Hindawi
Journal of Electrical and Computer Engineering
Volume 2023, Article ID 9853289, 1 page
https://doi.org/10.1155/2023/9853289

*Retraction*

# Retracted: Intrusion Detection-Data Security Protection Scheme Based on Particle Swarm-BP Network Algorithm in Cloud Computing Environment

**Journal of Electrical and Computer Engineering**

Received 19 December 2023; Accepted 19 December 2023; Published 20 December 2023



This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

## References

[1] Z. Wang and X. Chen, "Intrusion Detection-Data Security Protection Scheme Based on Particle Swarm-BP Network Algorithm in Cloud Computing Environment," *Journal of Electrical and Computer Engineering*, vol. 2023, Article ID 1128545, 10 pages, 2023.

Hindawi
Journal of Electrical and Computer Engineering
Volume 2023, Article ID 1128545, 10 pages
https://doi.org/10.1155/2023/1128545



*Research Article*

# Intrusion Detection-Data Security Protection Scheme Based on Particle Swarm-BP Network Algorithm in Cloud Computing Environment

**Zhun Wang and Xue Chen**

*School of Engineering, Guangzhou College of Technology and Business, Guangzhou, Guangdong 510850, China*

Correspondence should be addressed to Zhun Wang; wangzhun@gzgs.edu.cn

Received 10 May 2022; Revised 8 June 2022; Accepted 20 June 2022; Published 24 March 2023

Academic Editor: Xuefeng Shao



Aiming at the problems of low detection rate and high false detection rate of intrusion detection algorithms in the traditional cloud computing environment, an intrusion detection-data security protection scheme based on particle swarm-BP network algorithm in a cloud computing environment is proposed. First, based on the four modules of data collection, data preprocessing, feature selection, and intrusion detection, the overall framework of the intrusion detection model is constructed by designing corresponding functions. Then, by introducing the decision tree algorithm, the overfitting is reduced and the data processing speed of the model is improved, and on this basis, the feature selection is carried out through the "gain rate" optimization method, which reduces the redundant information of the feature vector. Finally, by introducing the Particle Swarm Optimization (PSO) algorithm into the optimization of the initial weights and thresholds of the BP neural network, the BP neural network is improved based on the momentum factor and adaptive learning rate, and the high detection rate and low false detection rate are realized. Through simulation experiments, the proposed intrusion detection method and the other three methods are compared and analyzed under the same conditions. The results show that the detection rate and false detection rate of the method proposed in this paper are the best under five different types of sample data, the highest detection rate reaches 95.72%, and the lowest false detection rate drops to 2.03%. The performance of the proposed algorithm is better than that of the other two comparison algorithms.

## 1. Introduction

With the continuous development of network technology, cloud computing has become a pillar technology of Internet applications in various fields [1, 2]. However, the universality of cloud computing applications and their virtualization, isomerization, dynamic complexity, and other characteristics also lead to the cloud computing platform becoming the target of many hacker attacks, and the complex structure of the cloud computing platform also makes it face many security threats [3]. Compared with the intrusion behavior of ordinary computer and ordinary network environment, the intrusion behavior in the cloud environment has a faster attack speed and stronger destructiveness. Therefore, while individuals and enterprises use the low-cost services of the cloud computing platform to establish unique advantages for their products, they need to bear more risks, which damages the interests of cloud computing platform users to a certain extent [4, 5].

In view of the above new cloud security threats, how to ensure the information security of the cloud environment is an urgent problem to be solved under the current situation. Through the analysis and research on the security of a cloud computing environment, it is found that the establishment of an intrusion detection system in a cloud computing environment can well protect the security of the cloud computing platform [6, 7]. Intrusion detection system is a core technical means of dynamic network security protection at present [8]. It can monitor the computer system and surrounding network environment in real time, collect and

analyze the information of key locations, and on this basis, identify whether there is an illegal intrusion, respond to aggressive intrusion, and effectively protect the security and integrity of the system resources. As an effective supplement to a firewall, an intrusion detection system is of great significance for information security in a cloud environment [9–11]. However, the characteristics of a large amount of data and high concurrent access in the cloud computing environment lead to the decline of the ability of traditional intrusion detection technology to detect the intrusion behavior in the cloud computing environment and the slow response speed. Therefore, the research on Intrusion Detection Technology in a cloud computing environment has far-reaching significance.

The rest of this paper is arranged as follows: the second chapter introduces the related work in this field; the third chapter describes the intrusion detection-data security protection algorithm; in Chapter 4, experiments are designed to verify the performance of the proposed algorithm; the fifth chapter is the conclusion.

## 2. Related Research

An intrusion detection system is a means to protect the security of a computer system. It can judge whether intrusion behavior occurs through analysis and calculation and take effective measures to protect the security and integrity of system resources. However, a cloud computing environment has the characteristics of a large amount of data and high concurrent access. The traditional intrusion detection technology can not detect intrusion behavior in a cloud computing environment quickly and effectively. Therefore, the research on intrusion detection technology in a cloud computing environment is of far-reaching significance. Considering the research method of cloud computing intrusion detection technology based on the BP neural network, reference [12] proposed a new algorithm that used an improved artificial colony algorithm to optimize BP neural network algorithm for intrusion detection. However, this method can not carry out real-time detection and occupies a large memory space. In reference [13], an intrusion detection system based on the Hadoop cloud node was proposed by improving the genetic algorithm and neural network algorithm. However, this method does not effectively reduce the false-positive rate of intrusion detection. For the mobile cloud involving heterogeneous client networks, the existing intrusion detection schemes have high computational complexity or need to update rules frequently, which seriously affects the effectiveness of intrusion detection. Reference [14] proposed an intrusion detection scheme based on machine learning that can customize the complexity to meet the requirements of client networks. However, this method can not avoid the disadvantage of single point deployment, and the utilization rate of server resources is low. Reference [15] developed an effective framework to monitor the attack rate of the whole network and provided various solutions to protect the cloud server from attack. However, this method needs to establish a complete network framework in advance for the existing schemes, which has great limitations. Reference [16] established an intrusion detection algorithm based on an integrated Support Vector Machine with a packet agent by dividing the sample stream into data streams, which are interrelated and can accurately reflect the intrusion behavior, especially the packets of persistent intrusion. However, the detection efficiency of this algorithm is low and the speed is slow. Aiming at the problem that the traditional intrusion detection system in the cloud is vulnerable to attack and can not maintain a balance between sensitivity and accuracy, reference [17] proposed a network intrusion detection system in the cloud computing environment. However, this method is difficult to adapt to a network with a large amount of data. Aiming at the problem of low detection rate of traditional intrusion detection algorithms in the cloud environment, reference [18] proposed a novel intrusion detection system by combining fuzzy C-means clustering algorithm with Support Vector Machine, which improved the accuracy of the detection system in a computing environment to a certain extent. However, this method can not accurately give the specific classification identification and can not effectively control the false detection rate.

