IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 4, 2020, accepted March 17, 2020, date of publication March 23, 2020, date of current version April 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982418

Network Intrusion Detection Based on PSO-Xgboost Model

HUI JIANG[1], ZHENG HE[2,3], GANG YE[2,3], AND HUYIN ZHANG[1]

[1]School of Computer Science, Wuhan University, Wuhan 430072, China
[2]National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan 430072, China
[3]Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, Wuhan 430072, China

Corresponding author: Zheng He (hezheng@whu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61772386 and Grant 61862015, and in part by the State Grid Hubei Electric Power Company Ltd. under Grant SGHBDK00DWJS1800134.

ABSTRACT Network intrusion detection system (NIDS) is a commonly used tool to detect attacks and protect networks, while one of its general limitations is the false positive issue. On the basis of our comparative experiments and analysis for the characteristics of the particle swarm optimization (PSO) and Xgboost, this paper proposes the PSO-Xgboost model given its overall higher classification accuracy than other alternative models such like Xgboost, Random Forest, Bagging and Adaboost. Firstly, a classification model based on Xgboost is constructed, and then PSO is used to adaptively search for the optimal structure of Xgboost. The benchmark NSL-KDD dataset is used to evaluate the proposed model. Our experimental results demonstrate that PSO-Xgboost model outperforms other comparative models in precision, recall, macro-average (macro) and mean average precision (mAP), especially when identifying minority groups of attacks like U2R and R2L. This work also provides experimental arguments for the application of swarm intelligence in NIDS.

INDEX TERMS Intrusion detection, PSO-Xgboost, ensemble learning, particle swarm optimization.

I. INTRODUCTION

With the rapid development of the Internet, artificial intelligence and big data technologies, network security confronts more complicated threats than ever before. The requirement for a more powerful and effective network intrusion detection system (NIDS) [1] is on the rise. A NIDS is to convert intrusion detection into pattern recognition and classification by using related algorithms to collect, clean, model and classify various behaviors in the network [2].

Network intrusion detection can be broadly divided into two categories by the methods it uses: anomaly detection and misuse detection. Anomaly detection builds a normal network traffic behavior model. The behaviors that do not match the normal model are defined as intrusions. This kind of detection can identify unknown attacks. Conversely, misuse detection builds an intrusion model based on abnormal behaviors. The behaviors that match this model are defined as intrusions.

Superior to the traditional defense systems like firewall, NIDS can capture data packets, extract their features and compare them with known attack patterns. With these functions, NIDS can be monitored in real-time. However, NIDS has some general limitations at its early stage: its false positive rate is often high; the system occupies too many resources; its capability to detect unknown attacks is poor, and thus manual intervention is required. In recent years, researchers in related fields have achieved remarkable advancements to improve NIDS by introducing machine learning, data mining, and other technologies to the systems, such as the restricted Boltzmann machines applied to Dos attack detection [3], artificial immune system approaches [4], and the application of autoencoder and SVM [5]. These efforts make NIDS more and more reproducible and adaptive.

Ensemble learning is a popular trend for machine learning, and shows more stable performance than a single model. Random Forest model, as one of the major ensemble models, can achieve better performance in most cases especially in classification problems. Xgboost model, proposed in 2016 by Chen and Guestrin [6] with the University of Washington uses tree structures and introduces second-order derivatives and regularization terms, which improves the efficiency of the algorithm, and has extraordinary performance on detecting minority groups of attacks. Xgboost has been widely used by

The associate editor coordinating the review of this manuscript and approving it for publication was Vicente Alarcon-Aquino.

academia and industry users [7], [8]. However, in a machine learning system, the parameters of the model determine the performance of the model to a large extent. There are many parameters in the Xgboost model, if human experience is used to set the parameters, the complexity of the work is increased. Swarm intelligence algorithms optimize targets based on population behavior. Its core is to achieve complex functions through cooperation between individuals, and PSO is one of the most widely used swarm intelligence algorithms.

This paper proposes a PSO-Xgboost model that combines swarm intelligence optimization with machine learning algorithm. In this hybrid model, the parameters of the Xgboost model is optimized by using the good search ability of PSO. The main contributions of this work are as follows:

1) We develop a novel model PSO-Xgboost based on Xgboost by using PSO to adaptively optimize its parameters. This can effectively improve the performance of network intrusion detection, including the improvement of the detection accuracy of various types of attacks, especially on minority groups of attacks.
2) To evaluate the PSO-Xgboost model's performance, we measure not only the overall metrics but also the metrics of each class, and compare them with Xgboost and other ensemble learning models (like Random Forest and Bagging). Our evaluations use NSL-KDD dataset as the benchmark.

The rest of the paper is organized as follows. Section II reviews the related works in the field. Section III elaborates the technological theories applied and the construction of PSO-Xgboost model. The experimental comparison and performance evaluation are presented in Section IV that prove the effectiveness of the proposed model. Section V concludes the work.

II. RELATED WORK

Many machine learning algorithms have been applied in NIDS, including K-Nearest Neighbor (KNN) [9], artificial neural networks (ANN) [10], support vector machine (SVM) [11], and Naive Bayesian (NB) [12]. However, when using these algorithms, the data needs to be processed for missing values, the results of the algorithms are not stable enough, and the program runs slowly because the amount of data is large. The Xgboost model has a better processing mechanism for the above problems. In [13], the Xgboost algorithm was applied to network intrusion detection, the advantages of the Xgboost model over other classification models was evaluated. The results showed that the Xgboost model performs the highest accuracy on the NSL-KDD dataset, which is better than SVM, NB and Random Forest. However, the parameter setting of the Xgboost model is to find a set of parameters which make the model perform best by fixing the values of several parameters and performing a finite number of exhaustive method on other parameters. Su *et al.* [14] proposed to apply the improved smote algorithm to the imbalanced KDD99 dataset. By oversampling the minority classes to improve model performance. This method improves the detection rate from the dataset level, but further work can be done at the model level.

