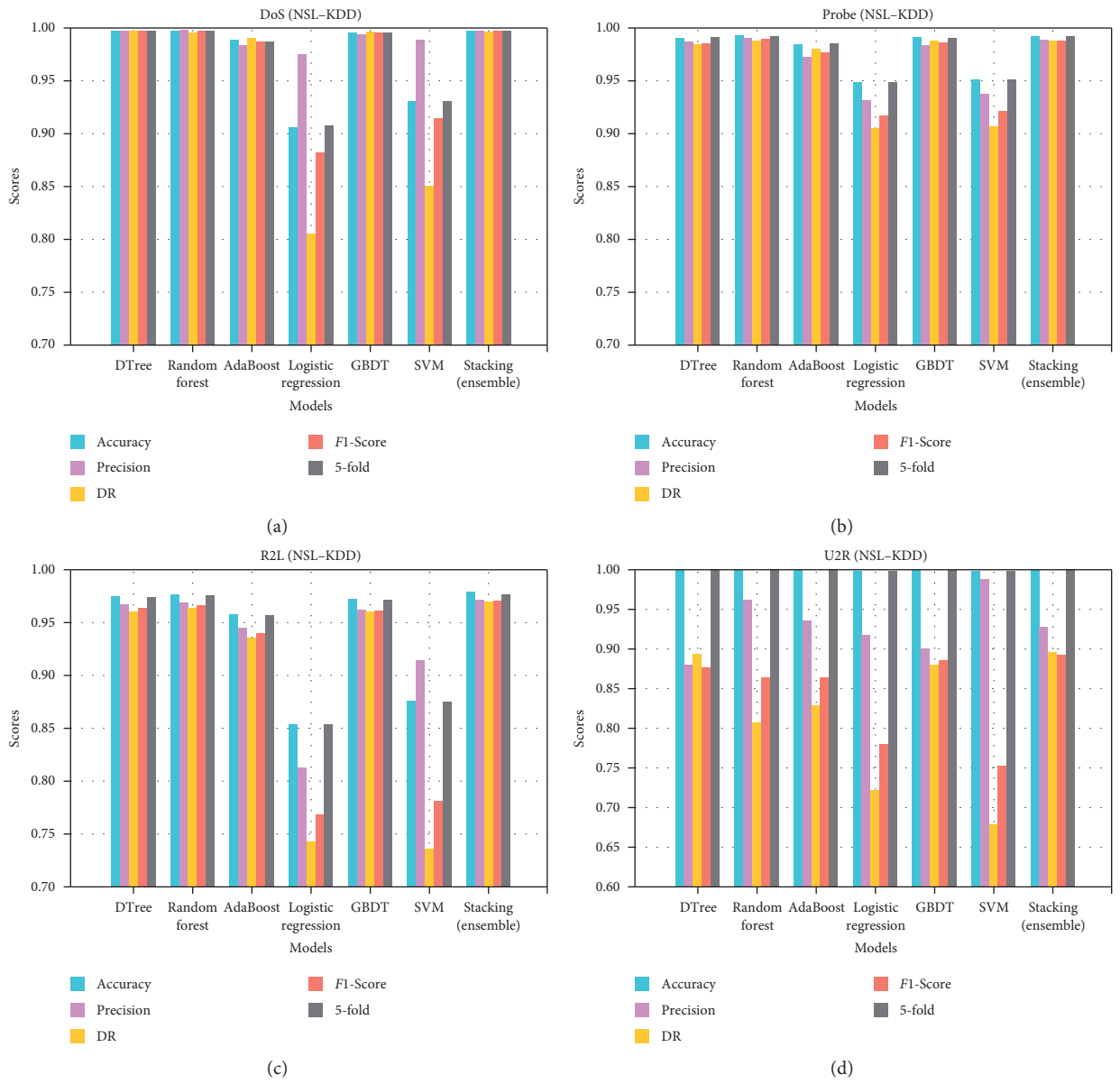

Figure 4: Comparison results in ACC, PRE, DR, and F1-score: (a) DoS, (b) Probe, (c) R2L, and (d) U2R.

Table 5: Accuracy of several machine learning algorithms on KDD CUP 99.