Through the research on the existing intrusion detection technology, it is found that the existing intrusion detection technology has made good progress, but there are still some shortcomings, such as the imbalance of data set samples, which is easy to be ignored in machine learning. This paper will focus on the problems of "poor detection effect of detection algorithm on a few attacks caused by unbalanced samples" and "high false detection rate" and propose an intrusion detection-data security protection scheme based on the Particle Swarm Optimization-BP network algorithm in the cloud computing environment. The basic ideas are as follows: ① using decision tree algorithm to reduce the intermediate parameters and overfitting of the convolutional neural network model. ② Particle Swarm Optimization (PSO) is introduced into the optimization of initial weight and threshold of BP neural network. ③ BP neural network is improved by momentum factor and adaptive learning rate. By introducing the PSO algorithm to optimize BP neural network, the intrusion detection rate can be effectively improved. Using momentum factor and adaptive learning rate in the detection model can effectively reduce the false detection rate of intrusion detection. Compared with traditional intrusion detection methods, the contributions of the proposed method lie in the following:

(1) After the decision tree is fully grown, the feature extraction is realized by pruning the algorithm on the tree, and the selected features are screened again in combination with the "gain rate," which improves the detection performance of the algorithm.

(2) PSO algorithm is introduced into BP neural network to optimize its initial weight and threshold, which improves the intrusion detection rate.

(3) Based on the momentum factor and adaptive learning rate, the detection model is improved to reduce the false detection rate to a certain extent.

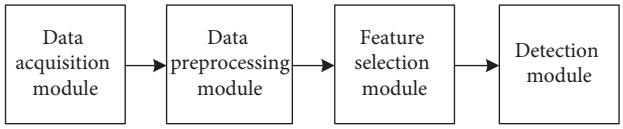


Figure 1: Overall scheme of intrusion detection model.

## 3. The Proposed Intrusion Detection-Data Security Protection Algorithm

*3.1. Overall Scheme Design of Intrusion Detection Model.* First, build the overall scheme of the intrusion detection model, as shown in Figure 1.

The intrusion detection model in Figure 1 consists of four basic modules, which are, respectively, data acquisition module, data preprocessing module, feature selection module, and detection module.

The function of data acquisition module is to design a data acquisition scheme based on flow for different types of attacks. Each collected data includes 32 original data features and 16 synthetic features. For traditional network attacks, the NSL-KDD data set is used for experiments.

In the data preprocessing module, there are different data types and dimensions between the different features of the original input data. In order to facilitate the calculation, the module digitizes the input data and unifies the dimensions.

The function of the feature selection module is to reduce the redundant information of the feature vector, improve the training speed of the intrusion detection model, simplify the neural network model, and improve detection efficiency.

The detection module is the core of the intrusion detection model. Its function is to realize the high-precision classification of different network behaviors.

*3.2. Design of Data Acquisition Module.* In the field of intrusion detection, the imbalance of samples in the data set will have a serious impact on the classification model, but it is often ignored. Sample imbalance refers to the large difference in the number of different types of samples in the data set, in which the large number is the majority of samples and the small number is the minority of samples. In fact, the essence of intrusion detection is to use classifiers to classify data. When there is a problem of unbalanced data samples in the training set, the classification effect of the trained model on a minority of samples will be very poor. In practical applications, the classifier is often very important for the results of a minority of samples. For example, in intrusion detection, if a few attacks are not detected, they will be regarded as normal traffic and implement the corresponding release strategy, and the harm to the system can not be estimated.

Solving the problem of intrusion detection caused by unbalanced data samples is the focus of the intrusion detection algorithm. Generally, there are the following two methods: (1) Improve the algorithm. Solve the problems caused by sample imbalance from the algorithm level, such as introducing sensitive cost function into the algorithm; (2) Solve from the data level. That is, before training the model, adjust the distribution of data samples in the training set. Here, the problem of sample imbalance is solved from the data level in the data acquisition module. Generally, there are two methods to deal with imbalance samples from the data level: random undersampling and random oversampling:

(1) Random undersampling refers to randomly taking a part of the samples from the majority of samples and discarding them. This method is simple and convenient, but there may be useful information in the lost data, which may lead to the loss of useful information and the phenomenon of underfitting in the training process. This leads to the poor effect of the model trained with this data set.

(2) Random oversampling refers to randomly replicating the minority of samples in the data set so as to increase the number of minority samples in the data set. However, this method needs to replicate minority samples repeatedly, which will increase the probability of overfitting of the classifier and lead to a decline in classifier performance.

