Wu *et al.*
EURASIP Journal on Advances in Signal Processing (2022) 2022:39
https://doi.org/10.1186/s13634-022-00871-6

EURASIP Journal on Advances
in Signal Processing

RESEARCH **Open Access**

Intrusion detection system combined enhanced random forest with SMOTE algorithm

Tao Wu[1], Honghui Fan[2], Hongjin Zhu[2*], Congzhe You[2], Hongyan Zhou[1] and Xianzhen Huang[1]

*Correspondence:
zhuhongjin@jsut.edu.cn
[2] School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China
Full list of author information is available at the end of the article

Abstract

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

Keywords: Network intrusion detection, Data imbalance, SMOTE algorithm, Enhanced random forest, Similarity, NSL-KDD

1 Introduction

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

It is notable that using decision forests has poor decision performance, which negatively affects the final voting results and model predictions. To solve this problem, this paper proposes an intrusion network detection model based on the enhanced random forest and SMOTE algorithms. In the first stage of data preprocessing, the SMOTE technique was employed to analyze the minority class samples and manually synthesize new samples based on the minority class samples to add to the dataset, and it was further improved by using K-means to make the sample dataset more convergent to the cluster center. The decision tree with good classification performance in the random forest model was calculated and selected for similarity calculation in the second stage. Before generating a new random forest model, we analyzed the types of network attacks and corrected the prediction results of voting processing through reasonable use of the similarity matrix of network attacks. Finally, the enhanced random forest model was trained on the processed NSL-KDD dataset in this paper, and the detection effect achieved the desired results.

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

2 Related work

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

The development of intrusion detection systems needs to be traced back to 1986 when the research group of Dorothy E. Denning et al. successfully implemented the first intrusion detection model, and subsequent research focused on feature extraction, classifier optimization, and data preprocessing [3]. Among them, the widely used classification algorithms include support vector machine (SVM), random forest (RF), K-nearest neighbors (K-NN), and other classification algorithms.

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

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

To further optimize the detection model for detection accuracy and false alarm rate, numerous researchers have optimized feature selection through research. Among them, Sumaiya et al. combined multiclass support vector machines with Chi-square feature selection. Through a series of test experiments, the results showed that the bonding method can achieve some improvements compared with other studies [6]. Peng et al. optimized the data type for the net attack by combining the minibatch means combined with principal component analysis (PCA) through analyzing the clustering algorithm [7].

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

The classification optimization algorithms proposed by many scholars prompt us to attempt to further optimize the random forest classification algorithm to achieve detection and classification of malicious attacks on the internet. However, too many studies focused on the overall detection accuracy and false alarm rate metrics, but they neglected the imbalance between training data types, and the proportional differences between data samples constantly affect the detection performance, which leads to a decrease in detection accuracy and an increase in false alarm rate for fewer sample types.

Many researchers have attempted to solve the data imbalance problem by processing the proportion of training data types, including oversampling and undersampling methods [9].

OFek et al. proposed a fast clustering method by combining undersampling techniques with clustering algorithms by analyzing the original clustering algorithms [10]. Based on this, the training results were weighted, and the algorithm achieved good results with certain applicability and effectiveness in processing the binary classification problem of the dataset.

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

Additionally, Yan et al. implemented a mean SMOTE (M-SMOTE) algorithm through their research on the SMOTE algorithm and verified the effectiveness of the algorithm in the classification process of unbalanced network data [12]. In general, although many studies focus on optimizing outstanding machine learning algorithms and the overall training metrics of the dataset, and the optimization methods include feature selection, data preprocessing, and classifier optimization, we can still make appropriate improvements in this area to obtain improved detection results.