	Decision tree	Random forest	AdaBoost	Logistic regression	GBDT	SVM	Stacking
DoS	0.98944	0.9975	0.98899	0.90641	0.99598	0.9307	0.99755
Probe	0.99521	0.99316	0.98483	0.94881	0.99093	0.95145	0.99412
R2L	0.94047	0.97626	0.95816	0.85392	0.97253	0.87607	0.97920
U2R	0.99652	0.99693	0.99683	0.9955	0.99693	0.9955	0.99744

Table 6: Accuracy of several machine learning algorithms on NSL-KDD.

	Decision tree	Random forest	AdaBoost	Logistic regression	GBDT	SVM	Stacking
DoS	0.99802	0.99689	0.98682	0.92374	0.99451	0.95318	0.99744
Probe	0.98975	0.98231	0.98337	0.95274	0.98988	0.94291	0.99204
R2L	0.97004	0.96664	0.96385	0.90723	0.97924	0.91325	0.98207
U2R	0.98973	0.99601	0.98937	0.99644	0.99702	0.98961	0.99765

Table 7: Accuracy for each class of different methods.

Author	Dataset	DoS	Probe	R2L	U2R	Average
Our method	KDD CUP 99	0.9976	0.9941	0.9792	0.9974	**0.9921**
Our method	NSL-KDD	0.9974	0.9920	**0.9821**	**0.9977**	**0.9923**
Hussain	NSL-KDD	**1.0000**	**0.9990**	0.7740	0.8860	0.9148
Akashdeep	KDD CUP 99	0.9993	0.9879	0.9190	0.8660	0.9431
Jia	KDD CUP 99	0.9990	0.9818	0.9706	0.8182	0.9424
Jia	NSL-KDD	0.9867	0.9773	0.9694	0.8182	0.9379
Sun	KDD CUP 99	0.9741	0.9313	0.1124	0.2542	0.5680
Andresini	KDD CUP 99	—	—	—	—	0.9249
Jiang	NSL-KDD	0.9621	0.6856	0.6045	0.6132	0.8358

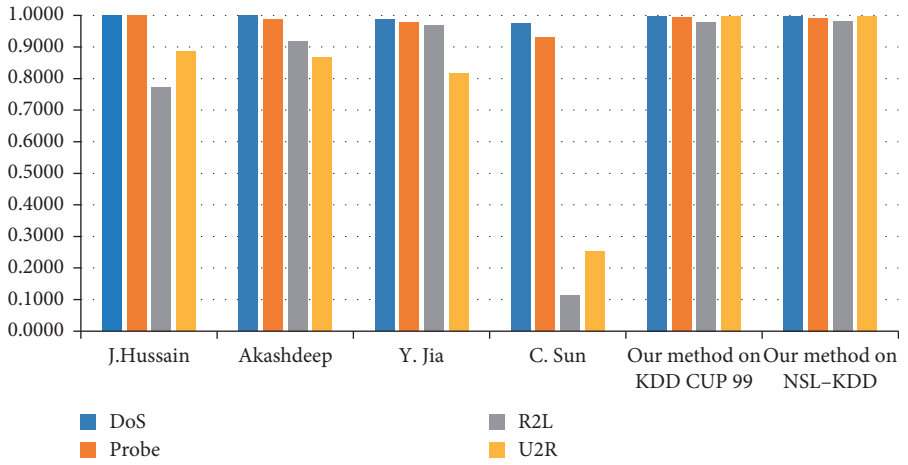

Figure 5: Our method compared with other previous methods.

features, it far exceeds the detection results of the algorithms proposed by Sun et al. [34] and Hussain et al. [16].

Although Akashdeep et al. [18] also proposed feature reduction by feature ordering based on information gain and correlation, however, the preprocessing work is done manually from the levels of information gain and correlation to identify useful and useless features. Jia et al. [25] classified records of NSL-KDD and KDD CUP 99 datasets with a deep learning method, but the U2R testing results are lower.

