IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 22, 2021, accepted November 22, 2021, date of publication December 1, 2021,
date of current version December 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3132052

Research on Optimization of GWO-BP Model for Cloud Server Load Prediction

KE HOU[1], MINGCHENG GUO[1], XINHAO LI[1], AND HE ZHANG[2]
[1]School of Economics and Management, Xi'an Shiyou University, Xi'an 710065, China
[2]Department of Petroleum Engineering, University of Louisiana at Lafayette, Lafayette, LA 70503, USA

Corresponding author: Ke Hou (kehou@188.com)

This work was supported in part by the Key Research and Development Projects of Shaanxi Province under Grant 2021GY083, and in part by the Key Scientific Research Program Funded by Shaanxi Provincial Education Department under Grant 21JK044.

ABSTRACT To improve the accuracy of cloud server resource load prediction, particle swarm optimization (PSO) algorithm, gray wolf optimization (GWO) algorithm and BP neural network are studied in-depth and applied. Firstly, the PSO algorithm is introduced to optimize the location update method in the search process of gray wolf. Secondly, the convex function is introduced to improve the linear convergence of the traditional GWO algorithm. Then the optimized GWO algorithm is used to further improve the assignment of weights and thresholds in the traditional BP neural network model, to construct a multi-stage optimized cloud server load prediction model, referred to as PSO- GWO-BP prediction model. Finally, the performance of the PSO- GWO-BP prediction model is verified by comparison experiments.

INDEX TERMS BP neural network, particle swarm optimization, gray wolf optimizer, cloud server.

I. INTRODUCTION

With the increasing maturity of cloud computing technology, more and more enterprises and individual users are choosing cloud data center hosting to meet the needs of data processing, computing, storage, and other tasks in their daily work. The continuous and rapid growth of cloud platform users and differentiated user needs has led to increasingly prominent resource management problems such as load imbalance among computing nodes [1]. To effectively solve such problems, the resource scheduling system must have the ability to predict accurately resource load of cloud server, strengthen further the rational allocation of resources, and improve resource utilization and cloud server performance [2].

Resource load prediction belongs to the research field of resource management and provides a reliable basis for the design of resource scheduling algorithms. The research in the field of cloud server resource management mainly focuses on resource scheduling and load prediction [3]. The cloud server resource scheduling system rationally allocates and dynamically reclaims computing and storage resources according to the requirements of user application. For example, the sudden peak period of cloud service access requests can easily lead to a significant reduction in service quality caused by the resource contention [4]. However, the peak period of cloud service usage usually has a certain regularity in a long term. Through the analysis and mining of historical data, the resource load prediction for a period in the future can be realized [5], to provide a basis for resource scheduling and ensure the service performance and quality of the cloud server.

The purpose of this paper is to build a resource load prediction model for cloud server with high prediction accuracy. Through the comparative analysis of existing research methods, BP neural network is used as the base model in this paper. The particle swarm optimization (PSO) algorithm and the gray wolf optimization (GWO) algorithm are introduced to improve the traditional BP neural network model. Firstly, the PSO algorithm is used to improve the location update method of the GWO algorithm, and then the linear convergence of the traditional GWO algorithm is optimized by the convex function. Finally, the optimized GWO algorithm is used to improve the calculation method of initial weight and threshold in the traditional BP neural network. Therefore, a multi-stage optimized PSO-GWO-BP model for cloud server load prediction is proposed and constructed, which is referred to as PSO-GWO-BP prediction model for short. In addition, when calculating the resource load value, this paper uses the entropy weight method to design the weight of five factors and multiplies the weight by the indexes, and then sum them to obtain the resource load value. To verify the availability and performance of the model, this paper

The associate editor coordinating the review of this manuscript and approving it for publication was Venkateshkumar M.

compared it with the support vector machine (SVM) model, the traditional BP neural network model, the BAS-BP model, the GWO-BP model, and the SSA-BP model. The experimental results proved that the comprehensive performance of the PSO-GWO-BP prediction model is better than the above five models. Therefore, the PSO-GWO-BP model can more accurately and effectively predict the overall trend of cloud server resource load changes, which is conducive to improve the intelligent management level of server monitoring and performance optimization in cloud data centers.

The rest of this paper is arranged as follows: the second part is the literature review. The third section is the introduction of GWO model and PSO model. The fourth part uses the traditional BP neural network model as the basic model and studies and proposes the PSO-GWO-BP prediction model. The fifth part verifies the availability and performance of the PSO-GWO-BP prediction model through the comparison experiments. The sixth part summarizes the paper.

II. LITERATURE REVIEW

Scientific and effective cloud service load prediction methods can provide a strong basis for resource scheduling decisions, to indirectly improve the utilization of cloud server resources and enhance its service performance. Through the analysis of the literature in recent years, it is found that the existing research on resource load prediction can be roughly divided into two categories.

The first category is the resource load prediction based on traditional statistical methods. The research on load prediction using statistical methods can be traced back to 1990. Hesterberg used weighted least squares linear regression technology for the load prediction [6]. Since then, many researchers have applied other types of regression statistical methods to load prediction. These methods can be divided into two categories: linear regression prediction methods and time series prediction methods. The linear regression prediction model is a kind of statistical method to analyze and find the causal relationship between resource load value and the key factors which affect resource load based on historical data, and then build a mathematical model to predict the change of resource load in the future. For example, Yang *et al.* used a linear regression method to predict the workload in the next period and designed an automatic scaling mechanism to scale virtual resources according to cloud workload conditions [7]. But this method assumes that the change trend of cloud service load is linear in the short term. The time series prediction model is a type of statistical method to study and figure out the change law of the data based on the statistical analysis of the past time series data and predict the resource load in the future according to this law. For example, Wang Xu and Chen Xiaoyi used the analytic hierarchy process to synthesize the utilization of CPU, memory and disk and obtained the comprehensive resource utilization of the power information system, and then used the auto-regressive integrated moving average (ARIMA) time series prediction method to predict the resource utilization and system response time. The prediction results are used to judge the load status of the system [8]. But this method has higher requirements for the stability of the data. Usually, with changing resource load, data has composite characteristics which are linear and non-linear [9]. The disadvantage of pure linear regression prediction is that it cannot accurately describe the characteristics of resource load change. Although time series prediction can analyze the changing trend of resource load, it is limited to short-term resource usage prediction. In addition, resource load prediction models on traditional statistical methods have poor data processing capabilities facing massive data.

