Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 6, pp. 5171–5183, DOI 10.12694/scpe.v25i6.3261

NETWORK DATA INTRUSION DETECTION AND DATA FEATURE EXTRACTION OF ELECTROMECHANICAL FACILITIES FROM MACHINE LEARNING

TING XU*, LIJUN WANG†, YANHONG HU‡ AND XUMING TONG§

Abstract. With the rapid development of Internet technology, network security issues have become more complex and changeable. Various intrusion methods threaten the information network environment in the Electromechanical Facility (EF) system. This paper focuses on EF to study the relevant detection methods and data feature extraction in complex network intrusions. Firstly, four common machine learning algorithms are used to calculate the data set. The advantages and disadvantages of each algorithm are analyzed after tuning and comparison. Secondly, a network intrusion detection algorithm is proposed based on Recursive Feature Elimination (RFE) principal component analysis. It uses RFE to reduce the number of features and improve the elimination judgment index to align with the detection requirements of information network datasets. Finally, a fault diagnosis method is proposed based on empirical pattern decomposition and support vector machine under Renyi entropy complexity measurement. This method trains and identifies the Renyi entropy of several basic pattern components obtained by decomposing empirical patterns as feature vectors. The results show that the RFE method judged by random forest removes irrelevant features, and the evaluation index is improved to align with the network dataset's detection requirements. It reduces the data dimension, reduces the operation time, and improves the accuracy of a few attack types. The comprehensive final detection effect is better than other algorithms. Additionally, the embedded operating system construction method based on the protection mechanism realizes the separate storage of the operating system and key data. Also, it can prevent the network system from being maliciously invaded, ensuring the stability of the instrument operation under harsh working conditions.

Key words: Machine learning, electromechanical facility, network data intrusion, data feature extraction, detection algorithm

1. Introduction.

With the rapid development of mobile Internet information technology such as cloud computing, big data, the Internet of Things, and artificial intelligence, traditional industries have been transformed into digitalization. The information network has achieved full coverage, popularization, and application in various fields and industries. The resulting information data has also grown exponentially. Big data has become an emerging resource [1]. Data growth can also bring some problems. In the Electromechanical Facility (EF) system, the information and control system are installed between each facility, which integrates computing, communication, and electromechanical systems to form a highly interconnected intelligent network information system. However, human factors attack the EF network, and the information system of the electromechanical enterprise is destroyed. Physical attacks can arbitrarily rewrite web pages and delete information data, which will seriously affect the integrity of information networks. In addition, various intrusion and cyber-attack tools and methods also exist in information system networks. Virus Trojan implants, Distributed Denial of Service attacks, phishing, and vulnerability attacks in the network all threaten the system's network security. Cybersecurity issues are also on the rise [2]. Based on this, this paper introduces the research results of Machine Learning (ML) in big data into intrusion detection for the network intrusion of EF. Various ML algorithms include Random Forest (RF), Gradient Boosting Decision Tree (GBDT), Adaptive Boosting (AdaBoost), and eXtreme Gradient Boosting (XGBoost). Among them, the Decision Tree (DT) is easy to understand and interpret, can be visualized and analyzed, and can easily extract rules. RF is used to classify and predict using multiple tree classifiers. Meanwhile, it can process high-dimensional datasets and solve nonlinear problems. The neural network has high classification accuracy, strong learning ability, and robustness and fault tolerance to noisy data, which provides good help for the research of network security intrusion detection.

*Hebei North University, Zhangjiakou, Hebei, 075132, China (TingXu61@126.com)
†Hebei North University, Zhangjiakou, Hebei, 075132, China (LijunWang36@163.com)
‡Hebei North University, Zhangjiakou, Hebei, 075132, China (YanhongHu7@126.com)
§Hebei North University, Zhangjiakou, Hebei, 075132, China (Corresponding author's e-mail:XumingTong7@163.com)

2. Literature Review. Kobayashi wrote a technical report called "Computer Security Threat Monitoring and Surveillance," which first coined the term "threat." It is similar to the meaning of intrusion anomaly, indicating potentially unauthorized access to the system, resulting in a vulnerability of the entire system [3]. Thirimanne combined both statistical and rule-based techniques to design a new system. The system model can cope with real-time intrusion detection to become a new network defense measure. This system has a great effect on intrusion detection research [4]. Then, Moustafa wrote a paper titled "A Network Security Monitor." Network flows were used directly as a data source for the first time. Monitors were used to obtain the Transmission Control Protocol/Internet Protocol Address packets. The system could also be detected when the data is not converted into a uniform format. From this, network intrusion detection began to derive [5].

Shen used binarization techniques to decompose the original dataset into subsets of binary classes. Then, the Synthetic Minority Oversampling Technique (SMOTE) algorithm was applied to each subset of the unbalanced binary class to obtain balanced data. Finally, an RF classifier was used to achieve the classification goal [6]. Zhang proposed a Copy/Paste Detector-SMOTE algorithm. The neighbor set with high correlation was determined from the characteristics of the small sample and the surrounding sample distribution of the training set. The neighbor set was expanded into a new sample set by the SMOTE algorithm, which had a good effect on processing the unbalanced dataset [7]. Jahangir combined the Principal Component Analysis (PCA) algorithm and the SMOTE algorithm to denoise and reduce the dimensionality of the data set before interpolating the data. Modeling with the RF algorithm could improve the classification performance of unbalanced datasets [8]. Tong further distinguished boundary samples by Borderline-smote, generating different numbers of synthetic samples for different boundary samples, further improving Borderline-smote [9]. Wang proposed an upsampling method for secondary synthesis. The first synthesis was performed on samples that contained important information in the support selection of minority samples. Then, according to how the centroid of minority samples was distributed on samples in the neighborhood, the synthesis range of the second sample was optimized to form a secondary synthesis [10].