*3.3. Design of Data Preprocessing Module.* The original data in the cloud computing environment has a huge scale and complex formats. If the format of the original data is not preprocessed, it can not be directly used in the intrusion detection model. The format of data in the NSL-KDD data set does not meet the input requirements of the intrusion detection model, so the original data should be preprocessed first. It mainly includes sparse feature merging, string feature digitization, and numerical normalization.

(1) *Sparse Feature Merging*. During data preprocessing, there will be a phenomenon where multiple values of a feature column correspond to the same label. For example, there are multiple values of feature *T*, in which the label corresponding to the two data values of A and B is *t*. It is regarded as a sparse feature. Different values are meaningless for the training of the model. In order to reduce the computational cost and the probability of misclassification, sparse features are combined. In the NSL-KDD data set used here, the corresponding data samples of the data with values of *S0* and *rsto* in the *service* feature column are both *Neptune*. Therefore, the two features are combined and the other sparse features are operated in the same way. This can reduce the imbalance of the samples in the data set and better analyze the classification results of the classifier.

(2) *String Feature Digitization*. The values of many features in the data are nonnumeric and cannot be directly used in machine learning algorithms. Therefore, these string features need to be converted into numeric first. There are some string features in the data features that cannot be used directly, so the values of these features need to be processed. The string types in the NSL-KDD data set are as follows: protocol type feature column with 3 values, service

type feature column with 70 values, and status flag feature column with 11 values. Here, one hot coding is used to convert these values into numerical type, and the data of the original data set is changed from 41 dimensions to 122 dimensions.

(3) *Numerical Normalization*. In practical application, there may be different dimensions between data features, which cannot be calculated directly. Normalization is needed to eliminate the influence of data units on calculation and map the data to [0,1] interval. Relevant research shows that data normalization can accelerate the speed of gradient descent to find the optimal solution and may improve accuracy. The dimensions of sample feature attributes in the data set are different, and direct calculation will lead to a deviation of the results. Here, use formula (1) to normalize the data and map the value of the data to the [0,1] interval.

$$d_i = \frac{d - d_{\min}}{d_{\max} - d_{\min}}. \tag{1}$$

In formula (1), $d$ represents the value of a dimension in the data, $d_{\min}$ represents the minimum value in the dimension, $d_{\max}$ represents the maximum value in the dimension, and $d_i$ represents the data after normalization.

*3.4. Feature Selection Module Design.* In order to reduce the number of intermediate parameters of the convolutional neural network model, reduce overfitting, and improve the data processing speed of the model, data feature screening is a common method in the field of machine learning. At present, the commonly used feature selection algorithms include genetic algorithm, principal component analysis algorithm, decision tree algorithm, and so on. The genetic algorithm first generates a batch of random feature subsets, sorts these subsets according to the evaluation function, and then multiplies the next generation subsets through crossover, mutation, and other operations. Then, the evaluation function filters all subsets, deletes the last subset, and selects the best subset after multiple generations of reproduction. The algorithm is too dependent on randomness, and the convergence speed is slow, so it is not suitable for intrusion detection-data sets. Principal component analysis (PCA) maps high-dimensional data to low-dimensional space through linear operation and expects the maximum variance in the projection dimension. However, the algorithm does not distinguish the physical meaning of the data, so it is not suitable for this experiment. A decision tree algorithm is a commonly used machine learning algorithm, which usually solves the problem of binary classification. Here, the methods of decision tree screening and "gain rate" optimization are used for feature selection.

In the process of feature selection, after the decision tree is fully grown, the pruning algorithm is carried out on the tree. Finally, the features of branches in the decision tree are the result of feature selection. However, the decision tree generally uses "information gain" as the evaluation function, which will lead to "majority bias" in feature screening. That is, it tends to select features with more attributes. In order to solve this problem, the "gain rate" is used for secondary screening of the selected features. The flow chart of the feature screening method is shown in Figure 2.

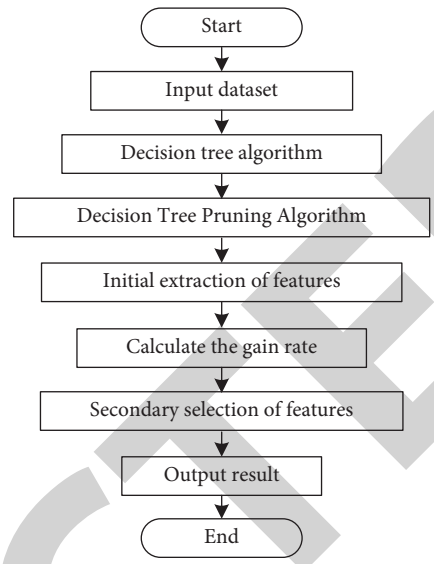


Figure 2: Feature screening process.

In Figure 2, first input the data set, then run the decision tree algorithm according to the content of the data set and generate the decision tree. On this basis, the postpruning technology is used to prune the generated decision tree, and the features are selected for the first time according to the pruning results. Finally, the gain rate of the first feature screening is calculated, and the features are sorted according to the gain rate. The largest features are selected as the results of the second feature screening.