The second category is the resource load prediction based on artificial intelligence methods. The research of artificial intelligence technology in load prediction can be traced back to around 1990 [10], which mainly involves the application of single methods such as support vector machines (SVM), feed-forward neural networks, and Bayes methods. Due to shortcomings in these methods, many scholars developed some combination algorithms. For example, to improve the accuracy of SVM in resource load prediction, Zhao Li designed a combination function for SVM to improve its learning ability [11]. Cortez *et al.* established a load prediction model on random forest to predict the actual resource load by analyzing the characteristics of virtual machine load data [12]. This model proved that the advance prediction can not only improve the utilization of resources, but also prevent the depletion of physical resources. Di designed a prediction model based on the Bayes method, which realized the prediction of load fluctuation in long-term intervals. The result of experiments has verified that the model has higher accuracy than auto-regressive prediction, moving average and other methods [13]. Bey *et al.* proposed a fuzzy reasoning system based on fuzzy clustering and adaptive network, but its prediction result is affected by the number of clusters [14]. Qian Shengpan *et al.* proposed a multi-step online prediction model based on a deep cyclic neural network encoder-decoder, which can predict the future multi-step host load value by collecting online real-time data [15]. In the experiment of this model, only the CPU utilization is considered. However, the load of the host is affected by many factors, so the generalization ability of this method needs to be verified. The advantage of artificial intelligence method is that the accuracy of the prediction results is generally better than traditional linear regression and time series prediction models, and it is more suitable for large-scale data processing. Because the performance of these models is easily affected by model parameter setting, research on improvement of model initialization has become a hot topic. From this point of view, this paper studies the optimization of cloud service load prediction model.

At present, the mainstream of artificial intelligence methods in the field of prediction research includes logistic regression, Naive Bayes, random forest, SVM and Back

Propagation neural network (BP neural network). The Logistic regression algorithm cannot be used to solve nonlinear problems [16]. Naive Bayes algorithm is limited to classification prediction [17]. The random forest algorithm cannot make prediction beyond the data range of the training set, which may lead to over-fitting while modeling data containing certain specific noises [18]. While the support vector machine faces a large-scale data set, the storage and calculation of the data will consume a lot of machine memory and computing time [19]. The BP neural network is a kind of neural networks with multiple feed-forward layers, which includes input layer, hidden layer and output layer, and there are interconnections among the nodes. BP neural network has strong nonlinear processing ability, generalization ability, fault tolerance, adaptability, and self-learning ability. The model is not only easy to implement, but also widely applied. At present, BP neural network has been successfully applied to solve prediction problems in different fields, especially in future resource use prediction [20], traffic prediction [21], demand prediction [22] of cloud data center, etc. Compared with other artificial intelligence methods, the application and performance of BP neural network are more prominent in prediction research. Based on the above analysis, this paper uses the BP neural network model as the basic model of load prediction, and then studies and optimizes it. BP neural network is very sensitive to the initial weight and initial threshold of each layer in the network, which will have a great impact on its accuracy. These weights and thresholds are assigned randomly in the traditional BP neural network. Although the whole model will continuously adjust the weights through error back propagation to find the optimal weights and thresholds, it is easy to fall into local optimization.

To solve the above problems, GWO algorithm is selected to assign initial weight and threshold of BP neural network. The GWO algorithm has the advantages of global optimization, few control parameters, and easy implementation. It is widely used in model parameter optimization and has been proved to have significant effects on the optimization of BP neural network models [23].

After GWO algorithm's own characteristics and convergence mode are further studied, it is found that the GWO algorithm itself may converge prematurely and fall into a local optimum [24]. Therefore, this paper improves the GWO algorithm from two aspects. On the one hand, due to the characteristics of strong memory and frequent communication of particles, PSO algorithm is used to strengthen the communication ability in the optimization process of GWO algorithm, effectively preventing premature convergence. On the other hand, the convex function is introduced to replace the original linear decreasing convergence, preventing GWO algorithm from falling into local optimization. Finally, performances of the PSO-GWO-BP model in prediction accuracy, convergence speed and stability are verified by comparative experiments.

III. RELATED THEORIES

A. GRAY WOLF OPTIMIZATION ALGORITHM

The Gray Wolf Optimization (GWO) algorithm is a new population intelligence optimization algorithm proposed by Mirjailili *et al.* The core of the algorithm is simulating the hunting process of the gray wolf [25]. Because of its simplicity and ease of implementation and few control parameters, the GWO algorithm is widely used in many fields: economic transportation [26]; workshop scheduling [27]; optimization of model parameters, such as PID controllers [28], Support Vector Machine (SVM); hybrid algorithm design, such as the gray wolf-bat (GWO-BA) optimization algorithm [29], hybrid gray wolf-genetic (GWO-GA) optimization algorithm [30].

In the GWO algorithm, wolves are divided into four ranks, from high to low: the first rank are named as wolves α, the second rank as wolves β, the third rank as wolves δ, and the fourth rank as wolves ω. Wolves of each rank must strictly obey the leadership of the previous rank. The gray wolf pack determines the prey position, evaluates the distance, and adjusts the best hunting position, and then repeats this process until the prey is captured successfully. The whole algorithm contains three main stages: encirclement, hunting, and attack. Its specific process is as follows.