The innovations in this paper are as follows. First, the network data intrusion detection of EF is studied. Besides, various algorithms are applied for comparative experiments. In terms of accuracy, the advantages and disadvantages of each algorithm in the attack detection effect in the data set are analyzed. Second, in terms of running time, each algorithm's calculation and running time are counted. The two are combined to evaluate, and the algorithm with the better experimental effect is selected as the basic algorithm. Third, data feature extraction based on network intrusion detection can prevent the network system from being maliciously intruded on and ensure the stability of instrument operation under harsh working conditions.

3. Establishment of optimization model of construction parameters.

3.1. Algorithms related to network data intrusion of EF.

3.1.1. RF algorithm. RF uses resampling technology. There are put back from the training set repeated random selection of some data to form a new data set. DT is used for training, and a certain number of DT is used to form an RF. The final result is determined by the number of votes cast in the DT [11]. The essence is to improve the algorithm for DT conduct. Multiple DTs are combined from a single DT. Each tree is independent of the other. A DT is built according to the different samples taken. So, a single tree may not be very effective in classification. Still, the test of the sample by the forest formed by multiple trees will result in a more accurate classification after statistics.

The total number of training sets of the RF algorithm is N. A single DT is to randomly take n training samples from the training set, and bootstrap has a putback sample. When the input feature of the training sample is M, M is much greater than m. Each DT splits according to characteristics. m features are randomly selected out of M. If the Gini coefficient is used as the basis for division, one of the characteristic attributes is selected to split until all the characteristic attributes have been used.

When using Gini index splitting, if there are m samples of different classes in the training sample set T, the Gini index of the sample set is:

$$gini(T) = 1 - \sum_{i=1}^{m} p_i^2 \# \tag{3.1}$$

In Equation 3.1, p_i is the probability of the type i sample. For sample T contains l sample subsets $T_1, T_2, ..., T_l$, the subset contains a sample number of $N_1, N_2, ..., N_l$, respectively. The splitting Gini coefficient is:

$$gini_{spht}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)\# + \cdots + \frac{N_1}{N} gini(T_1)\# \tag{3.2}$$

3.1.2. GBDT algorithm. GBDT is also a type of ensemble learning. Compared with the traditional AdaBoost algorithm, it is different from iterating again and again by updating the weight of the error rate of the previous weak learner. GBDT uses a forward distribution algorithm. The weak learner regression tree model is used, and the iterative method is also different from AdaBoost [12].

When using an iteration of GBDT, if the strong learner obtained by the previous iteration is $f_{t-1}(x)$, the loss function is:

$$L(y, f_{t-1}(x)) = L(y, f_{t-1}(x) + h_t(x))\# \tag{3.3}$$

Each round of iteration will find the weak learner $h_t(x)$ on the CART regression tree model to minimize the loss function each time. The resulting DT should minimize sample loss.

The multivariate GBDT classification algorithm code is as follows.

If there are K classes, the log-likelihood loss function is:

$$L(y, f(x)) = -\sum_{k=1}^{K} y_k log p_k(x)\# \tag{3.4}$$

Suppose there are k sample output categories, = 1. The expression for the k-type probability $p_k(x)$ is:

$$p_k(x) = \frac{exp(f_k(x))}{\sum_{k=1}^{K} exp(f_l(x))}\# \tag{3.5}$$

From the above two equations, it can be concluded that the negative gradient error of the class l corresponding to the ith sample of round t is:

$$r_{til} = -[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}]_{f_k(x)=f_{l,t-1}(x_i)}\# = y_{il} - p_{l,t-1}(x_i)\# \tag{3.6}$$

In Equation 3.6, the error on the sample is the probability that sample i is the true l-category minus the probability of prediction at round t-1. The optimal negative gradient fit value for each leaf node in the resulting DT is:

$$c_{tjl} = argmin \sum_{i=1}^{m} \sum_{k=1}^{K} L(y_k, f_{l,t-1}(x_i)) + \sum_{j=1}^{J} c_{tjl}(x_i \in R_{tjl})\#\#\# \tag{3.7}$$

Since the above equation is difficult to optimize, it is generally used as an approximation. Then, Equation 3.8 is obtained.

$$c_{tjl} = \frac{K+1}{K} \frac{\sum_{x_i \in R_{til}} r_{til}}{\sum_{x_i \in R_{til}} |r_{til}|(1 - |r_{til}|)}\# \tag{3.8}$$

In calculating negative gradient and the best negative gradient fitting of leaf nodes, the operations of multi-classification, binary classification, and regression algorithms are similar.

3.1.3. AdaBoost algorithm. The AdaBoost algorithm adaptively enhances the error samples of the previous basic classifier and retrains the next basic classifier with weighted samples. Additionally, the latest weak path is added to each operation. The operation is stopped when a pre-specified range of error rates is met, or the maximum iteration value has been run [13].

The AdaBoost classification algorithm process is as follows.

The input is the sample set $T = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$. The output is {-1,1}. Then, it is a weak classifier algorithm, and the number of weak classifier iterations is K. The output is the final strong classifier.

(1) The initialized sample set weights are as follows.

$$D(1) = (\omega_{11}, \omega_{12}, \cdots, \omega_{1m}); \omega_{1i} = \frac{1}{m}; i = 1, 2, \cdots, m\# \tag{3.9}$$

(2) When k = 1, 2,...,K,

(a) For the dataset adding the weight D_k, the generated weak classifier is $D_k(x)$.

