Hindawi
Security and Communication Networks
Volume 2019, Article ID 7130868, 11 pages
https://doi.org/10.1155/2019/7130868

Research Article

Building an Effective Intrusion Detection System by Using Hybrid Data Optimization Based on Machine Learning Algorithms

Jiadong Ren,[1] Jiawei Guo,[1] Wang Qian,[1] Huang Yuan,[2] Xiaobing Hao,[1] and Hu Jingjing[3]

[1]*Computer Virtual Technology and System Integration Laboratory of Hebei Province, College of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei, 066000, China*
[2]*Hebei University of Engineering, School of Information & Electrical Engineering, Hebei Handan, 056038, China*
[3]*Beijing Key Laboratory of Software Security Engineering Technique, Beijing Institute of Technology, 5 South Zhongguancun Street, Haidian District, Beijing, 100081, China*

Correspondence should be addressed to Huang Yuan; huangyuan722@163.com

Received 7 January 2019; Revised 7 May 2019; Accepted 16 May 2019; Published 16 June 2019

Academic Editor: Mamoun Alazab

Intrusion detection system (IDS) can effectively identify anomaly behaviors in the network; however, it still has low detection rate and high false alarm rate especially for anomalies with fewer records. In this paper, we propose an effective IDS by using hybrid data optimization which consists of two parts: data sampling and feature selection, called DO_IDS. In data sampling, the Isolation Forest (iForest) is used to eliminate outliers, genetic algorithm (GA) to optimize the sampling ratio, and the Random Forest (RF) classifier as the evaluation criteria to obtain the optimal training dataset. In feature selection, GA and RF are used again to obtain the optimal feature subset. Finally, an intrusion detection system based on RF is built using the optimal training dataset obtained by data sampling and the features selected by feature selection. The experiment will be carried out on the UNSW-NB15 dataset. Compared with other algorithms, the model has obvious advantages in detecting rare anomaly behaviors.

1. Introduction

With the rapid development of the Internet, the issue of network security has also received more and more attention. Research on the detection of anomaly behavior in the network is an important topic in the field of network security. IDSs are used to analyze network data and detect anomaly behaviors in the network. IDSs are generally classified into two categories: signature-based and anomaly-based detection systems [1]. Signature-based intrusion detection systems [2, 3], such as Snort intrusion detection systems [3], are designed to detect intrusion by building anomaly behavior character libraries and matching network data. These IDSs have high detection rate, but they are difficult to identify new attacks in the network. Anomaly-based intrusion detection systems establish models according to normal network behavior and conduct intrusion detection based on whether the behaviors are dedicated from the normal behavior. Such IDSs have an excellent recognition efficiency for unknown types of anomaly behavior, but their overall detection rate is low and has a high false alarm rate.

In order to improve the detection rate of IDSs and reduce the false alarm rate, researchers have done a lot of work, trying to apply a variety of methods of data mining and machine learning on IDSs. For example, SVM and neural network models are applied to the research of intrusion detection [4]. Koc et al. propose a Hidden Naïve Bayes model (HNB) to build intrusion detection system [5], which shows that the HNB model exhibits a superior overall performance with traditional Naïve Bayes. LP Rajeswari et al. propose a multiple level hybrid classifier to build IDS that uses a combination of tree classifiers of Enhanced C4.5 [6], which can be trained with unlabeled data and detects previously "unseen" attacks. In addition to the improvement of traditional classification

methods, some researches focus on the selection of data records.

However, the huge amount of network data and the unbalanced distribution of normal and anomaly behaviors lead to the problems of low detection rate and high false alarm rate in most IDSs. In this paper, an effective IDS by using hybrid data optimization data consists of sampling and feature selection is proposed. Data sampling is to delete outliers in dataset and reduce the negative impact of unbalanced data distribution on Intrusion detection. Feature selection is to search for features that best reflect the difference between anomalous behaviors and normal behaviors and delete useless features to enhance the detection performance of IDS. And an effective IDS based on data sampling and feature selection is built by using RF algorithm.

The organization of this paper is as follows. Section 2 outlines the related works. Section 3 introduces the operational principle of iForest, GA, and RF which will be applied in DO_IDS. Section 4 introduces the building of DO_IDS in detail. Section 5 describes and analyzes the experiments. Section 6 summarizes and elaborates DO_IDS.

2. Related Work

Data sampling can solve the problem of unbalanced distribution of network data. Data sampling includes oversampling and undersampling. When the data is insufficient for analysis, the oversampling method balances the data by increasing the rare samples, such as SMOTE algorithm. In contrast, undersampling deals with a dataset by reducing some samples, such as EasyEnsemble and BalanceCascade proposed by Liu et al [7].

By using sampling method to extract representative training data and combining with machine learning method, the performance of IDS can be improved effectively. Enamul et al. use sampling technique to select representative dataset and Least Squares SVM to identify anomalous network data [8], proving that data sampling can improve the accuracy and speed of intrusion detection. Alyaseen et al. combine modified K-means with machine learning methods to build intrusion detection models [9–11]. The modified K-means method can discover similar structures and models between datasets to compress datasets with higher quality. Integrating K-means with C4.5 to construct the classifier of intrusion detection model can greatly reduce the running time of intrusion detection system [9]; with SVM algorithm it can effectively improve performance for detecting DoS anomaly [10]; and with hybrid model of SVM and extreme learning machine (ELM) it can improve accuracy and efficiency of IDS [11].