(1) Encirclement stage: The main purpose is to identify the target prey and then encircle them. The mathematical model is as follows.

$$D = |C \cdot X_j(i) - X(i)| \tag{1}$$

where i is the number of current iterations; $X_j(i)$ is the position of prey in the current i generation; $X(i)$ is the position of individual gray wolf in the current i generation; so D denotes the distance between the current prey and the gray wolf. The position transformation formula of the gray wolf is as follows.

$$X(i+1) = X_j(i) - A \cdot D \tag{2}$$

$$A = 2a \cdot r_1 - a \tag{3}$$

$$a = 2 - 2 \cdot i/i_{max} \tag{4}$$

$$C = 2 \cdot r_2 \tag{5}$$

where A and C are both coefficients; $X_j(i+1)$ and $D_\alpha = |C_1 \cdot X_\alpha - X(i)|$ is a random number in the interval $[0, 1]$; a is the convergence factor and it is inversely proportional to the number of iterations, whose value decreases linearly from 2 to 0.

(2) Hunting stage: after the encirclement, wolves α will lead wolves β and wolves δ to hunt, while wolves ω will be guided by wolves α, β and δ to update position. Where $X_j(i+1)$ is the optimal solution for $i+1$ iterations, and the mathematical model of specific location information transformation is as follows.

$$D_\alpha = |C_1 \cdot X_\alpha - X(i)| \tag{6}$$

$$D_\beta = |C_2 \cdot X_\beta - X(i)| \tag{7}$$

$$D_\delta = |C_3 \cdot X_\delta - X(i)| \tag{8}$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha \tag{9}$$

$$X_2 = X_\beta - A_2 \cdot D_\beta \tag{10}$$
$$X_3 = X_\delta - A_3 \cdot D_\delta \tag{11}$$
$$X_j(i+1) = \frac{X_1 + X_2 + X_3}{3} \tag{12}$$

where C_1, C_2, C_3 are coefficients that can be calculated according to formula (5) to obtain specific values.

(3) Attack stage: The target is captured, and the optimal solution is obtained. As the convergence factor a decreases linearly from 2 to 0, the value of A varies within $[-2, 2]$ in formulas (2) and (3). When $|A| \geq 1$, it means that it is still in the global search stage; while $|A| < 1$, the wolf pack will launch an attack to capture the targets which are already locked. The flowchart of GWO algorithm is shown in Figure 1.

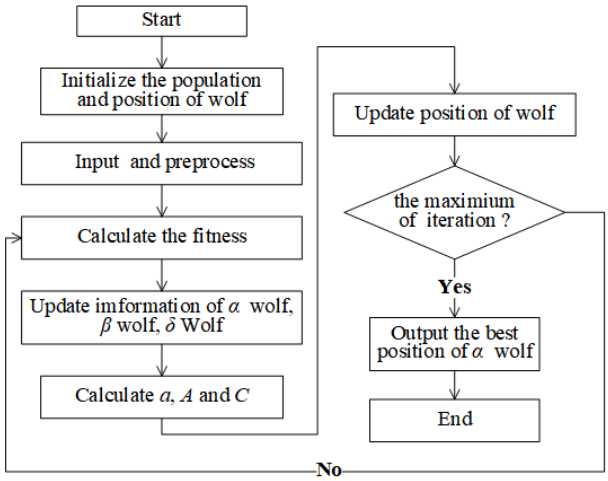


FIGURE 1. Flowchart of GWO algorithm.

B. PARTICLE SWARM OPTIMIZATION ALGORITHM

The Particle Swarm Optimization (PSO) algorithm is also named as the particle swarm algorithm. In this algorithm, particles keep searching for optimal positions in space and storing the known information. In the process of finding the optimal solution, the optimal solution is found quickly by constantly communicating with other particles, and these positions are called *pbest* (the best of the particles) and *gbest* (the global best). The iterative update process of the particle swarm is as follows: Firstly, the initial population is generated at random; then, the particles start to find *pbest* and *gbest*, and the information about the positions of the particles in the swarm is continuously stored in memory. Finally, the iterative model of position update of all particles is as follows.

$$p_j^{i+1} = p_j^i + v_j^{i+1} \tag{13}$$
$$v_j^{i+1} = v_j^i + c_1 r_1 (pbest_j - p_j^i) + c_2 r_2 (gbest_j - p_j^i) \tag{14}$$

where i is the particles in the particle swarm; j is the iteration step performed; r_1 and r_2 is a random number between 0 and 1. The coefficients c_1 and c_2 are the optimization parameters; v is the update speed; and p^i is the best position information obtained by particle i. The flowchart of PSO algorithm is shown in Figure 2.

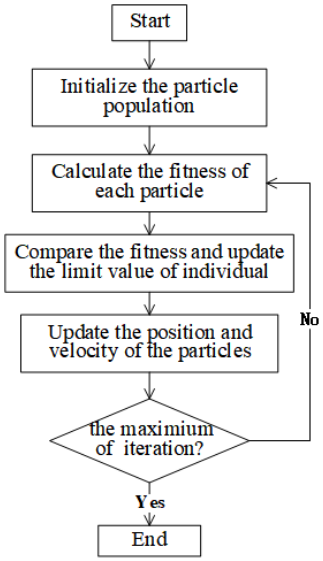


FIGURE 2. Flowchart of PSO algorithm.

IV. RESOURCE LOAD PREDICTION MODEL FOR CLOUD SERVER

A. RESOURCE LOAD MODEL

The resource load is used to reflect the current working status of the cloud server. The pressure on the processing capacity of the cloud server is increasing with the resource load value, and vice versa. The load situation of cloud server is affected by many factors. This paper mainly considers five factors on the resource load [31]: CPU utilization, memory utilization, disk space utilization, the number of incoming network packets and the number of outgoing network packets. The load value at each time point is calculated: firstly, the entropy weighting method is used to objectively assign weights to the above five factors. Then the weight of each factor is multiplied by the resource utilization rate of the factor at each moment. Finally, the resource load value of each moment is calculated by Equation (15).

$$L = \omega_1 L_1 + \omega_2 L_2 + \omega_3 L_3 + \omega_4 L_4 + \omega_5 L_5 \tag{15}$$
$$\omega_1 + \omega_2 + \omega_3 + \omega_4 + \omega_5 = 1 \tag{16}$$

where $L_1 \ldots L_5$ denote the five factors which include CPU utilization, memory utilization, disk space utilization, number of incoming network packets and number of outgoing network packets, respectively; $\omega_1 \ldots \omega_5$ is the weight of each factor. To make the weight determination more objectively, this experiment uses the entropy weighting method to assign weights to individual factors.