(b) The classification error rate of $D_k(x)$ is:

$$C_k = P(G_k(x_i) \neq y_i) = \sum_{i=1}^{m} w_{ki} I(G_k(x_i) \neq y_i) \tag{3.10}$$

(c) The coefficients of the weak classifier are calculated as:

$$\beta_k = \frac{1}{2} log \frac{1 - C_k}{C_k} + log(R - 1) \tag{3.11}$$

In Equation 3.11, R is the number of categories. From the above equation, if it is a binary classification, R=2. If it is a quintuple classification, R=5.

(d) The weight distribution of the updated sample set is:

$$w_{k+1,i} = \frac{w_{ki}}{Z_k} \beta xp(-C_k y_i G_k(x_i))\# \tag{3.12}$$

Here, Z_k is the normalization factor, as shown in the following equation.

(3) The final classifier is constructed as:

$$f(x) = sign \sum_{k=1}^{K} \beta_k G_k(x)\# \tag{3.13}$$

3.1.4. XGBoost algorithm. XGBoost is an open-source ML project. It is one of the boosting algorithms, a good classifier that integrates many tree models. Algorithms and engineering improvements are carried out efficiently based on GBDT to make it more powerful and suitable for a larger range to improve the tree model [14].

The main flow of the XGBoost algorithm is as follows.

The input is the sample set $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$. The maximum number of iterations is K, the loss function is L, and the regularization coefficient is λ, γ.

The output is the strong learner f(x).

For the number of iteration rounds k = 1,2,..., k, there are:

(1) The first-order derivative g_{ki} of the loss function L based on $f_{k-1}(x_i)$ and the second-order derivative h_{ki} are calculated for the ith sample in the current round. Besides, $i = 1, 2, \cdots, m$. All samples contain first-order derivatives and $G_i = \sum_{i=1}^{m} g_{ki}$ and second-order derivatives and $H_i = \sum_{i=1}^{m} h_{ki}$.

(2) When splitting is attempted on a node, the default fraction equals zero. G and H are the sums of the first and second-order derivatives of the node to be split. For the feature sequence number, q=1,2,..., Q.

(a)$G_L = 0, H_L = 0$;

Table 3.1: Dataset distribution

Data set	U2R	R2L	DOS	Detect attacks	Normal attack
Training set	48	1263	38729	4519	98371
Percentage of training sets	0.02%	0.31%	81.32%	0.91%	21.69%
Test set	265	17651	235916	4476	62098
Percentage of the test set	0.08%	6.21%	71.79%	2.65%	22.95%

(Note: DOS: Disk Operating System; R2L: Remote-to-Login; U2R: User-To-Root)

(b.1) The k features are arranged sequentially, and the ith sample is called in order. After the output samples are placed into the left subtree, the first and second-order derivatives of the left and right subtrees are summed, as shown in the following equation.

$$\begin{aligned} G_L &= G + g_{ki}\# \\ G_R &= G + G_L \\ H_L &= H + h_{ki} \\ H_R &= H + H_L\# \end{aligned} \tag{3.14}$$

(b.2) Try to update the maximum score, as shown in Equation 3.15.

$$score = max(score, \frac{1}{2}\frac{G_L^2}{H_L + \lambda} + \frac{1}{2}\frac{G_R^2}{H_R + \lambda} - \frac{1}{2}\frac{(G_L + G_R)^2}{H_L + H_R + \lambda} - \gamma) \tag{3.15}$$

(3) The feature and eigenvalue split subtrees are divided according to the best results.
(4) If the maximum score is zero, the DT creation is finished. All the leaf regions w_{kj} are calculated, and the weak learner $h_k(x)$ can be obtained. The strong learner $f_k(x)$ is adjusted. Then, the next weak learner is calculated iteratively. If it is not zero, then step two is continued to try to branch the DT again.

3.2. Data sources and features. Marker datasets are necessary to train and evaluate anomaly-based network intrusion detection systems. Knowledge of the underlying packet- and stream-based network data is required. The Data Mining and Knowledge Discovery 1999 (KDD 99) sample set contains five million data, but the dataset provides 10% of the training and testing subsets. Here, 10% of the training set in the KDD dataset is used as an experimental study [15]. Its sample category distribution table is shown in Table 3.1.

In the experiments, the categories of each type of attack are correspondingly labeled for the convenience of calculation. The attack categories in the network dataset can be divided into five major categories, subdivided into 42 attack types. There are 21 types in the training set. The new 18 types of attacks that do not appear are displayed in the test set. This allows for testing the model's adaptability to the experiment's new environment. Whether the designed model can accurately identify the type of attack when a new attack appears in the outside world is an important indicator to evaluate the system's capability.

3.3. Recursive Feature Elimination (RFE) cross-validation. The main idea of RFE is to build models repeatedly. After each round of model construction, the first n features with the least correlation are eliminated, and the subsequent features are re-screened to obtain the feature importance ranking. After traversing all the features, the optimal feature set is selected. This process is the process of eliminating the features in turn [16]. Figure 3.1 shows the schematic of ten cross-validations.

In the experiment, ten cross-validations will be used to ensure the stability of the experimental effect. The dataset is divided into ten copies, with only one as the test set and the remaining nine for training. Figure 3.2 demonstrates a schematic diagram of the One vs One (OVO) decomposition.