In recent years, many researchers have achieved better performance by combining swarm intelligence algorithms with other methods to solve prediction problems [15]–[17]. A work done by Qiao *et al.* [15] proposed a model combining improved whale optimization algorithm (IWOA) with relevance vector machine (RVM). This IWOA-RVM model was applied to short-term prediction of natural gas load, and demonstrated higher prediction accuracy than other models. In [17], an improved whale algorithm was used to optimize important parameters of the Volterra adaptive filter and make predictions of natural gas consumption. Responding to the classification problem, Wang *et al.* [18] proposed to use PSO to automatically search the optimal architecture of convolutional neural networks (CNNs). Their work used a novel encoding strategy to easily encode CNN layers, and designed a disabled layer to achieve variable-length particles. The results showed that the proposed approach had higher classification accuracy than other existing algorithms. Li *et al.* [19] used PSO to optimize the penalty factor and kernel function of SVM, and then used the PSO-SVM algorithm to classify bridge cable pictures, which has a good application in the detection of bridge cables.

Researchers have also done a lot of work to improve swarm intelligence algorithms. For example, high-dimensional functions easily falling into the local optimal solution is a common problem for optimization, so it is important to improve the global search ability of optimization algorithms [20]–[22]. In [22], for example, chaotic mapping method was introduced into dolphin swarm algorithm (DSA) to improve the global search ability of the optimization algorithm. The results showed the effectiveness of the chaotic dolphin swarm algorithm (CDSA) based on Kent map.

In most cases, combining two algorithms can effectively improve performance [23], [24]. PSO and its variants are widely used in network intrusion detection. Tan *et al.* [25] proposed to use PSO algorithm to optimize deep belief network (DBN) and apply it to network intrusion detection. Their results showed that the effect of PSO was superior to intelligent optimization algorithms such as genetic algorithm (GA) and simulated annealing (SA), thus the PSO-DBN model also performed other machine learning models. Sakr *et al.* [26] applied the PSO-SVM algorithm to network intrusion detection in cloud computing, by using binary-based PSO (BPSO) for network feature selection, and standard-based PSO (SPSO) to adjust control parameters of SVM. The results showed that the BPSO-SPSO-SVM model achieved higher detection accuracy and lower false alarm rates (FARs).

However, in [25], [26], the results only gave the overall metrics of the algorithm on the dataset, including accuracy, f-measures and so on, but not the performance of the algorithm in each class. In network intrusion detection, common types of attacks are easier to detect, such as Dos attacks; while rare types of attacks, such as R2L and U2R, are more difficult

to detect, because the system does not have enough database about the features of these attacks. Therefore, the accuracy of detecting the minority classes is particularly important in network intrusion detection.

To summarize, many network intrusion detection methods based on machine learning have been proposed in recent years, but most of them still have some limitations, such as:

- The algorithm itself does not have a good missing value and overfitting processing mechanism.
- The combination of swarm intelligence and machine learning methods can achieve good performance in network intrusion detection, but the experimental results are mostly comparative analysis of overall metrics without considering the effects of each class, and the performance on minority classes tends to be more important.

Based on the above studies and findings, we propose the combination of PSO and Xgboost, and evaluate the performance of this hybrid model on each class. By using PSO to effectively optimize the Xgboost model, and using the PSO-Xgboost algorithm to evaluate the NSL-KDD dataset, we find that the performance of the algorithm can be effectively improved on most classes.

III. INTRUSION DETECTION BASED ON PSO-XGBOOST

A. XGBOOST

The Xgboost algorithm is based on GBDT [27]. Compared with GBDT, the advantage of Xgboost is that it supports linear classifiers and performs Taylor expansion for the cost function by introducing the second derivative to make the results more accurate. There are the following principles of Xgboost.

Xgboost model uses additive training method to optimize the objective function, which means the optimization process of the latter step relies on the result of its previous step. The t-th objective function of the model can be expressed as

$$obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \Omega(f_t) + constant \quad (1)$$

where l represents the loss term of the t-th round, *constant* represents a constant term, and Ω is the regularization term of model, shown as

$$\Omega(f_t) = \gamma \cdot T_t + \lambda \frac{1}{2} \sum_{j=1}^{T} w_j^2 \quad (2)$$

both γ and λ are customization parameters. Generally, the larger these two values are, the simpler the structure of the tree is. And the problems of overfitting can be effectively solved.

Perform a second-order Taylor expansion on (1). This process is given by

$$obj^{(t)} = \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}_i^{t-1}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + constant \quad (3)$$

where g is the first derivative, and h is the second derivative. They can be described as

$$g_i = \partial_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1}) \quad (4)$$

$$h_i = \partial^2_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1}) \quad (5)$$

Substitute (2), (4), (5) into (3) and take the derivative. Then solutions can be obtained from (6) and (7) as

$$w_j^* = -\frac{\sum g_i}{\sum h_i + \lambda} \quad (6)$$

$$obj^* = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum g_i)^2}{\sum h_i + \lambda} + \gamma \cdot T \quad (7)$$

where obj^* represents the score of loss function. The smaller the score, the better the structure of the tree. w_j^* refers to the solution of weights.

B. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is proposed by Wang *et al.* [28]. The idea is derived from the research of bird swarm foraging behavior. Particles are the simulation of birds. Each particle can be regarded as a search individual in the N-dimensional search space. The current position of each particle is a candidate solution to the problem. Each particle has two attributes: velocity and position. Velocity represents the step of movement, and position denotes the direction. The optimal solution found by each particle is taken as the individual optimal, and the optimal solution of all particles is regarded as the global optimal. By iterating many times like this, the velocity and position are continuously updated, and the iteration will exit when the termination conditions are met.