Some researchers also focus their research on feature selection. Feature selection includes three methods: filter, wrapper, and embedded. Filter method evaluates each feature according to its divergence or correlation and sets threshold to select feature, which is irrelevant to the classification performance of classifier [12]. Wrapper method selects features or excludes features according to the objective function, which is usually the effect of classification [13]. Embedding method first trains some machine learning models to obtain the weights of each feature and then selects features according to weights, such as Decision Tree [14].

When it is found that some features can contribute more for classification but some make classification confused, feature selection is paid more attention. Wang et al. transform the original features using the logarithms of the marginal density ratios and obtain new and better-quality transformed features [15], which improves the performance of an SVM-based detection model. Vajiheh Hajisalem et al. propose a hybrid classification method based on artificial bee colony (ABC) and artificial fish swarm (AFS) algorithm [16], using fuzzy C-means clustering (FCM) and correlation-based feature selection (CFS) techniques for training data. George et al. apply SVM and PCA to anomaly detection of network data [17]. It is proved that PCA can effectively improve classification effect of SVM and increase model training speed. Raman et al. propose the combination of hypergraph, GA, and support vector machine to implement IDS [18]. Hypergraph and GA are used to perform parameter estimation of SVM and feature selection. Support vector machine is used to detect anomalous network behaviors after feature selection. It is proved that the combination of feature selection method and SVM can improve the accuracy of classifier. Genetic algorithm, which is also applied in this paper, is a heuristic search algorithm used to solve optimization in the field of computer science and artificial intelligence and is widely used in various directions, such as global optimization, parameter optimization, and feature selection [19]. At the same time, many scholars also apply genetic algorithm to network security. Khammassi et al. use GA and logistic regression algorithm to select the optimal feature subset [20] and prove that the feature subset selected by the method is effective for intrusion detection through different decision tree algorithms. Hamamoto A H et al. combine GA and fuzzy logic to detect anomalous events in network and prove that fuzzy logic can improve accuracy [21]. In their work, GA is used to generate a digital signature of network segment using flow analysis and fuzzy logic scheme is applied to decide whether an instance represents an anomaly or not. Faris H et al. propose an intelligent detection system that is based on GA and Random Weight Network to deal with email spam detection tasks [22], and the experimental results confirm that the proposed system can achieve remarkable results in terms of accuracy, precision, and recall. Vijayanand R et al. propose a novel intrusion detection system with GA based feature selection and multiple support vector machine classifiers for wireless mesh networks [23]. The system proposed by them exhibits a high accuracy of attack detection and is suitable for intrusion detection in wireless mesh networks.

3. Preliminary

This section introduces the genetic algorithm, iForest algorithm, and RF algorithm that will be used in the next section.

3.1. Isolation Forest (iForest). IForest algorithm is proposed by Liu, Fei et al. in 2012 [24, 25]; this algorithm is a tree-based outlier detection model with linear time complexity

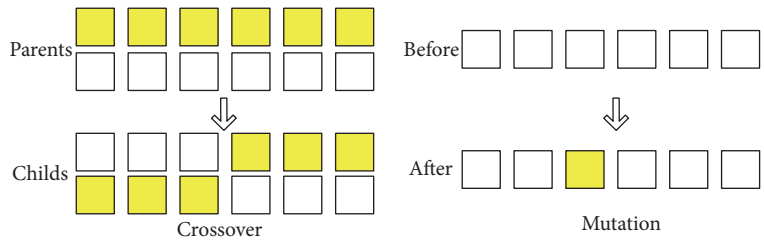

Figure 1: Crossover and mutation.

and high precision and suitable for high-dimensional and large amount of data.

Because anomalies are "less and different," they are more vulnerable to be isolated. In a data oriented random tree, records are recursively cut until all records are isolated. This random partition makes outlier record as a shorter path length because records with distinguishable attribute values are more likely to be separated in early partitions. IForest consists of some iTrees (Isolation Tree). Each iTree is a binary tree. The implementation steps are as follows:

(1) Randomly select a fixed number of sample points from training data as subsamples and put them in the root node of the tree.

(2) Randomly specify an attribute and randomly generate a cutting point p in the current node data, cutting point is generated between the maximum and minimum value of the specified attribute in the current node data.

(3) A hyperplane is generated from this cutting point, and the data space of the current node is divided into two subspaces: the data less than p in the specified attribute is put into the left child of the current node, and the data greater than or equal to p is put into the right child of the current node.

(4) Recursively execute steps 2 and 3, until the child node has only one record or the iTree has reached the defined height.

After getting these iTrees, the training of iForest is terminate, and then we can evaluate the testing data using the generated iForest. For a testing record, let it traverse each iTree and then calculate height of the records that eventually fall on each tree. Then we can get the average height of the record in each tree. If the average height is less than the given threshold, then the record is considered an outlier.

3.2. Genetic Algorithm. Genetic algorithm mainly includes four parts: chromosome encoding, initial population generating, fitness calculating, and genetic operator design.

(1) Chromosome Designing. GA expresses the solution space data as genotype string structure before optimization searching. Different combinations of these string structure constitute different chromosomes, and each chromosome represents a possible solution.