For unidentified samples, the OVO method trains the binary classifier with a classification algorithm to distinguish between paired classes. Reintegration confidence is the probability that the classifier will classify an

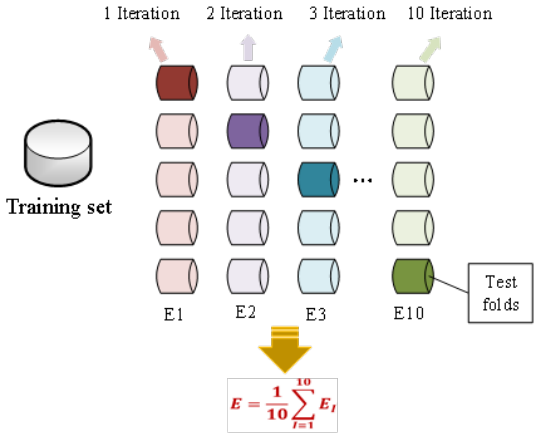

Fig. 3.1: Schematic of ten cross-validations

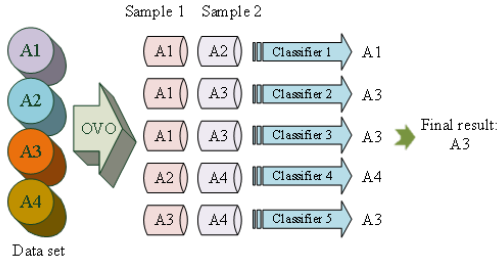

Fig. 3.2: Schematic diagram of the OVO decomposition

unknown sample that should be classified as j into i. Performing the above operation for each paired class will result in a complete score matrix [17]. Figure 3.3 presents the information network intrusion detection model of the OVO algorithm.

A K-means-based hybrid imbalance processing method is also used to deal with unbalanced data to improve the recall rate of network intrusion detection for minority attacks. The entire operation flow chart is shown in Figure 3.4.

The classification algorithm used in the model is the OVO intrusion detection model based on the REF Cross Validation-PCA proposed in the previous section. It has an attack imbalance characteristic for the selected dataset. Firstly, the data is preprocessed, the special symbols are numeric, and some data values are considered large for standardization. Then, the K-means-based hybrid imbalance processing method is used to sample the attack categories to obtain a balanced data set. The model proposed in the previous section is used to train this data set. Finally, the detection results of network intrusion attacks are obtained, which are compared with the unprocessed unbalanced data operation model [18].

3.4. Information analysis and feature parameter extraction technology. Given the current problems of facility condition monitoring and the lack of effective data communication between the facility condition monitoring and the facility management system itself and between them, the embedded condition monitoring and diagnosis system for facility management under the network architecture developed during the specific implementation of the project is shown in Figure 3.5.

The whole system is based on Client/Server (C/S) and Browser/Server hybrid architecture. According to the specific needs of the enterprise, the system provides a variety of commonly used network databases for

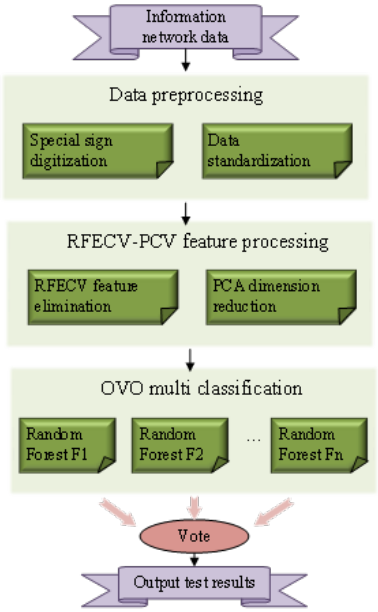

Fig. 3.3: RFECV-PCA-OVO algorithm-based intrusion detection model for information networks

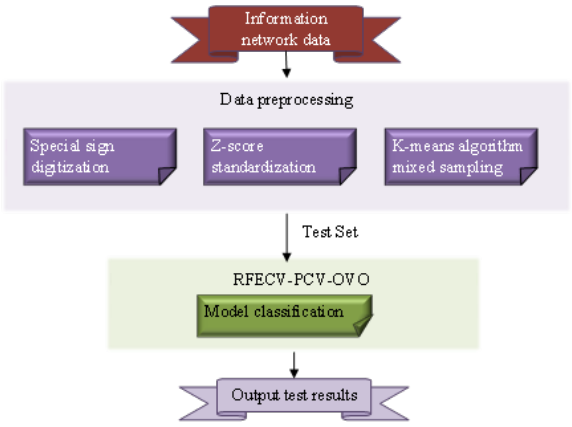

Fig. 3.4: Flow diagram of intrusion detection model based on K-means hybrid method

selection. The entire system framework can be divided into three networks as shown in the figure above. Figure 6 shows the workflow diagram of feature parameter extraction.