The larger inertial weight is good for jumping out of local optimal and the smaller inertial weight is conductive to accurate local search of the search space. The former is convenient for global search and the latter is good for algorithm convergence. Therefore, the inertial weight can be linearly decreased. The process of PSO can be described as follows:

1) Randomly initialize the velocity and position of particles in the velocity and search space.
2) Define the fitness function. Each particle has its own individual optimal solution, and the global optimal is generated from these individual optimal solutions. Then the current global optimal is compared to historical global optimal, and the comparison result will determine whether to update the global optimal.
3) The update of each particle's velocity and position is expressed as

$$V_{id} = \omega V_{id} + C_1 random(0, 1)(P_{id} - X_{id}) + C_2 random(0, 1)(P_{gd} - X_{id}) \quad (8)$$

$$X_{id} = X_{id} + V_{id} \quad (9)$$

where C_1 and C_2 are the individual and social learning factors, P_{id} is the d-th dimension of individual optimal of the i-th particle, and P_{gd} is the d-th dimension of global optimal. ω is inertial weight, its linearly decreasing strategy can be expressed as

$$\omega = \frac{\omega_{max} + (iter - iter_i) \times (\omega_{max} - \omega_{min})}{iter} \quad (10)$$

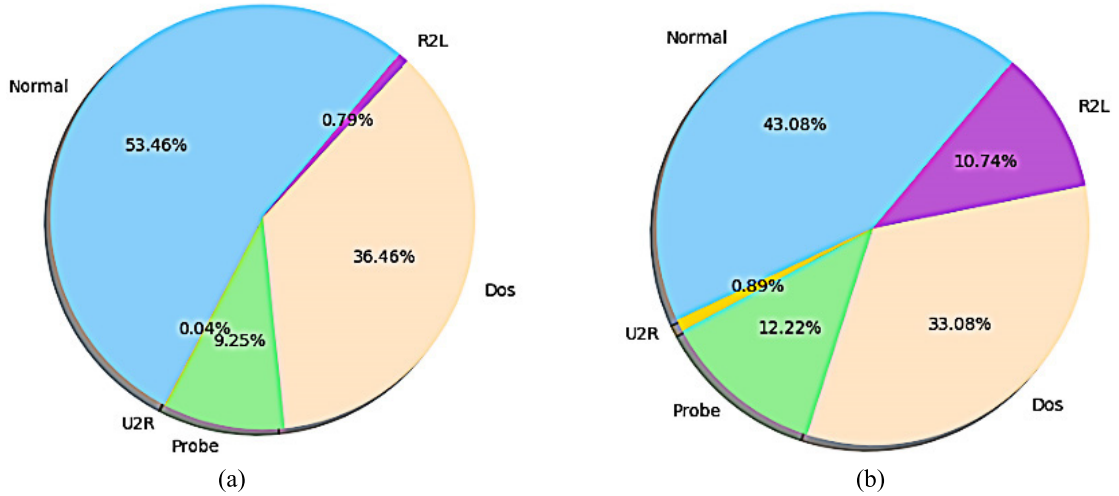


FIGURE 1. Distribution of classes in the dataset. (a) Distribution of classes in the training data (total 125973). (b) Distribution of classes in the test data (total 22544).

where *iter* is the maximum number of iterations, $iter_i$ is the current number of iterations, ω_{max} and ω_{min} are the maximum and minimum value of ω respectively.

C. DATA PREPROCESSING

Being an optimized version from KDD99, NSL-KDD [29], [30] dataset overcome the inherent problems on the dataset. By removing the redundant and duplicate records, the classifier tends not be biased towards more frequent recordings and shall have higher detection accuracy. There are four classes of attacks in the NSL-KDD dataset: Probe, Dos, U2R, and R2L. Under each attack class, there might be multiple attack behaviors such as nmap, smurf and so on. The distributions of classes in the training and test data are shown in Fig. 1(a) and Fig. 1(b), respectively.

Among 41 features of the dataset, there are 9 discrete features and 32 continuous features. Because different features may have different measurement methods, in order to avoid the impact of the unit of measurement, the data needs to be standardized. Suppose there are m records in the dataset, X_{ij} represents the i-th eigenvalue of the j-th data, where $1 \leq i \leq m$. Then continuous data can be standardized by

$$\hat{X}_{ij} = \frac{X_{ij} - AVG_j}{STAD_j} \tag{11}$$

where AVG_j represents the average of the j-th feature data in the dataset, and $STAD_j$ represents the average absolute error of the j-th feature column data. They are expressed as

$$AVG_j = \frac{1}{m}\sum_{i=1}^{m} X_{ij} \tag{12}$$

$$STAD_j = \frac{1}{m}\sum_{i=1}^{m} |X_{ij} - AVG_j| \tag{13}$$

After data standardization, it is then normalized by using maximum and minimum normalization method, which is given by

$$\hat{X}'_{ij} = \frac{\hat{X}_{ij} - X_{min}}{X_{max} - X_{min}} \tag{14}$$

where X_{min} and X_{max} are the minimum and the maximum values of the j-th feature.

D. MODEL METRICS

For one class, it can be regarded as positive cases, and others will be regarded as negative cases. In this experiment, we used common metrics like precision, recall and f-measures to evaluate the model. In addition, the precision-recall (P-R) curve is also used as one of the evaluation metrics. Instead, receiver operating characteristics (ROC) curve is insensitive to class distribution, but P-R curve can capture the impact of a large number of negative cases on the performance of the model [31]–[34]. This means that when the model needs to classify the minority class such as U2R, the remaining four classes are regarded as a large number of negative cases. The P-R curve can correctly reflect the model's performance on U2R, but ROC cannot achieve this purpose.

Average precision (AP) represents the area calculated from the P-R curve. The larger the AP value of a class, the better the performance of the model on it. Other metrics such as mean average precision (mAP) and macro-averaging (macro) are also used to describe the integrative performance of the model on all classes.

E. PSO-XGBOOST MODEL

Xgboost model contains general parameters, booster parameters and learning target parameters. In this experiment, we use PSO to optimize six parameters that have a great influence on the model: learning rate (eta), maximum tree depth (max_depth), minimum leaf weight (min_child_weight), gamma, subsample and colsample_bytree. The information of each parameter is shown in Table 1.