(2) Initial Population Generating. Each population contains a certain number of chromosomes, and each chromosome represents a possible solution. The chromosomes are initially generated randomly.

(3) Fitness Calculating. The fitness function indicates the superiority or inferiority of the individual. For different problems, the definition of fitness function is different.

(4) Genetic Operator Design. Genetic operators include three operators: selection, crossover, and mutation. Selection operation refers to reserving individuals with high fitness. Roulette wheel strategy is commonly used in selection operation. Roulette wheel strategy is based on the fitness of each chromosome in the proportion of the total fitness to get a survival probability, the chromosome with this probability to decide whether to inherit to the next generation. Survival probability is shown in Formula (1).

$$P\left(X_i\right) = \frac{f\left(X_i\right)}{\sum_{j=1}^{N} f\left(X_j\right)} \tag{1}$$

$f(X_i)$ is the fitness for ith chromosome X_i.

Crossover operation is the most important genetic operation in GA. It refers to exchanging genes between two chromosomes, resulting in the generating of two new chromosomes. The mutation begins by randomly selecting a chromosome in a population and randomly changing the value of a gene with a certain probability for the selected chromosome. The crossover and mutation operation are shown in Figure 1, and the genetic operation flow is shown in Figure 2.

3.3. Random Forest. Random Forest is an ensemble supervised machine learning algorithm, which was first proposed by Leo [26]. Its classification performance is better than other single classifier models in most cases and it can handle both binary classification problems and multiclassification problems. The main idea of RF is to use randomly sampling with replacement to construct multiple decision trees, and the final result is obtained by voting. The process of constructing RF is as follows.

(1) Using randomly sampling with replacement to extract samples from dataset and obtain a training subset.

(2) For the training subset, m features are randomly extracted from the feature set without replacement

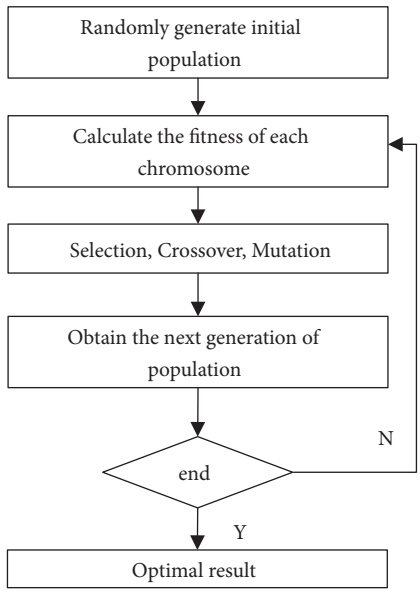

Figure 2: Process of GA.

as the basis for splitting each node in the decision tree. From the root node, a complete decision tree is generated from top to bottom.

(3) The k decision trees are generated by executing (1) and (2) repeatedly K times. RF classifier is obtained by combining these decision trees. The result of classification is voted by these decision trees.

4. Proposed DO_IDS

In the network, the normal behavior of users is more than the anomalous behavior, which makes the data distribution of normal behaviors and anomalous behaviors unbalanced. In order to enhance the detection performance of IDS, a hybrid data optimizing method based on multiply machine learning algorithms is proposed in this paper. The data optimizing method consists of two parts: data sampling and feature selection. (i) Data sampling: in this part, iForest outlier detection method is used to sample the data, GA is used to optimize the sampling ratio globally, and the classification performance of RF on candidate sampled data is used as the evaluation indicator. The purpose of data sampling is to search the optimal training dataset and reduce the imbalance of dataset. (ii) Feature Selection: in this paper, the method of integrating GA with RF is used to select features. Like data sampling, GA is used as a search strategy to specify candidate feature subset, and the classification performance of RF as evaluation indicator of candidate feature subset. The purpose of feature selection is to find the best feature subset that can maximize the performance of the detection. Once the optimal training dataset and the optimal feature subset are selected, those will be taken into the classifier training phase which employs RF algorithm. The whole process is shown as Figure 3.

Table 1: Confusion matrix.

		Predicted	
		Anomalous	Normal
Actual	Anomalous	TP	FN
	Normal	FP	TN

4.1. Data Sampling. The purpose of data sampling is to delete outliers in data and reduce the negative impact of outliers on detection performance. So in this paper, iForest, which can detect outliers quickly and effectively [27], is used to detect and delete outliers in network data at a given ratio, and the data obtained is the sampled data. In order to determine the best sampling ratio of each category, GA is used to optimize the sampling ratio of each category and the performance of RF classification is used to evaluate candidate sampled data. The description of data sampling in detail is as follows.

In data sampling, a chromosome sequence $X = \{x_1, x_2, \ldots, x_k\}$, k is the number of classes of network behaviors, $x_i \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ is a gene of chromosome, represents the ratio of outliers of class i detected by performing iForest.

In the classification problem, the fitness function is usually set as the accuracy of the classifier. In this paper, the fitness function is assumed to be the F1_score. F1_score is a harmonic function that takes both precision and recall into account. Calculation of F1_score is shown as follows.

$$F1\_score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

Among them, True Positive (TP) is the number of actual anomalous records classified as anomalous ones, True Negative (TN) is the number of actual normal records classified as normal ones, False Positive (FP) is the number of actual normal records classified as anomalous ones, and False Negative (FN) is the number of actual anomalous records classified as normal ones. Confusion matrix is shown as Table 1.