The calculation process of "information gain" is shown in the following formula:

$$G(S, x) = H(S) - \sum_{i=1}^{I} \frac{|S_i|}{|S|} H\left(S^i\right). \tag{2}$$

In formula (2), $p_j$ represents the proportion of the class $j$ samples in the sample set $S$, $j = (1, 2, 3, \ldots, J)$. $x$ represents a certain attribute. The possible value of the attribute $x$ is $\{x_1, x_2, x_3, \ldots, x_I\}$. $I$ represents the number of possible values of the attribute $x$. Using the attribute $x$ to take the value of the sample will produce $I$ different types. It is assumed that type $i$ contains all the samples in $S$ whose value is $x_i$ on $x$, which is recorded as $S_i$. $H(S)$ represents "information entropy," which reflects the purity of the sample. The smaller the $H(S)$ value, the higher the purity of $S$. The calculation method of $H(S)$ is shown in the following formula:

$$H(S) = -\sum_{j=1}^{|J|} p_j \log_2 p_j. \tag{3}$$

The calculation formula of gain rate is shown in the following formula:

$$R(S,x) = \frac{G(S,x)}{T(x)}. \quad (4)$$

In formula (4), $G(S,x)$ represents "information gain" and $T(x)$ represents "inherent value" of attributes $x$. The more types of $x$ attributes, the greater the $T(x)$ value. The calculation method is shown in the following formula:

$$T(x) = -\sum_{i=1}^{I} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}. \quad (5)$$

The essence of the gain rate is to divide "information gain" by a constant. The value of the constant depends on the number of feature attributes. The more features, the greater the constant. In this way, the problem of "majority tendency" in "information gain" can be solved.

The decision-making process of the decision tree is shown in Figure 3.

### 3.5. Design of Detection Module

*3.5.1. Particle Swarm Optimization Algorithm.* PSO is a stochastic optimization technology based on swarm intelligence. The specific steps of the PSO algorithm are as follows:

(1) *Initialize Particle Swarm.* Determine the population size $G$, randomly initialize the position $P_k$ and the speed $V_k$ of each particle, and set the initial value of inertia weight $\alpha$, learning factor $\beta_1$ and $\beta_2$, and the maximum number of iterations $D$.

(2) *Fitness Function Evaluation.* Design an appropriate fitness function $f$ according to the problem to be solved, then calculate the fitness value $f(P_k)$ of each particle, and evaluate the particles according to the fitness value.

(3) For each particle, its individual optimal position $P_0$ and corresponding optimal fitness value $F_0$ are recorded. In each iteration, if the new fitness value is better than $F_0$, the new position and fitness value are used to replace $P_0$ and $F_0$.

(4) For particle swarm optimization, the global optimal position $P_b$ and corresponding global optimal fitness value $F_b$ in all particles are recorded. In each iteration, if the new fitness value is better than $F_b$, the new position and fitness value are used to replace $P_b$ and $F_b$.

(5) Update the speed and position of particles.

(6) Judge whether the maximum number of iterations is reached or the global optimal fitness value does not change. If it is satisfied, the global optimal solution is output, and the algorithm ends; otherwise, the number of iterations is increased by one, and jump to step (2) to continue the next iteration.

*3.5.2. Improved BP Detection Algorithm Based on PSO.* The performance of traditional BP algorithm largely depends on the initial weight and threshold, so it is necessary to improve the BP network to improve the convergence speed of the network. Here, the PSO algorithm is introduced into the optimization of the initial weight and threshold of BP, and an improved BP detection algorithm ALR-PSO-BP based on adaptive learning rate and PSO is proposed.

The specific process of the ALR-PSO-BP algorithm is as follows:

(1) Initialize the BP neural network training sample $T$, arbitrarily set the network weight and threshold, calculate the error function $E$ of the actual output value $U_a$ of each node and the corresponding expected value $V_a$ according to the following formula, and set the accuracy.

$$E = \frac{1}{2} \sum_t \sum_a (V_a - U_a)^2. \quad (6)$$

(2) Initialize the parameters of the particle swarm, calculate the dimension W of the particle, initialize the population, and generate the initial position and speed of each particle. The encoding of particle position is shown in Figure 4.

(3) Calculate the fitness value of each particle according to the following formula and compare it with the current best fitness value. If the current fitness value is better, $F_0$ will be updated; otherwise, $F_0$ will be maintained. Then, compare $F_0$ with the global optimal value $F_b$. If $F_0$ is better, update $F_b$; otherwise, maintain $F_b$.

$$f(P_k) = \frac{1}{1 + 1/2 \sum_{k=1}^{K} (t_{i0} - t_i)^2}. \quad (7)$$

In formula (7), $K$ represents the number of training samples of ALR-PSO-BP. $t_{i0}$ represents the $k$-th expected output. $t_i$ represents the $k$-th actual output.

(4) Update the inertia weight, and then update the position and velocity of the particles.

(5) If the current iteration reaches the maximum number or the error has been within the given range, the iterative process is ended, and the current global extreme value $P_b$ is the initial weight and threshold of BP neural network. Otherwise, go to (3).

(6) Based on the optimized initial weight and threshold obtained in step (5), the BP neural network is trained, and the intrusion detection model is established.

The basic flow chart of the ALR-PSO-BP detection algorithm is shown in Figure 5.

## 4. Experiments and Analysis

Six ordinary computers are used to build a 6-node Hadoop cloud computing platform in the laboratory. The environment configuration of each node is the same, including hardware configuration and software configuration. The specific node environment configuration is shown in Table 1.