3 The proposed intrusion detection model

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

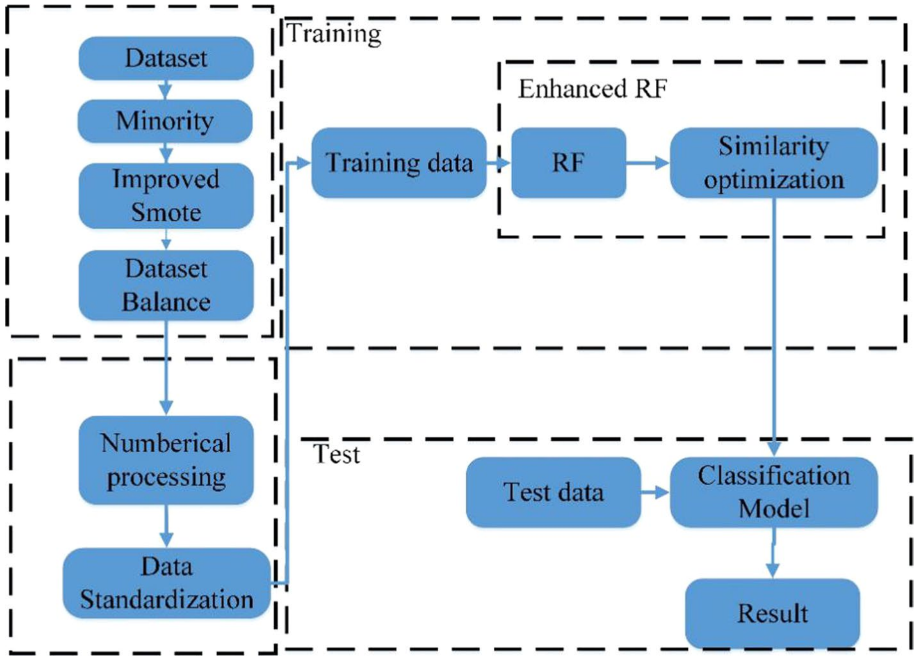


Fig. 1 Model architecture

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

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

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

(4) In the next section, correction and optimization of detection results are performed by calculating the similarity relationship between sample types of network attack

data. The process started with a preliminary determination of the cybersecurity attack dataset by enhanced random forest and determined the type of attack by majority voting. In the next step, this paper made accuracy judgments were made based on the key features, and if the judgment results indicated that the activity type was not reasonable, the results were corrected based on the attack type similarity matrix.

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

3.1 Construction of a balanced dataset based on the K-means clustering algorithm and smote sampling technique

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

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

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

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

$$x_{\text{new}} = x + \text{rand}(0,1) \times |x - x_n| \tag{1}$$

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

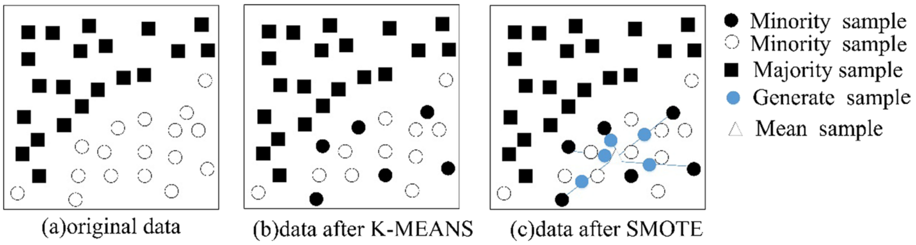
(1) The minority samples data are analyzed, the number of clustered sample centers T is determined, and then the target samples are selected by K-means clustering based on this value.
(2) For the target sample determined in the above step, a random sample is selected, and the k nearest neighbors of this sample in the target sample are calculated.
(3) The samples are also selected from the k nearest neighbors based on the preanalyzed data and oversampling rate N is set. The mean value between all samples is calculated, and then a new sample is generated between that value and the neighboring samples following the steps shown in Eq. (2).

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

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

3.2 IDS based on enhanced random forest

The random forest algorithm has relatively high detection accuracy compared to other classification algorithms, and the algorithm is more tolerant of noisy samples, which has resulted in numerous theoretical and experimental studies focusing on the use of



(a)original data (b)data after K-MEANS (c)data after SMOTE

Fig. 2 Sampling processing

these algorithms. As a combinatorial classifier algorithm, by learning the basic idea of the bagging algorithm to obtain N bootstrap training samples with a put-back sampling of the original dataset, the disguised augmentation of the samples used for training can be achieved. This approach effectively reduces the probability of overfitting. The dataset obtained by the above operation is fed into a decision tree model for training, and the final model is combined to generate a forest classification model. The model predicts the results by majority voting. The overall flow of random forest is shown in Fig. 3.

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

The horizontal and vertical axes of the receiver operating characteristic (ROC) curve depict the proportion of predicted types that are consistent with positive samples and actual types, respectively. The value of the area under the curve (AUC) is the area between the ROC curve and the coordinate axis. It serves as the corresponding probability value, and it can indicate the superiority of the classifier performance by comparing the high or low value, which is the reason that it is a criterion for internal performance optimization. In addition, the structural similarity between classifiers can be further calculated based on calculating classifier nodes and branches. Based on this, the structural similarity between classifiers was further calculated, which could be roughly classified as