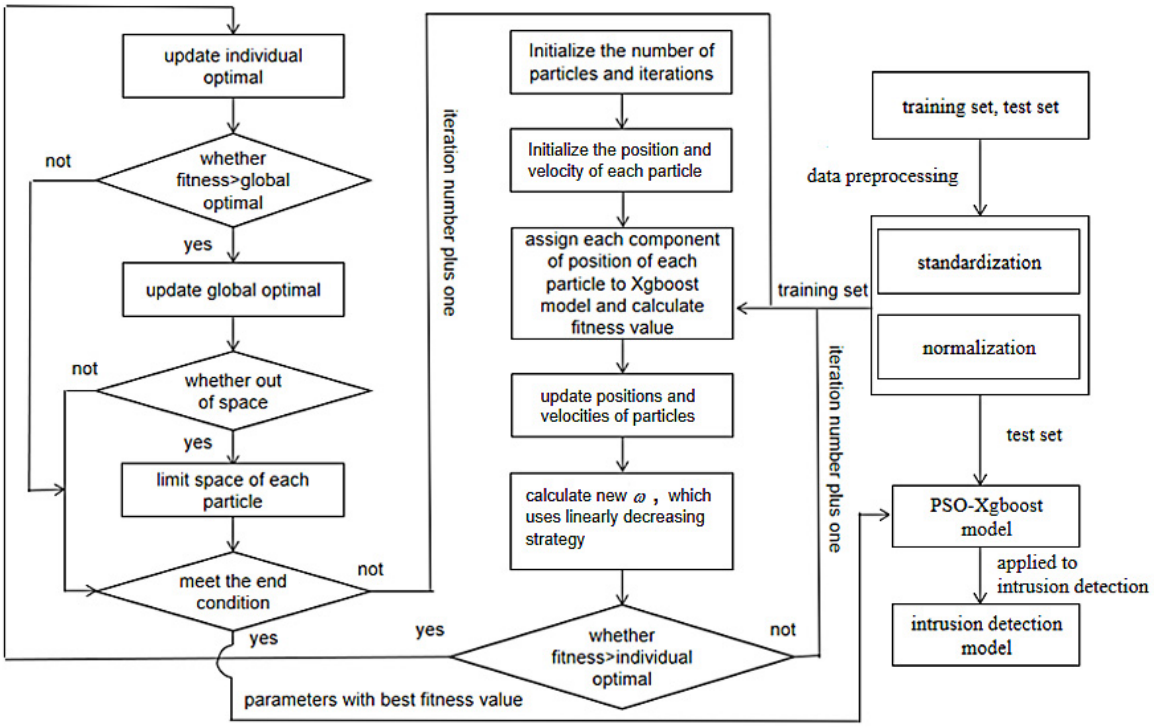


FIGURE 2. The pipeline of the proposed PSO-Xgboost model.

TABLE 1. Information about parameters of the Xgboost model.

Parameters	default value	range	explanations
eta	0.3	[0,1]	Learning rate, reducing the weight of each step.
max_depth	6	[0,∞]	Maximum depth of tree. The larger the value, the model will learn more specific and local samples.
min_child_weight	1	[0,∞]	Minimum leaf weight, when the value is large, the model can avoid learning the local optimal solution.
gamma	0	[0,∞]	Related to loss function.
subsample	1	(0,1]	Control the proportion of random sampling of each tree to prevent overfitting.
colsample_bytree	1	(0,1]	Control the proportion of sampling features.

The pipeline of the PSO-Xgboost model is shown in Fig. 2. The steps for constructing the PSO-Xgboost model are as follows:

1) Determine dimensions according to the number of parameters to be optimized, then randomly initialize positions and velocities of particles. The position attribute of each particle is a 6-dimensional vector, and its range refers to the entire search space. And the components of each dimension correspond to different Xgboost parameters, so the initialized ranges of each dimension are different. The position vector of the i-th particle at time t can be expressed as

$$P_{i(t)} = [p_{i(t)}^{eta}, p_{i(t)}^{\max\_depth}, p_{i(t)}^{\min\_child\_weight}, p_{i(t)}^{gamma}, p_{i(t)}^{subsample}, p_{i(t)}^{colsample\_bytree}] \quad (15)$$

Since all particles move in the same search space, the velocity can be initialized to the (0, 1) range in each dimension when $t = 0$. The velocity vector of the i-th particle at time t can be expressed as follow

$$V_{i(t)} = [v_{i(t)}^{eta}, v_{i(t)}^{\max\_depth}, v_{i(t)}^{\min\_child\_weight}, v_{i(t)}^{gamma}, v_{i(t)}^{subsample}, v_{i(t)}^{colsample\_bytree}] \quad (16)$$

We then assign the position vector to the corresponding parameters of the model, and take the performance on the training set as the initial fitness value. The fitness value of the i-th particle at time t is

$$F_{i(t)} = (P_{i(t)} \rightarrow Xgboost|_{trainingset})_{[metric=P-Rcurve]} \quad (17)$$

For the i-th particle, its individual optimal at time t can be expressed as

$$Pbest_{i(t)} = \max\left(F_{i(j)}\right), \quad 0 \le j \le t \quad (18)$$

Suppose there are m particles, the global optimal at time t is written as

$$Gbest_{(t)} = \max\left(Pbest_{k(t)}\right), 1 \le k \le m \quad (19)$$

TABLE 2. Parameter settings of PSO.

Parameters	value
particle numbers	100
maximum number of iterations	80
local learning factor C1	2
global learning factor C2	2
decreasing range of inertia weight	(0.4, 0.9)
particle dimensions	6

TABLE 3. Three sets of optimal parameters and the corresponding fitness values.

Parameters	1	2	3
eta	0.33	0.16	0.48
max_depth	4	5	4
min_child_weight	0.29	0.86	3.80
gamma	0.03	0.65	0.95
subsample	0.31	0.14	0.47
colsample_tree	0.14	0.84	0.01
fitness value	0.8730	0.8719	0.8648

2) Update the position, velocity and inertia weight of each particle according to (8) to (10). Given that the position may exceed the range of the search space after the movement of each particle, it is necessary to control the boundary. Then we assign the new position to the model and calculate the new fitness value. After comparing with the historical fitness, we determine which individual optimal values of the particles to be updated and judge whether to update the global optimal values.