Overall, our method has better performance in NSL-KDD and KDD CUP 99 datasets. Our method has higher accuracy in the detection of attacks. In order to show our improvement, the average accuracy of multiple attacks is calculated in Table 7. In addition, Figure 5 shows the visualization in Table 7, which obviously reflects that our method has a very balanced accuracy for each type of attack.

6. Conclusions and Future Scope

In this paper, a novel intelligent intrusion detection system based on Stacking is proposed, and it used a DT-RFE algorithm to extract less features. Our method can improve and optimize the dataset and increase the resource utilization through deleting uncorrelated and redundant records. When the accuracy of a single machine learning model is difficult to improve, Stacking can be used to stack machine learning models to improve accuracy. Overall, our method has higher DR and lower FPR and has higher recognition accuracy for each type of attack. The designed IDS shows that feature reduction can reduce the system size and shorten training time. The lower time complexity resulted through the above measures can make the system performance better. Our method in the IDS can execute security functions in networks, organizations, and social groups with critical security.

Although the current work can optimize the feature set and acquire some excellent achievements, the R2L detection rate is still less than ideal, and the accuracy, DR, and FPR are relatively low. It can improve the disadvantages through a variety of extensions. According to the interactive network intrusion detection data 3D visualization method proposed in [35], it geometrically visualizes the relationship between every two different types of network traffic so as to explain the composition of the intrusion detection dataset in a more intuitive way. By using a fast convergence learning algorithm to fast check DR, the performance of the system can be further improved. And how we apply stacking into unsupervised learning is also a future work that needs to be explored. The above research will be the focus of our future work.

Data Availability

The data used to support the results of this study are provided in given links in this article. The datasets can be obtained from the online websites [36, 37]: KDD

CUP 99 dataset, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [36], and NSL-KDD dataset, https://www.unb.ca/cic/datasets/nsl.html [37].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the Scientific Research Foundation of Shandong University of Science and Technology for Recruited Talents, grant no. 0104060511314, and the National Key Research and Development Program of China, grant no. 2017YFC0804406.

References

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

[2] National Computer Network Emergency Technical Processing Coordination Center, *The 2018 China Internet Network Security Report*, People's Posts and Telecommunications Press, Beijing, China, 2019.

[3] L. N. Tidjon, M. Frappier, and A. Mammar, "Intrusion detection systems: a cross-domain overview," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3639–3681, 2019.

[4] J. H. Lee, J. H. Lee, S. G. Sohn et al., "Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system," in *Proceedings of the 10th International Conference on Advanced Communication Technology*, pp. 1170–1175, Gangwon-Do, Republic of Kore, February 2008.

[5] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive bayes vs. decision trees in intrusion detection systems," in *Proceedings of the 2004 ACM Symposium on Applied Computing*, pp. 420–424, Nicosia, Cyprus, March 2004.

[6] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[7] M. Shojafar, R. Taheri, Z. Pooranian, R. Javidan, A. Miri, and Y. Jararweh, "Automatic clustering of attacks in intrusion detection systems," in *Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–8, Abu Dhabi, UAE, May 2019.

[8] G. Andresini, A. Appice, N. D. Mauro, C. Loglisci, and D. Malerba, "Multi-channel deep feature learning for intrusion detection," *IEEE Access*, vol. 8, pp. 53346–53359, 2020.

[9] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, vol. 8, pp. 32464–32476, 2020.

[10] W. Liang, K.-C. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2020.

[11] H. Yao, D. Fu, P. Zhang, M. Li, and Y. Liu, "MSML: a novel multilevel semi-supervised machine learning framework for intrusion detection system," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1949–1959, 2019.

[12] K. Kim, M. E. Aminanto, and Tanuwidjaja, *Network Intrusion Detection Using Deep Learning: A Feature Learning Approach*, Springer, Berlin, Germany, 2018.

[13] V. P. Singh, R. Pathak, S. Tiwari, and K. Kaur, "Content-based image retrieval based on supervised learning and statistical-based moments," *Modern Physics Letters B*, vol. 33, no. 19, Article ID 1950213, 2019.

[14] A. A. Aburomman and M. B. Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360–372, 2016.

[15] P.-F. Marteau, "Sequence covering for efficient host-based intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 994–1006, 2019.