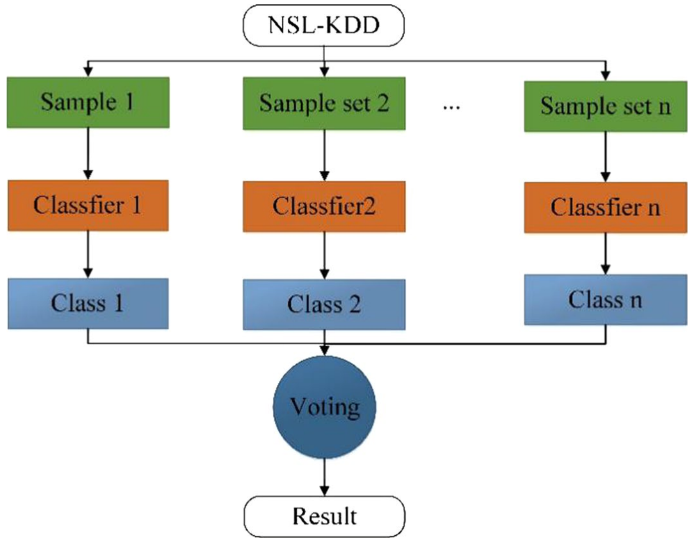


Fig. 3 Random forest classifier

similar (more than 80%) or dissimilar (less than 40%) by setting a certain threshold value as the judgment criterion. After the similarity comparison by the above steps, the values were transformed into matrix form, and the secondary optimization process was carried out according to the difference in classification and the AUC level. The classifiers with strong individual classification ability and low similarity were selected to form a classification model by combining them.

The details of the enhanced random forest model involved in this paper are explained as follows:

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

$$\{\mathrm{IOB}_1, \mathrm{IOB}_2, \ldots, \mathrm{IOB}_W\} \tag{3}$$

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

(3) The acquired classification models are optimized based on the similarity, and the classifiers with high similarity and poor classification performance are emptied. The computational approach taken in this paper focuses on calculating and applying structural similarity, and this class of methods learned and borrowed from Bakirli's [15] multitree intersimilarity optimization method. By analyzing and utilizing the decision tree storage structure form, the classifier is transformed and decomposed into the corresponding rule set and candidate rule set. The similarity $\mathrm{similarity}_{c1,c2}$ between multiple trees is derived by comparing split nodes among them and generating the similarity matrix Matrix. The set of classifiers (4) with high similarity and poor overall performance is selected by setting the corresponding thresholds:

$$\left\{\mathrm{classifier}_1, \mathrm{classifier}_2, \ldots, \mathrm{classifier}_Q\right\} \tag{4}$$

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

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

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

In this paper, the voting session results were processed with corrections. Certain criteria needed to be established because the detection accuracy of the classification model was not completely accurate, and if there was a misclassification during the detection process, the method could be used to correct the detection results. Therefore, some activity rules needed to be predefined. The voting result $F(X)$ was obtained after the majority voting of the result by the combined classifier, and the result of this determination was within a reasonable range if the attribute characteristics of the type were within the predefined activity rules. If the attribute characteristics of the activity did not match the set activity rules, it was necessary to calculate the similarity relationship between the decision results to generate the SimMatrix. Finally, a relatively more reasonable decision could be chosen based on the probability value SimMatrix obtained. This operation focused on the following components:

1. Setting activity rules.
2. Generating the activity similarity matrix.

3.2.1 Setting activity rules

Through analyzing the NSL-KDD dataset samples, more critical features and candidate features were found, which could be used as the basis for setting the corresponding activity rules. The number of NSL-KDD attribute features was relatively large at 41, and the attack type labels were roughly divided into two categories, including normal and anomaly. The anomaly data types could be divided into four major categories, including denial of service attacks (DOS), probe, users to root attacks (U2R), and remote to local attacks (R2L).

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

On this basis, the inherent features of the dataset were selected through information gain. The attributes selected for the active rule setting included service, src_bytes, dst_bytes, hot, num_file_creations, dst_host_srv_count, and dst_host _same_src_port_rate.

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

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

$$T = \begin{cases} F(x) & F(x) \in T_{\text{ser}} \\ \max\limits_{j} \text{AcCorr}_{[\text{classifer}(x)][j]} & F(x) \notin T_{\text{ser}} \end{cases} \tag{6}$$

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

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

3.2.2 Generating the activity similarity matrix

To analyze the intrusion detection data samples, it was clear that the malicious network attack types had certain similar property operations, such as DOS, U2R, PROBE, R2L, and other attack types, which could lead to the reduction in Src_byte and dst_byte byte values. U2R- and R2L-type attacks could be detected by hot, num_failed_logins feature behavior, and malicious interactions had a strong correlation in time. Therefore, the analysis of the correlation between attack types could be performed in advance, and in this way, the correction operation of the random forest determination results was achieved. A large number of experiments for intrusion type detection were conducted in this paper, and on this basis, comparisons were made with the real type to generate the corresponding activity matrix in Fig. 4.