B. APPLICATION OF PSO AND CONVEX FUNCTION

Although the GWO algorithm has a high convergence speed in the process of finding the best, it mainly relies on the leadership of wolves α, β, and δ, and this way makes the whole wolf pack communicate little. And it is prone to problems such as premature convergence [24]. In addition, in the GWO algorithm, the convergence speed of the gray wolf should be different for each stage of its main task, so the gray wolf search for the optimum is extremely dependent on the convergence factor a. It can be seen from formula (5) that the convergence factor converges in a linear decreasing way, and the convergence speed of the whole process is constant, which may lead to the wolf pack missing the search range and premature convergence into local optimum [32]. To overcome these two shortcomings, the following approach is used for optimization in this paper.

1) USING PSO ALGORITHM TO OPTIMIZE THE POSITION UPDATE OF WOLF PACK

The PSO algorithm with the characteristics of memory and frequent communication can be used to compensate for the shortcoming of early convergence, which is caused by the low communication of wolves in the optimization process of gray wolves. According to the above PSO mathematical model, the update of the position of gray wolves can be improved as follows.

$$X_j^{i+1} = X_j^i + v_j^{i+1} \tag{17}$$

$$v_j^{i+1} = \omega(v_j^i + c_1 r_1(X_1 - X_j^i) + c_2 r_2(X_2 - X_j^i) + c_3 r_3(X_3 - X_j^i)) \tag{18}$$

$$D_\alpha = |C_1 \times X_\alpha - \omega \times X|$$

$$D_\beta = \left|C_2 \times X_\beta - \omega \times X\right|$$

$$D_\delta = |C_3 \times X_\delta - \omega \times X| \tag{19}$$

$$\omega = 0.5 + rand\ (0, 1)\ /2 \tag{20}$$

where c_1, c_2, and c_3 are optimization parameters and ω is inertia coefficient. The pseudo-code is shown in Table 1.

2) INTRODUCING CONVEX FUNCTION TO IMPROVE THE TRADITIONAL LINEAR CONVERGENCE OF GWO ALGORITHM

Considering that the wolf pack mainly conducts prey target search in the early stage, which requires global search, so the convergence speed should be slowed down; the later stage is to capture the prey, which requires local search, and the convergence speed should be accelerated to prevent the prey from escaping. Therefore, convex function is introduced in this paper to improve the convergence of the GWO algorithm as follows.

$$a = \frac{a_{\max} - a_{\min}}{e^{-1} - 1}\left(e^{-1} - e^{-\left(\frac{t}{t_{\max}}\right)^3}\right) \tag{21}$$

where $a_{\max}$ is equal to 2, which is the maximum of a; $a_{\min}$ is equal to 0, which is the minimum of a. t denotes the current iteration number; $t_{\max}$ denotes the maximum iteration number. The comparison of the convergence before and after the improvement is shown in Figure 3.

TABLE 1. Pseudocode of PSO-GWO.

```
Create population randomly
Set a small probability rate: p
Fix maximum iterations and initiate iteration count itr = 0
Run PSO for fitness evaluation of all particles
Sort and index fitness of each particle
if itr = itr_max
    Stop
    else update particle velocity and position
end if
for current particle
  if rand (0,1) < p, then set value a, A, & C (for avoiding local minima)
    else run PSO for fitness evaluation of all particles
  end if
Evaluate the fitness of all wolves
Update X_α, X_β, and X_δ

    if itr < itr_max
      Calculate X_{t+1}
      Substitute this position to PSO particles
      Run PSO
      else update the position of wolf
    end if
end for
```

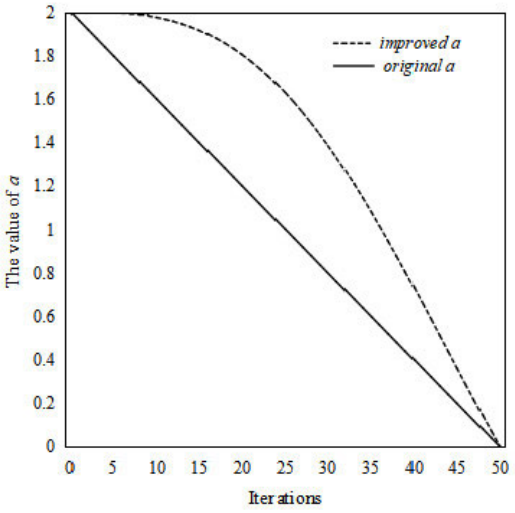


FIGURE 3. Comparison of the convergence before and after the improvement.

C. FLOWCHART OF PSO-GWO-BP MODEL

Based on the above analysis and discussion, this paper designs and proposes the PSO-GWO-BP prediction model based on PSO algorithm, GWO algorithm and BP neural network. By analyzing the GWO model, we found that the class system and leadership style of the wolf pack may cause premature convergence in the optimization process, and the

traditional convergence of this model makes it easy to mistake in the search of the wolf pack. Therefore, this paper first introduces the PSO algorithm and the convex functions to solve the above two problems respectively. The improved GWO algorithm is applied to the traditional BP neural network model as a new way of assigning weights and thresholds. Then a new cloud server resource load prediction model is constructed, which is named as the PSO-GWO-BP prediction model. The basic framework and execution of the PSO-GWO-BP model are shown in Figure 4.

Step1: Initialize the PSO-GWO-BP model, and determine the topological structure of PSO-GWO-BP.