For the genetic operator of GA, in the part data sampling, crossover operation means that the same gene of any two chromosomes exchanges with a certain probability. Mutation operation means changing a gene of chromosomes by adding or subtracting 0.1 with a certain probability. The roulette wheel is applied as a selection function.

In this stage, the algorithm description is shown as Algorithm 1, and Algorithm 2 is the calculation of chromosome fitness in the stage of data sampling. where X_{best} is the chromosome with the highest fitness in the final population. $D_{outlier}$ is the set of outliers detected by iForest. D_{train_best} is the optimal training dataset obtained in data sampling.

The first step is to randomly generate a population P composed of N chromosomes. In order to get the next generation of population, GA is applied for population P. Firstly, perform

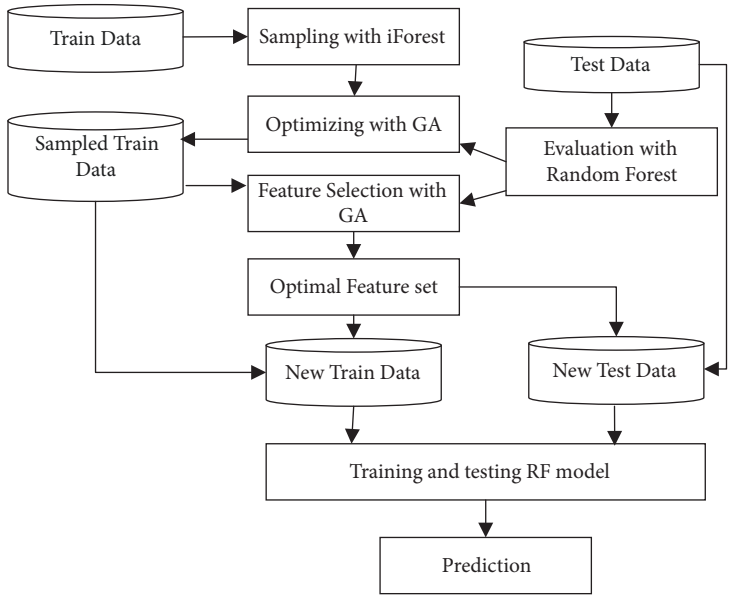

Figure 3: Process of DO_IDS.

Input: Original training dataset D_{train}, testing dataset D_{test}
Output: New training dataset D_{train_best}
Generate initial population P
While not reach terminating condition
  Calculate_Fitness_Sample (D_{train}, D_{test}, P)
  Selection(P)
  /*$\rho_{crossover}$, $\rho_{mutation}$ are probabilities of crossover and mutation respectively.*/
  Crossover (P, $\rho_{crossover\_sampling}$)
  Mutation (P, $\rho_{mutation\_sampling}$)
End While
X_{best} = Chromosome with the highest fitness in P
$D_{outlier}$ = iFoest(D_{train}, X_{best})
$D_{train\_best} = D_{train} - D_{outlier}$
End

Algorithm 1: GA_iForest_RF_Sample ( ).

Input: Original Dataset D_{train}, D_{test}, Chromosome population $P = \{X_1, \ldots, X_n\}$
Output: Fitness set $\{f(X_1), f(X_2), \ldots, f(X_n)\}$
for $i = 1$: n
  $D_{train\_temp} = D_{train}$ – iForest (D_{train}, X_i)
  Train Random Forest classifier rf by D_{train_temp}
  Test rf based on D_{test} and get classification
  Calculate F1-score $F1$ based on actual class and classification
  $f(X_i) = F1$
End

Algorithm 2: Calculate_Fitness_Sample( ).

the selection operation to retain the optimal individuals and calculate the fitness of each chromosome. Secondly, two chromosomes are randomly assigned to perform crossover operation with probability $\rho_{crossover\_sampling}$ and perform mutation operation with probability $\rho_{mutation\_sampling}$. In this way, a new population can be obtained. Finally, implement the above process iteratively until the iteration termination condition is reached and then we can get the best chromosome X_{best}. Performing iForest on training dataset D_{train} according to X_{best} can get outlier dataset $D_{outlier}$. The optimal training

Input: New training dataset D_{train_best}, testing dataset D_{test}
Output: Optimal feature subset F_{best}
Generate initial population $P = \{X_1, X_2, \ldots, X_n\}$
While not reach terminating condition
  Calculate fitness (D_{train_best}, D_{test}, P)
  Selection(P)
  /*$\rho_{crossover}$, $\rho_{mutation}$ are probabilities of crossover and mutation respectively.*/
  Crossover (P, $\rho_{crossover\_FS}$)
  Mutation (P, $\rho_{mutation\_FS}$)
End While
X_{best} = Chromosome with the highest fitness in P
F_{best} = Convert X_{best} to feature number
End

Algorithm 3: GA_RF_FS ( ).

Input: D_{train_best}, D_{test}, Chromosome population $P = \{X_1, \ldots, X_n\}$
Output: Fitness set $\{f(X_1), f(X_2), \ldots, f(X_n)\}$
for $i = 1: n$
  Extract data from D_{train_best}, D_{test} based on X_i and get D'_{train_best}, D'_{test}
  Train Random Forest classifier rf based on D'_{train_best}
  Test rf based on D'_{test}, and get classification
  Calculate F1-score $F1$ based on actual class and classification
  $f(X_i) = F1$
end

Algorithm 4: Calculate_Fitness_FS().

dataset D_{train_best} can be obtained by deleting $D_{outlier}$ from D_{train}.