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

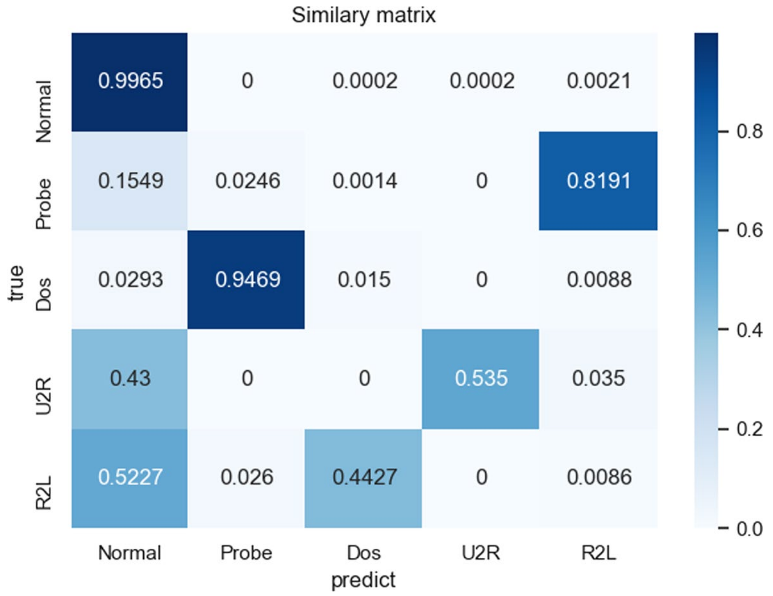


Fig. 4 Attack-type similarity matrix

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

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

4 Results and discussion

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

4.1 Dataset description

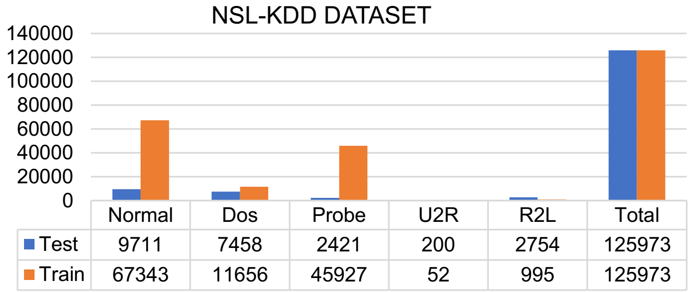
Through analyzing the experimental datasets used in similar studies, one that is better known in the field of intrusion detection and has obvious disadvantages is the KDD-cup99 dataset. The main problem in this dataset is that the system training data, which consists of up to 75% redundant data, tends to be somewhat misleading to the classifier. This allows the classifier to focus on records with more frequent occurrences and to learn relatively less from minority classes of data, including R2L and U2R, which has a great impact on the model detection effect and makes the classification results have a more obvious bias.

Considering the above factors, this paper to adopted an NSL-KDD dataset optimized for frequent records in the KDD dataset. As shown in the table, the overall data types of the NSL-KDD dataset can be broadly classified into two categories, including normal and anomaly. The exception types in the dataset can be subdivided into four major categories and many subtypes, including DOS, probe, U2R, and R2L. Table 1 shows the distribution of the specific type composition of the dataset.

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

Table 1 Attack-type distribution

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



	Normal	Dos	Probe	U2R	R2L	Total
■ Test	9711	7458	2421	200	2754	125973
■ Train	67343	11656	45927	52	995	125973

Fig. 5 Overall NSL-KDD data statistics

4.2 Data preprocessing

To further reduce the computational overhead time and ensure processing important attribute information, it was necessary to numerically transform the attributes that were not directly available in the original dataset and perform data normalization operations on the key data. The number of attribute features in the original dataset was up to 41, including basic features such as duration, content features such as host, and time and host-based network traffic statistics such as count and dst_host_count. Among them, preprocessing operations focused on processing character-based attributes, including protocol_type, service, and flag. First, numeric values were assigned to the tail column data types of each data sample, including the following five types: 0 for normal type, 1 for probe-type attack, 2 for DOS type attack, 3 for U2R type attack, and 4 for R2L type attack. Then, the character-based values were transformed into binary code features for easy identification and processing by one-hot encoding; for example, the protocol_type field was preprocessed to represent the TCP protocol using [1, 0, 0]. Finally, to prevent the model performance from being affected by data processing overflow problems during training due to overly large data values, a normalized processing operation for the original data was necessary and mapped to the [0, 1] interval range.