In the facility condition monitoring and diagnosis network in C/S mode, the facility monitor uses the embedded data acquisition analyzer to download the facility inspection path from the remote server, extracts the facility status data according to the downloaded path information, and saves it to the analyzer. At this time, the monitor can not only use the signal analysis method provided in the analyzer for on-site data analysis but also upload the collected data to the corresponding measurement point directory specified on the server [19]. The facility monitoring personnel or experts in the monitoring center can obtain the uploaded facility status data from the remote server as the client user of the system after permission verification. The method of signal analysis and diagnosis in the client intrusion software provided by the system on the computer is run. The health status of the facility is extracted and analyzed. If necessary, corresponding maintenance decisions

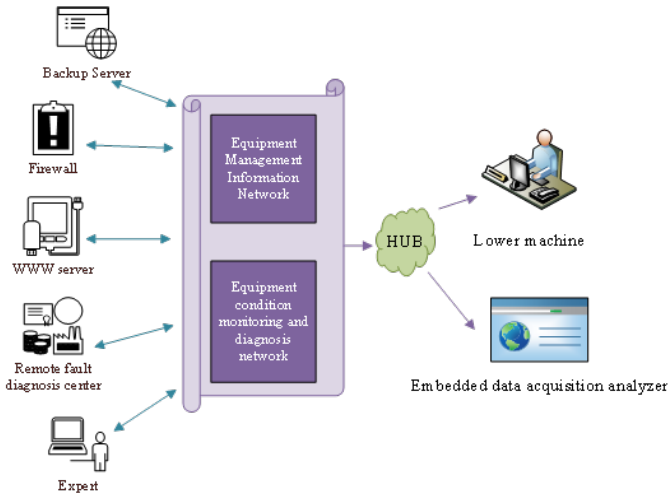

Fig. 3.5: Feature parameter extraction system framework

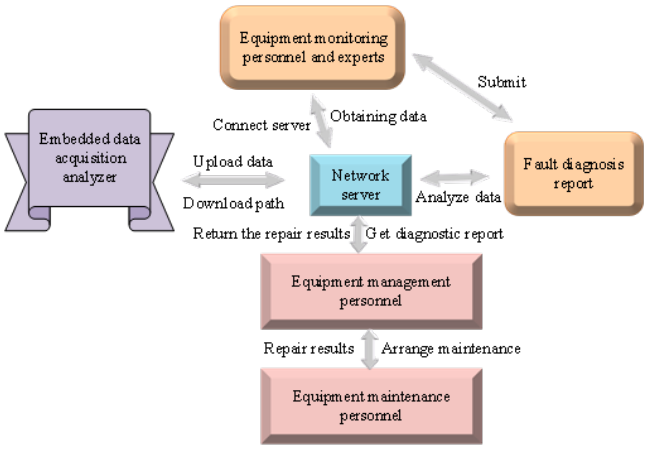

Fig. 3.6: System workflow

can be made, and facility maintenance plan reports can be submitted to the network database.

4. Test of construction parameter optimization model.

4.1. Comparison of network intrusion simulation results. The final result is parameter-dependent for the accuracy of various algorithms. Tuning is required to have a better model. The main parameters are the number of DT N_estimators, the maximum depth Max_depth, the minimum number of samples of leaf nodes Min_samples_leaf, and the minimum number of samples required for internal node subdivision Min_samples_split. The optimal values of the simulation parameters of the four algorithms are revealed in Figure 4.1.

The simulation test here is carried out in the Windows 10 environment, using the python programming language under Jupyter in Anaconda as a simulation experiment platform. The simulation environment for hardware conditions is Inter(R)Core(TM) i5-8500 CPU 3.00GHz, memory size 8.00GB, 64-bit operating system. After the parameter tuning of various algorithms, the running effects of various algorithms are compared. The advantages and disadvantages of each are analyzed, and the algorithms with a better effect on the network

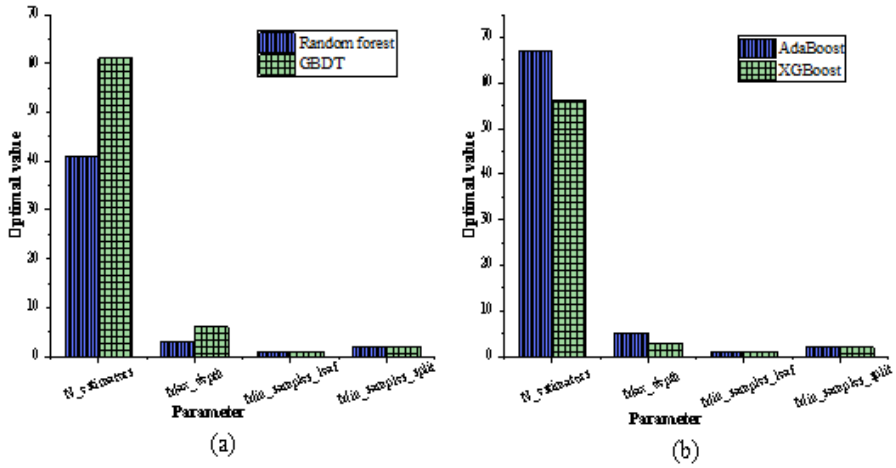

Fig. 4.1: Optimal number of main parameters of four types of algorithms

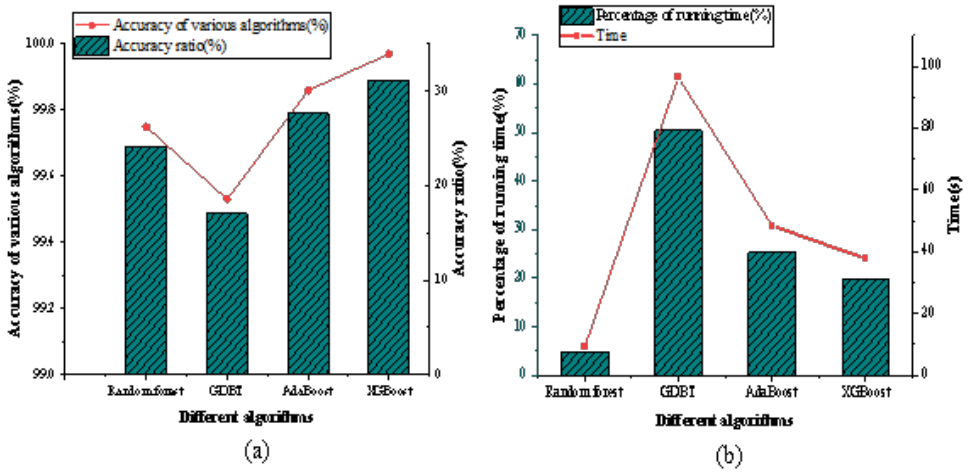

Fig. 4.2: Results from operation (a) are the accuracy of each type of algorithm; (b) is the algorithm running time

security dataset are screened out. After running, the result is shown in Figure 4.2.