4.2. Feature Selection. In the research of intrusion detection, redundant features can degrade detection performance, so more and more researchers focus on feature selection [2, 16, 18, 20]. The process of feature selection in this paper is similar to the data sampling. The difference lies mainly in chromosome designing and mutation. In data sampling, the chromosome contains the number of classes in the dataset, and each gene is a floating-point number, representing the ratio of outliers to be eliminated. In feature selection, the chromosome is a binary string, $X = \{x_1, x_2, \ldots, x_m\}$, $x_i \in \{0, 1\}$, $1 \leq i \leq m$, m is the number of feature, $x_1 = 1$ represents the ith feature is selected, and $x_1 = 0$ represents not. The detailed steps are shown in Algorithm 3, and Algorithm 4 is the calculation of chromosome fitness in the feature selection stage.

4.3. Classifier Training. According to the data sampling and feature selection, the optimal training dataset and the optimal feature subset can be obtained. Dimension reduction is performed on the optimal training set according to the optimal feature subset. Because RF classifier can handle multiclassification problems [28], we can further identify the classes of anomalous behaviors. Assuming that, let normal behaviors be one class, and there are k classes of anomalous behaviors; then, the whole network dataset can be composed of $k + 1$ classes. For each class, data sampling and feature selection methods are used to get optimal training dataset and optimal feature subset; there will be $k + 1$ classifiers for all the class trained by their corresponding data. Finally, the final classification is voted by the $k + 1$ classifiers.

5. Experimental Results and Analysis

5.1. Experimental Settings and Dataset Description. Experiments are performed on a PC with Intel(R) Core(TM) i5-4460 at 3.6 GHz CPU and 8GB memory, running on Windows 10. Programs are coded in Python using Pycharm2017 environment on the version of Anaconda3.

The parameters used in the algorithm are obtained by empirical value and set as follows.

In genetic algorithm, population initiation N = 100, the crossover probability $\rho_{crossover} = 0.5$, the mutation probability $\rho_{crossover} = 0.1$, and the termination condition (the number of descendants inherited) G = 50. In data optimization, considering the efficiency factor, the numbers of components of iForest and RF are set as 10. In classifier training, the number of decision trees of RF is set as 200.

The UNSW-NB15 dataset is created by the cyber security research group at the Australian Centre for Cyber Security (ACCS) recently [29]. The dataset contains 2, 540,044 records with 42 attributes, which is divided into training set and testing set. The training set contains 175,341 records, while the test set contains 82,332 records. The parameters of the dataset

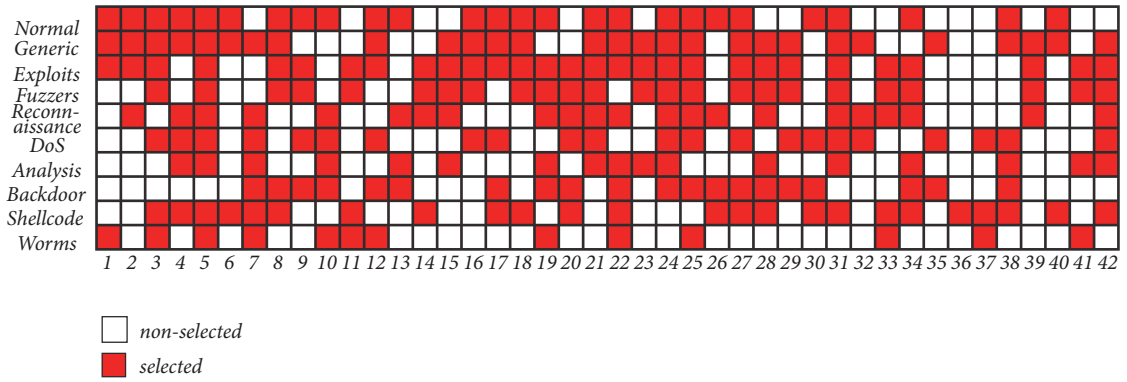

Figure 4: Optimal feature subset for each class of anomalous behaviors.

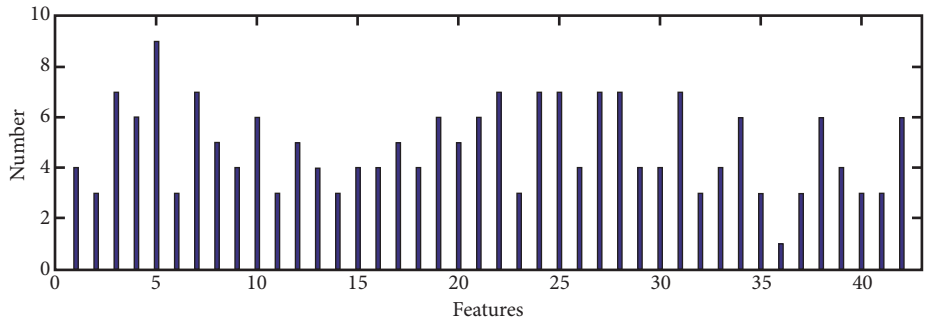

Figure 5: The number of times each feature is selected.