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

where $r_{\min}$ represents the minimum value of the current attribute feature, $r_{\max}$ represents the maximum value of the current attribute feature, and r_n represents the value after normalization.

4.3 Evaluation metrics

To obtain a comprehensive understanding of the overall model classification and the performance effect of classifying fewer classes of samples. The performance evaluation metrics selected in this paper include accuracy, recall, precision and $F1$-score, which were commonly used in similar studies, in addition to the commonly used accuracy rate. All of the above indicators were derived from analyzing and applying the basic attributes of the confusion matrix, including true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), and the specific explanation of each indicator is shown below:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

$$F1\text{ - score } = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{12}$$

where TP (true positive) is the quantity of real intrusion data that is detected as intrusion data and TN (true negative) is the quantity of data that is correctly discriminated as normal behavior. FN (false negative) is the quantity of intrusion data that is recognized as normal behavior, and FP (false positive) is the quantity of data where normal behavior is identified as intrusion data. Accuracy can evaluate the model performance. However, when unbalanced sample data are analyzed, the model does not evaluate the data correctly. Therefore, it is necessary to add accuracy and recall to the evaluation.

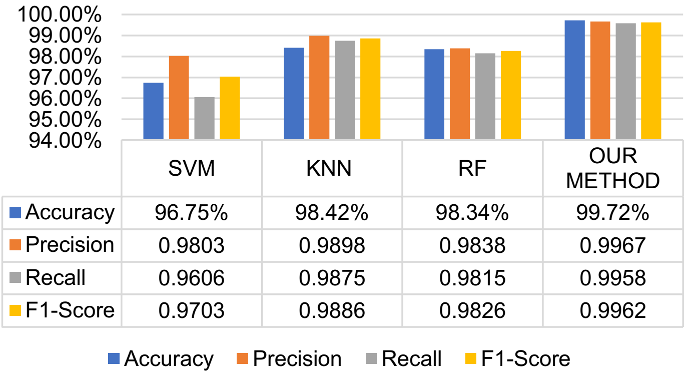
Precision is the ratio of data correctly identified as intrusions to those predicted to be intrusions. Recall is the ratio of data correctly identified as intrusive to data actually identified as intrusive. The $F1$-score is the harmonic average of precision and recall, which responds to the generalization ability of the model, and a model with a high $F1$-score is easier to apply to practical problems.

4.4 Experimental results on NSL-KDD

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

From Table 2, it can be seen that the enhanced random forest method proposed in this paper is effective in the accuracy of various types of attacks. The accuracy is high in all categories and only slightly lower than KNN in two of the attack types.

Therefore, the model proposed in this paper performs better overall in terms of accuracy rate. In terms of recall, the method in this paper outperforms other comparative algorithms in general and is only slightly lower than KNN in the normal category, but the model in this paper performs better in a few categories in terms of recall, indicating that the method in this paper to solve the data imbalance problem is effective.



	SVM	KNN	RF	OUR METHOD
■ Accuracy	96.75%	98.42%	98.34%	99.72%
■ Precision	0.9803	0.9898	0.9838	0.9967
■ Recall	0.9606	0.9875	0.9815	0.9958
■ F1-Score	0.9703	0.9886	0.9826	0.9962

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

In Table 3, the method in this paper is generally better than other comparison experiments in terms of the *F*1 score, and the *F*1 scores concentrate the precision and recall rates, which illustrate the effectiveness and feasibility of the method in this paper.

In Table 4 and Fig. 7, the experimental results on the test set KDDTest + show that the proposed method had a high recall on all attack types and had a significant advantage over other detection models, with obtained recall rates of 84.56, 77.53, 26.50 and 30.63, respectively. Compared with the original random forest model, there was a large improvement in all types except for the attack type, which was slightly lower, 11.08, 19.5 and 9.26.

Table 5 shows the statistical results of the *F*1-SCORE metrics for attack type to better evaluate the combined detection performance of each detection model for the four attack types. The model proposed in this paper obtained the highest *F*1-score of 79.98, 6.02, 6.50 and 30.63 for DoS, probe, U2R and R2L, respectively. Compared with other modeling methods, all types of results significantly improved. The *F*1 score concentrates the precision and recall rates, which indicate the effectiveness and feasibility of the method in this paper.

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

The classification algorithms were evaluated in the NSL-KDD dataset after being processed by the sampling method proposed in this paper, and the classification results are clearly shown in Figs. 8 and 9. The proposed hybrid method combined the K-means clustering method with the SMOTE sampling technique, which improved the detection effect of each classifier for minority class samples and effectively alleviated the problem of data imbalance. Therefore, the proposed method in this paper optimizes the intrusion detection dataset to a certain extent, and the sampled dataset has some practical significance compared with the original dataset.