Step2: Calculate the initial weights and thresholds of the PSO-GWO-BP model.

Step3: Call the improved GWO algorithm. The above-mentioned initial weights and thresholds are used as optimization targets.

Step4: Run the improved GWO algorithm. Firstly, relevant parameters are initialized, such as the size of wolf pack, maximum of iteration times, the range of searching range. Then the fitness value of gray wolf individuals is calculated. Finally, α Wolf, β Wolf and δ Wolf are determined by comparing the fitness values.

Step5: Calculate the convergence factor and update the position. Firstly, the convex function is used to calculate the convergence factor by formula (21). Then the values of A and C are updated according to formulas (3) and (5). Finally, the PSO algorithm is used to update the position of the gray wolf individual as shown in formulas (13) and (14).

Step6: Determine whether the termination condition (iteration times or error) is reached. If not, Step 4 and Step 5 will be repeated until the conditions are met. If yes, the position of the α wolf is output and mapped to the optimal initial weights and thresholds of the PSO-GWO-BP model.

Step7: Continue the main process of the PSO-GWO-BP model. Firstly, the training part of the PSO-GWO-BP model is executed with the above-mentioned optimal initial weights and thresholds until the training termination conditions are met. Then the result is output, and the entire execution of the PSO-GWO-BP model completes.

D. EVALUATION CRITERIA OF PREDICTION MODEL

To verify the performance of the PSO-GWO-BP resource load prediction model for the cloud server proposed in this paper, the experiment will compare SVM model, traditional BP neural network model, BAS-BP model [33], GWO-BP model [34], and SSA-BP model [35] with it on the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Mean Square Error (MSE) [36]. The performance evaluation indexes of the six models are given by formulas (22) - (24). In addition, this paper also uses formula (24) in the calculation of the fitness value.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \tag{22}$$

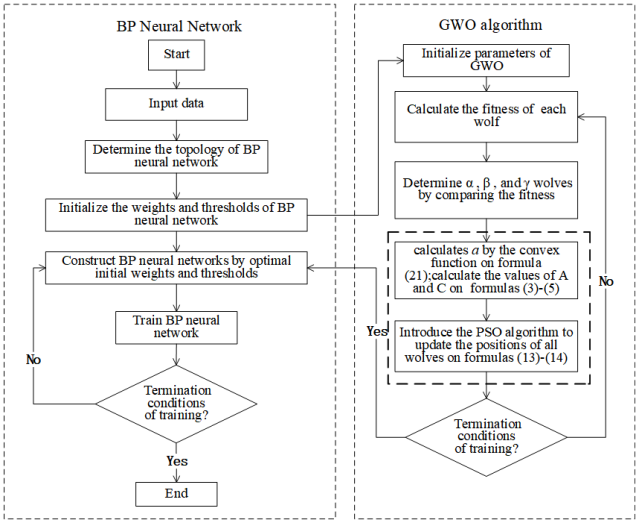


FIGURE 4. Framework diagram of PSO-GWO-BP prediction model.

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{f_i - y_i}{y_i}\right| \tag{23}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2 \tag{24}$$

In formulas (22)-(24), n is the number of samples predicted; f_i and y_i are the predicted and actual values of the resource load respectively.

V. SIMULATION EXPERIMENTS

A. EXPERIMENTAL DATA

The dataset used in this experiment is derived from the Alibaba cluster-trace-v2018 trace dataset [37]. This dataset records the resource usage of each machine at different point in time and information about the instances in the batch workload. The usage of resources such as CPU, memory, and disk space for each machine at different times of the day is used in this experiment.

B. DATA PRE-PROCESSING

This experiment randomly selects a machine in the machine usage dataset for a certain day to detect the trace data. Firstly, the five factors which include CPU utilization, memory utilization, disk space utilization, number of incoming network packets and number of outgoing network packets are used as the evaluation factors of the resource load of the cloud server. Then the entropy weighting method is used to calculate the information entropy as well as the weight of each factor, as shown in Table 2. The entropy weighting method uses the variability among data to assign weights, and the method can reflect the weights between each factor more objectively. Finally, the weight of each factor is multiplied by the actual value of each factor and summed to obtain the resource load value of the cloud server. By analyzing the original data, it can be found that the resource utilization changes within the

same minute are relatively small. Therefore, to reflect the variability of the data, this experiment samples the data in minutes and selects the peak within each minute as the load value of node each minute. Among them, the percentages of training data and test data are 80% and 20% respectively.

TABLE 2. Information entropy and weights.

	CPU	Memory	Disks	Number of incoming network packets	Number of outgoing network packets
Information entropy	2.38	2.16	2.27	2.32	2.07
Weights	0.22	0.19	0.21	0.21	0.17

This experiment takes CPU utilization, memory utilization, disk space utilization, the number of incoming network packets and outgoing network packets as the input values, and the cloud server resource load value as the output value. To avoid the possible instability of the input data and the problem of the magnitude among the data affecting the model training effect, this experiment normalizes the data so that all of them are between [−1,1]. The principle is as follows.

$$Y = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{25}$$

where X is the original input; $X_{\max}$ and $X_{\min}$ is the maximum and minimum values in each row of the original data.

C. DETERMINING STRUCTURE AND PARAMETER OF NEURAL NETWORK

Based on the number of input and output data, it can be determined that the number of neurons in the input and output layers of the network are 5 and 1 respectively. The number of neurons in the hidden layer is calculated as follows.

$$M = \sqrt{N + L} + \alpha \tag{26}$$

where M denotes the number of neurons in the hidden layer, N denotes the number of neurons in the input layer, L denotes the number of neurons in the output layer, and α is a constant which is between [1], [10].

According to formula (26), the number of neurons in the hidden layer can be determined at [4], [13]; the whole network structure includes three layers. After several training experiments, it is found that the best output is achieved when the number of neurons in the hidden layer is 11. The maximum number of iterations is 50; the training error target is 0.001; and the learning rate is 0.01.

