Journal of Computers Vol. 36 No. 1, February 2025, pp. 15-28
https://doi.org/10.63367/19911599202502360100 2

Research on DDoS Attack Detection Based on GBDT-SVM Model in SDN Architecture

Ruo Zhang and Guiqin Yang*

School of Electronic information and Engineering, Lanzhou Jiaotong University, Lanzhou, Gansu, 730070, China
253852013@qq.com, yangguiqin@mail.lzjtu.cn

Received 19 March 2024; Revised 30 May 2024; Accepted 10 August 2024

Abstract. Distributed Denial of Service (DDoS) attack is one of the significant threats to network security currently. The emerging network architecture Software-Defined Networking (SDN) with its centralized control and programmability makes it susceptible to malicious attacks, leading to network paralysis. In response to this issue, this paper proposes a hybrid machine learning model based on Support Vector Machine (SVM) and Gradient Boosting Decision Tree (GBDT) to detect attack traffic. The combination of GBDT and SVM enables dual-stage classification detection. Initially, GBDT conducts preliminary classification on large-scale data and filters misclassified samples. Subsequently, these filtered samples are inputted into the SVM classifier. Leveraging SVM's robust generalization performance between training and testing data and its advantage in detecting anomalous traffic, further classification of data is achieved to accomplish attack detection. The integration of GBDT-SVM helps reduce misclassification of data samples by SVM that are close to the decision boundary during detection. Experimental results demonstrate that compared to other methods, the GBDT-SVM model achieves higher detection efficiency, with an average detection rate of up to 98.1%, lower false positive rates, thus enhancing detection accuracy and efficiency.

Keywords: SDN, DDoS, SVM, GBDT-SVM

1 Introduction

Nowadays, as society continues to develop in the direction of intelligence, the number of devices accessing the Internet has increased dramatically worldwide, and the distributed control in the traditional network is difficult to manage the huge number of devices and cannot meet the complex business requirements at the same time, which has a greater impact on the speed of the development of innovative business [1]. The proposal of SDN solves the limitations imposed by traditional networks, but the over-reliance on the centralized control of the global network by controllers makes it a major target for network hackers to launch malicious attacks. Large-scale DDoS attacks can cause network congestion, affecting normal data flow and exhausting the resources of victimized devices, resulting in their inability to provide normal network services. According to the DDoS Attacks Status and Trends Report 2023 published by Zayo, a security firm, DDoS attacks increased by 387% in the second quarter of 2023 compared to the first quarter [2]. DDoS is considered as one of the most disastrous attacks that target government and business organizations [3], so solving the threat posed by DDoS attacks on SDN networks has become a key direction of research nowadays.

To solve this problem, this paper proposes a machine learning fusion model that combines gradient boosting decision tree and support vector machine DDoS attacks are characterized by high traffic volume and long attack time. To address this feature, this paper combines GBDT and SVM for two-layer detection to improve the detection accuracy. In the first set of detections, GBDT is trained to detect and classify large-scale data using a stepwise tree model generated through an iterative approach, which is an iterative decision tree algorithm consisting of multiple decision trees, where the conclusions of all the trees are summed up as the final solution [4]. When used in combination with SVM, GBDT can help reduce the performance problems that may occur when SVM handles large-scale data. GBDT itself can handle nonlinear relationships and high-dimensional features, which allows it to effectively capture the features and patterns of complex data in the first layer of preprocessing,

* Corresponding Author

and GBDT can clean up the dataset by eliminating the data samples that may lead to misclassification, improving the accuracy and efficiency of subsequent accuracy and efficiency of the model. In the second-level detection stage, the data input to the SVM has already been preprocessed by GBDT, which can give full play to the SVM's advantages in dealing with small-scale, nonlinear, and high-dimensional data. The SVM can focus more on optimizing the boundaries and fine-tuning the classification in this case, thus further improving the detection accuracy. The experimental results show that the GBDT-SVM model has a higher detection efficiency than other methods, with an average detection rate of up to 98.1% and a lower false alarm rate, which improves the detection accuracy and efficiency, and is able to detect DDoS attacks in real time.

The main sections of the paper are organized as follows. In Section II, we describe the research and analysis of the work related to DDoS detection. In Section III, we detail the introduction of the model used in this paper. In Section IV, in order to validate the algorithm of this paper, we build the Mininet platform under Linux system, implant the detection model of this paper into the Ryu controller to realize real-time detection, and compare the five performance metrics so as to analyze and evaluate the performance of the detection model. Section V gives the conclusion and outlook.

2 Related Work

In recent years, studies for DDoS attack detection in SDN networks are mainly categorized into two types: statistical learning-based and machine learning-based. Statistical learning-based anomaly detection methods utilize information entropy to reflect the randomness of data traffic, and in machine learning-based anomaly detection methods, supervised learning, unsupervised learning, and neural network methods perform better than statistical learning in terms of time overhead and detection rate. Therefore, in this paper, the more effective machine learning is chosen to accomplish the detection.

So far, numerous scholars have conducted extensive studies on machine learning-based DDoS attack detection. Hadem et al. [5] utilized Packet-in packets sent from OpenFlow switches, from which IP traceability was performed to counter network attacks and machine learning support vector machine SVM was used to detect the collected IP information; Yang et al. [6] proposed a method used to mitigate DDoS attacks in campus networks, opting for a detection framework based on the machine learning SVM model to identify and defend against attacks. SVM has advantages in solving small sample data, but it is difficult to deal with large-scale data; J. Liu et al. [7] proposed a detection method based on the C4.5 decision trees for identifying DDoS attacks in the network, But DDoS attacks usually involve a large amount of network traffic data, which may have a very high dimen-sionality, and C4.5 may face dimensionality catastrophe problems when dealing with high-dimensional data, leading to overfitting or poor modeling;