In addition, we compared the proposed method with excellent research methods to further show the superiority of the proposed method. The comparison results obtained

Table 2 Comparison of performance with algorithms on the training set (5-class)

Type	Accuracy (%)				Precision (%)				Recall (%)			
	KNN	SVM	RF	Our method	KNN	SVM	RF	Our method	KNN	SVM	RF	Our method
Normal	92.47	99.63	98.96	99.91	99.32	98.02	98.41	99.68	99.65	95.62	98.19	99.57
Probe	94.23	99.68	98.74	99.74	97.42	96.77	98.02	96.02	96.19	94.97	84.06	98.54
DOS	99.65	99.57	99.59	99.99	99.76	89.95	96.25	99.88	99.79	94.23	97.87	99.93
U2R	99.96	99.88	99.91	99.92	91.72	76.42	77.54	82.37	76.25	82.22	78.75	91.25
R2L	96.08	99.45	93.37	99.89	93.85	87.26	89.25	96.32	84.49	79.85	77.09	93.10

Table 3 Comparison of the *F*1 score with the traditional machine learning method (5-class)

Attack type	*F*1-score (%)			
	KNN	SVM	RF	Our method
Normal	99.49	95.62	98.30	99.62
Probe	96.81	94.97	90.51	97.35
DOS	99.78	94.23	97.05	99.91
U2R	78.89	79.21	76.98	86.58
R2L	88.93	83.39	82.72	94.68

are described in detail in Table 6, where the dataset used, the classification model, the accuracy rate, and other factors were used as aspects of the comparison. As described in the table, in terms of the overall accuracy, the enhanced random forest method used in this paper and the random forest method used in the literature [19] had higher accuracies of 99.72 and 99.4, respectively. Second, compared with the classification methods such as SVM used in the literature [22], the detection accuracy of this paper improved.

5 Conclusion

In this paper, we analyzed the attack types and similarities of malicious intrusion attacks in the NSL-KDD dataset, and then an intrusion detection system model was proposed and discussed based on the enhanced random forest and SMOTE algorithm. First, training samples were equalized by combining the K-means algorithm with the SMOTE algorithm to some extent to compensate for the undertraining of smaller-scale samples. Then, the optimization of the similarity between decision trees was used to further enhance the random forest detection performance. Initial detection was obtained by enhancing random forest, and finally, the results were further corrected by the intrusion attack similarity. This paper evaluated the enhanced random forest algorithm on the NSL-KDD dataset and achieved a relatively ideal effect. In the future, our study will further optimize the model accuracy and computational overhead time through feature extraction and classifier selection. Intrusion detection systems have great research

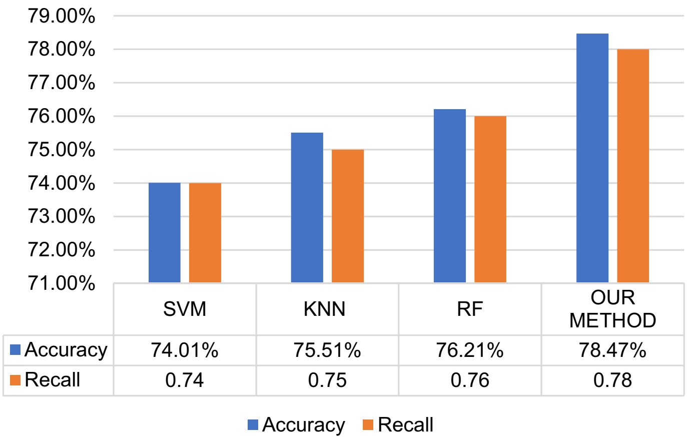


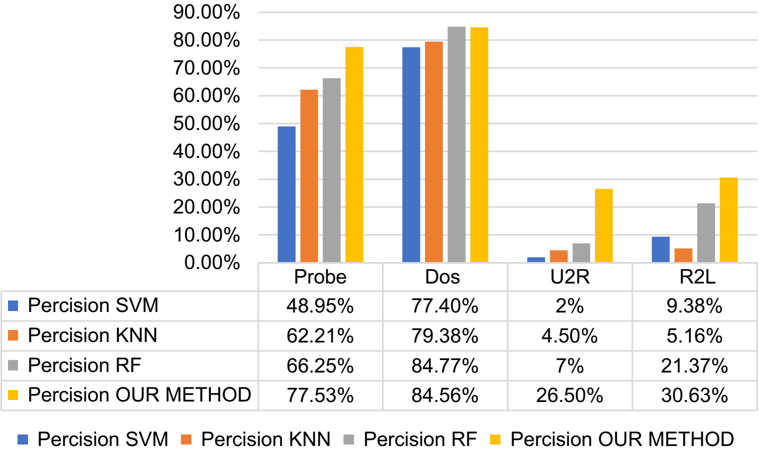
Fig. 7 Comparison of algorithms on KDDTest+