D. ANALYSIS OF EXPERIMENTAL RESULTS

The structure and parameter settings of BAS-BP model, GWO-BP model, SSA-BP model, and PSO-GWO-BP model are consistent with the traditional BP neural network. The population numbers and iteration times of these five models are set to 20 and 50. The maximum of iteration times of SVM is set to 50, the value range of the penalty coefficient in SVM model is [0,100], and the value range of the parameter g of the kernel function is [0,100].

1) ANALYZING THE PREDICTION ACCURACY OF MODEL

To control the stability of experiment results and reduce the impact of emergencies, the error values are created by the average values of the ten experiments with SVM, BP, BAS-BP, GWO-BP, SSA-BP, and PSO-GWO-BP models, as shown in Figure 5 and Table 3. In Figure 5, error values of prediction of the above six models are compared with each of test samples. The error of the prediction results of the PSO-GWO-BP model is less than the other five models and its fluctuation isn't obvious, so PSO-GWO-BP model's prediction is more stable. In Table 3, MAE, MAPE, and MSE values of the above six models are summarized, which analyze overall errors of test samples. The MAE, MAPE and MSE values of the PSO-GWO-BP model are lower than the other five models. According to the above analysis, it can be determined that the prediction accuracy of the PSO-GWO-BP model is better than the other five models.

TABLE 3. Comparison of overall errors of prediction.

	MAE	MAPE	MSE
SVM	6.40E-04	1.33E-03	6.16E-04
BP	2.21E-02	2.87E-02	7.06E-04
BAS-BP	9.94E-03	1.26E-02	1.59E-04
GWO-BP	2.71E-03	3.53E-03	1.29E-05
SSA-BP	2.16E-03	2.83E-03	9.92E-06
PSO-GWO-BP	1.65E-03	2.13E-03	5.42E-06

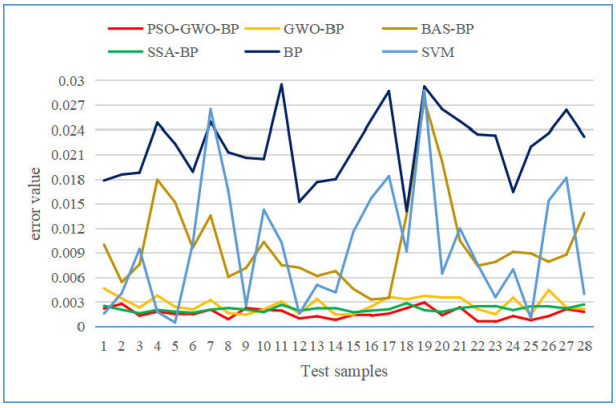


FIGURE 5. Comparison of prediction error with each group of test samples.

2) ANALYSIS OF CONVERGENCE SPEED AND STABILITY OF MODEL

BAS-BP model, GWO-BP model, SSA-BP model, and PSO-GWO-BP model are improved prediction models based on traditional BP neural network. The time consumed in the optimization process of these models determines their respective convergence speeds. Commonly, it can be judged based on the time or number of iteration while the fitness

curve reaches a plateau. In this paper, the number of iterations is used to evaluate the convergence speed of each prediction model. Meanwhile, the change trend of the fitness curve can intuitively reflect the stability of the prediction model, and then it is easy to judge whether the model falls into the local optimum during the optimization process.

As shown in Figure 6, analyzing of the convergence speed and stability of the above five models, it is indicated that the fitness value of the BAS-BP model is fluctuating, and its fluctuation range is large. So it is clear that the convergence speed of the model is too slow, and it is not stable. The fitness value of the GWO-BP model decreases rapidly in the beginning 10 iterations, and then gradually becomes flat. Compared with the fitness curve of the BAS-BP model, the GWO-BP model has better convergence speed and stability. But the convergence speed and stability of the GWO-BP model are worse compared with the SSA-BP model and the PSO-GWO-BP model. The fitness curve of the SSA-BP model tends to be flat after 5 iterations, indicating that its convergence speed is faster, and its stability is strong. However, the minimum fitness value of the model has not reached the minimum shown in Figure 6, indicating that the model may fall into a local optimum. Until 25 iterations, the fitness value of the PSO-GWO-BP model becomes very slow and stabilized, and its fitness values are lowest than the BAS-BP model, GWO-BP model, and SSA-BP model. Therefore, the convergence speed and stability of the PSO-GWO-BP model are much better than the above-mentioned models.

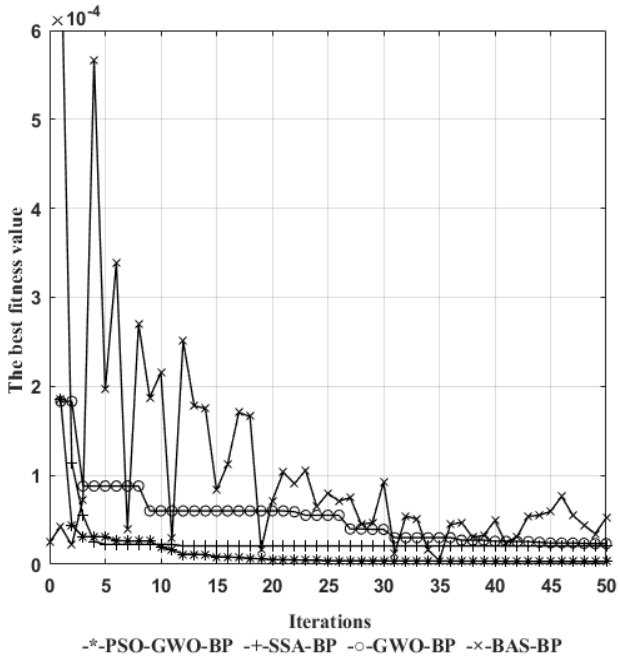


FIGURE 6. Fitness values and iteration numbers of BAS-BP, GWO-BP, SSA-BP, and PSO-GWO-BP.