Due to the relatively limited diversity of machine learning algorithms used in the training modules of the aforementioned detection systems, and the improvement in training accuracy is not significant, hybrid machine learning models have also been widely used in the field of attack detection in recent years. You Fu [8] proposed a DDoS attack detection method based on conditional entropy and decision tree, which uses conditional entropy to determine the current network state, extracts six important features used for traffic detection by analyzing the characteristics of DDoS attacks in SDN, and classifies the network traffic using the C4.5 decision tree algorithm to achieve the detection of DDoS attacks in SDN. The processing of high-dimensional data by the C4.5 decision tree might lead to overfitting or excessive model complexity, especially when there is correlation between the features, the decision tree may not be able to fully utilize the conditional entropy for effective feature selection and splitting.; Bai et al. [9] proposed a lightweight distributed edge computing architecture, OCM, which combines a deep learning methods. Leveraging the advantages of Long Short-Term Memory (LSTM) networks for global information detection, they employed an optimized bidirectional Long Short-Term Memory (Bi-LSTM) based detection method to detect attacks in networks, The LSTM algorithm and lightweight distributed edge computing architecture OCM may face problems such as computational resource limitations and increased communication overhead; Yu Chen [10] proposed a detection algorithm based on the combination of genetic algorithm and gradient boosting tree in detecting DDoS attacks, which is divided into two phases, in the first phase, genetic algorithm and decision tree algorithm are utilized for feature extraction, from which the optimal subset of features is selected; in the second phase, the gradient boosting tree-based algorithm is used for the detection of DDoS attacks on a subset of the selected features. But both genetic algorithms and GBDT have high computational complexity and will require more computational resources and time to train and deploy the model; Li et al. [11] proposed a statistical-based method to extract Rényi entropy features and set dynamic thresholds to judge

suspicious traffic; based on the integrated self-encoder algorithm, a more accurate DDoS attack judgment is performed on suspicious traffic. The two-layer detection model not only improves the detection effect and solves the problem of high false alarm rate, but also effectively shortens the detection time, thus reducing the consumption of computational resources. The sliding window, dynamic threshold interval and output layer RMSE threshold used in the article need to be tuned for their parameters, and the tuning process requires labeled datasets for experimentation and adjustment; S. Wu [12] proposed a DDoS attack detection algorithm based on the federated Tree Coding for DDoS attack detection, which utilizes the GBDT classification model as an encoder for the feature data. The model combining GBDT and MLP needs to be trained locally at each participant, and then the updated model parameters are aggregated, which may affect the generalization ability and detection accuracy of the final model if the participant's data is unevenly distributed or the amount of data is insufficient.

3 GBDT-SVM Fusion Detection Model

3.1 Gradient Boosting Decision Tree

Ensemble learning is a kind of learning method to collaborate multiple "individual learners" to accomplish the task. Its principle is to integrate and combine multiple weak learners to form a strong learner, thereby mitigating the issues such as poor generalization ability and overfitting that may arise in single learners. Gradient Boosting Decision Tree (GBDT) is a boosting-based ensemble learning algorithm used for classification and regression problems, first proposed by Friedman in 1999 [13]. Its core idea involves generating multiple weak classifiers through iterative rounds, where after each iteration, the negative gradient of the loss function is after each round as an approximation of the residuals [14]. It generates predictive models in the form of an ensemble of base decision trees, especially Classification And Regression Tree (CART). The core idea of this model is to build weak classifiers step-by-step, where the training objective of each classifier is to reduce the residuals generated by the previous classifier in the prediction, iterating in a serial fashion with gradients in the direction of residual reduction. At the end of the entire training process, the model weights and combines the outputs of all the weak classifiers in order to form the final model classifier. Compared with traditional classifiers, it can effectively handle nonlinear data and provide better prediction performance when dealing with various types of data and complex problems, as illustrated in Fig. 1.

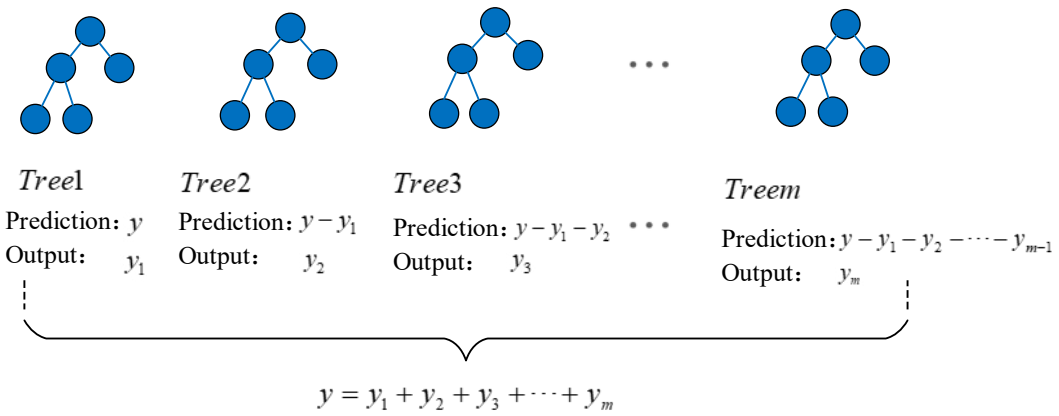


Fig. 1. GBDT schematic

The algorithmic process of GBDT is as follows:
First initialize the weak classifier:

$$f_0(x)=\arg\min\sum_{i=1}^{N}L(y_i,\rho)\ . \tag{1}$$

y_i is the actual value of the sample; ρ is a constant.

Loss function:

$$L\left(y, f\left(x\right)\right) = \left(y - f\left(x\right)\right)^2 \ . \tag{2}$$

Determine the ρ constant so that the initial prediction loss is minimized. Then the total loss for all N samples is:

$$L_{all} = \sum_{i=1}^{N} L\left(y_i, f_m\left(x_i\right)\right) \ . \tag{3}$$

In Eq. (3), y_i and $f_m(x_i)$ are the actual value of the sample and the predicted value of the mth model, respectively.

