CIT. Journal of Computing and Information Technology, Vol. 31, No. 4, December 2023, 219–231
doi: 10.20532/cit.2023.1005771

An Intrusion Detection Model Based on Extra Trees Algorithm with Dimensionality Reduction and Oversampling

Li Yin and Yijun Chen

Informatization Center, Nantong University, Nantong, Jiangsu, China

With the advancement of the university information process, more and more application systems are running on the campus network, and the information system becomes larger and more complex. With the rapid growth of network users and the popularization and deepening of computer knowledge, the campus network has been transformed from an experimental network for education and scientific research into an operational network that attaches equal importance to education, scientific research, and service. As the most important transmission carrier of digital information, how to ensure its security has become an urgent issue for colleges and universities. Therefore, this paper uses advanced intrusion detection technology to design the corresponding model to solve the security problem of the campus network. When traditional machine learning algorithms train network intrusion data sets, they are prone to problems such as too many feature dimensions, overfitting, and imbalance of data sets, which lead to lower accuracy and low time efficiency of intrusion detection algorithms. In order to solve the above problems, this paper proposes an intrusion detection model based on Extra Trees, which uses linear discriminant analysis to reduce the dimension of data, then uses oversampling to reduce the influence of imbalance of sample categories of the network intrusion dataset, and finally uses Extra Trees algorithm to train the model. The experimental results show that after LDA reduction and oversampling, using the Extra Trees classification model can improve the overall recognition performance of imbalanced data sets under multi-classification and satisfy the network intrusion detection.

ACM CCS (2012) Classification: Security and privacy → Intrusion/anomaly detection and malware mitigation

Computing methodologies → Machine learning → Machine learning approaches → Classification and regression trees

Keywords: Intrusion detection, Random Forest, Extra Trees

1. Introduction

While the Internet brings great convenience to people, it also brings dangers. Criminals use network configuration errors, software vulnerabilities, and various intrusion tools to try to undermine network security for personal gain, exposing people to loss of data, money, time, and legal risks. Network security is increasingly becoming a serious issue of global concern. Many machine learning algorithms cannot accurately identify network intrusions because there are many category features, the dimensions obtained after data preprocessing are high, and most of the network intrusion data sets are imbalanced.

2. Literature Review

S. Forrest *et al.* [1] designed an intrusion detection system based on the principle of artificial immunity, and the behavior is divided into two categories: normal and abnormal. Based on the system mobilization data, the intrusion of various viruses was deeply studied.

Y. Ye *et al.* [2] propose an intrusion detection method based on pattern matching, which not only describes the characteristics of intrusions but also describes the structural relationship between systems.

H. Zhao *et al.* [3] apply an Artificial Neural Network (ANN) to misuse intrusion detection and abnormal intrusion detection, and Multi-Layer Perception (MLP) and Self-Organizing Map (SOM) models are adopted. MLP is used to record intrusion characteristics and respond to input signals. SOM preprocesses the data and makes it into MLP input data by performing dimensionality reduction operations on high-dimensional data.

P. R. Kanna *et al.* [4] apply the Hidden Markov Model (HMM) in a statistical model to an intrusion detection system. Q. Liu *et al.* [5] applied a wavelet transformation for the detection of denial of service attacks and achieved remarkable results. Y. Chen *et al.* [6] adopt Singular Value Decomposition to reduce the dimension of data and apply Spectrum Analysis to intrusion detection.

S. Cheng *et al.* [7] study the application of neural networks and Support Vector Machine (SVM) in intrusion detection. Experiments show that the optimized multi-objective SVM algorithm can minimize the empirical risk of intrusion detection in terms of accuracy, detection time, and scalability. It has a better detection effect and generalization ability than a simple neural network algorithm.

Since 1997, deep learning has developed very rapidly and has become the biggest breakthrough in the field of machine learning. It has been widely used in computer vision, speech recognition, and other fields, causing researchers to study its application in intrusion detection. A. Thakkar *et al.* [8] attempt to use deep learning models to train intrusion features, and the results running on the benchmark network intrusion dataset NSL-KDD are satisfactory. M. Mayuranathan *et al.* [9] propose a diverse deep learning framework where an autoencoder superimposes multiple layers of restricted Boltzmann machines (RBMs) and a layer of associative memory to detect new unknown malware through Windows API calls extracted from PE profiles. A. Javaid *et al.* [10] apply deep learning methods to the field of anomaly detection. Experiments on the NSL-KDD dataset show that the deep learning method has better results in intrusion detection.

S. Gurumurthy *et al.* [11] use a Convolutional Neural Network (CNN) for network intrusion detection. Network traffic is modeled as a time series, and TCP/IP packets are modeled using supervised learning methods within a predefined time range. The validity of the network structure in intrusion detection is also proved in the KDD99 data set.

Z. Li *et al.* [12] propose an effective deep learning method based on the deep self-learning framework for feature learning and dimensionality reduction, which effectively improves the accuracy of support vector machines against attacks and reduces the detection time of network intrusion detection systems.

S. Gurumurthy *et al.* [13] propose a coupled hidden Markov model based on recursion and conditional probability. M. Al-Qatf *et al.* [14] propose a new fitness function based on a fuzzy genetic algorithm and describe its application in the field of intrusion detection. N. Montazeri Ghahjaverestan *et al.* [15] propose an intrusion detection system based on neural fuzzy classifiers and uses a genetic algorithm to optimize the decision-making process of fuzzy logic.