Ten operation iterations are performed on the algorithm to prevent the chance of one operation. In Figure 4.3, the abscissa is the number of operations. It can be seen that although the accuracy results of each type of algorithm are a little biased, the detection rate is a very high value. In addition to accuracy, the efficiency of algorithm operation is also very important. As long as it is maintained in real-time, it can effectively block intrusion attacks and ensure the reliable operation of the system.

After one run and ten iterations, it is found that the RF has the least operation time, 9.19s. Several of the remaining algorithms are more than four times this time. Combined with the accuracy, it can be concluded that the RF algorithm has a better effect on the detection effect of network security datasets than other types of algorithms.

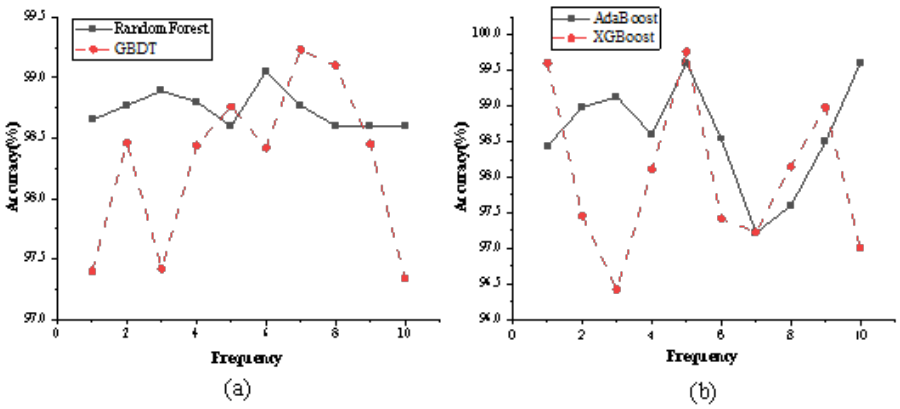

Fig. 4.3: Ten iterations of the operation (a) are RF and GDBT results; (b) are AdaBoost and XGBoost results

4.2. Comparison of network attack detection. After analyzing the entire dataset, it is found that DOS attacks account for a large proportion of the dataset, while U2R attacks account for a very small proportion. This can lead to an imbalance in the data set so that the data detection results are heavily biased towards the data type that accounts for many data types. Therefore, the final result is more one-sided. Therefore, when observing the application of each algorithm model on the dataset, it is necessary to view the accuracy and recall of various attack types to analyze the operation effect more comprehensively. A summary of the program output is shown in Figure 4.4.

The number of DOS attacks of the first category is relatively large in the dataset, about 80%. The various algorithms are not much different and work very well, staying at the same level. However, the GBDT algorithm has some shortcomings in terms of accuracy. The number of positive cases judged to be true accounts for a small proportion of all examples. The second NORMAL category is a more normal number of normal data in the data set, about 20%, similar to the DOS category of attacks. The RF, AdaBoost, and XGBoost algorithms work well, while the GBDT algorithm lacks recall. It indicates that the ratio of positive cases judged by the classifier to be true to the total positive cases is low, and the effect is not very good. The number of PROBE-type attacks in the third category is relatively small in the dataset, about 0.83%. The effect of each type of model is a little bit lower than the first two types of attacks, and the overall detection effect is quite good. The number of R2L attacks in the fourth category accounts for very little of the data set, holding only 0.23%. The algorithms in each category dropped to less than 80%. The number of U2R attacks in the fifth category is very small in the dataset, occupying only 0.01%. Models of all types are the least effective in detecting such attacks.

4.3. Network intrusion faults based on feature extraction analysis. According to a large number of experimental analyses and comparisons in previous studies, it is found that the first seven components after the decomposition of the original signal contain most of the information of the signal. Therefore, only the Renyi entropy of the first seven decomposition components is found to reflect the complexity of data feature extraction under different network intrusion states. The test results are given in Figure 4.5.

It can be seen that through this simple Renyi entropy measurement method, it is already possible to separate the three types of states. It can be seen from the analysis that under the normal working conditions of the network, the energy distribution of the intrusion signal in each frequency band is relatively uniform, the uncertainty and complexity of the energy distribution are large, and the Entropy of Renyi is also greater. For outer ring faults, the data will be more concentrated in the natural frequency band, with less relative uncertainty and less complexity. As a result, Renyi entropy is also smaller. The network shock caused by cage facility failure is less severe than the outer ring due to the small natural excitation frequency. The energy distribution is relatively divergent, and the uncertainty is relatively increased. Therefore, its Renyi entropy also increases accordingly. However, on the whole, the size of the Renyi entropy in the event of a facility failure