Table 4 Comparison of performance of algorithms on KDDTest+

Type	Accuracy (%)				Precision (%)				Recall (%)			
	KNN	SVM	RF	Our method	KNN	SVM	RF	Our method	KNN	SVM	RF	Our method
DOS	73.26	71.81	74.38	77.81	76.14	73.80	75.11	75.87	78.38	77.40	84.77	84.56
Probe	67.86	72.8	78.21	80.63	73.82	43.83	68.64	74.56	62.21	48.85	66.25	77.53
U2R	6.5	0	7	10	2	4.50	7	26.50	5	2.00	7	26.50
R2L	1.27	0	3.41	11.47	8.38	5.16	21.37	30.63	5.16	8.38	21.37	30.63

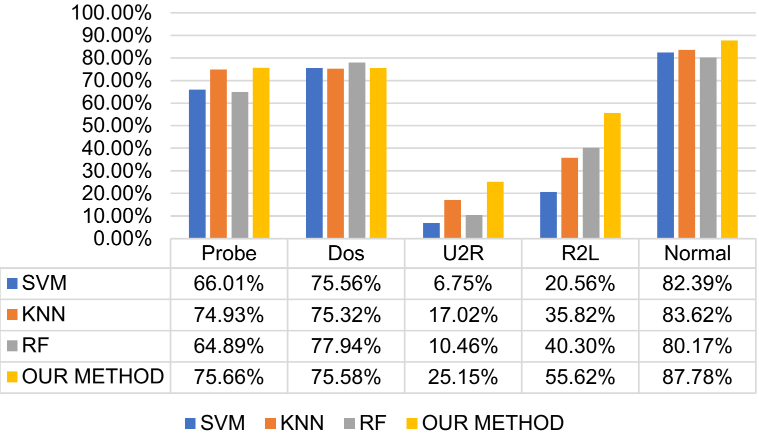
Table 5 Comparison of *F*1-score with the traditional machine learning method on KDDTest+

Type	*F*1-score (%)			
	KNN	SVM	RF	Our method
DOS	77.24	75.56	79.65	79.98
Probe	67.20	46.20	67.42	76.02
U2R	3.09	2.77	7.00	26.50
R2L	6.93	6.39	21.37	30.63



	Probe	Dos	U2R	R2L
■ Percision SVM	48.95%	77.40%	2%	9.38%
■ Percision KNN	62.21%	79.38%	4.50%	5.16%
■ Percision RF	66.25%	84.77%	7%	21.37%
■ Percision OUR METHOD	77.53%	84.56%	26.50%	30.63%

Fig. 8 Multiclass comparison of algorithms on the original dataset



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

Fig. 9 Multiclass comparison of algorithms on the sampling dataset

significance as an important method for defending against malicious activities, and the use of ensemble learning methods can further improve detection accuracy and robustness, so machine learning technology has an important role in advancing research in the network security field.

Table 6 Comparison of the proposed model with the state-of-the-art on the NSL-KDD

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

Abbreviations

IDS: Intrusion detection systems; SMOTE: Synthetic minority oversampling technique; SVM: Support vector machine; RF: Random forest; K-NN: K-Nearest neighbors; PSO: Particle swarm optimization; PCA: Principal component analysis; M-SMOTE: Mean SMOTE; AUC: Area under the curve; ROC: Receiver operating characteristic; IOB: In of bag; DOS: Denial of service attacks; U2R: Users to root attacks; R2L: Remote to local attacks; FPR: False positive rate; TP: True positive; TN: True negative; FP: False positive; FN: False negative.

Acknowledgements

Not applicable.

Author contributions

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

Funding

This work was supported by the National Science Fund of China (Nos. 61806088, 61902160), the QingLan Project of Jiangsu Province, and the Natural Science Research Project of Higher Education Institutions in Jiangsu Province (No. 20KJA520007).

Availability of data and materials

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

Declarations

Competing interests

The author declares that they have no competing interests.