3) The algorithm iterates and terminates when it meets the maximum number of iterations or convergence. Then it outputs the optimal fitness value and the corresponding optimal position.

The parameters of the PSO itself mainly include the number of particles, dimensions of each particle, maximum number of iterations, local learning factor, global learning factor, and inertia weight. The parameter settings are tabulated in Table 2. The fitness function uses the area calculated from the P-R curve on the dataset. Table 3 shows three sets of optimal experimental data. It can be seen that the first set of parameters performs best, where eta is 0.33, max_depth is 4, min_child_weight is 0.29, gamma is 0.03, subsample is 0.31, and colsample_bytree is 0.14. This set of parameters will be used for experimental comparisons.

IV. EXPERIMENTAL COMPARISONS AND PERFORMANCE EVALUATION

A. EXPERIMENTAL RESULTS

The experimental environment of this paper is based on Python 3.6.3. This section describes a set of parameters with the best performance for experimental comparisons and analysis, where eta is 0.33, max_depth is 4, min_child_weight is 0.29, gamma is 0.03, subsample is 0.31, and colsample_tree is 0.14. Then the PSO-Xgboost model is applied to the NSL-KDD testing set. Some metrics of each class are shown in Table 4. From these metrics, we can see U2R has the highest precision, but its recall is also the lowest. This means that although the samples predicted by the classifier as positive are all from U2R, but the number of samples is very small. In other words, a large number of samples from U2R are predicted by the model as other classes. F-measures can be a good trade-off between precision and recall. Therefore, we need to consider the accuracy, recall rate and f-measures comprehensively. It can be seen that the performance of PSO-Xgboost model on Normal, Probe and Dos is better, while not well on the latter two (classes perform worse).

TABLE 4. Confusion matrix of optimal parameters on testing set.

Class	precision	recall	F-score	support
Normal	0.66	0.97	0.78	9771
Probe	0.80	0.52	0.63	2421
Dos	0.95	0.83	0.88	7458
U2R	1.00	0.01	0.01	200
R2L	0.96	0.05	0.09	2754
Avg./total	0.81	0.75	0.71	22544

TABLE 5. AP value in each class of PSO-Xgboost model.

	Normal	Probe	Dos	U2R	R2L
AP value	0.90	0.79	0.94	0.15	0.49

The calculation from the P-R curve of the PSO-Xgboost model in each class (namely the AP value) is shown in Table 5.

Compared with the area under curve (AUC) calculated by ROC, the AP value can better represent the performance of the model on minority classes. From the Table 4, we can see that the AP value is positively correlated with the f-measures mentioned above. Generally, the larger the f-measures of a class, the better the model performs on it. It can be seen that the AP values on U2R and R2L are lower than other three classes. The main reason is that some classes such as Normal, Probe and Dos in the NSL-KDD training set account for more than 99% of the total data, while U2R and R2L account for less than 1%. This means that the model is unable to learn the features of some potential distributions of U2R and R2L, so U2R and R2L in the test set cannot be well classified.

B. COMPARISON WITH XGBOOST

For a dataset of an imbalanced class distribution, even if the model has a poor performance on minority class, the ROC

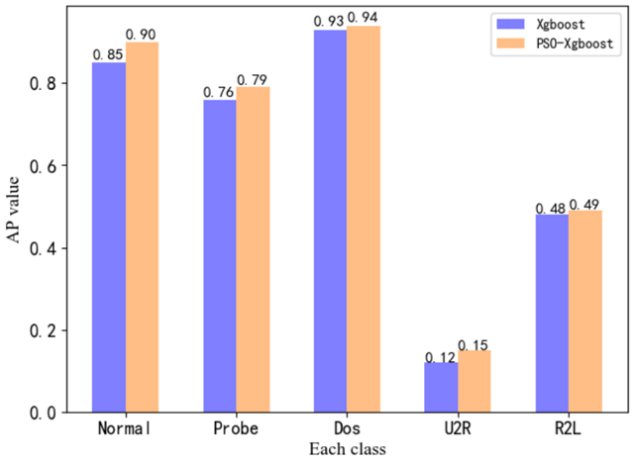


FIGURE 3. AP value for each class of PSO-Xgboost and Xgboost.

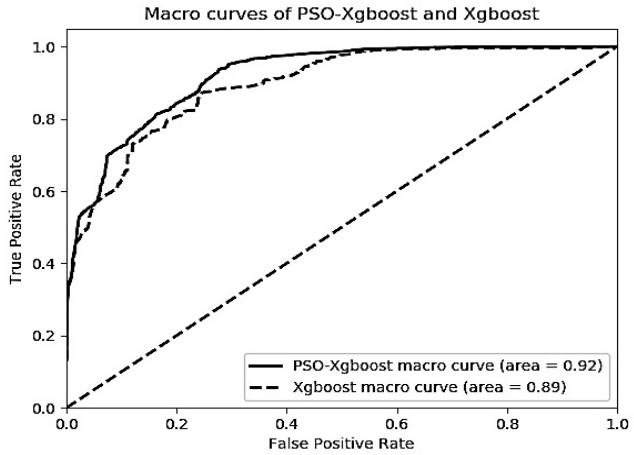


FIGURE 4. Comparison between macro curves of PSO-Xgboost and Xgboost.

curve of that class may be too optimistic, which is undoubtedly a misunderstanding of the model. The P-R curve can better reflect the performance of the model in minority classes. Therefore, this experiment also uses the AP value to compare the performance between two models. The AP values of each class are shown in Fig. 3.

The default parameter settings of the Xgboost model are as follows: eta is 0.3, max_depth is 6, min_child_weight is 1, gamma is 0, subsample is 1, and colsample_bytree is 1. Being optimized by PSO, the final parameters of PSO-Xgboost appear in an irregular form, and are not artificially specified in advance by grid search method. As can be seen from Fig. 3, both models have a good effect on the first three classes, while PSO-Xgboost performs better than Xgboost. In addition, compared with Xgboost, the AP value of PSO-Xgboost in each class is higher, which illustrates the effectiveness of PSO to improve model performance.