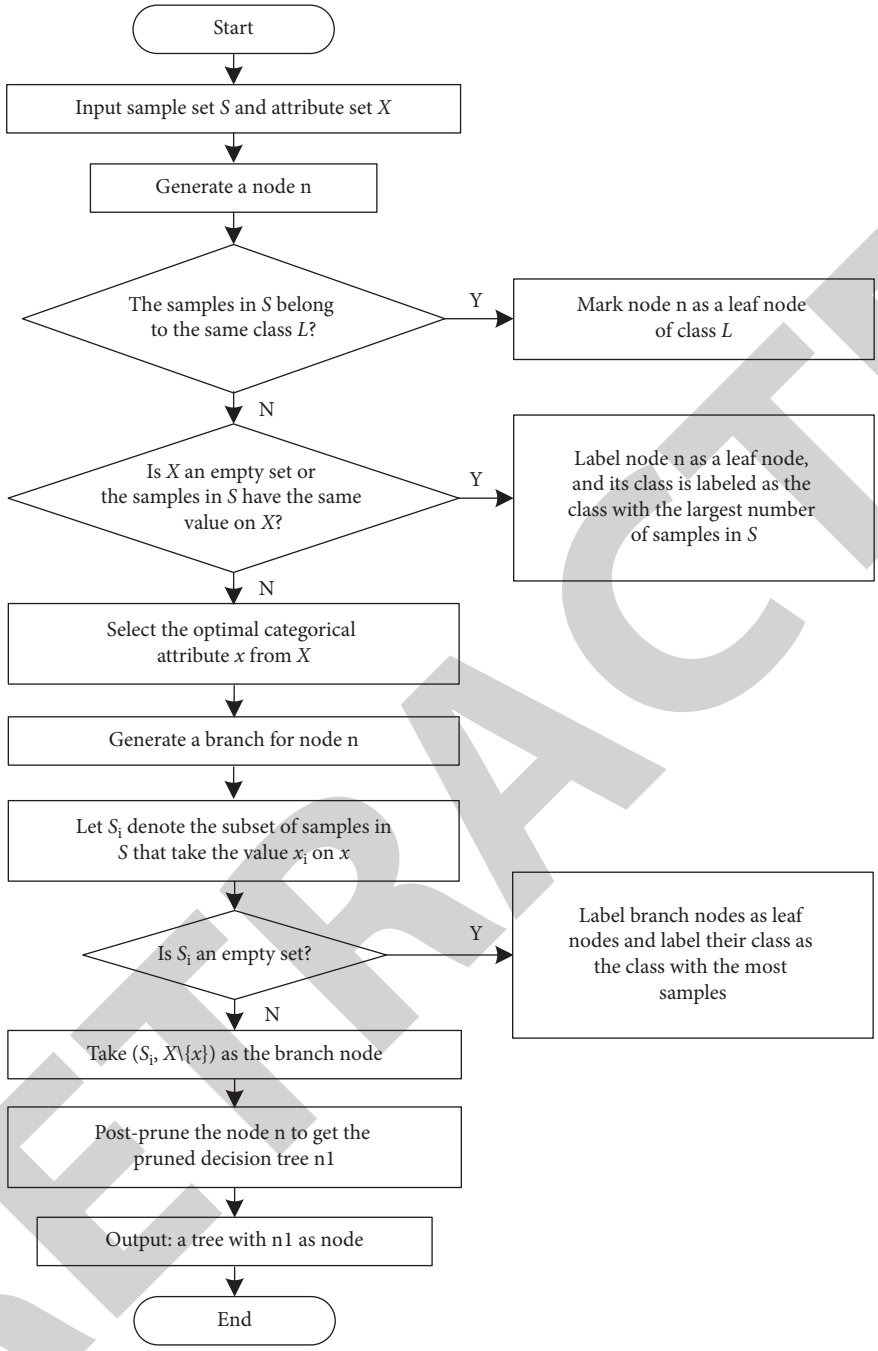


Figure 3: Decision tree decision process.

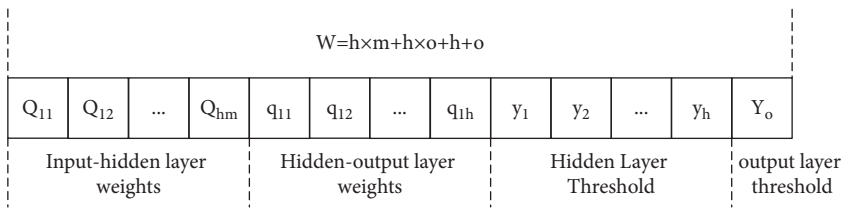


Figure 4: Particle position encoding.