[16] J. Hussain, S. Lalmuanawma, and L. Chhakchhuak, "A two-stage hybrid classification technique for network intrusion detection system," *International Journal of Computational Intelligence Systems*, vol. 9, no. 5, pp. 863–875, 2016.

[17] A. A. Aburomman and M. B. I. Reaz, "Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection," in *Proceedings of the 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 636–640, Xi'an, China, October 2016.

[18] I. M. Akashdeep, I. Manzoor, and N. Kumar, "A feature reduced intrusion detection system using ANN classifier," *Expert Systems with Applications*, vol. 88, pp. 249–257, 2017.

[19] B. Jia, Y. Ma, X. Huang, Z. Lin, and Y. Sun, "A novel real-time DDoS attack detection mechanism based on MDRA algorithm in big data," *Mathematical Problems in Engineering*, vol. 2016, Article ID 1467051, 10 pages, 2016.

[20] H. Musafer, A. Abuzneid, M. Faezipour et al., "An enhanced design of sparse autoencoder for latent features extraction based on trigonometric simplexes for network intrusion detection systems," *Electronics*, vol. 9, no. 2, p. 259, 2020.

[21] R. Taheri, M. Ghahramani, R. Javidan, M. Shojafar, Z. Pooranian, and M. Conti, "Similarity-based android malware detection using hamming distance of static binary features," *Future Generation Computer Systems*, vol. 105, pp. 230–247, 2020.

[22] M. Jin, Z. Xu, R. Li, and D. Wu, "Fuzzy ARTMAP ensemble based decision making and application," *Mathematical Problems in Engineering*, vol. 2013, Article ID 124263, 7 pages, 2013.

[23] S. Mohammadi and A. Namadchian, "A new deep learning approach for anomaly base IDS using memetic classifier," *International Journal of Computers Communications & Control*, vol. 12, no. 5, pp. 677–688, 2017.

[24] J. Gu, L. Wang, H. Wang, and S. Wang, "A novel approach to intrusion detection using SVM ensemble with feature augmentation," *Computers & Security*, vol. 86, pp. 53–62, 2019.

[25] Y. Jia, M. Wang, and Y. Wang, "Network intrusion detection algorithm based on deep neural network," *IET Information Security*, vol. 13, no. 1, pp. 48–53, 2019.

[26] Z. T. Jiang and T. S. Z. Zhou, "Intrusion detection method based on ADBN," *Application Research of Computers*, vol. 37, no. 9, p. 46, 2020.

[27] M. X. Lu, G. Z. Du, and Z. X. Ji, "Network intrusion detection based on deep transfer learning," *Application Research of Computers*, vol. 37, no. 9, p. 44, 2019.

[28] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[29] A. Gupta, D. Singh, and M. Kaur, "An efficient image encryption using non-dominated sorting genetic algorithm-III

based 4-D chaotic maps," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1309–1324, 2020.
[30] M. Kaur, H. K. Gianey, D. Singh, and M. Sabharwal, "Multi-objective differential evolution based random forest for e-health applications," *Modern Physics Letters B*, vol. 33, no. 5, Article ID 1950022, 2019.
[31] K. Siddique, Z. Akhtar, and F. Kim, "KDD Cup 99 data sets: a perspective on the role of data sets in network intrusion detection research," *Computer*, vol. 52, no. 2, pp. 41–51, 2019.
[32] M. Tavallaee, E. Bagheri, W. Lu et al., "A detailed analysis of the KDD CUP 99 data set," in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, Ottawa, Canada, July 2009.
[33] M. T. Lincoln Laboratory, https://www.ll.mit.edu/.
[34] C. Sun, K. Lv, C. Z. Hu et al., "A double-layer detection and classification approach for network attacks," in *Proceedings of the 27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–8, Hangzhou, China, July 2018.
[35] W. Zong, Y.-W. Chow, and W. Susilo, "Interactive three-dimensional visualization of network intrusion detection data for machine learning," *Future Generation Computer Systems*, vol. 102, pp. 292–306, 2020.
[36] KDD CUP 99 Dataset, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[37] NSL-KDD Dataset, https://www.unb.ca/cic/datasets/nsl.html.