In conclusion, the PSO-GWO-BP model proposed in this paper outperforms SVM model, traditional BP neural network model, BAS-BP model, GWO-BP model, and SSA-BP model in several aspects which involve prediction accuracy, convergence speed and stability.

VI. CONCLUSION

To avoid the one-sidedness of using a single indicator to reflect the resource load of cloud server, this paper adopts the entropy weighting method to calculate the weight coefficients of five factors affecting the resource load, and then calculates the expected value of the resource load. In addition, this paper uses the PSO algorithm and the GWO search algorithm for the optimization of traditional BP neural network, and designs and constructs the PSO-GWO-BP cloud server resource load prediction model. Experiments are conducted using the Alibaba-cluster-trace-2018 dataset, and through the comparative analysis of the prediction results with SVM model, traditional BP neural network model, BAS-BP model, GWO-BP model, SSA-BP model, and PSO-GWO-BP model, it is verified that the PSO-GWO-BP model proposed in this paper is much better than the above five models. Therefore, this model can predict the overall trend of cloud server resource load more accurately and can be applied to monitoring and performance optimization of cloud servers in data centers. The PSO-GWO-BP model is limited by the data acquisition method and the implementation environment of the algorithm and can only process offline data at present. In the future research work, experimental conditions will be upgraded, and then the PSO-GWO-BP model will be improved and carried out on real-time computing platform.

REFERENCES

[1] H. Zhong, Y. Fang, and J. Cui, "LBBSRT: An efficient SDN load balancing scheme based on server response time," *Future Gener. Comput. Syst.*, vol. 2015, no. 1, pp. 1–9, 2017.

[2] J. C. Zhu, L. Deng, C. J. Liang, and M. Yan, "Analysis prediction of host resource load in the cloud," *Mini-Comput. Syst.*, vol. 42, no. 3, pp. 153–167, 2021.

[3] B. Liu, J. Guo, and C. Li, "Workload forecasting based elastic resource management in edge cloud," *Comput. Ind. Eng.*, vol. 139, pp. 106–136, Jan. 2020.

[4] F. P. Tso, S. Jouet, and D. P. Pezaros, "Network and server resource management strategies for data centre infrastructures: A survey," *Comput. Netw.*, vol. 106, pp. 209–225, Sep. 2016.

[5] S. Gupta and A. D. Dileep, "Relevance feedback based online learning model for resource bottleneck prediction in cloud servers," *Neurocomputing*, vol. 402, pp. 307–322, Aug. 2020.

[6] A. D. Papalexopoulos and T. C. Hesterberg, "A regression-based approach to short-term system load forecasting," *IEEE Trans. Power Syst.*, vol. 5, no. 4, pp. 1535–1547, Nov. 1990.

[7] T.-H. Li, "A hierarchical framework for modeling and forecasting web server workload," *J. Amer. Stat. Assoc.*, vol. 100, no. 471, pp. 748–763, Sep. 2005.

[8] X. Wang and X. Y. Chen, "Research and application of load forecasting for power information system based on AHP and ARIMA algorithms," *Power Grid Clean Energy*, vol. 33, no. 8, pp. 20–25, 2017.

[9] T. Lin, J. K. Feng, Z. X. Hao, and S. Q. Huang, "Research on cloud computing resource load prediction based on combined prediction model," *Comput. Eng. Sci.*, vol. 42, no. 7, pp. 1168–1173, 2020.

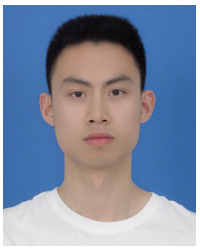
[10] T. Czernichow, A. Piras, and K. Imhof, "Short term electrical load forecasting with artificial neural networks," *Eng. Intell. Syst. Elect. Eng. Commun.*, vol. 4, pp. 85–99, Apr. 1996.

[11] L. Zhao, "Support vector machine-based cloud computing resource load prediction model," *J. Nanjing Univ. Technol.*, vol. 42, no. 6, pp. 687–692, 2018.