The macro and mAP curves are shown in Fig. 4 and Fig. 5, respectively. As can be seen from Fig. 4, the macro curve of PSO-Xgboost almost encompasses the macro curve of Xgboost. However, in Fig. 5, the mAP curves of

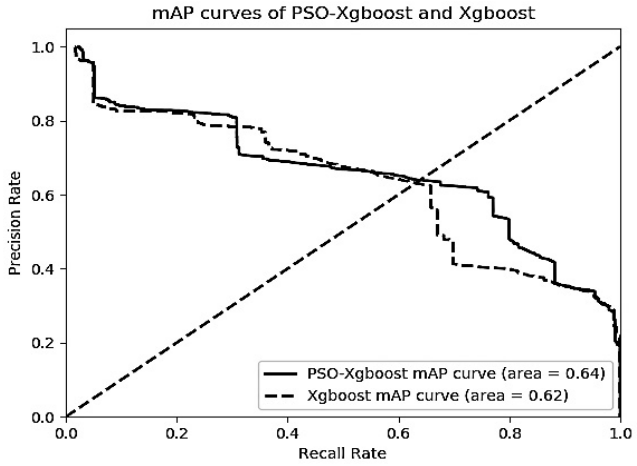


FIGURE 5. Comparison between mAP curves of PSO-Xgboost and Xgboost.

PSO-Xgboost and Xgboost intersect at multiple points. This means Xgboost is better than PSO-Xgboost in one region, and PSO-Xgboost is better than Xgboost in another region. It is difficult to intuitively judge which model is better based on the curve alone, therefore, we evaluate the comprehensive performance of the model by the area enclosed by the curve.

It can be seen that the area of PSO-Xgboost in macro and mAP curves are larger than Xgboost, where the area of macro curve is 3% higher than Xgboost, and the area of mAP curve is 2% higher than Xgboost. In the context of network intrusion detection, small improvements for performance sometimes enable the system to detect more attacks that are difficult to detect, and so more effective defense measures can be taken. Evidently, PSO can effectively optimize the parameters of the Xgboost model, thereby improving the classification performance on dataset.

C. COMPARISONS WITH OTHER MODELS

To verify the effectiveness of PSO-Xgboost, this section compares PSO-Xgboost with some commonly used ensemble learning models, such as Random Forest, Bagging and Adaboost. The experiment still uses the AP value calculated by the P-R curve as the classifier's evaluation on each class.

Fig. 6 tabulates the comparison results. From Fig. 6, we can see that for the classes with a large number of training samples (such as Normal, Probe, and Dos), both models have a better detection effect than the other two classes. On the Normal, Probe and Dos classes, the performance gap between Bagging, Random Forest and PSO-Xgboost is not large. On the one hand, PSO-Xgboost has the highest AP values on Normal and Dos, followed by Random Forest. One the other hand, Random Forest and Bagging get the highest AP value on Probe, which is 1% higher PSO-Xgboost. However, on minority classes such as U2R and R2L, PSO-Xgboost has an obvious advantage over other models. On U2R, PSO-Xgboost performs the best, followed by Bagging, and the AP value of PSO-Xgboost is 6% higher than Bagging.

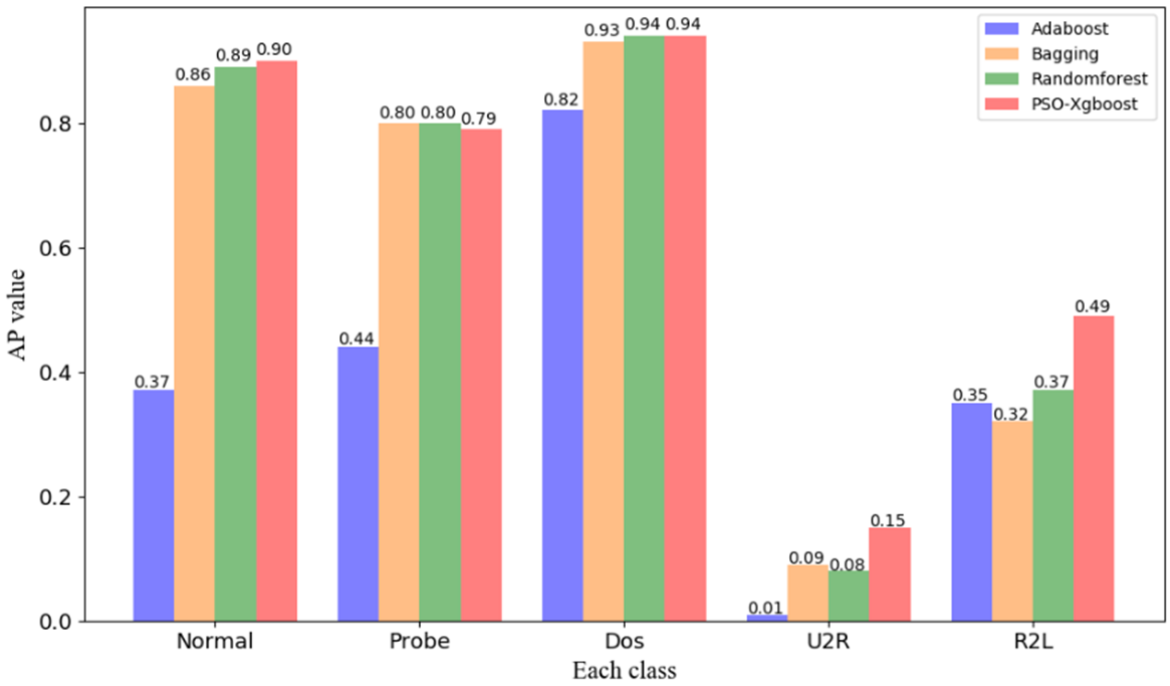


FIGURE 6. AP value for each class of PSO-Xgboost and other ensemble models.