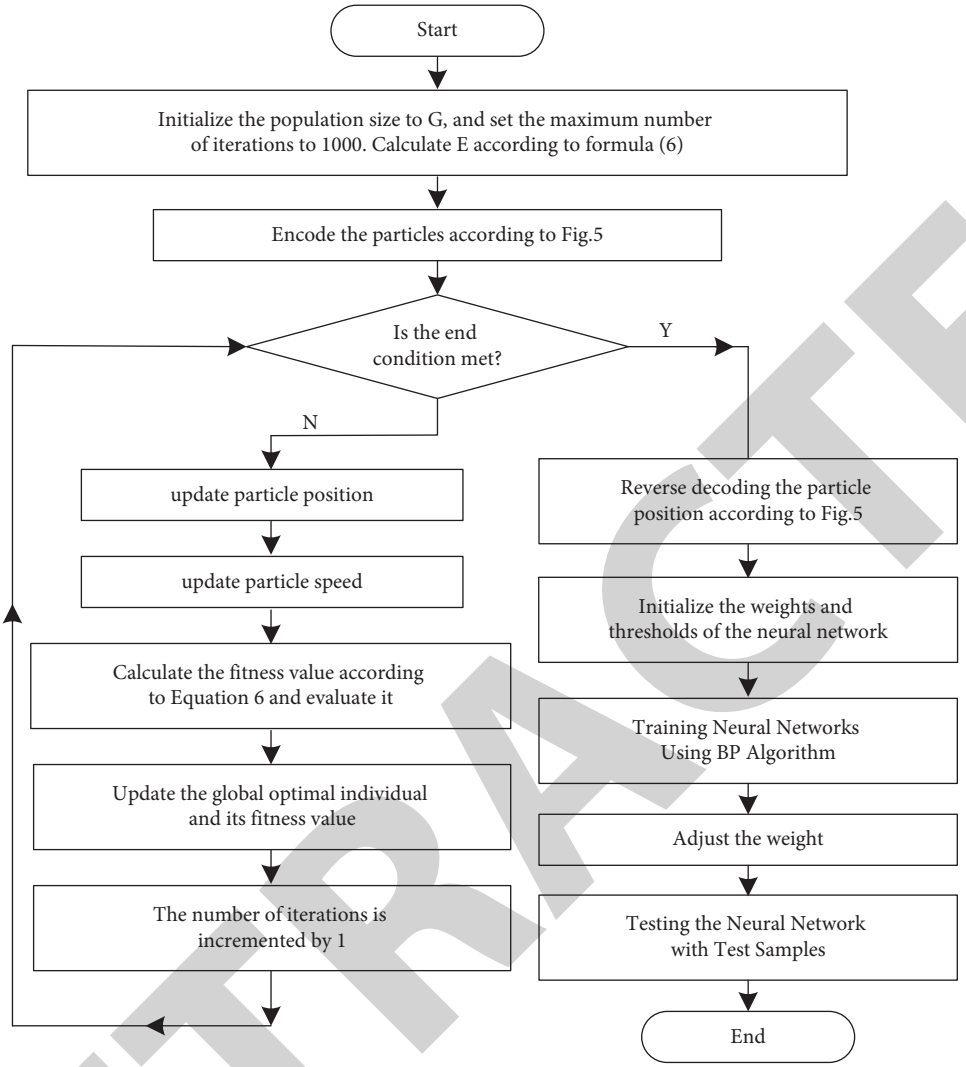


Figure 5: Flow chart of ALR-PSO-BP detection algorithm.

Table 1: Experimental operating environment.

| Name | Model |
|---|---|
| Operating system | Linux ubuntu12.04 |
| Hard disk | 500 GB |
| RAM | 4 GB |
| Processor | Intel(R) core(TM) i5 CPU |
| Hadoop version | Hadoop 2.2.0 |
| JDK version | JDK1.6.0 |

Table 2: Parameters of PSO algorithm.

| Parameters | Value |
|---|---|
| $\beta_1$ | 1 |
| $\beta_2$ | 0.8 |
| $\alpha$ | 0.8 |
| $Z_{\max}$ | 2 |
| $Z_{\min}$ | −2 |

The settings of PSO algorithm parameters are shown in Table 2.

In Table 2, $\alpha$ represents the initial value of inertia weight, $\beta_1$ and $\beta_2$ represent the learning factors, and $Z_{\max}$ and $Z_{\min}$ represent the maximum and minimum values of population search space.

The number of iterations of the BP algorithm is set to 1500. According to many comparative experiments, the learning rate of the BP neural network is set to 0.01 and the momentum factor is set to 0.008.

*4.1. Data Set and Evaluation Indices.* Scholars from Columbia University used data mining technology to process the 9-week TCPdump network connection data, and obtained the KDD-CUP99 data set. In the experiments, the data set NSL-KDD without duplicate records is obtained by processing the KDD-CUP99 data set. Each data in the NSL-KDD data set has 45 columns. The first 43 columns are the features of the data, the 44th column is the data label, and the 45th column is the number of occurrences of the data type. Since column 45 has no effect on the experiment, it is excluded. There are 40 sample types in the NSL-KDD data set.

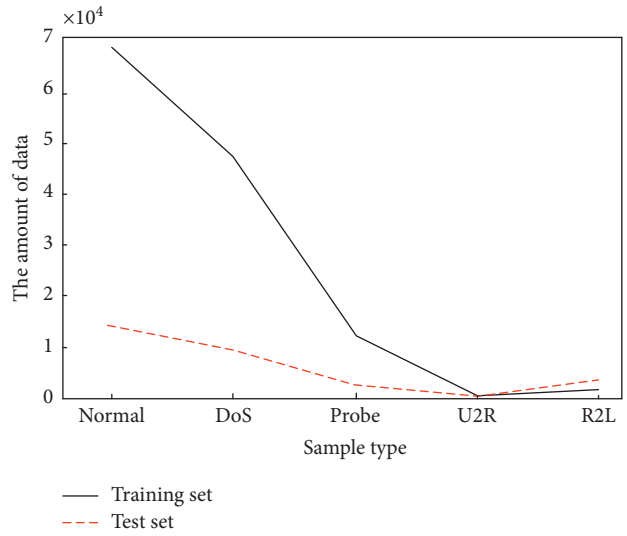


Figure 6: Distribution of sample types in the data set.

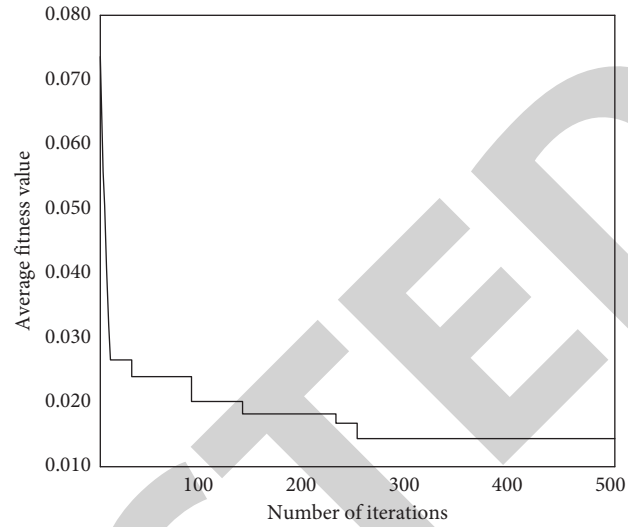


Figure 7: Fitness value of ALR-PSO-BP detection algorithm.

After summarizing the subcategories, they are divided into five main categories: Normal, DoS, R2L, U2R, and Probe. The test set has attack types that are not in the training set, which can well test the generalization performance of the intrusion detection model.