R. Sivakami *et al.* [16] use multi-layer perceptron and reverse error propagation algorithm to detect shell code. Intrusion detection methods based on machine learning can learn data features better than intrusion detection methods based on statistics and game theory, so the overall accuracy is also higher. The hybrid intrusion detection model constructed by combining various machine learning methods can further improve the detection accuracy. Many comparison algorithms in this paper are based on this hybrid intrusion detection model.

Through the study of the above literature, it is found that the main focus of previous studies is to improve the accuracy of the classification algorithm, but a good algorithm should not only have high classification accuracy but also have a certain time efficiency guarantee.

Based on the above analysis, this paper proposes an intrusion detection model based on an improved Extra Trees algorithm and trains the network intrusion data set through reduction and oversampling, which can not only ensure the overall recognition performance of the multi-classification model but also meet the time efficiency of the practical application of network intrusion detection.

3. Research Method

3.1. Imbalanced data sets

The learning result of the classification algorithm for imbalanced data can be represented by a confusion matrix. Take the binary classification problem as an example: a few classes are defined as 1, and most classes are defined as 0. The confusion matrix is a matrix with two rows and two columns. The confusion matrix of the binary classification problem is shown in Table 1, where the rows represent the true values, and the columns represent the predicted values.

Based on the confusion matrix, the precision, recall and harmonic average (F1_Score) can be calculated. The calculation method is as follows:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{1}$$

$$\mathrm{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

$$\mathrm{F1\_score} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{4}$$

The precision represents how many of the predicted positive samples are actually positive; The recall rate represents how many positive cases in the sample were predicted correctly. The change curve of precision and recall rate with decision threshold is shown in Figure 1. The intersection point of the two is F1_Score, which is the harmonic average of precision and recall and can take both precision and recall rate into account.

Table 1. Confusion matrix for binary classification problems.

	0	**1**
0	Prediction Negative correct (TN)	Prediction Positive fault (FP)
1	Prediction Negative fault (FN)	Prediction Positive correct (TP)

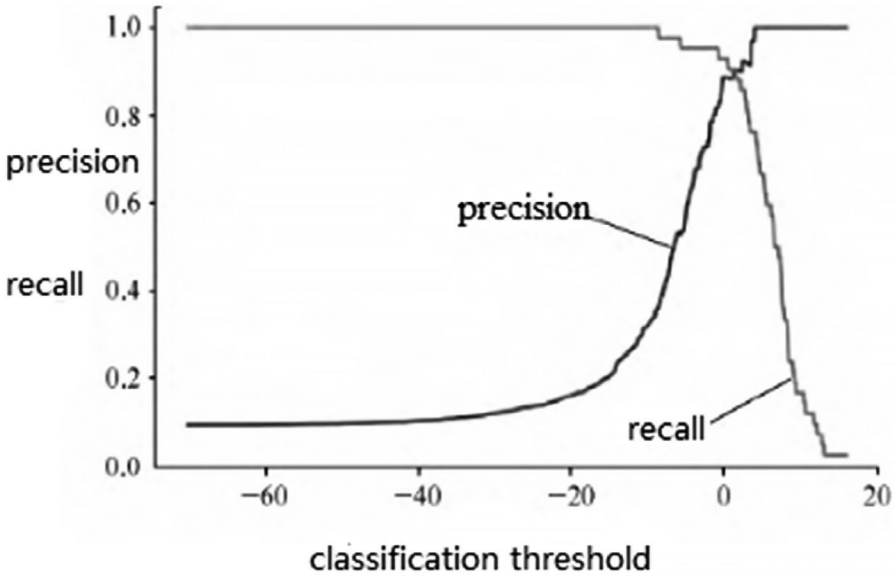


Figure 1. Change of accuracy and recall rate with threshold.

To deal with the imbalanced classification issue, the SMOTE algorithm is combined with Tomek Links. First, the SMOTE algorithm is used to oversample the original sample, and the Tomek Link method is used to delete the Tomek Link pairs on the new data set constructed by the SMOTE algorithm, to improve the sticking point problem, which can effectively alleviate the overfitting of the model. The Tomek Link pair is the distance between the majority sample point x and the minority sample point y. If no third point z makes or holds, then (x, y) is called a Tomek Link pair. Figure 2 shows the deletion of a Tomek Link pair. SMOTE can effectively resolve the existing problems:

1. Quality problems of synthetic samples,
2. Fuzzy class boundary problem.

1. For each sample in the minority class, the Euclidean distance is used as the standard to calculate its distance to all minority class samples, and its K-nearest neighbors are obtained.
2. For each minority sample, a number of samples are randomly selected from its K-nearest neighbors, and a new sample is constructed according to Formula (5):

$$x_{\text{new}} = x + \alpha \cdot (\overline{X} - x) \tag{5}$$

where, α is the sampling ratio; $\overline{X}$ is the K nearest neighbor mean.

3.2. LDA with Random Forest and Extra Trees

Linear discriminant analysis (LDA) is a linear dimensionality reduction method that minimizes the spread matrix within the class, meaning that each sample after LDA dimensionality reduction is classified [17]. The algorithm flow is as follows:

1. The n-dimensional input matrix X is normalized and $\overline{X}$ is obtained.
2. For each category of the input matrix $\overline{X}$, calculate the n-dimensional mean vector.
3. The mean value vector is used to construct the interclass spread matrix S1 and the intra-class spread matrix S2.
4. Calculate the eigenvalues and eigenvectors of the matrix S1-1S2. By selecting the eigenvectors corresponding to the first K eigenvalues, the N·K transformation matrix P is constructed.
5. Use equation (6) to map the transformation matrix P to a new feature subspace, and obtain the sample Q after dimensionality reduction:

$$Q = P^T \cdot \overline{X} \tag{6}$$

In the Random Forest (RF) algorithm the decision tree is based on a tree structure that uses an impurity index (Gini coefficient or information entropy) to continuously divide the data set into less uncertain subsets. When the decision tree is used for classification, the samples to be classified are judged from the root node of the tree structure, and according to the judgment