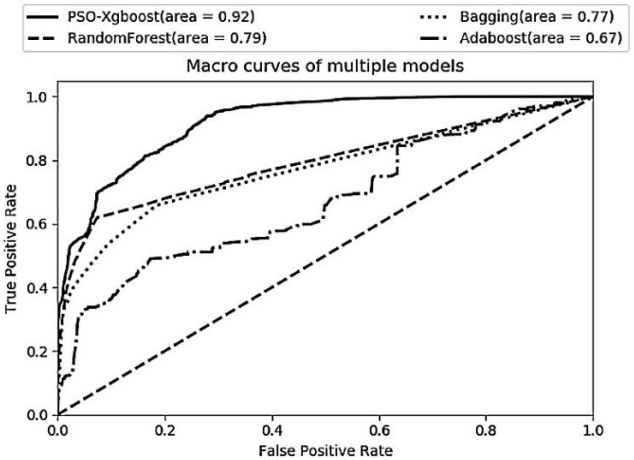


FIGURE 7. Comparison between macro curves of PSO-Xgboost and other models.

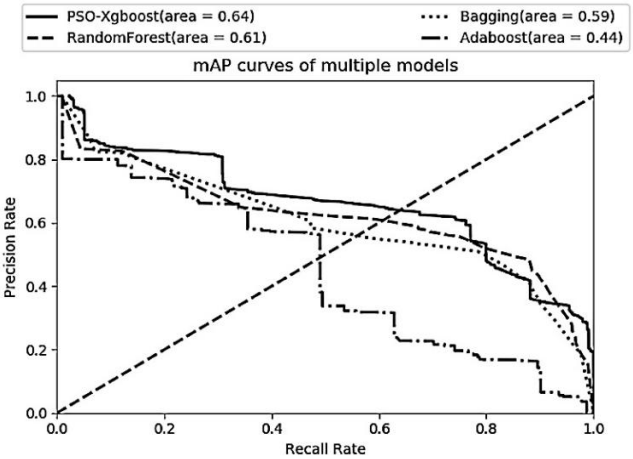


FIGURE 8. Comparison between mAP curves of PSO-Xgboost and other models.

Similar to R2L, the AP value of PSO-Xgboost is 12% higher than Random Forest.

Compared with other models, PSO-Xgboost has a higher detection rate on minority attack types, which is particularly important for the capability of NIDS to detect unknown attacks.

The macro and mAP curves of these models are shown in Fig. 7 and Fig. 8, respectively.

In Fig. 7, the macro curve of PSO-Xgboost encompasses the macro curve of other models. For quantitative analysis, we calculate the area under the macro curve. It can be seen from the data that PSO-Xgboost performs better than other models, followed by Random Forest, and the area of PSO-Xgboost is 13% higher than Random Forest.

As for mAP values, curves of the four models in Fig. 8 intersect, where the area of PSO-Xgboost is the largest, with a value of 0.64, followed by Random Forest with a value of 0.61.

V. CONCLUSION

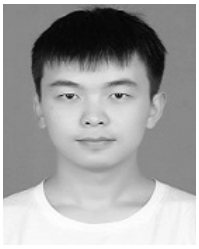
Improving the detection accuracy of NIDS is an important issue in the field of network security. Xgboost model can be effectively applied to overcome multi-classification problems. PSO can achieve approximate optimal solution at a fast rate. This paper discusses the method of optimizing the Xgboost model by PSO. The experimental results show that our model has higher mAP and macro value compared with other models such as Random Forest, Bagging and Adaboost.

In terms of AP value of each class, the advantages of our model are more obvious in minority classes such as U2R and R2L. The proposed method in this paper provides an idea of swarm intelligence applications in NIDS, which can also be applied to solve other classification problems.

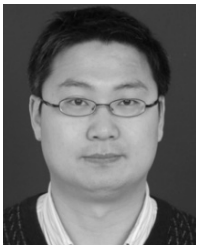
There are still also some functions in this model to be improved. If the number of particles or iterations is small, the algorithm is prone to fall into a local optimal solution. Instead, if the number of particles or iterations is large, the global optimum can be found more efficiently, but the computational will spend long time. In the future, we will conduct further research regarding the above issues.