The purpose of iteration is to minimize the loss value and find the direction where the gradient falls the fastest, the negative gradient of the iterative calculation is calculated as follows:

$$-g\left(x_i\right) = \frac{\partial L\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)} \ . \tag{4}$$

Construct a fit function as $h(x_i; \alpha)$, to fit the negative gradient $-g(x_i)$, α for the residual coefficients.

$$\alpha_m = \arg\min \sum_{i=1}^{N} \left(-g\left(x_i\right) - \beta h\left(x_i; \alpha\right)\right)^2 \ . \tag{5}$$

In Eq. (5), α_m is the residual parameter; $g(x_i)$ is the gradient; β is the coefficient; and $h(x_i; \alpha)$ is the negative gradient fitting function. Next, the optimal value of β_m is calculated:

$$\beta_m = \arg\min \sum_{i=1}^{N} L\left(y_i, f_{m-1}\left(x_i\right) + \beta h\left(x_i; \alpha_m\right)\right) \ . \tag{6}$$

In Eq. (6), β_m is the weighting coefficient, and $f_{m-1}(x_i)$ is the fitting function for the m-1st iteration; ultimately, the results of the calculations are merged into the model to update the prediction function for the next round:

$$f_m\left(x\right) = f_{m-1}\left(x\right) + \beta_m h_m\left(x; \alpha_m\right) \ . \tag{7}$$

$f_m(x)$ is the fitting function for the current iteration, and $h_m(x; \alpha_m)$ is the negative gradient fitting function for the m-1st iteration.

Iteration continues until the prediction residuals of the final round reach zero or become very small, at which point the iteration is terminated. Then, the predicted results from all rounds are summed to obtain the final prediction result. Using the integrated learning approach for DDoS traffic datasets with temporal characteristics, the bias of the model can be consistently reduced to achieve better fitting results. Compared to other machine learning methods, this approach can construct the tree structure faster, thus shortening the training and detection time of the algorithm and enhancing its applicability.

3.2 Support Vector Machine

According to the principle of structural risk minimization in statistical learning theory, Cortes et al. proposed a new supervised machine learning method called SVM in 1995 [15]. The principle is to construct an objective function to distinguish the patterns of different categories as much as possible, which has strong adaptive ability to fresh samples, and can solve the classification problems of high dimensionality and nonlinearity better. As a binary classification algorithm, SVM exhibits good generalization and processing capabilities for imbalanced

data, with high classification accuracy. SVM is used to classify the data sample set and seek an optimal hyperplane, the data traffic is divided into two types: normal traffic and abnormal traffic, and the optimal classification performance is achieved by seeking the maximum interval hyperplane that can distinguish the data perfectly, and the hyperplane is illustrated in Fig. 2.

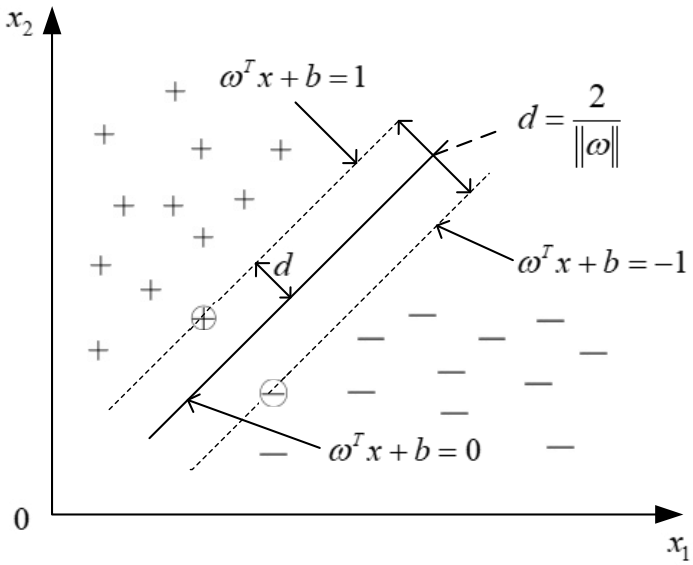


Fig. 2. SVM for classifying hyperplanes

In the sample space, dividing the hyperplane can be described by the linear equation Eq. (8):

$$\omega^T x + b = 0 \ . \tag{8}$$

In the above equation, $\omega = (\omega_1; \omega_2; ...; \omega_d)$ is the normal vector, which determines the direction of the hyperplane, and b is the displacement term, which determines the distance between the hyperplane and the origin. Obviously, the dividing hyperplane can be determined by the normal vector ω and the displacement b, which will be denoted as (ω, b) in the following. The distance from any flow data x in the sample space to the hyperplane (ω, b) can be written as:

$$r = \frac{\left|\omega^T x + b\right|}{\|\omega\|} \ . \tag{9}$$

For ease of calculation, it is common to set $r = \frac{1}{\|\omega\|}$.

Suppose that the hyperplane (ω, b) classifies the training samples correctly, That is, for $(x_i, y_i) \in D$, if $y_i = +1$, then there is $\omega^T x_i + b > 0$; if $y_i = -1$, then there is $\omega^T x_i + b < 0$. Definition of formula (10):

$$\begin{cases} \omega^T x_i + b \geq +1, \ y_i = +1; \\ \omega^T x_i + b \leq -1, \ y_i = -1. \end{cases} \ . \tag{10}$$

The nearest training sample points to the hyperplane that make the equal sign of the above equation hold are called “support vectors”, and the sum of the distances of the two dissimilar support vectors to the hyperplane is represented as:

$$d = \frac{2}{\|\omega\|} \ . \tag{11}$$

Equation (11) is called “margin”.

In order to maximize the distance between normal and abnormal flows is to find the dividing hyperplane with the maximum interval, i.e., to find ω and b that satisfy the above equation such that the value of d is maximized. Thus the optimization problem of SVM is converted into in order to obtain the maximum value of the distance between normal and abnormal flows, i.e., to solve for the minimum value of $||\omega||$ such that $2r$ is maximum. Therefore the optimization problem of SVM can be converted into Eq. (12).

$$\begin{aligned} &\max_{\omega,b} \frac{2}{\|\omega\|} \\ &s.t. y_i\left(\omega^T x_i + b\right) \geq 1,\ i = 1, 2, \cdots, m \end{aligned} \ . \tag{12}$$

The Lagrange optimization method is to dualize the optimal hyperplane problem by introducing the Lagrange multiplier α_i. If $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $\alpha_i \geq 0$ (i =1, ..., n) are satisfied, the above constrained problem can be defined as Eq. (13)

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(x_i \cdot x_j\right) \ . \tag{13}$$

The final optimal SVM classifier is obtained as in Eq. (14)

$$f(x) = \operatorname{sgn}\left\{\sum_{i=1} \alpha_i y_i \left(x_i \cdot x\right) + b\right\} \ . \tag{14}$$

The filtered set of n-dimensional traffic features is used to classify the DDoS attack detection using a trained SVM classifier to differentiate the traffic into normal and abnormal categories.