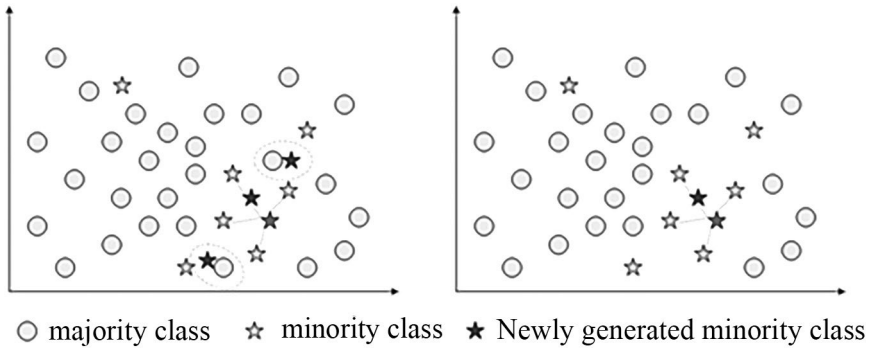


Figure 2. oversampling process

results, the next node is determined until the leaf node is reached, so as to obtain the classification result of the samples [18].

Extremely Randomized Trees (Extra Trees, ET) is an ensemble learning classification model built using a decision tree as a base classifier. The specific process is as follows:

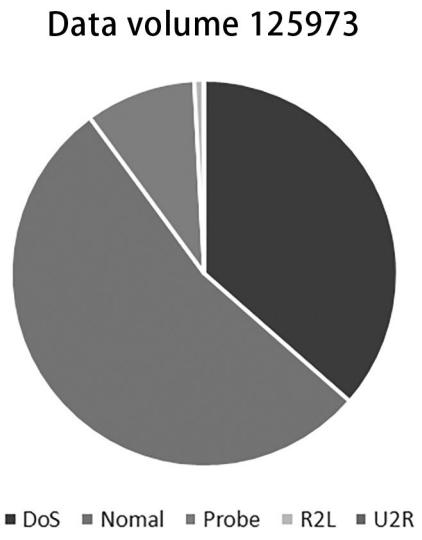
1. Build decision tree: In the process of forming a decision tree, each node randomly selects a feature for division and then splits downward continuously until a decision tree is formed.
2. Build a decision tree forest: Repeat step (1) to build a large number of decision trees to form a forest.
3. Input the test samples into the constructed forest: use each decision tree for classification, and then get the final classification result.

3.3. Intrusion detection model based on Random Forest and Extra Trees

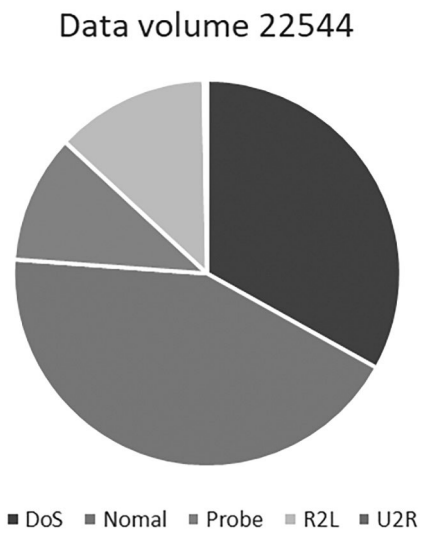
The NSL-KDD dataset contains five types of traffic, one of which is normal traffic on the network, and the other four are intrusion traffic, namely Dos, Probe, R2L, and U2R. The statistical diagram of the number of samples of various categories in the training set (KDDT rain+) and test set (KDDTeat+) of NLS-KDD is shown in Figure 3.

As shown in Figure 3, the distribution of samples in the data set is extremely imbalanced, in which the number of R2L and U2R in the training set samples only accounts for 0.790% and 0.041% of the total samples respectively.

The first step is data preprocessing. Each sample data in the NSL-KDD dataset is composed of 43 columns, the first 41 columns are the inherent characteristics of network traffic itself, the 4 second column is the label representing the data type, and the last column is used to mark the number of times that the sample can be correctly classified after classification by different machine learning models, which has nothing to do with the training and testing of the data set, so it needs to be deleted from the data. Non-numerical data cannot be processed by the machine learning algorithm, so the three non-numerical features Protocol, Type, Service, and Flag in 41-dimensional features need to be numerically transformed, and the method adopted is one-hot coding [19]. Take Protocol and Type features as examples. The four states are TCP, UDP, IP and ICMP, which are represented by four-bit binary numbers instead of non-numeric values, as shown in Table 2.



(a) NLS-KDD training data set sample distribution.



(b) NLS-KDD test set data sample distribution.

Figure 3. NLS-KDD sample statistics.

Table 2. Non-numerical feature one-hot encoding.

status	One-hot encoding
TCP	1000
UDP	0100
ICMP	0010
IP	0001

The features Service and Flag are separately thermally encoded in the same way and are represented in 70-bit and 11-bit binary, respectively. After numerical processing, all features can be represented numerically, but the value range of some features is very different, which will affect the convergence speed of subsequent model training, and it is necessary to conduct normalization processing. The normalization method adopted here is Z-Score normalization:

$$x'_i = \frac{x_i - \mu}{\sigma} \tag{7}$$

For a feature, x_i represents the eigenvalue of the JTH sample; x'_i is the value of this feature in i sample after normalization; μ mean values of this feature on all samples. σ Standard deviation of this feature on all samples. After unique thermal coding and normalization, the 41-dimensional features of the NSL-KDD dataset are mapped to 122 features.