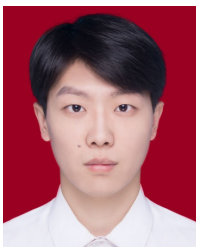
[12] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini, "Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms," in *Proc. 26th Symp. Oper. Syst. Princ.*, Oct. 2017, pp. 153–167.
[13] S. Di, D. Kondo, and W. Cirne, "Google hostload prediction based on Bayesian model with optimized feature combination," *J. Parallel Distrib. Comput.*, vol. 74, no. 1, pp. 1820–1832, Jan. 2014.
[14] H. Wang and Y. Luo, "Research and implementation of dynamic scheduling algorithm for virtual machines based on load prediction," *Comput. Eng. Sci.*, vol. 38, no. 10, pp. 1974–1979,2016.
[15] J. Xia, P. G. Zhou, P. G. Xie, Z. Li, H. Xu, and G. Wang, "A load prediction method in Femtocell networks," *J. Chongqing Univ. Posts Telecommun.*, vol. 31, no. 3, pp. 382–389, 2019.
[16] Y. K. Lai, X. Y. Chen, and H. Liu, "Research on software defect prediction based on Bayesian logistic regression," *Comput. Eng. Appl.*, vol. 55, no. 11, pp. 204–208+220,2019.
[17] M. Wang and D. X. Li, "Handwritten digit recognition based on naive Bayes and improved VGG-1," *Mod. Electron.*, vol. 43, no. 12, pp. 176–181 and 186, 2020.
[18] C. S. Zhu and S. H. Li, "Random forest regression model based on improved fruit fly optimization algorithm and its application to wind speed forecasting," *J. Lanzhou Univ. Technol.*, vol. 47, no. 4, pp. 83–90, 2021.
[19] Z. Y. Cui and X. L. Geng, "Application research of SVM algorithm based on RF and quantum particle swarm optimization," *Comput. Integr. Manuf. Syst.*, pp. 1–13, 2021. [Online]. Available: http://kns.cnki.net/kcms/detail/11.5946.TP.20210910.1824.008.html
[20] S. Gupta, A. D. Dileep, and T. A. Gonsalves, "A joint feature selection framework for multivariate resource usage prediction in cloud servers using stability and prediction performance," *J. Supercomput.*, vol. 74, no. 11, pp. 6033–6068, Nov. 2018.
[21] H. L. Dai and Y. D. Luo, "Optimized back propagation neural network model based on bat algorithm for wireless network traffic prediction," *Comput. Appl.*, vol. 41, no. S1, pp. 185–188, 2021.
[22] Q. Li, J. Chen, and B. Xu, "Research on product demand forecasting in manufacturing industry under the condition of multi-product mutual restriction," *Comput. Appl. Softw.*, vol. 38, no. 3, pp. 59–69, 2021.
[23] D. Y. Xu and S. Ding, "Research on improving GWO optimized SVM for short-term prediction of cloud computing resource load," *Comput. Eng. Appl.*, vol. 53, no. 7, pp. 68–73, 2017.
[24] L. Wen and S. Xu, "A novel grey wolf optimizer for global optimization problems," in *Proc. Adv. Inf. Manage., Commun., Electron. Autom. Control Conf.*, Dec. 2016, pp. 1266–1270.
[25] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[26] F. Men and X. Jiang, "An improved grey wolf optimization algorithm for solving the low carbon transport scheduling problem in open pit mines," *Ind. Mining Automat.*, vol. 46, no. 12, pp. 90–94, 2020.
[27] T. H. Jiang, "Hybrid grey wolf optimization algorithm for solving flexible job shop scheduling problem," *Control Decis.*, vol. 33, no. 3, pp. 503–508,2018.
[28] A. B. L. D. Medeiros, "Application of the gray wolf (GWO) algorithm in the tuning of a PID controller in a feedback control system," *J. Eng. Technol. Ind. Appl.*, vol. 3, no. 12, pp. 56–62, 2017.
[29] M. ElGayyar, E. Emary, N. H. Sweilam, and M. Abdelazeem, "A hybrid grey wolf-bat algorithm for global optimization," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl.*, 2018, pp. 3–12.
[30] M. A. Tawhid and A. F. Ali, "A Hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function," *Memetic Comput.*, vol. 9, no. 4, pp. 1–13, 2017.
[31] E. H. Liao, N. Shu, J. W. Li, and H. W. Pang, "A cloud server performance prediction model based on temporal convolutional networks," *J. South China Normal Univ.*, vol. 52, no. 4, pp. 107–113, 2020.
[32] M. F. Hassanin, A. M. Shoeb, and A. E. Hassanien, "Grey wolf optimizer-based back-propagation neural network algorithm," in *Proc. 12th Int. Comput. Eng. Conf. (ICENCO)*, Dec. 2016, pp. 213–218.
[33] S. Q. Jiang and Q. Liu, "The BP neural network optimized by Beetle Antenna Search Algorithm for storm surge prediction," in *Proc. 30th Int. Ocean Polar Eng. Conf.*, 2020, pp. 2725–2729.
[34] G. Li, T. Hu, and D. Bai, "BP neural network improved by sparrow search algorithm in predicting debonding strain of FRP-strengthened RC beams," *Adv. Civil Eng.*, vol. 2021, pp. 1–13, May 2021.
[35] W. D. Zhou and Y. Q. Xia, "Virtual machine performance prediction using broad learning system based on Compression Factor," *Acta Automatica Sinica.*, vol. 45, no. 6, pp. 1–11, 2021.
[36] S. Padhy, S. Panda, and S. Mahapatra, "A modified GWO technique based cascade PI-PD controller for AGC of power systems in presence of plug in electric vehicles," *Eng. Sci. Technol., Int. J.*, vol. 20, no. 2, pp. 427–442, Apr. 2017.
[37] Alibaba. *Cluster Data Collected From Production Clusters in Alibaba for Cluster Management Research*. Accessed: Dec. 13, 2019. [Online]. Available: https://github.com/alibaba/clusterdata/tree/master/cluster-trace-v2018

KE HOU was born in Jinan, Shandong, China, in 1977. He received the B.S. degree in metal material and heat treatment, the M.S. degree in weapon systems and utilization engineering, and the Ph.D. degree in mechanical engineering from Xi'an Technological University, Xi'an, China, in 2000, 2005, and 2016, respectively.

From 2007 to 2015, he was a Lecturer with Xi'an Shiyou University, Xi'an, where he has been an Assistant Professor, since 2016. From 2018 to 2019, he was a Visiting Scholar with the College of Engineering, University of Louisiana at Lafayette, USA. His research interests include machine learning, data-intensive computing, and digital image processing.



MINGCHENG GUO was born in Deyang, Sichuan, China, in 1996. He received the B.S. degree in logistics management from Mianyang Normal University, China, in 2019. He is currently pursuing the M.S. degree in management science and engineering with Xi'an Shiyou University, China. His research interests include algorithm optimization, machine learning, and big data analysis.



XINHAO LI was born in Luoyang, Henan, China, in 1998. He received the B.S. degree in information management and information system from Xi'an Shiyou University, China, in 2020, where he is currently pursuing the M.S. degree in management science and engineering.

His research interests include big data and intelligent system based on cloud platform.

HE ZHANG received the B.S. degree in petroleum engineering from the China University of Petroleum, in 2008, the master's degree in mathematics from Texas A&M University-Commerce, USA, in 2016, and the master's degree in petroleum engineering from the University of Louisiana at Lafayette, USA, in 2019, where he is currently pursuing the Ph.D. degree. He is also a Researcher with the University of Louisiana at Lafayette. He was also a Drilling Supervisor with six years of experience in engineering at China National Offshore Oil Corporation. His research interests include rock mechanics, hydraulic fracturing, numerical simulation, machine learning, and deep learning.