Scikit-learn is an open-source python library for machine learning, which is built on top of NumPy, SciPy, and Matplotlib. It provides implementations of various machine learning algorithms, including classification, regression, and clustering. In this paper, we focus on constructing SVM algorithm models using Scikit-learn. Before the model training, the Support Vector Classifier (SVC) learning model is loaded in the form of a list, and subsequently a large amount of data is loaded for training the SVC learning model.

3.3 GBDT-SVM Model Detection

The training process of the GBDT-SVM fusion model is as follows:

1. Train the GBDT model using the original model training set to generate a series of decision trees to construct a strong classifier.

2. Use the trained GBDT model to predict the original data, instead of outputting the classification probability, the position of the leaf node to which the predicted value of each tree belongs is output. This position information is used as new feature values to trip the new data.

3. Encode the new data, i.e., the position of the node to which the sample output belongs is labeled as 1, to obtain the position labeling vector ω_i for each sample, and the outputs of all the samples form a sparse matrix labeled with the position of the leaf node of the output of each decision tree.

4. Use ω_i as the new training data for training the SVM model.

The GBDT-SVM algorithm combines the advantages of gradient boosting tree and SVM, which has good classification performance and saves time overhead; the decision tree constructed only extracts its feature information gain partially unaffected by the overfitting phenomenon; and the classification process does not increase time complexity due to filtering the redundant features.

The training process is illustrated in Fig. 3.

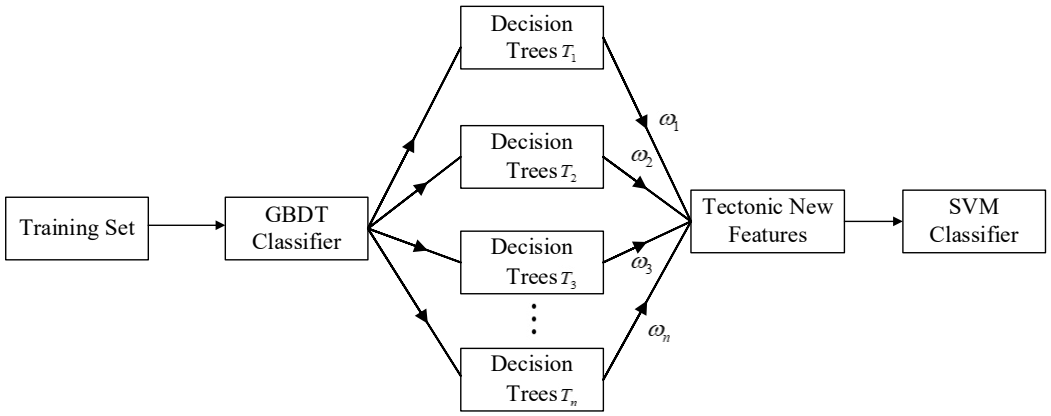


Fig. 3. Diagram of GBDT-SVM model training process

The attack detection framework consists of three parts: the traffic collection module, feature extraction module, and attack detection module, as shown in Fig. 4:

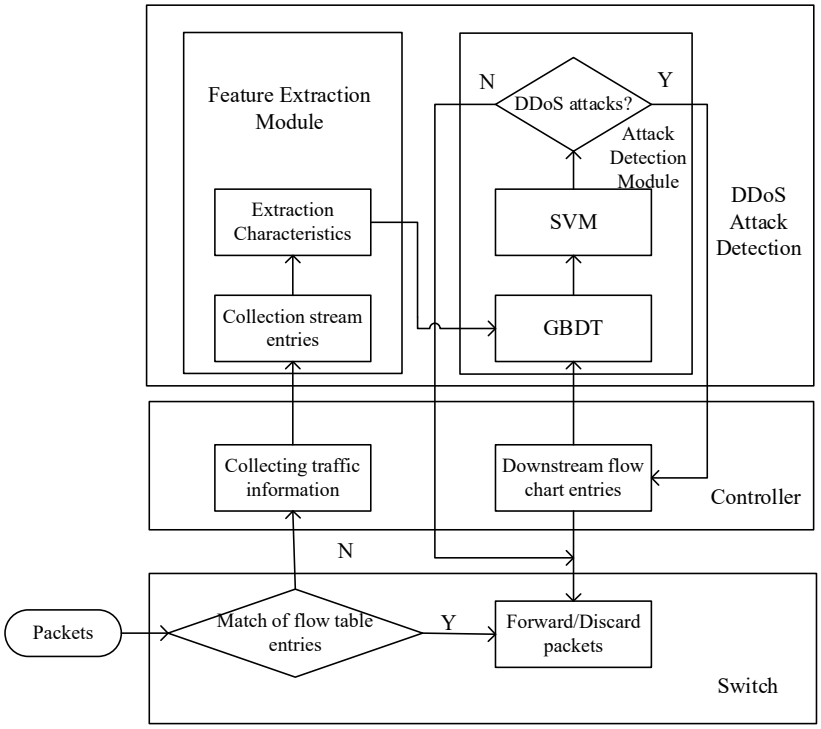


Fig. 4. A DDoS attack detection framework based on GBDT-SVM modeling

1. The traffic collection module is used to collect network traffic data, including network packets, log files, or other forms of data can be parsed through network monitoring, packet capturing on network devices, or log parsing.

2. The feature extraction module performs data preprocessing and extracts five features, such as average packet count per flow, flow packet average bit count, port growth rate, flow growth rate, and source IP growth rate, from the collected experimental traffic data based on the network traffic characteristics under DDoS attacks to

ensure that the system can effectively identify potential attacks behaviors in real-time traffic.

3. The attack detection module is the core part of the whole framework, which is responsible for analyzing and detecting the extracted feature data to identify potential attacks.