Next, LDA reduction and the SMOTETomek algorithm are used for sampling. As mentioned above, the pre-processed data features are up to 122 dimensions, which need to be reduced. Common dimensionality reduction methods include LDA and principal component analysis (PCA). This paper gives the reasons for choosing LDA for data dimensionality reduction in 4.1. In addition, the sample distribution of the training set was extremely imbalanced, and the R2L and U2R data accounted for less than 1% of the total samples, so they could not directly participate in the model training, and data sampling was required. In this paper, the SMOTE-Tomek algorithm was used to oversample the imbalanced data set.

The training and testing process based on the Random Forest and Extra Trees models is conducted as follows. At present, the most relevant approaches divide KDDTrain+ of NSL-KDD data set into a training set and test set in proportion, and then conduct model training and test, and then use KDDTest+ for model verification. This paper directly uses KDDTrain+ as the training set and KDDTest+ as the test set, which can more truly reflect the classification ability of the model [20]. The training and testing process of the intrusion detection model is as follows:

1. Data import: The network intrusion data set of NSL-KDD (KDDTrain+, KDDTest+) is imported, and the subcategories of network intrusion are merged into five categories, with KDDTrain+ as the training set and KDDTest+ as the test set.
2. Data preprocessing: the non-numerical features are encoded by unique heat, and then the numerical features are normalized.
3. Data dimensionality reduction: the training set and test will be preprocessed Set LDA dimension reduction.
4. Data sampling: Oversampling the test set with SMOTE algorithm.
5. Base classifier training: The model is trained by a decision tree to compare the influence of oversampling on classification performance.
6. Construction of machine learning models: The decision tree is used as the base classifier to build a Random Forest and Extra Trees, and the over-sampled data is used as the model input to vote on the classification results and get the final classification results.
7. Model comparison: Extra Trees and Random Forest models were compared, and classification performance index F1_Score and model training convergence time index were compared [21].

Model training parameters are set as follows:

1. Basic parameters of Extra Trees: the impurity index is set as the Gini coefficient; The number of trees (n_estimators) is set to 100. The maximum depth of the tree (max_depth) is set to 10.
2. Basic parameters of Random Forest: the impurity index is set as Gini coefficient; The number of trees (n_estimators) is set to 100. The maximum depth of the tree (max_depth) is set to 10.

4. Results

4.1. Analysis of Data Dimension Reduction Results

The pre-processed KDDTrain+ data sets were reduced by PCA and LDA respectively. The experimental results show that the sample data of Probe and DOS categories are relatively dispersed after PCA reduction, while the data of different categories are separated as far as possible after LDA reduction. LDA can reasonably use classification labels to make the projected dimensions more discriminative, which can be used for dimension reduction and classification. In this paper, LDA is used for data dimensionality reduction. After analyzing the data sampling results, SMOTE oversampling was conducted on the training set after LDA dimensionality reduction, and the sample distribution was obtained as shown in Figure 4.

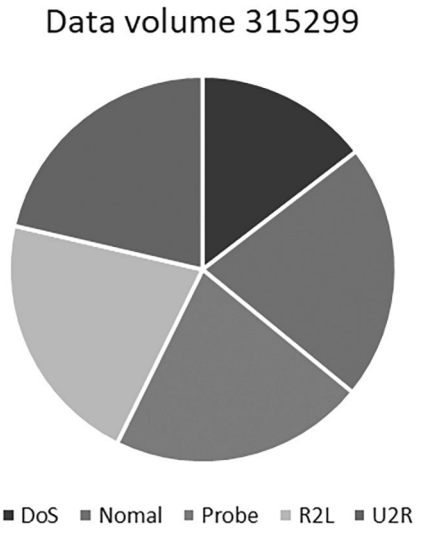


Figure 4. Sample distribution of training set after oversampling.

In order to verify the influence of data sampling on the model, the decision tree was selected as the base classifier, the unsampled test set and SMOTE over-sampled test set were used as the input of the base classifier, and the confusion matrix was obtained, as shown in Figure 5 and Figure 6.

In order to visually see the impact of data sampling on the classification of imbalanced sample categories, the classification accuracy of the confusion matrix in Figure 5 and 6 is converted into a value as shown in Table 3.

In Table 3, after using SMOTE oversampling, the accuracy of the imbalanced category R2L increased from 5.21% to 23.83%, and the accuracy of the DoS and Probe categories also increased. SMOTE oversampling can improve the overall recognition performance of the multi-model.

Table 3. Decision tree classification accuracy.

type	Precision before sampling %	Precision after sampling %
DoS	78.69	79.11
Normal	97.23	95.36
Probe	72.33	80.03
R2L	6.34	25.57
U2R	39.9	37.47