Author details

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

Received: 23 February 2021 Accepted: 15 April 2022
Published online: 07 May 2022

References

1. G. Fernandes, J.J.P.C. Rodrigues, L.F. Carvalho, A comprehensive survey on network anomaly detection. Telecommun. Syst. **70**(3), 447–489 (2019). https://doi.org/10.1007/s11235-018-0475-8
2. D. Ramotsoela, A. Abu-Mahfouz, G. Hancke, A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study. Sensors **18**(8), 2491 (2018). https://doi.org/10.3390/s18082491
3. D.E. Denning, An intrusion–detection model. IEEE Trans. Softw. Eng. **2**, 222–232 (1987). https://doi.org/10.1109/TSE.1987.232894
4. F. Zhao, Detection method of LSSVM network intrusion based on hybrid kernel function. Mod. Electron. Tech. **21**, 027 (2015)
5. S.J. Horng, M.Y. Su, Y.H. Chen, A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Syst. Appl. **38**(1), 306–313 (2011). https://doi.org/10.1016/j.eswa.2010.06.066
6. P. Tao, Z. Sun, Z. Sun, An improved intrusion detection algorithm based on GA and SVM. IEEE Access **6**, 13624–13631 (2018). https://doi.org/10.1109/ACCESS.2018.2810198
7. K. Peng, V.C.M. Leung, Q. Huang, Clustering approach based on mini batch kmeans for intrusion detection system over big data. IEEE Access **6**, 11897–11906 (2018). https://doi.org/10.1109/ACCESS.2018.2810267
8. R.M. Elbasiony, E.A. Sallam, T.E. Eltobely, A hybrid network intrusion detection framework based on random forests and weighted K-means. Ain Shams Eng. J. **4**(4), 753–762 (2013). https://doi.org/10.1016/j.asej.2013.01.003

9. J.L. Leevy, T.M. Khoshgoftaar, R.A. Bauder, A survey on addressing high-class imbalance in big data. J. Big Data **5**(1), 1–30 (2018). https://doi.org/10.1186/s40537-018-0151-6
10. N. Ofek, L. Rokach, R. Stern, Fast-CBUS: a fast clustering-based undersampling method for addressing the class imbalance problem. Neurocomputing **243**, 88–102 (2017). https://doi.org/10.1016/j.neucom.2017.03.011
11. X. Ma, W. Shi, AESMOTE: adversarial reinforcement learning with SMOTE for anomaly detection. IEEE Trans. Netw. Sci. Eng. (2020). https://doi.org/10.1109/TNSE.2020.3004312
12. B. Yan, G. Han, Y. Huang, New traffic classification method for imbalanced network data. J. Comput. Appl. **38**(1), 20–25 (2018)
13. G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newsl. **6**(1), 20–29 (2004). https://doi.org/10.1145/1007730.1007735
14. H. Han, W. Wang, B. Mao, in *International Conference on Intelligent Computing. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning*, vol. 3644 (Springer, 2015), pp. 878–887. https://doi.org/10.1007/11538059_9
15. G. Bakirli, D. Birant, DTreeSim: a new approach to compute decision tree similarity using re-mining. Turk. J. Electr. Eng. Comput. Sci. **25**(1), 108–125 (2017). https://doi.org/10.3906/elk-1504-234
16. S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, Cyber intrusion detection by combined feature selection algorithm. J. Inf. Secur. Appl. **44**, 80–88 (2019). https://doi.org/10.1016/j.jisa.2018.11.007
17. B. Selvakumar, K. Muneeswaran, Firefly algorithm based feature selection for network intrusion detection. Comput. Secur. **81**, 148–155 (2019). https://doi.org/10.1016/j.cose.2018.11.005
18. R.C. Staudemeyer, C.W. Omlin, Extracting salient features for network intrusion detection using machine learning methods. S. Afr. Comput. J. **52**(1), 82–96 (2014). https://doi.org/10.18489/sacj.v52i0.200
19. A. Golrang, A.M. Golrang, S.Y. Yayilgan, A novel hybrid IDS based on modified NSGAII-ANN and random forest. Electronics **9**(4), 577 (2020). https://doi.org/10.3390/electronics9040577
20. J. Gao, S. Chai, B. Zhang, Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis. Energies **12**(7), 1223 (2019). https://doi.org/10.3390/en12071223
21. M. Belouch, S. El Hadaj, M. Idhammad, A two-stage classifier approach using reptree algorithm for network intrusion detection. Int. J. Adv. Comput. Sci. Appl. **8**(6), 389–394 (2017). https://doi.org/10.14569/IJACSA.2017.080651
22. F. Salo, A.B. Nassif, A. Essex, Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. Comput. Netw. **148**, 164–175 (2019). https://doi.org/10.1016/j.comnet.2018.11.010

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.