Table 2: Parameters of UNSW-NB15.

Number	Class	Size	Distribution (%)
1	Normal	56,000	31.94
2	Generic	40,000	22.81
3	Exploits	33,393	19.04
4	Fuzzers	18,184	10.37
5	DoS	12,264	6.99
6	Reconnaissance	10,491	5.98
7	Analysis	2,000	1.14
8	Backdoor	1,746	1
9	Shellcode	1,133	0.65
10	Worms	130	0.07
	Totals	175,341	100

are shown in Table 2, and the feature description is shown in Table 3.

5.2. Experimental Results. The optimal sampling ratio of each class obtained during data sampling is shown in Table 4, where the data volumes of Analysis, Backdoor, Shellcode, and Worms are too small for sampling, so they are not sampled.

Table 5 and Figure 4 show the optimal feature subset of each class of anomalous behaviors. It can be noted that Normal class has the largest number of features in the subset of optimal features, the number of its optimal features is 26, the least is Worms, and the number is 13. Compared with the total number of Features 42, all the classes have achieved considerable dimensionality reduction. Figure 5 shows the selected times of each feature. We can see that the 5th feature has been selected the most; all the classes regard it as an important feature except the class "Backdoor."

5.3. Comparison with Other Methods. Table 6 shows the confusion matrix of all classes over the UNSW-NB15 dataset using DO_IDS. To verify the effectiveness of the data optimization proposed in this paper, the precision, recall, and F1_score obtained by testing the proposed model are shown in the Table 7 and compared with the simple RF classifier without data sampling and feature selection. Obviously, except for the slight decrease in the precision of Worms and DoS and the recall of Exploits and Shellcode, the precision and recall in other classed have improved significantly, especially for the anomaly behavior with less records, such as Analysis, Backdoor, Shellcode, and Worms. It can be seen that DO_IDS has achieved good performance on the detection of network anomaly behavior with unbalanced data distribution.

Table 8 shows the comparison of accuracy and false alarm rate (FAR) of all classes between simple RF and DO_IDS. FAR refers to the proportion of anomaly behaviors classified as normal to all anomaly behaviors. In the research of IDS, FAR is a significantly important evaluation indicator because in the network data, the number of normal behaviors is far more than the number of anomalous behaviors; even if all network data are classified as normal behavior, the accuracy can reach a high level. As we can see from table 8, both simple RF and DO_IDS have high classification accuracy in each class, but FAR of DO_IDS is obviously better than

Table 3: Feature set of UNSW-NB15.

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

Table 4: Optimal sampling ratio for each class of anomalous behaviors.

Class name	Sampling Ratio					
	Normal	Generic	Exploits	Fuzzers	Reconnaissance	DoS
Normal	1.0	1.0	0.9	0.9	0.8	0.9
Generic	1.0	0.8	1.0	1.0	1.0	1.0
Exploits	1.0	1.0	1.0	1.0	0.9	1.0
Fuzzers	0.9	0.9	1.0	0.8	0.9	0.8
Reconnaissance	0.9	0.9	1.0	1.0	1.0	1.0
DoS	1.0	0.7	1.0	0.8	1.0	1.0
Analysis	0.9	1.0	0.8	0.7	0.9	0.9
Backdoor	1.0	0.9	1.0	0.7	1.0	0.9
Shellcode	0.9	0.6	0.9	1.0	1.0	0.9
Worms	1.0	0.7	1.0	0.7	0.8	1.0

Table 5: Optimal feature subset for each class of anomalous behaviors.

Class name	Sequence number of Features	Features Number
Normal	1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 13, 16, 17, 18, 19, 21, 22,24, 25, 26, 27, 30, 31, 34, 38, 40	26
Generic	1, 3, 4, 5, 6, 7, 8, 15, 16, 23, 24, 27, 28, 29, 32, 35, 38,39, 40	19
Exploits	1, 2, 3, 5, 12, 17, 18, 21, 22, 25, 27, 28, 31, 39, 42	15
Fuzzers	3, 5, 8, 9, 11, 14, 15, 16, 18, 19, 20, 21, 23, 24, 25, 27, 28, 29, 31, 33, 34, 39, 41, 42	24
Reconnaissance	2, 4, 5, 7, 10, 13, 14, 15, 19, 20, 21, 22, 24, 25, 26, 28, 31, 32, 33, 34, 39, 42	22
DoS	3, 4, 5, 7, 9, 10, 12, 16, 17, 20, 21, 24, 25, 27, 29, 30, 31, 32, 35, 37, 38, 42	22
Analysis	4, 5, 7, 10, 13, 15, 19, 21, 22, 23, 24, 28, 31, 34, 38, 41, 42	17
Backdoor	7, 8, 9, 10, 12, 13, 17, 19, 20, 22, 24, 25, 26, 27, 28, 29, 30, 34, 35, 38	20
Shellcode	3, 4, 5, 6, 7, 8, 11, 14, 17, 18, 20, 22, 26, 27, 28, 30, 31, 33, 34, 36, 37, 38, 40, 42	24
Worms	1, 3, 5, 7, 10, 11, 12, 19, 22, 25, 33, 37, 41	13

Table 6: Confusion matrix of all classes over the UNSW-NB15 dataset using DO_IDS.