4 Experiments and Analysis of Results

4.1 Experimental Environment

This experiment is conducted in an Ubuntu16.04 environment, using the Mininet simulation platform to build a simple network topology for generating network traffic, to build a simulation of the real network environment, the network topology consists of a Ryu controller, three Open vSwitch switches, and five hosts configured under each switch, and the southbound protocol is chosen to be OpenFlow1.3. The topology is shown in Fig. 5.

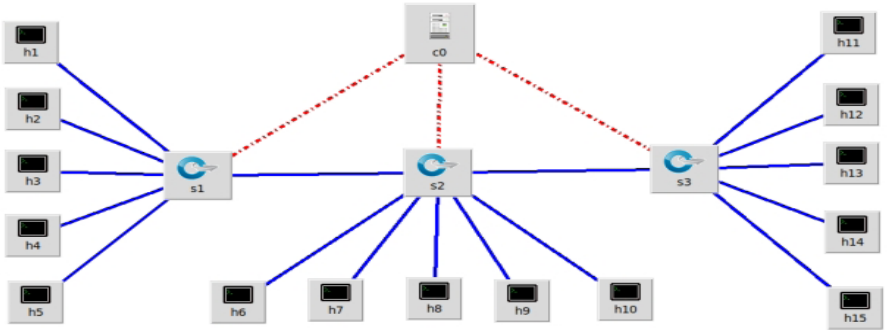


Fig. 5. Experimental topology diagram

The GBDT-SVM model detects DDoS attacks using real network traffic as the experimental data set. After data collection and feature extraction, marking bits are set according to the type of data traffic, where normal traffic is marked as “0” and attack traffic is marked as “1”, as shown in Fig. 6. In order to simulate the attack data, the trafgen toolkit of Netsniff-ng is used in this paper for the simulation of Syn-Flood attack. This type of attack traffic shows a significant reduction in the average number of packets in the stream and the average number of bits in the stream packet compared to normal traffic, and at the same time, the port growth rate, stream growth rate and source IP growth rate are increased by at least 20 times, which shows that the Syn-Flood attack is characterized by a smaller number of packets but a larger amount of data traffic. The system mainly extracts five features in the feature extraction module, namely, the average number of packets, the average bit number of packets, the port growth rate, the flow growth rate and the source IP growth rate.

```
2023-09-01 19:16:56 0.41935483870967744 67.59677419354838 5.4 6.0 1.5 0
2023-09-01 19:17:06 0.5576923076923077 115.84615384615384 5.9 6.2 1.5 0
2023-09-01 19:17:16 0.5901639344262295 110.9672131147541 4.8 5.2 1.4 0
2023-09-01 19:17:26 0.26229508196721313 53.967213114754095 4.7 6.1 1.5 0
2023-09-01 19:17:36 0.39655172413793105 53.741379310344826 5.1 6.1 1.5 0
2023-09-01 19:24:31 0.008188331627430911 1.0235414534288638 97.0 97.4 91.8 1
2023-09-01 19:24:41 0.011156186612576065 1.1196754563894524 97.3 97.7 92.1 1
2023-09-01 19:24:51 0.003080082135523614 0.17248459958932238 98.7 98.6 93.6 1
2023-09-01 19:25:01 0.012084592145015106 1.7683786505538772 96.7 97.4 91.8 1
2023-09-01 19:25:11 0.009259259259259259 1.1512345679012346 99.0 99.3 94.2 1
```

Fig. 6. Real-time collection of experimental data traffic

1. Stream average packet count: usually refers to the average number of packets in a network stream. Calculating the average packet count of a flow can help evaluate the activity level and traffic characteristics of a network. As shown in the following equation, *APF* represents the stream average packet count, *packetNum* is the number of packets contained in each stream table, and *N* represents the total number of different stream table entries.

$$APF \frac{\sum_{i=1}^{N} packetNum}{N} . \tag{15}$$

2. The average number of packets: the number of bits contained in each packet in the flow table, which represents the average size of the data stream transmitted in a certain time. In the following formula, *ABF* represents the average number of packet bits, *pcount* represents the number of packets in each flow table, and *bcount* represents the total number of bits contained in each flow table.

$$ABF = \frac{\sum bcount}{\sum pcount} . \tag{16}$$

3. Port Speed Increase: It is the increase of the data transmission rate of the ports on the network equipment. *GRP* denotes the port growth rate, $port_i$ denotes the number of flow tables containing different ports in the i-th in f_n , f_n denotes the total number of flow tables collected in time *T*, and *T* is the flow table collection period.

$$GRP = \frac{\sum_{i=0}^{f_n} port_i}{T} . \tag{17}$$

4. Stream growth rate: Used to describe the size of data streams in a network or the rate of change of data transmission over time. When an IP spoofing attack in a DDoS attack is initiated, the growth rate of single and paired streams in the stream table is greatly affected. This characteristic indicates the rate at which the attacked party receives traffic per unit of time. *GRF* denotes the stream growth rate, and $flow_i$ is the i-th single-stream number in f_n .

$$GRF = \frac{\sum_{i=0}^{f_n} flow_i}{T} . \tag{18}$$

5. Source IP Growth Rate: The growth rate of the source IP address in the data stream in the network, which can be used to monitor the change of the activity of a specific source IP address in the network, and to understand the change of the traffic of a specific source over time. As shown in the following equation, *GRIP* denotes the source IP growth rate, and IP_flow_i denotes the number of flow tables with the same source IP address as the i-th source IP address in f_n .

$$GRIP = \frac{\sum_{i=0}^{f_n} IP\_flow_i}{T} . \tag{19}$$

4.2 Experimental Evaluation Indicators

To better illustrate the performance of the GBDT-SVM model, the accuracy rate *Acc*, precision rate *P*, recall rate *R*, false alarm rate *F* and detection are used as the length of time used for evaluation metrics. *TP* denotes the number of attack traffic that the model correctly classifies as attack traffic, *FP* denotes the number of attack traffic that the model incorrectly identifies as normal traffic, *TN* denotes the number of normal traffic that the model cor-

rectly identifies as normal traffic, and *FN* denotes the number of normal traffic that the model traffic is incorrectly predicted as the number of attack traffic.