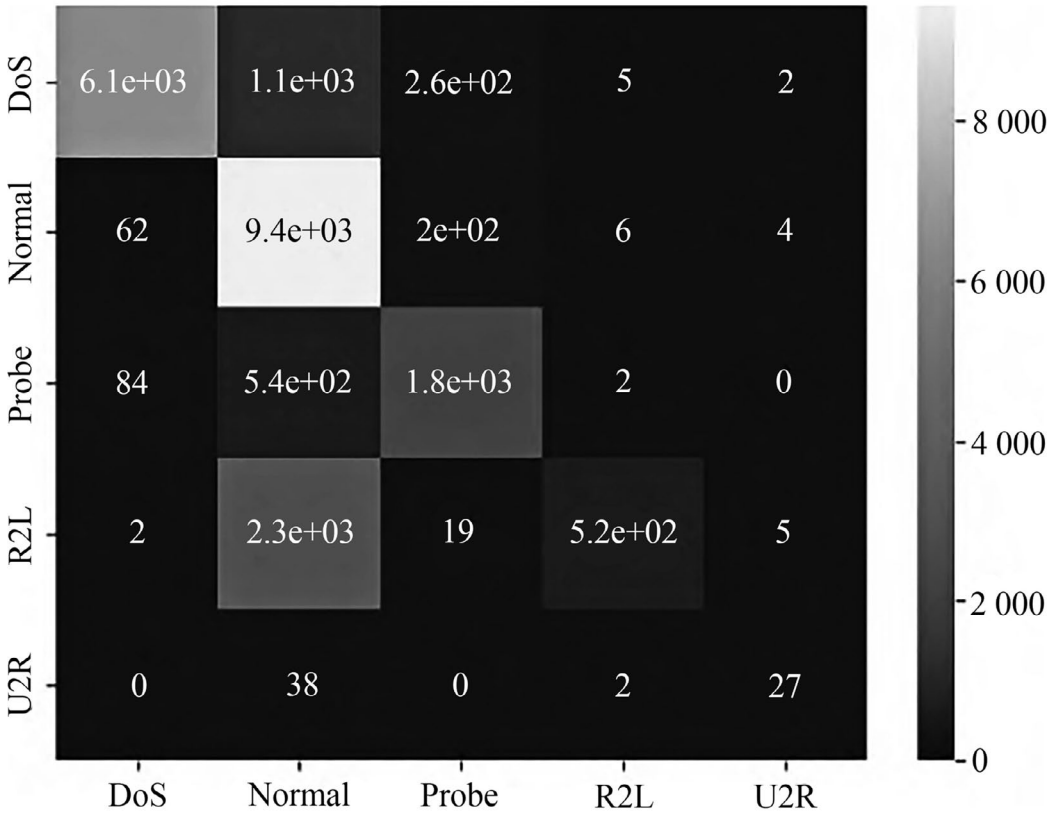


Figure 5. Confusion matrix of base classifier (decision tree) before sampling.

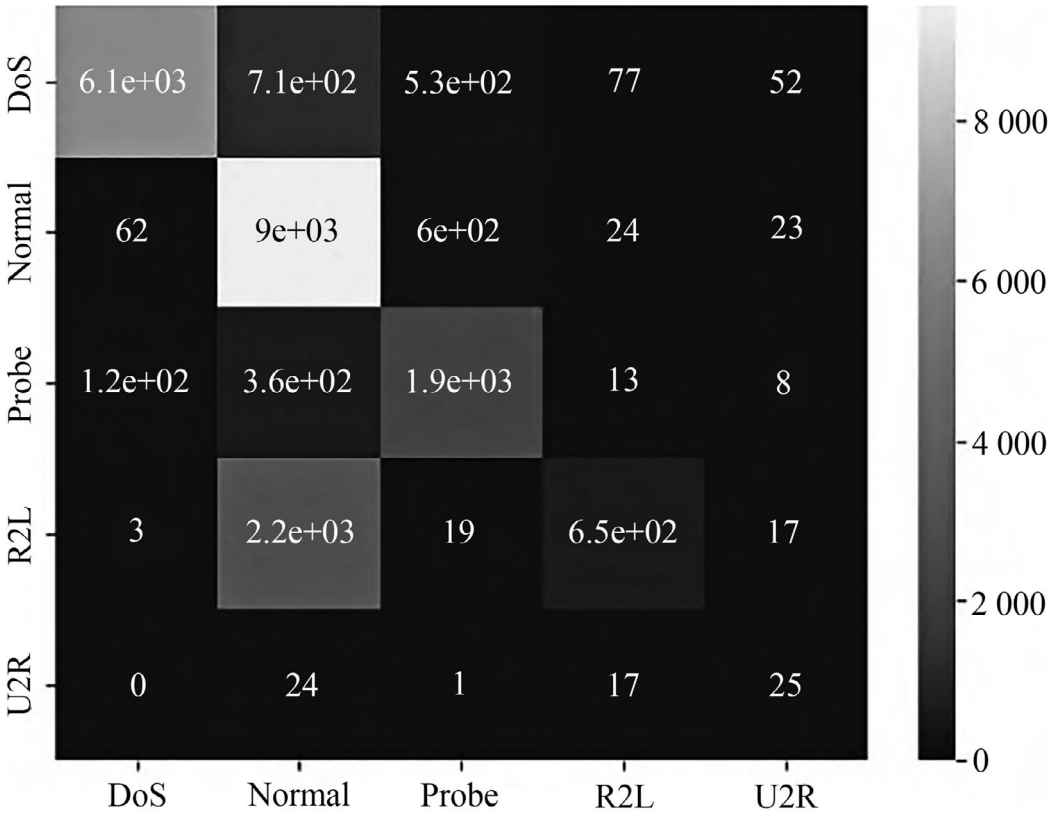


Figure 6. Confusion matrix of base classifier (decision tree).

4.2. Experimental Results of Extra Trees Model

In the previous section, the decision tree model was used as the base classifier, and then the Extra Trees algorithm was used to build the ensemble learning model. The pre-sampling data set and post-sampling data set were used as the input to the model for model training and the corresponding test data set was used for testing. The confusion matrix obtained is shown in Figure 7 and Figure 8. The classification accuracy of the confusion matrix in Figure 7 and Figure 8 is converted into a value as shown in Table 4.