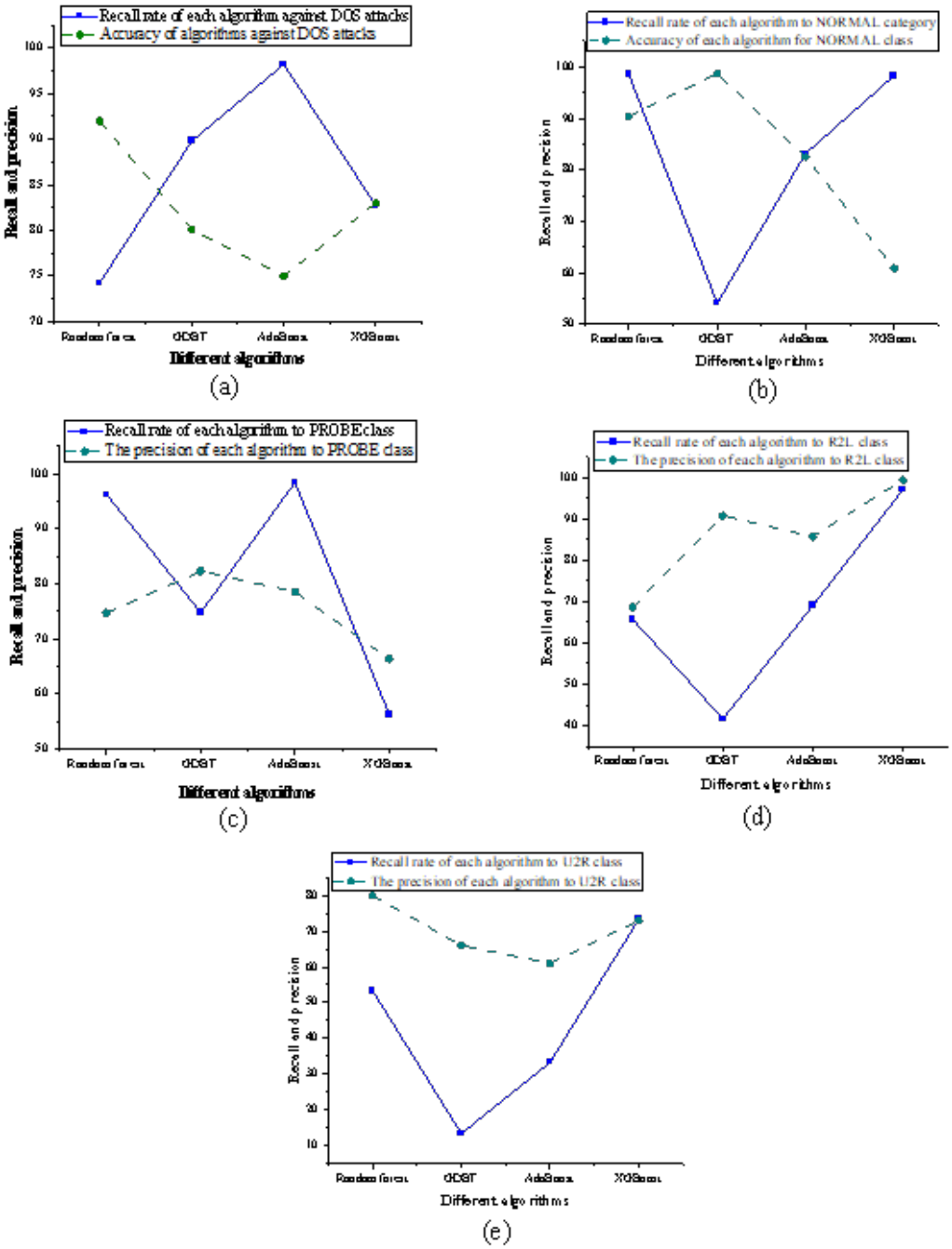

Fig. 4.4: Detection results of various algorithmic attacks

should be between normal and outer ring failure. The Renyi entropy in different states is obtained by adding the Renyi entropy of each energy distribution. Although the above three states can be distinguished, this distinction is not very obvious. The difference in Renyi entropy between different states is not very large.

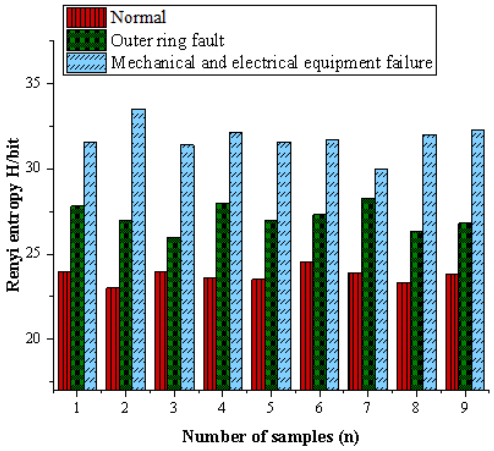

Fig. 4.5: Feature extraction detection results

5. Conclusion. Traditional EF network intrusion detection technology and means have become ineffective with the increase in data volume. Various attack techniques are constantly optimized, and the attack methods are changeable. It is easy to have problems such as false positive rates and false negative rates during analysis. ML methods in big data can be better applied to network security intrusion detection. It can make up for the detection deficiency of traditional information networks and extract data characteristics for facility failure problems to analyze fault problems. This paper mainly focuses on the problems of insufficient recall and accuracy in detecting a small number of attack data in the network data of EF. The following conclusions are drawn: (1) The operation time of the RF is 9.19s. Among the four algorithms, the running time is the least to reduce the data dimension and operation time. Furthermore, the accuracy rate of a few attack types has been improved, which is more in line with the detection requirements of network datasets. (2) Among the five types of network attacks, the number of extractions of DOS attacks is the best, accounting for a relatively large proportion of the dataset, about 80%. The various algorithms are not much different and work well. (3) Renyi entropy can vary depending on the complexity of network intrusions. The embedded operation of the protection mechanism effectively resists different degrees of external intrusion, realizes the separate storage of the operating system and key data, and ensures the stability of the operation of the EF. The disadvantage is that real EF network data cannot be obtained due to the limitations of the experimental environment of the facility. The selected data set is the KDD 99 data set. Although good classification effects have been achieved, it has not been verified whether other EF system data sets can also achieve these effects. Subsequent work will attempt to obtain real data so that the model can be applied to actual information network intrusion detection.