Actual	Predicted										Recall
	1	2	3	4	5	6	7	8	9	10	
1	35778	9	298	131	17	39	510	0	217	1	0.967
2	84	18291	419	2	1	41	0	9	21	3	0.969
3	716	13	7381	5	280	1832	166	580	154	5	0.663
4	2613	5	175	2307	6	857	12	60	27	0	0.381
5	83	1	223	0	2867	171	20	85	46	0	0.820
6	254	6	1128	3	55	1887	133	554	68	1	0.461
7	178	0	41	0	0	380	41	37	0	0	0.061
8	127	0	48	0	0	166	0	235	7	0	0.403
9	65	0	14	0	1	3	0	0	295	0	0.780
10	4	1	3	0	0	0	0	0	1	35	0.795
Precision	0.897	0.998	0.759	0.942	0.888	0.351	0.046	0.151	0.352	0.778	

Table 7: Comparison between DO_IDS and RF on Precision, Recall, and F1-score.

	Precision		Recall		F1-score	
	RF	DO_IDS	RF	DO_IDS	RF	DO_IDS
Normal	0.859	**0.897**	0.876	**0.967**	0.867	**0.930**
Generic	0.997	**0.998**	0.967	**0.969**	0.982	**0.983**
Exploits	0.687	**0.759**	**0.697**	0.663	0.692	**0.708**
Fuzzers	0.055	**0.942**	0.029	**0.381**	0.038	**0.542**
Reconnaissance	0.886	**0.888**	0.814	**0.820**	0.849	**0.853**
DoS	**0.327**	0.351	0.417	**0.461**	0.367	**0.399**
Analysis	0.002	**0.046**	0.003	**0.061**	0.002	**0.053**
Backdoor	0.040	**0.151**	0.063	**0.403**	0.049	**0.219**
Shellcode	0.242	**0.352**	**0.817**	0.780	0.373	**0.486**
Worms	**0.800**	0.778	0.182	**0.795**	0.296	**0.787**

Table 8: Comparison between DO_IDS and RF on accuracy and FAR.

	Accuracy		FAR	
	RF	DO_IDS	RF	DO_IDS
Normal	0.865	**0.935**	0.124	**0.033**
Generic	0.989	**1.0**	0.033	**0.031**
Exploits	0.902	**0.926**	**0.303**	0.337
Fuzzers	0.876	**0.953**	0.971	**0.619**
Reconnaissance	0.984	**0.988**	0.849	**0.180**
DoS	0.915	**0.931**	0.583	**0.539**
Analysis	0.972	**0.982**	0.997	**0.939**
Backdoor	0.978	**0.980**	0.937	**0.597**
Shellcode	0.984	**0.992**	**0.183**	0.220
Worms	0.999	**1.0**	0.218	**0.205**

Table 9: The overall comparison between DO_IDS and RF.

Method	Accuracy	FAR	Macros precision	Macros recall	Macros F1_score
RF	0.865	0.124	0.489	0.487	0.488
DO_IDS	0.928	0.033	0.616	0.630	0.623

that of simple RF. From an integrated view, Table 9 shows the overall comparison of the whole dataset without specific class distinction between DO_IDS and RF without data optimization. All the above metrics are defined as follows.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TM} \tag{5}$$

$$FAR = \frac{FP}{FP + TN} \tag{6}$$

$$Macros\ precision = \frac{1}{k}\sum_{i=1}^{k} precision_i \tag{7}$$

$$Macros\ recall = \frac{1}{k}\sum_{i=1}^{k} recall_i \tag{8}$$

$$Macros\ F1\_score = \frac{2 * Macros\ precision * Macros\ recall}{Macros\ precision + Macros\ recall} \tag{9}$$

where k is the number of classes.

Table 10: The comparison between DO_IDS and other models.

Method	Accuracy	FAR
KNN	0.760	0.195
SVM	0.625	0.007
LR	0.832	0.185
NB	0.821	0.186
MLP	0.813	0.211
EM	0.785	0.238
DT	0.856	0.158
RF	0.865	0.124
AdaBoost	0.861	0.116
RUSBoost [30]	0.844	0.131
GA-LR [13]	0.814	0.639
DO_IDS	**0.928**	**0.330**

Table 10 shows comparison between the proposed method and other machine learning methods. Since most of the classic machine learning algorithms only focus on binary classification, we choose accuracy and FAR as

Table 11: Comparison of DO_IDS, DT, AdaBoost, and RUSBoost.

Method	Accuracy	FAR	Macros precision	Macros recall	Macros f1-score
DT	0.866	0.083	0.415	0.427	0.421
AdaBoost	0.920	0.074	0.558	0.556	0.557
RUSBoost	0.923	0.047	0.580	0.590	0.585
DO_IDS	0.928	0.033	0.616	0.630	0.623

universal evaluation indicators to evaluate their abilities to distinguish between normal and anomalous behaviors. In Table 10, it is obvious that the proposed method has a great improvement in both accuracy and FAR. However, the IDSs based on traditional machine learning algorithms have the problem of high FAR, which is mostly because of the lack of consideration about dataset imbalance.