Compared with decision trees, extreme random trees after ensemble learning can greatly improve the recognition ability of imbalanced sample categories (R2L, U2R), especially for U2R categories that are difficult to be detected by decision trees, and the classification accuracy rate after sampling can be increased to 73.13%.

Table 4. Extreme random tree classification precision.

type	Precision before sampling %	Precision after sampling %
DoS	78.35	78.21
Normal	97.12	94.75
Probe	81.72	92.17
R2L	4.55	85.99
U2R	30.52	70.44

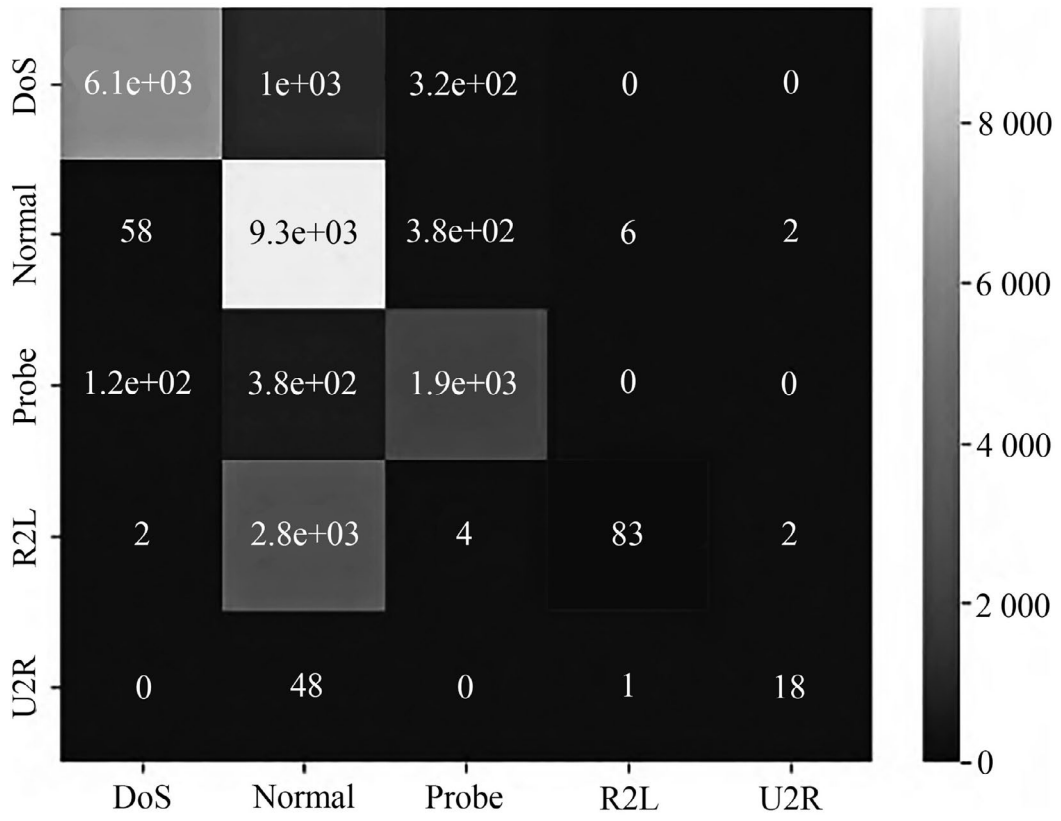


Figure 7. Confusion matrix of Extra Trees before sampling.

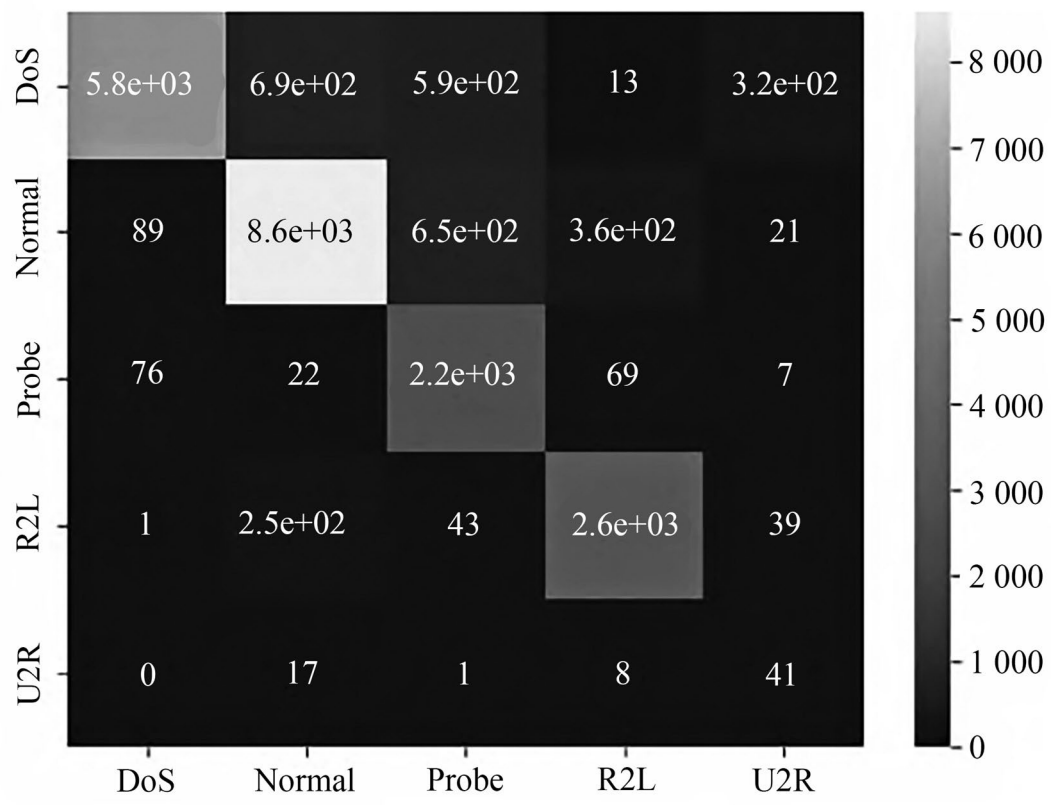


Figure 8. Confusion matrix of extreme random tree (after sampling).