There are 69542 data of Normal type, 46892 data of DoS type, 13549 data of Probe type, 58 data of U2R type, and 1024 data of R2L type in the training set. In the test set, there are 9868 data of Normal type, 9967 data of DoS type, 2817 data of Probe type, 206 data of U2R type, and 2547 data of R2L type. The distribution of sample types of this data set is shown in Figure 6. As can be seen from Figure 6, there is a sample imbalance problem in the data set.

70% of samples are randomly selected from the data set as the training data of the BP neural network detection model, and the remaining 30% of samples are used as the test data of the detection model. The detection indices of the experimental test include detection rate and false detection rate, and their calculation methods are shown in the following formulas, respectively:

$$R_D = \frac{Y_T}{Y_S}, \tag{8}$$

$$R_F = \frac{Y_{FT}}{Y_S}. \tag{9}$$

In formulas (8) and (9), $R_D$ represents the detection rate, $R_F$ represents the false detection rate, $Y_T$ represents the number of samples being tested, $Y_{FT}$ represents the number of normal data detected by mistake, and $Y_S$ represents the total number of samples.

*4.2. Experimental Results and Analysis.* The following is a simulation experiment for the proposed intrusion detection algorithm based on the Particle Swarm Optimization-BP network algorithm in the cloud computing environment. The training data is used to test the change trend of the fitness value of the proposed ALR-PSO-BP detection algorithm for 50 times and then take the average value. The final result is shown in Figure 7.

It can be seen from the curve trend in Figure 7 that after about 260 iterations, the fitness value of the proposed algorithm has decreased to a stable level, which is about 0.015.

Based on the test data, 10 simulation tests and calculations are carried out for the detection rate of the proposed algorithm. The final results obtained by the ALR-PSO-BP detection algorithm, PSO algorithm, and BP neural network algorithm are shown in Figure 8.

As can be seen from Figure 8, whether it is a simple PSO algorithm or BP neural network algorithm, the final intrusion detection rate is much lower than the proposed ALR-PSO-BP algorithm. In 10 experiments, the lowest detection rate of the proposed algorithm is 94.82%, the highest is 95.93%, and the average detection rate is 95.13%. After the decision tree is fully grown, the feature extraction is realized by pruning algorithm on the tree, and the selected features are screened again in combination with the "gain rate," which improves the detection performance of the algorithm.

Next, taking the detection rate and false detection rate as the evaluation criteria, the proposed ALR-PSO-BP detection algorithm is compared and analyzed with the algorithms in reference [16–18] in the case of various types of samples, and 10 tests are carried out, respectively, to obtain the average value. The final calculation results of the average detection rate and false detection rate of different algorithms are shown in Tables 3 and 4.

It can be seen from Tables 3 and 4 that when five same types of sample data are used, respectively, the proposed intrusion detection method is superior to the other three comparison methods in terms of detection rate and false detection rate. Among the five types of samples, the lowest average detection rate of the proposed algorithm is 95.28%, and the highest average detection rate is 95.72%, which is greatly improved compared with the other three comparison algorithms. The highest average false detection rate of the proposed algorithm in the five types of samples is 2.33%, and

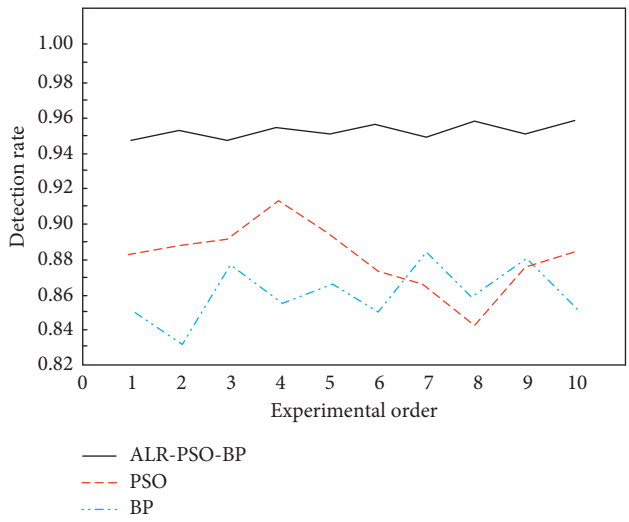


Figure 8: Detection rates of different algorithms.

Table 3: Average detection rate of different algorithms.

| Algorithms | Sample type | | | | |
|---|---|---|---|---|---|
| | Normal (%) | DoS (%) | Probe (%) | U2R (%) | R2L (%) |
| ALR-PSO-BP | 95.35 | 95.28 | 95.72 | 95.41 | 95.65 |
| Reference [16] | 90.38 | 90.11 | 90.82 | 90.23 | 89.95 |
| Reference [17] | 92.39 | 92.66 | 92.63 | 92.51 | 92.39 |
| Reference [18] | 91.44 | 91.32 | 91.71 | 91.92 | 91.28 |

Table 4: Average false detection rate of different algorithms.

| Algorithms | Sample type | | | | |
|---|---|---|---|---|---|
| | Normal (%) | DoS (%) | Probe (%) | U2R (%) | R2L (%) |
| ALR-PSO-BP | 2.03 | 2.14 | 2.08 | 2.33 | 2.24 |
| Reference [16] | 5.22 | 5.15 | 5.31 | 5.09 | 5.17 |
| Reference [17] | 4.11 | 4.52 | 4.42 | 4.25 | 4.46 |
| Reference [18] | 7.05 | 7.17 | 7.09 | 7.11 | 6.92 |

the lowest average false detection rate is 2.03%, which is lower than the other three comparison algorithms. Because the PSO algorithm is introduced into BP neural network, its initial weight and threshold are further optimized. While accelerating the convergence speed of the BP neural network, the problem of falling into local optimization is solved, and the detection rate of intrusion detection algorithm is improved. The introduction of momentum factor and adaptive learning rate method can better combine the global optimization ability of PSO algorithm and the advantages of gradient descent local search of BP algorithm and reduce the false detection rate of intrusion algorithm to a certain extent.