1. *Acc* is an accuracy measure of the overall correctness of the model, indicating the number of correctly predicted data as a proportion of all data.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \ . \tag{20}$$

2. Accuracy rate *P* is the proportion of correctly predicted positive examples to all predicted positive examples. In DDoS refers to the proportion of all data predicted as attack traffic that is detected correctly.

$$P = \frac{TP}{TP + FN} \ . \tag{21}$$

3. The recall rate *R* represents the ratio of the number of samples correctly detected as attack traffic to the total number of samples in the attack traffic, and it measures the ability of the model to recognize positively classified samples.

$$R = \frac{TP}{TP + FP} \ . \tag{22}$$

4. The false alarm rate *F* is expressed as the ratio of the number of samples of normal traffic incorrectly detected as attack traffic to the total number of samples of normal traffic. A lower *FPR* indicates that the model is less likely to incorrectly predict negative samples as positive.

$$F = \frac{FN}{TN + FN} \ . \tag{23}$$

5. Detection time *Time* is used to represent the length of time the model takes to detect all the attacks. t_1 denotes the moment when the detection starts and t_2 denotes the moment when the detection ends.

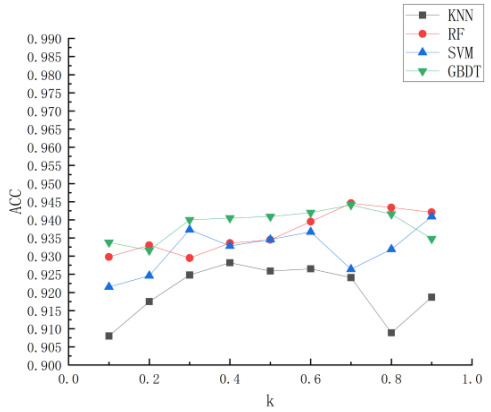
$$Time = t_2 - t_1 \ . \tag{24}$$

4.3 Analysis of Experimental Results

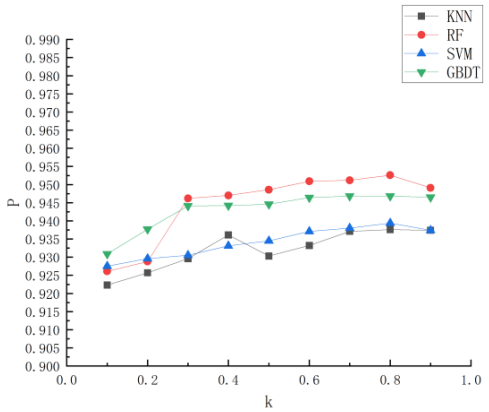
In order to more clearly represent the detection effect of GBDT-SVM model, the experimental steps in this chapter are divided into two steps, firstly, individual machine learning models (KNN, RF, SVM, GBDT) are compared for effect detection. The second step combines the SVM algorithm with KNN, RF and GBDT respectively and analyzes and compares the detection effect of the three hybrid models KNN-SVM, RF-SVM and GBDT-SVM. The comparison of four individual models on four detection metrics is shown in Fig. 7. Define *k* as the proportion of test data D_1 to the total data *D*.

$$k = \frac{D_1}{D} \ . \tag{25}$$

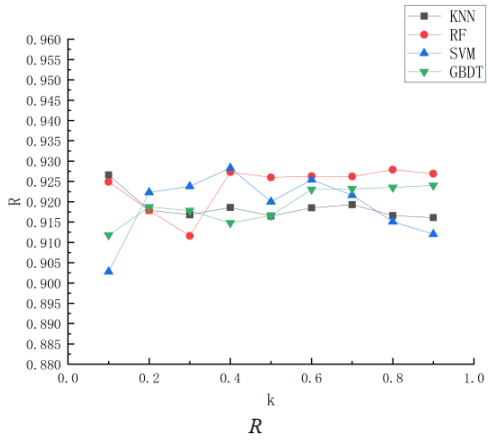
The detection time of KNN is three times higher than the other three models, so line graphs are not used to show the comparison effect. From Table 1 and the above line graphs, it can be seen that the GBDT algorithm has a higher accuracy than the other three algorithms, a lower false alarm rate than the other three algorithms, a recall rate that is on par with RF higher than KNN and SVM, and the time used for detection is slightly higher than SVM. The accuracy rate *Acc* is higher compared to the other three, being 0.0168 higher than KNN, 0.0004 higher than RF, and 0.0052 higher than SVM, the time used for detection is lower than that of KNN, and the false alarm rate is lower than that of SVM. KNN is 0.8717 seconds less than RF and 0.0206 seconds less than KNN, while the False alarm rate *F* is lower compared to the others, 0.015 lower than KNN, 0.011 lower than RF, and 0.0084 lower than SVM, which concludes that the detection efficiency of GBDT is better than other single models.



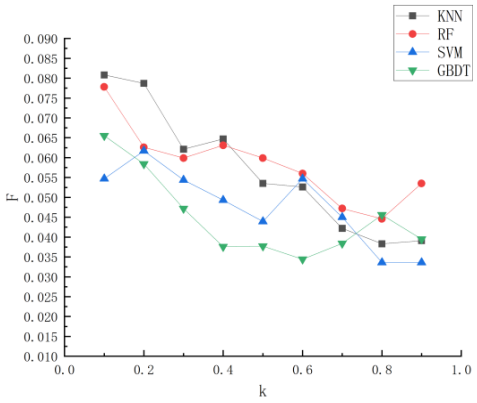
(a) Single model Acc comparison line graph



(b) Single model P comparison line graph



R



F

Fig. 7. The comparison of the above four individual models on four detection metrics

Table 1. Average of individual machine learning models *Acc*, *P*, *R*, *F* and *Time*

Performance indicators	KNN	RF	SVM	GBDT
Average of *Acc*	0.9203	0.9367	0.9319	0.9388
Average of *P*	0.9322	0.9445	0.9341	0.9431
Average of *R*	0.9174	0.9194	0.9190	0.9193
Average of *F*	0.0561	0.0521	0.0495	0.0411
Average of *Time*	1.3009	0.4498	0.4227	0.4292

In order to further validate the advantages of GBDT algorithm for detecting DDoS attacks, this paper fuses the two machine learning algorithms for the detection of hybrid models, because SVM has good generalization and processing ability in dealing with unbalanced data, and due to the characteristics of the DDoS attack traffic, the percentage of the normal traffic and the attack traffic is not balanced, so the SVM algorithm can be very well applied to the detection of DDoS attacks. Therefore, we choose to fuse other algorithms with SVM, and the hybrid models that are analyzed in the paper are KNN-SVM, RF-SVM, and GBDT-SVM. when increasing from 0.1 to 0.9, the line graph of comparison of each index is shown in the Fig. 8.