5. Discussion

In this section, we compare Extra Trees and Random Forest models on the sampled data set. F1_Score is used as the evaluation index of model classification performance, and the convergence time of model training is used as the evaluation index of model efficiency. Table 5 and Figure 9 are obtained by comparing Extra Trees and Random Forest models.

As shown in Table 5, Extra Trees and Random Forest are very close in overall classification performance, but Figure 9 shows that the training convergence time of the Extra Trees model is much lower than that of Random Forest, and it is more suitable for the real-time requirements of network intrusion detection. Finally, the detection accuracy of the intrusion detection model based on Extra Trees is compared with other newly proposed intrusion detection models on KDDTest+, as shown in Table 6. It can be found that compared with other methods, the model proposed in this paper has a higher detection accuracy on the KDDTest+ data set.

6.Conclusion

Based on the improved Extra Trees Model, intrusion detection is studied and demonstrated in this paper. Dimensionality reduction is used to preprocess data to greatly improve the time efficiency during data classifier training. Oversampling is used to preprocess data to ensure the effect of dimensionality reduction and improve the accuracy of the classifier during training. The experimental results show that the problem of excessive input feature dimensions can be solved by LDA dimensionality reduction. SMOTE oversampling can improve the classification effect of network intrusion detection imbalance and imbalanced categories in the data set. In terms of model selection, the improved Extra Trees model based on ensemble learning can meet the requirement of time efficiency of network intrusion detection while ensuring the overall classification performance of the model.

Table 5. F1_Score comparison between Random Forest and Extra Trees.

Type	Random Forest	Extra Trees
DoS	0.84	0.88
Normal	0.79	0.80
Probe	0.74	0.74
R2L	0.35	0.34
U2R	0.31	0.31

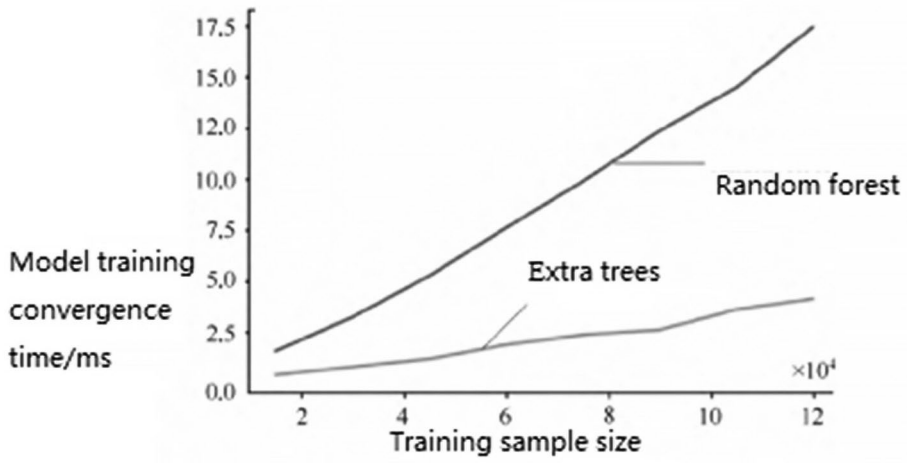


Figure 9. Comparison of training time between Random Forest and Extra Trees.

Table 6. The precision of each model is compared.

Literature	Model	Precision %
This paper	Extra Trees	86.13
Ambusaidi	SVM (Feature selection method)	78.42
Majjed	SVM (Sparse self-coding)	81.44
Gao	Ensemble learning (voting method)	85.3

References

[1] S. Forrest *et al.*, "A Sense of Self for Unix Processes", in *Proceedings of the 1996 IEEE Symposium on Security and Privacy, IEEE*, 1996, pp. 120–128. http://dx.doi.org/10.1109/SECPRI.1996.502675

[2] Y. Ye *et al.*, "DeepAM: A Heterogeneous Deep Learning Framework for Intelligent Malware Detection", *Knowledge and Information Systems*, vol. 54, no. 2, pp. 265–285, 2018. http://dx.doi.org/2018.10.1007/s10115-017-1058-9

[3] H. Zhao *et al.*, "Artificial Intelligence based Ensemble Approach for Intrusion Detection Systems", *Journal of Visual Communication and Image Representation*, vol. 71, p. 102736, 2020. http://dx.doi.org/10.1016/j.jvcir.2019.102736

[4] P. R. Kanna and P. Santhi, "Unified Deep Learning approach for Efficient Intrusion Detection System using Integrated Spatial–Temporal Features", *Knowledge-Based Systems*, vol. 226, p. 107132, 2021. http://dx.doi.org/10.1016/j.knosys.2021.107132

[5] Q. Liu *et al.*, "A Multi-task Based Deep Learning Approach for Intrusion Detection", *Knowledge-Based Systems*, vol. 238, p. 238, 2022. http://dx.doi.org/10.1016/j.knosys.2021.107852