## 5. Conclusion

Aiming at the problems of low detection rate, high false-positive rate and slow computing speed of intrusion detection algorithm in the cloud computing environment, an intrusion detection-data security protection scheme based on the Particle Swarm Optimization-BP network algorithm in cloud computing environment is proposed. The experimental results show that the introduction of a decision tree algorithm and pruning algorithm on the tree after the decision tree has fully grown can reduce the overfitting and improve the data processing speed of the model. The secondary screening of the selected features combined with the "gain rate" can effectively avoid the "majority bias."

Through the summary of the work done in this paper, it is found that although the detection method proposed in this paper can solve some problems in the existing technology and achieve good detection results, there are still shortcomings, and there is still much room for improvement:

(1) The model proposed in this paper improves the detection effect of minority classes by oversampling the samples of minority classes and solves the bad impact of unbalanced samples in the data set on the classification algorithm from the data level. In the future, we will consider using a variety of oversampling techniques for comparison so as to better solve the problems brought by unbalanced samples to machine learning.

(2) The data set used in this paper is nsl-kdd. In the future, we will consider using the updated data set for experiments to further verify the performance of the model.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the Teaching Research and Reform Project of Higher Education of Guangdong Province (No. 20201224642) and the Cooperative Education of Project Higher Education Department of the Ministry of Education (No. 202101035008).

## References

[1] M. S. Akshaya and G. Padmavathi, "A survey on various intrusion detection system tools and methods in cloud computing," in *Proceedings of the 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 439–445, New Delhi, India, March 2019.

[2] S. Singh, M. Kubendiran, and A. K. Sangaiah, "A review on intrusion detection approaches in cloud security systems," *International Journal of Grid and Utility Computing*, vol. 10, no. 4, pp. 361–374, 2019.

[3] S. Alam, M. Shuaib, and A. Samad, "A collaborative study of intrusion detection and prevention techniques in cloud computing," in *Proceedings of the International Conference on*

*Innovative Computing and Communications (ICICC)*, pp. 231–240, New Delhi, India, March 2019.

[4] A. Bakshi and Sunanda, "A comparative analysis of different intrusion detection techniques in cloud computing," in *Proceedings of the 2nd International Conference on Advanced Informatics for Computing Research (ICAICR)*, pp. 358–378, Gurugram, India, December 2020.

[5] S. Ghribi, A. M. Makhlouf, F. Zarai, and M. Guizani, "Fog-cloud distributed intrusion detection and cooperation," *TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES*, vol. 23, no. 1, pp. 364–371, 2019.

[6] S. M. Alturfi, B. Al-Musawi, and H. A. Marhoon, "An advanced classification of cloud computing security techniques: a survey," in *Proceedings of the 8th International Conference on Applied Science and Technology (ICAST)*, pp. 501–505, Karbala, Iran, December 2020.

[7] A. Aldribi, I. Traore, B. Moa, and O. Nwamuo, "Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking," *Computers & Security*, vol. 88, no. 5, pp. 49–57, 2019.

[8] P. Lou, G. T. Lu, X. M. Jiang, Z. Xiao, J. Hu, and J. Yan, "Cyber intrusion detection through association rule mining on multi-source logs," *Applied Intelligence*, vol. 51, no. 6, pp. 4043–4057, 2020.

[9] M. Liu, Z. Xue, and X. J. He, "Two-tier intrusion detection framework for embedded systems," *IEEE CONSUMER ELECTRONICS MAGAZINE*, vol. 10, no. 5, pp. 102–108, 2021.

[10] W. Elmasry, A. Akbulut, and A. H. Zaim, "A design of an integrated cloud-based intrusion detection system with third party cloud service," *OPEN COMPUTER SCIENCE*, vol. 11, no. 1, pp. 365–379, 2021.

[11] V. Chang, L. Golightly, P. Modesti, and Q. A. Xu, L. M. Thao Doan, A survey on intrusion detection systems for fog and cloud computing," *Future Internet*, vol. 14, no. 3, pp. 164–171, 2022.

[12] L. B. Wen, "Cloud computing intrusion detection technology based on BP-NN," *Wireless Personal Communications*, vol. 13, no. 22, pp. 47–55, 2021.

[13] Y. S. Liu, L. Zhu, and F. Liu, "Optimal design of hadoop intrusion detection system based on neural network boosting algorithms," *Journal of Intelligent and Fuzzy Systems*, vol. 37, no. 5, pp. 6127–6138, 2019.

[14] S. Dey, Q. Ye, and S. Sampalli, "A machine learning based intrusion detection scheme for data fusion in mobile clouds involving heterogeneous client networks," *Information Fusion*, vol. 49, no. 3, pp. 205–215, 2019.

[15] M. Nadeem, A. Arshad, S. Riaz, S. S. Band, and A Mosavi, "Intercept the cloud network from brute force and DDoS attacks via intrusion detection and prevention system," *IEEE Access*, vol. 9, no. 12, Article ID 152300, 2021.

[16] J. X. Wei, C. Long, J. W. Li, and J. Zhao, "An intrusion detection algorithm based on bag representation with ensemble support vector machine in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 24, pp. 115–122, 2020.

[17] S. S. Sathiyadhas and M. C. V. S. Antony, "A network intrusion detection system in cloud computing environment using dragonfly improved invasive weed optimization integrated Shepard convolutional neural network," *International Journal of Adaptive Control and Signal Processing*, vol. 12, no. 5, pp. 304–312, 2022.

[18] A. N. Jaber and S. U. Rehman, "FCM-SVM based intrusion detection system for cloud computing environment," *Cluster Computing*, vol. 23, no. 4, pp. 3221–3231, 2020.