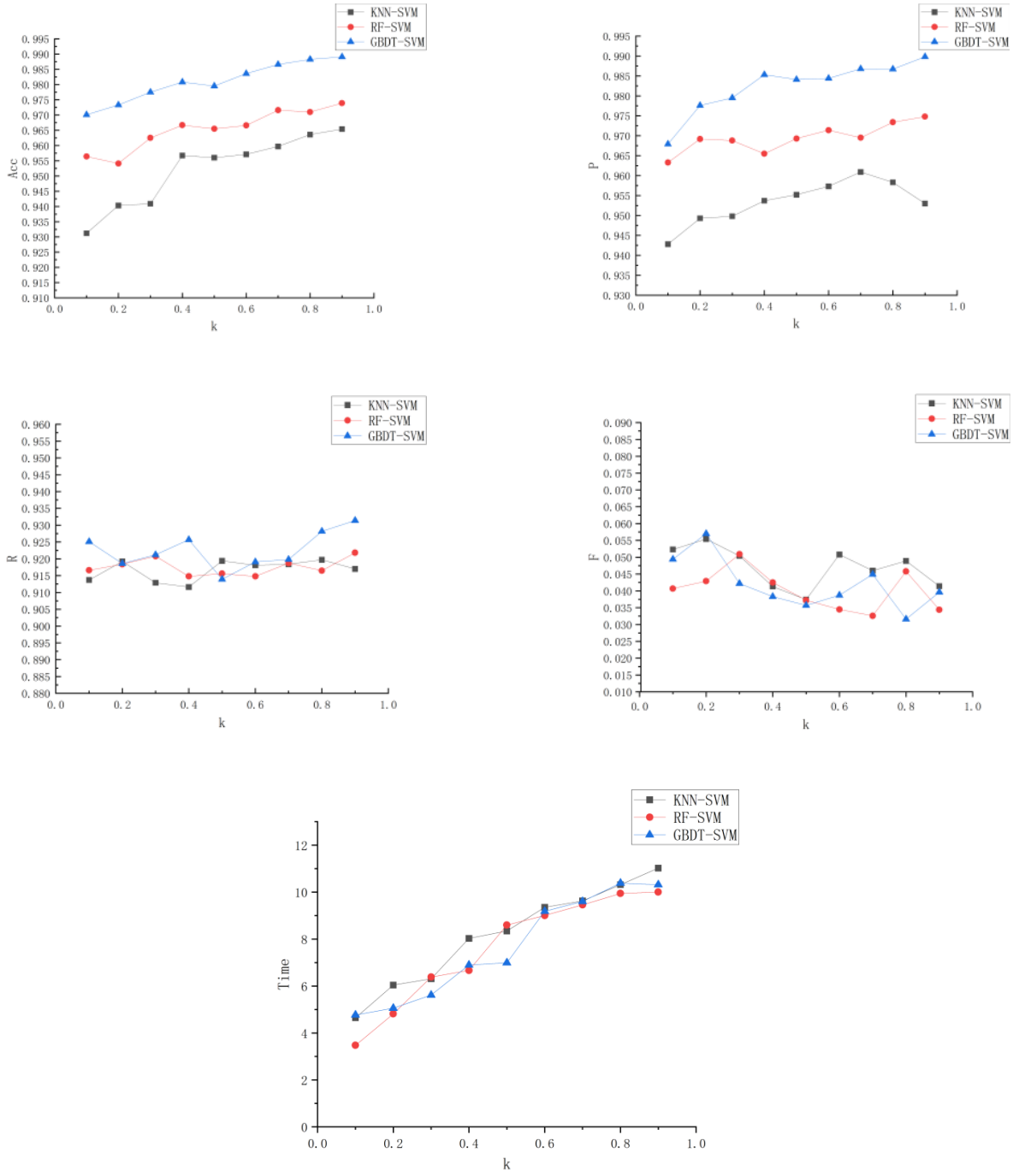


Fig. 8. The comparison of the above tree hybrid models in the five detections metrics

Table 2. Average of individual machine learning models *Acc*, *P*, *R*, *F* and *Time*

Performance indicators	KNN-SVM	RF-SVM	GBDT-SVM
Average of *Acc*	0.9523	0.9654	0.9810
Average of *P*	0.9534	0.9695	0.9825
Average of *R*	0.9167	0.9176	0.9226
Average of *F*	0.0471	0.0402	0.0419
Average of *Time*	8.1864	7.5399	7.6454

The above graphs visualize the comparison of the results of the four hybrid models, as can be seen in Table 2, the GBDT-SVM hybrid model has the best results in terms of *Acc*, *P*, *R*, *F* and *Time* used for detection is slightly higher than RF-SVM. precision GBDT-SVM is 0.0287 higher compared to KNN-SVM and 0.0156 higher compared to RF-SVM. False alarm rate is 0.0052 lower than KNN-SVM but 0.0017 higher than RF-SVM, detection time GBDT-SVM is 0.541s lower than KNN-SVM and 0.1055s higher than RF-SVM, in terms of detection efficiency the GBDT-SVM proposed in this paper is superior to the hybrid model of both KNN-SVM and RF-SVM, and it has better detection effect.

After the comparison of the above experiments it can be clearly seen that the hybrid model of GBDT-SVM has an average detection accuracy of up to 98.1%, and the detection efficiency is better than several other models, so this model is imported into the Ryu controller for real-time traffic detection. In order to observe the real-time detection situation more intuitively, three metrics are set including traffic type, detection correctness, and detection time. The correspondence between the detection situation and the metrics is shown in Table 3. When normal traffic is detected, as shown in Fig. 9. When abnormal attack traffic is detected, it is shown in Fig. 10.