[6] Y. Chen *et al.*, "Intrusion Detection Using Multi-objective Evolutionary Convolutional Neural Network for Internet of Things in Fog Computing", *Knowledge-Based Systems*, vol. 244, p. 108505, 2022. http://dx.doi.org/10.1016/j.knosys.2022.108505

[7] S. Cheng and Z. Wang, "Solve the IRP Problem with an Improved Discrete Differential Evolution Algorithm", *International Journal of Intelligent Information and Database Systems*, vol. 12, no. 1–2, pp. 20–31, 2019. http://dx.doi.org/10.1504/ijiids.2019.102324

[8] A. Thakkar and R. Lohiya, "Analyzing Fusion of Regularization Techniques in the Deep Learning-based Intrusion Detection System", *International Journal of Intelligent Systems*, vol. 36, no. 12, pp. 7340–7388, 2021. http://dx.doi.org/10.1002/int.22590

[9] M. Mayuranathan *et al.*, "An Efficient Optimal Security System for Intrusion Detection in Cloud Computing Environment Using Hybrid Deep Learning Technique", *Advances in Engineering Software*, vol. 173, no. 3, p. 103236, 2022. http://dx.doi.org/10.1016/j.advengsoft.2022.103236

[10] A. Javaid *et al.*, "A Deep Learning Approach for Network Intrusion Detection System", in *Proceedings of the International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), BICT'15*, 2016, pp. 21–26. http://dx.doi.org/10.4108/eai.3-12-2015.2262516

[11] S. Gurumurthy *et al.*, "Hybrid Pigeon Inspired Optimizer-gray Wolf Optimization for Network Intrusion Detection", *Journal of System and Management Sciences*, vol. 12, no. 4, pp. 383–397, 2022. http://dx.doi.org/10.33168/JSMS.2022.0423

[12] Z. Li *et al.*, "Intrusion Detection Using Convolutional Neural Networks for Representation Learning", in *Proceedings of the International Conference on Neural Information Processing, ICONIP 2017: Neural Information Processing*, 2017, pp. 858–866. http://dx.doi.org/10.1007/978-3-319-70139-4_87

[13] S. Gurumurthy *et al.*, "Hybrid Pigeon Inspired Optimizer-gray Wolf Optimization for Network Intrusion Detection", *Journal of System and Management Sciences*, vol. 12, no. 4, pp. 383–397, 2022. http://dx.doi.org/10.33168/JSMS.2022.0423

[14] M. Al-Qatf *et al.*, "Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection", *IEEE Access*, vol. 6, pp. 52843–52856, 2018. http://dx.doi.org/10.1109/ACCESS.2018.2869577

[15] N. Montazeri Ghahjaverestan *et al.*, "Coupled Hidden Markov Model-Based Method for Apnea Bradycardia Detection", *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 2, pp. 527–538, 2016. http://dx.doi.org/10.1109/JBHI.2015.2405075

[16] R. Sivakami *et al.*, "Dirichlet Feature Embedding with Adaptive Long Short Term Memory Model for Intrusion Detection System", *Journal of System and Management Sciences*, vol. 12, no. 4, pp. 398–412, 2022. http://dx.doi.org/10.33168/JSMS.2022.0424

[17] R. Sivakami *et al.*, "Dirichlet Feature Embedding with Adaptive Long Short Term Memory Model for Intrusion Detection System", *Journal of System and Management Sciences*, vol. 12, no. 4, pp. 398–412, 2022. http://dx.doi.org/10.33168/JSMS.2022.0424

[18] M. S. Abadeh *et al.*, "Intrusion Detection Using a Fuzzy Genetics-based Learning Algorithm", *Journal of Network & Computer Applications*, vol. 30, no. 1, pp. 414–428, 2007. http://dx.doi.org/10.1016/j.jnca.2005.05.002

[19] A. N. Toosi and M. Kahani, "A New Approach to Intrusion Detection Based on an Evolutionary Soft Computing Model Using Neuro-fuzzy Classifiers", *Computer Communications*, vol. 30, no. 10, pp. 2201–2212, 2007. http://dx.doi.org/10.1016/j.comcom.2007.05.002

[20] B. Zhong and S. Cheng, "A Fast Encryption Method of Social Network Privacy Data Based on Blockchain", *International Journal of Web Based Communities*, vol. 18, no. 3–4, pp. 345–356, 2022. http://dx.doi.org/10.1504/ijwbc.2022.125502

[21] S. Liu, "Research on Computational Methods and Algorithms for Dimensionality Reduction and Feature Selection in High-Dimensional Data", *Journal of Logistics, Informatics and Service Science*, vol. 10, no. 3, pp. 1–12, 2023. http://dx.doi.org/10.33168/JLISS.2023.0301

Received: December 2023
Revised: January 2024
Accepted: January 2024

Contact addresses:
Li Yin
Informatization Center
Nantong University
Nantong
Jiangsu
China
e-mail: ntuyinli@163.com

Yijun Chen*
Informatization Center
Nantong University
Nantong
Jiangsu
China
e-mail: ntucyj@163.com
*Corresponding author

Li Yin holds a Master's degree obtained from Yangzhou University in June 2009. Currently, she works at Nantong University, where she is actively involved in research in the field of Educational Informationization, computer-aided teaching, and social paperless exams.

Yijun Chen holds a Master's degree obtained from Yangzhou University in December 2009. Currently, he is affiliated with Nantong University, where he specializes in research areas such as Educational Informationization, Network Security, and Information System Development.