From the comparison in Table 10, it is obvious that the performances of DT, RUSBoost, and AdaBoost are close to RF, so, we further applied data optimization to these four algorithms to see which algorithm is the best in the combined performance with data optimization in Table 11. It can be seen from Table 11 that DO_IDS, that is, applying RF as the final classifier, is better overall.

6. Conclusion

In this paper, we have proposed a data optimization method to build IDS, named DO_IDS. The data optimization consists of two parts: data sampling and feature selection. In data sampling, iForest is used to sample data and integration of GA and RF is used to optimize sampling ratio. In feature selection, integration of GA and RF is used again to select the optimal feature subset. Classification is performed by using RF to build IDS. DO_IDS has been evaluated by using intrusion detection dataset UNSW-NB15.

DO_IDS is a RF classifier based algorithm with data optimization, through experimental comparison; DO_IDS performs much better than RF classifier in all the indicators selected in the paper, which indicates the advantage of data optimization in DO_IDS. Also, by comparing with traditional machine learning methods, it demonstrates that RF classifier is a much stronger classifier, so the combined effect of data optimization and RF classifier makes DO_IDS almost always the best among all especially in detecting the anomalous behaviors with less records, such as DoS, Analysis, Backdoor, and Worms. However, there are still improvements that can be focused on, like much time cost in the data optimization stage and support for online processing.

As a future work, since the proposed data optimization can effectively reduce impact of the unbalanced sample distribution on IDS and has shown encouraging performance, it could be further applied to other anomaly detection fields, such as fraud detection. In addition, because it takes a lot of time to train classifiers, the search strategy could be further optimized.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Key R&D Program of China under Grant No. 2016YFB0800700, the National Natural Science Foundation of China under Grant Nos. 61472341, 61772449, 61572420, 61807028, and 61802332.

References

[1] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pp. 120–132, USA, May 1999.

[2] H. P. Sasan and M. Sharma, "Intrusion detection using feature selection and machine learning algorithm with misuse detection," *International Journal of Computer Science and Information Technologies*, vol. 8, no. 1, pp. 17–25, 2016.

[3] J. E. Díaz-Verdejo, P. García-Teodoro, P. Muñoz, G. Maciá-Fernández, and F. De Toro, "A Snort-based approach for the development and deployment of hybrid IDS," *IEEE Latin America Transactions*, vol. 5, no. 6, pp. 386–392, 2007.

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

[5] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13492–13500, 2012.

[6] L. P. Rajeswari and A. Kannan, "An intrusion detection system based on multiple level hybrid classifier using enhanced C4.5," in *Proceedings of the International Conference on Signal Processing Communications and Networking (ICSCN '08)*, pp. 75–79, IEEE, January 2008.

[7] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 539–550, 2009.

[8] E. Kabir, J. Hu, H. Wang et al., "A novel statistical technique for intrusion detection systems," *Future Generation Computer Systems*, vol. 79, no. 1, pp. 303–318, 2017.

[9] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Hybrid modified k-means with c4.5 for intrusion detection systems in multiagent systems," *The Scientific World Journal*, vol. 2015, Article ID 294761, 14 pages, 2015.

[10] W. L. Al-Yaseen, Z. A. Othman, and M. Z. Nazri, "Intrusion detection system based on modified k-means and multi-level support vector machines," in *Proceedings of the International*

Conference on Soft Computing in Data Science 2015 Proceedings, pp. 265–274.

[11] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296–303, 2017.

[12] I. Guyon, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, pp. 31157–31182, 2003.

[13] J. Jelonek, K. Krawiec, and J. Stefanowski, "Comparative study of feature subset selection techniques for machine learning tasks," in *Proceedings of the International Symposium Intelligent Information Systems*, pp. 77–99, 1998.

[14] F. Esposito, D. Malerba, and G. Semeraro, "A comparative analysis of methods for pruning decision trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 476–491, 1997.

[15] H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowledge-Based Systems*, vol. 136, pp. 130–139, 2017.

[16] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018.

[17] A. George, "Anomaly detection based on machine learning dimensionality reduction using PCA and classification using SVM," *International Journal of Computer Applications*, vol. 47, no. 21, pp. 5–8, 2012.

[18] M. R. G. Raman, N. Somu, K. Kirthivasan et al., "An efficient intrusion detection system based on hypergraph - Genetic algorithm for parameter optimization and feature selection in support vector machine," *Knowledge-Based Systems*, vol. 134, pp. 1–12, 2017.

[19] C. Huang and C. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Systems with Applications*, vol. 31, no. 2, pp. 231–240, 2006.

[20] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255–277, 2017.

[21] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença, "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Systems with Applications*, vol. 92, pp. 390–402, 2018.

[22] H. Faris, A. M. Al-Zoubi, A. A. Heidari et al., "An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks," *Information Fusion*, vol. 48, pp. 67–83, 2019.

[23] R. Vijayanand, D. Devaraj, and B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Computers & Security*, vol. 77, pp. 304–314, 2018.

[24] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the 8th International Conference on Data Mining*, pp. 413–422, IEEE, 2009.

[25] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[26] L. Breiman, "Random forest," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[27] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[28] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proceedings of the Military Communications and Information Systems Conference*, pp. 1–6, IEEE, 2015.

[29] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Systems Security*, vol. 25, no. 1-3, pp. 18–31, 2016.

[30] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: a hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 40, no. 1, pp. 185–197, 2010.