REFERENCES

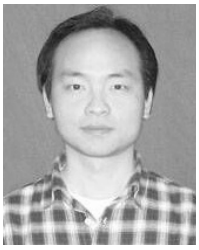
[1] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.
[2] T. Shi and Y. Zhao, "Overviews of network intrusion evasion and defense techniques," *Netinfo Secur.*, vol. 1, pp. 70–74, Jan. 2016.
[3] Y. Imamverdiyev and F. Abdullayeva, "Deep learning method for denial of service attack detection based on restricted Boltzmann machine," *Big Data*, vol. 6, no. 2, pp. 159–169, Jun. 2018.
[4] D. Hooks, X. Yuan, K. Roy, A. Esterline, and J. Hernandez, "Applying artificial immune system for intrusion detection," in *Proc. IEEE 4th Int. Conf. Big Data Comput. Service Appl. (BigDataService)*, Bamberg, Germany, Mar. 2018, pp. 287–292.
[5] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018.
[6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Francisco, CA, USA, Aug. 2016, pp. 785–794.
[7] H. Zheng, J. Yuan, and L. Chen, "Short-term load forecasting using EMD-LSTM neural networks with a xgboost algorithm for feature importance evaluation," *Energies*, vol. 10, no. 8, p. 1168, Aug. 2017.
[8] L. Torlay, M. Perrone-Bertolotti, E. Thomas, and M. Baciu, "Machine learning–XGBoost analysis of language networks to classify patients with epilepsy," *Brain Informat.*, vol. 4, no. 3, pp. 159–169, Apr. 2017.
[9] F. Chen, Z. Ye, C. Wang, L. Yan, and R. Wang, "A feature selection approach for network intrusion detection based on tree-seed algorithm and K-nearest neighbor," in *Proc. IEEE 4th Int. Symp. Wireless Syst. Int. Conf. Intell. Data Acquisition Adv. Comput. Syst. (IDAACS-SWS)*, Lviv, Ukraine, Sep. 2018, pp. 68–72.
[10] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," *ICT Express*, vol. 4, no. 2, pp. 95–99, Jun. 2018.
[11] J. Gu, L. Wang, H. Wang, and S. Wang, "A novel approach to intrusion detection using SVM ensemble with feature augmentation," *Comput. Secur.*, vol. 86, pp. 53–62, Sep. 2019.
[12] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," *Procedia Technol.*, vol. 4, pp. 119–128, Feb. 2012.
[13] S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Information*, vol. 9, no. 7, p. 149, Jun. 2018.
[14] P. Su, Y. Liu, and X. Song, "Research on intrusion detection method based on improved smote and XGBoost," in *Proc. 8th Int. Conf. Commun. Netw. Secur. (ICCNS)*, Nov. 2018, pp. 37–41.
[15] W. Qiao, K. Huang, M. Azimi, and S. Han, "A novel hybrid prediction model for hourly gas consumption in supply side based on improved whale optimization algorithm and relevance vector machine," *IEEE Access*, vol. 7, pp. 88218–88230, 2019.
[16] W. Qiao, H. Lu, G. Zhou, M. Azimi, Q. Yang, and W. Tian, "A hybrid algorithm for carbon dioxide emissions forecasting based on improved lion swarm optimizer," *J. Cleaner Prod.*, vol. 244, Jan. 2020, Art. no. 118612.
[17] W. Qiao, Z. Yang, Z. Kang, and Z. Pan, "Short-term natural gas consumption prediction based on volterra adaptive filter and improved whale optimization algorithm," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103323.
[18] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.
[19] X. Li, Y. Guo, and Y. Li, "Particle swarm optimization-based SVM for classification of cable surface defects of the cable-stayed bridges," *IEEE Access*, vol. 8, pp. 44485–44492, 2020.
[20] W. Qiao and Z. Yang, "An improved dolphin swarm algorithm based on kernel fuzzy C-Means in the application of solving the optimal problems of large-scale function," *IEEE Access*, vol. 8, pp. 2073–2089, 2020.
[21] W. Qiao and Z. Yang, "Solving large-scale function optimization problem by using a new metaheuristic algorithm based on quantum dolphin swarm algorithm," *IEEE Access*, vol. 7, pp. 138972–138989, 2019.
[22] W. Qiao and Z. Yang, "Modified dolphin swarm algorithm based on chaotic maps for solving high-dimensional function optimization problems," *IEEE Access*, vol. 7, pp. 110472–110486, 2019.
[23] W. Qiao, W. Tian, Y. Tian, Q. Yang, Y. Wang, and J. Zhang, "The forecasting of PM2.5 using a hybrid model based on wavelet transform and an improved deep learning algorithm," *IEEE Access*, vol. 7, pp. 142814–142825, 2019.
[24] W. Qiao and Z. Yang, "Forecast the electricity price of US using a wavelet transform-based hybrid model," *Energy*, vol. 193, Feb. 2020, Art. no. 116704.
[25] X. Tan, S. Su, Z. Zuo, X. Guo, and X. Sun, "Intrusion detection of UAVs based on the deep belief network optimized by PSO," *Sensors*, vol. 19, no. 24, p. 5529, Dec. 2019.
[26] M. M. Sakr, M. A. Tawfeeq, and A. B. El-Sisi, "Network intrusion detection system based PSO-SVM for cloud computing," *Int. J. Comput. Netw. Inf. Secur.*, vol. 11, no. 3, pp. 22–29, Mar. 2019.
[27] X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu, "Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning," *Electron. Commerce Res. Appl.*, vol. 31, pp. 24–39, Sep. 2018.
[28] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: An overview," *Soft Comput.*, vol. 22, no. 2, pp. 387–408, Jan. 2018.
[29] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, Jun. 2015.
[30] R. Bala and R. Nagpal, "A review on KDD CUP99 and NSL NSL-KDD dataset," *Int. J. Adv. Res. Comput. Sci.*, vol. 10, no. 2, pp. 64–67, Apr. 2019.
[31] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, Mar. 2015, Art. no. e0118432.
[32] B. Ozenne, F. Subtil, and D. Maucort-Boulch, "The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases," *J. Clin. Epidemiol.*, vol. 68, no. 8, pp. 855–859, Aug. 2015.
[33] H. R. Sofaer, J. A. Hoeting, and C. S. Jarnevich, "The area under the precision-recall curve as a performance metric for rare binary events," *Methods Ecol. Evol.*, vol. 10, no. 4, pp. 565–577, Dec. 2018.
[34] Z. Liu and H. D. Bondell, "Binormal Precision–Recall curves for optimal classification of imbalanced data," *Statist. Biosci.*, vol. 11, no. 1, pp. 141–161, Feb. 2019.



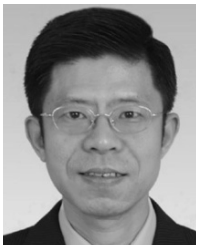
HUI JIANG is currently pursuing the master's degree with the School of Computer Science, Wuhan University, Wuhan, China. His current research interests include machine learning and network security.



ZHENG HE received the Ph.D. degree in computer science from Hiroshima University, Hiroshima, Japan, in 2007. He is currently a Lecturer with the School of Computer, Wuhan University, Wuhan, China. His research interests include data mining, image processing, and multimedia communications.



GANG YE received the Ph.D. degree in computer applied technology from Wuhan University, Wuhan, China, in 2009. His research interests include data mining, semantic web, and multimedia communications.



HUYIN ZHANG is currently a Professor in computer science with Wuhan University, where he directs the National Experimental Teaching Center of Virtual Simulation and has trained over 80 graduate students. He has authored ten monographs and textbooks and over 70 articles, of which more than 50 were SCI/EI/ISTP indexed. His main research interests focus on computer networks and communications, network QoS, data center software energy saving, high-performance computing, the next-generation Internet, network management, e-learning, and multimedia distance education.