6. Acknowledgement. This study was supported by Fundamental Research Funds for Provincial Universities in Hebei (Grant No. JYT2022019) and Cultivation Special Project of Scientific and Technological Innovation Ability of College Students of Hebei Education Department (Grant No. 22E50159D).

REFERENCES

[1] Garcia-Arroyo, J., & Osca, A. (2021) Big data contributions to human resource management: a systematic review[J]. The International Journal of Human Resource Management, 32(20), 4337-4362.

[2] Abbas, S. G., Vaccari, I., Hussain, F., Zahid, S., Fayyaz, U. U., Shah, G. A., ... & Cambiaso, E. (2021) Identifying and mitigating phishing attack threats in IoT use cases using a threat modelling approach[J]. Sensors, 21(14), 4816.

[3] Kobayashi, S. (2018) Contextual augmentation: Data augmentation by words with paradigmatic relations[J]. arXiv preprint arXiv:1805.06201.

[4] Thirimanne, S. P., Jayawardana, L., Yasakethu, L., Liyanaarachchi, P., & Hewage, C. (2022) Deep Neural Network Based Real-Time Intrusion Detection System[J]. SN Computer Science, 3(2), 1-12.

[5] Moustafa, N., Turnbull, B., & Choo, K. K. R. (2018) An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things[J]. IEEE Internet of Things Journal, 6(3), 4815-4830.
[6] Shen, F., Zhao, X., Kou, G., & Alsaadi, F. E. (2021) A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique[J]. Applied Soft Computing, 98, 106852.
[7] Zhang, N., Nex, F., Vosselman, G., & Kerle, N. (2022) Training a Disaster Victim Detection Network for UAV Search and Rescue Using Harmonious Composite Images[J]. Remote Sensing, 14(13), 2977.
[8] Jahangir, H., Tayarani, H., Baghali, S., Ahmadian, A., Elkamel, A., Golkar, M. A., & Castilla, M. (2019) A novel electricity price forecasting approach based on dimension reduction strategy and rough artificial neural networks[J]. IEEE Transactions on Industrial Informatics, 16(4), 2369-2381.
[9] Tong, J., Zhang, J., Dong, E., & Du, S. (2021) Severity Classification of Parkinson's Disease Based on Permutation-Variable Importance and Persistent Entropy[J]. Applied Sciences, 11(4), 1834.
[10] Wang, C., Ping, W., Bai, Q., Cui, H., Hensleigh, R., Wang, R., ... & Hu, L. (2020) A general method to synthesize and sinter bulk ceramics in seconds[J]. Science, 368(6490), 521-526.
[11] Zhou, X., Lu, P., Zheng, Z., Tolliver, D., & Keramati, A. (2020) Accident prediction accuracy assessment for highway-rail grade crossings using random forest algorithm compared with decision tree[J]. Reliability Engineering & System Safety, 200, 106931.
[12] Bai, J., Xue, H., Jiang, X., & Zhou, Y. (2022) Recognition of bovine milk somatic cells based on multi-feature extraction and a GBDT-AdaBoost fusion model[J]. Mathematical Biosciences and Engineering, 19(6), 5850-5866.
[13] Zelenkov, Y. (2019) Example-dependent cost-sensitive adaptive boosting[J]. Expert Systems with Applications, 135, 71-82.
[14] Yang, C. T., Chan, Y. W., Liu, J. C., Kristiani, E., & Lai, C. H. (2022) Cyberattacks detection and analysis in a network log system using XGBoost with ELK stack[J]. Soft Computing, 26(11), 5143-5157.
[15] Romero, C., & Ventura, S. (2020) Educational data mining and learning analytics: An updated survey[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(3), e1355.
[16] Wang, C., Pan, Y., Chen, J., Ouyang, Y., Rao, J., & Jiang, Q. (2020) Indicator element selection and geochemical anomaly mapping using recursive feature elimination and random forest methods in the Jingdezhen region of Jiangxi Province, South China[J]. Applied Geochemistry, 122, 104760.
[17] Chen, J., Zhuo, X., Xu, F., Wang, J., Zhang, D., & Zhang, L. (2020) A novel multi-classifier based on a density-dependent quantized binary tree LSSVM and the logistic global whale optimization algorithm[J]. Applied Intelligence, 50(11), 3808-3821.
[18] Geetha, R., Sivasubramanian, S., Kaliappan, M., Vimal, S., & Annamalai, S. (2019) Cervical cancer identification with synthetic minority oversampling technique and PCA analysis using random forest classifier[J]. Journal of medical systems, 43(9), 1-19.
[19] Hamdan, M., Hassan, E., Abdelaziz, A., Elhigazi, A., Mohammed, B., Khan, S., ... & Marsono, M. N. (2021) A comprehensive survey of load balancing techniques in software-defined network[J]. Journal of Network and Computer Applications, 174, 102856.

Edited by: Bradha Madhavan
Special issue on: High-performance Computing Algorithms for Material Sciences
Received: Jan 30, 2024
Accepted: Apr 5, 2024