Table 3. Real-time detection and indicators

Real-time situation	Type of data traffic	Detecting the correctness
Normal circumstances	Normal	Correct
Attacks on the situation	Attack	Correct

```
2024-02-29 18:46:34 0.7804878048780488 178.03252032520325 16.8 26.8 3.0 0 normal correct 0.00039649009704589844
2024-02-29 18:46:39 0.6434782608695652 106.7304347826087 15.6 24.6 3.0 0 normal correct 0.0005538463592529297
2024-02-29 18:46:44 0.5166666666666667 103.20833333333333 16.8 23.0 3.0 0 normal correct 0.0003478527069091797
2024-02-29 18:46:49 0.4649122807017544 112.90350877192982 17.0 24.0 3.0 0 normal correct 0.0003991127014160156
2024-02-29 18:46:54 0.3949579831932773 116.98319327731092 15.8 22.8 3.0 0 normal correct 0.00038909912109375
```

Fig. 9. Real-time detection of normal traffic

```
2024-02-29 19:17:59 0.03763987792472025 7.3570701932858595 188.8 196.6 175.0 1 attack correct 0.00038170814514160156
2024-02-29 19:18:04 0.05263157894736842 9.55312810327706 189.0 196.6 174.8 1 attack correct 0.0003800392150878906
2024-02-29 19:18:09 0.03670634920634921 7.300595238095238 193.4 201.4 180.0 1 attack correct 0.0002658367156982422
2024-02-29 19:18:14 0.05105105105105105 13.636636636636636 194.0 201.6 180.4 1 attack correct 0.0004136562347412109
2024-02-29 19:18:19 0.0712136409227683 17.311935807422266 192.8 199.8 178.8 1 attack correct 0.000335693359375
2024-02-29 19:18:24 0.04875621890547264 11.103482587064677 192.0 199.4 178.4 1 attack correct 0.0003769397735595703
```

Fig. 10. Real-time detection of DDoS attacks

The above results show that in this paper the model accomplishes the correct detection of DDoS attacks.

5 Conclusions and Outlook

In this paper, based on the problem of DDoS attack detection in SDN networks, a hybrid detection model combining GBDT and SVM is proposed and designed, and the main purpose is to improve the detection accuracy and accomplish efficient and correct real-time detection. In order to improve the accuracy of detection, SVM, which has advantages in detecting abnormal traffic, is chosen, SVM can effectively handle nonlinear and unbalanced data, and integrated learning GBDT is utilized to gradually correct the data samples misclassified by SVM near the decision surface through gradual learning of the data, and combining the prediction of multiple models to reduce the risk of misclassification by a single model. In this paper, a real SDN environment is simulated, and the GBDT-SVM model is put into the controller for real-time detection, and the experimental results show that the model can accomplish the detection of abnormal attack traffic, and the effect is better than the single model and the other two hybrid models. However, there are some shortcomings in the method; both GBDT and SVM require a large amount of computational resources in the training and inference process, especially when dealing with large-scale datasets. Combining the two may increase the computational burden and time cost of the system. The focus of subsequent research is to improve the execution efficiency of the algorithm to mitigate and defend

against DDoS attacks, so as to continue the research on DDoS attacks and achieve the overall detection and defense of the system.

References

[1] P.-F. Zhai, Research on DDoS Attack Detection and Defense Method in SDN, [dissertation] Xi'an: Xidian University, 2020.

[2] Security Protection Company Zayo, Protecting Your Business From Cyber Attacks: The State of DDoS Attacks DDoS Insights From Q1 & Q2. <https://www.zayo.com/resources/protecting-your-business-from-cyber-attacks>, 2023 (accessed 01.07.2023).

[3] R. Swami, M. Dave, V. Ranga, IQR-based approach for DDoS detection and mitigation in SDN, Defence Technology 25(2023) 76-87.

[4] J. Zhang, Q. Liang, R. Jiang, X. Li, A Feature Analysis Based Identifying Scheme Using GBDT for DDoS with Multiple Attack Vectors, Applied Sciences 9(21)(2019) 4633.

[5] P. Hadem, D.-K. Saikia, S. Moulik, An SDN-based Intrusion Detection System using SVM with Selective Logging for IP Traceback, Computer Networks 191(2021) 108015.

[6] L. Yang, H. Zhao, DDoS Attack Identification and Defense Using SDN Based on Machine Learning Method, in: Proc. 2018 15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN), 2018.

[7] J.-J. Liu, J. Wang, M.-L. Wang, Y. Wang, DDoS attack detection based on C4.5 in SDN, Computer Engineering and Applications 55(20)(2019) 84-88+127.

[8] Y. Fu, D.-S. Zou, A DDoS attack detection method based on conditional entropy and decision tree in SDN, Journal of Chongqing University 46(7)(2023) 1-8.

[9] J.-J. Bai, R.-C. Gu, Q.-H. Liu, A DDoS attack detection scheme based on Bi-LSTM in SDN, Computer Engineering and Science 45(2)(2023) 277-285.

[10] Y. Chen, Research on network traffic anomaly detection method based on combination learning, [dissertation] Qinhuangdao: Yanshan University, 2019.

[11] C.-J. Li, S.-P. Yin, H.-T. Chi, J. Yang, H.J. Geng, DDoS Attack Detection Model Based on Statistics and Ensemble Autoencoders in SDN, Computer Science 51(11)(2024) 389-399. https://doi.org/10.11896/jsjkx.230900028

[12] S. Wu, Research on DDoS attack detection method based on federated learning, Southeast University, 2022.

[13] J.-H. Friedman, Greedy function approximation: A gradient boosting machine, The Annals of Statistics 29(5)(2001) 1189-1232.

[14] L. Li, Y. Yu, S. Bai, J. Cheng, X. Chen, Towards effective network intrusion detection: a hybrid model integrating gini index and GBDT with PSO, Journal of Sensors 2018(1)(2018) 1-14.

[15] K.D. Teng, Q. Zhao, H.R. Tan, J.H. Zheng, Y.X. Dong, H.F. Shan, Emotion Classification Using EEG Signals Based on SVM-KNN Algorithm, Computer Systems & Applications 31(2)(2022) 298-304.