A Cloud Load Forecasting Model with Nonlinear Changes using Whale Optimization Algorithm Hybrid Strategy - extreme Learning Machine

Hua Peng
Sun Yat-Sen University
Wu-Shao Wen
Sun Yat-Sen University
Ming-Lang Tseng (tsengminglang@gmail.com)
Asia University
Ling-Ling Li
Hebei University of Technology

Research Article

Keywords: cloud load prediction, artificial intelligence optimizer, extreme learning machine model, hybrid optimization strategy, efficient resource management

Posted Date: April 19th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-412160/v1

A cloud load forecasting model with nonlinear changes using whale optimization algorithm hybrid strategy - extreme learning machine

Authorship

Hua Peng

- School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

E-mail: pengh27@mail2.sysu.edu.cn

Wu-Shao Wen

- School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

E-mail: wenwsh@mail.sysu.edu.cn

Ming-Lang Tseng * (Corresponding author)

- Institute of Innovation and Circular Economy, Asia University, Taiwan;
- Department of medical, China medical university Hospital, China medical university, Taiwan
- Faculty of Economics and Management, Universiti Kebangsaan Malaysia, Malaysia

*E-mail: tsengminglang@asia.edu.tw or tsengminglang@gmail.com

Ling-Ling Li

- State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300130, China

E-mail: lilingling@hebut.edu.cn

A cloud load forecasting model with nonlinear changes using whale optimization algorithm hybrid strategy - extreme learning machine

Abstract

This study proposes a novel cloud load prediction model and combines hybrid whale optimizer (HWOA) and extreme learning machine (ELM) together for strong nonlinear mapping ability. Accurate cloud load prediction improves the cloud service efficiency and serves as the foundation for network scheme due to traditional linear forecasting models are unable to predict cloud computing resources with nonlinear changes on massive multiplication and cloud computing data complexity, effectively. The proposed cloud load forecasting model is to employ HWOA optimizer to optimize the ELM model random parameters. The contributions of this study are as follows. (1) the HWOA optimizer is to solve the whale optimizer local extremum problem; (2) the proposed HWOA optimizer reduces the ELM random parameters on cloud load forecasting; (3) the convergence performance verifies the benchmark testing functions; and (4) three simulation experiments are conducted to test the cloud load forecast effect. The result indicated that the convergence analysis reveals the HWOA optimizer outperforms the prior optimizers. The proposed cloud load prediction model obtains better forecasting results. The mean absolute percentage error and root mean square error of the proposed model are less than 14% and 11, respectively. Accurate cloud load forecasting lays a foundation for effective deployment of cloud computing resources and maximization of economic benefits.

Keywords: cloud load prediction; artificial intelligence optimizer; extreme learning machine model; hybrid optimization strategy; efficient resource management

A cloud load forecasting model with nonlinear changes using whale optimization algorithm hybrid strategy - extreme learning machine

1. Introduction

Traditional data processing models have fallen into a bottleneck under massive, high-dimensional data with the rise of cloud computing technology (Rafique et al., 2021; Meenakshi et al., 2019). Cloud computing is applied to process massive data in a short time on the distributed parallel computing basis, and has strong network service ability (Kim and Jeong, 2017). Achieving efficient resource management has always been the most concerned issue for cloud service operators and providers (Khalilpourazari and Khalilpourazary, 2018; Ye et al., 2019; Mehrabi et al., 2021). Cloud load prediction is necessary for efficient network regulation in the cloud environment (Xu et al., 2017). More precise forecasting of cloud load realizes more resource utilization and faster service response to bring better energy efficiency and high-quality service. Meanwhile, accurate cloud load forecasting lays a foundation for effective deployment of cloud computing resources and maximization of economic benefits. This study argues on an accurate approach to cloud loading due to lack of stronger predictive performance to develop a cloud load forecasting model.

Prior studies have been formulated or improved a variety of cloud load prediction models (Ros et al., 2014; Juszczyk et al., 2019; Kumar et al., 2021). Optimization achievements are applied in various fields in cloud load prediction (Safavi et al., 2021; Mahmud et al., 2021). Yet, these forecasting models have focused on three categories such as single time series models, multiple time series models and machine learning models. The common prediction models on the basis of single time series mostly include moving average model (MA), autoregressive model (AR) and autoregressive average moving model (ARMA) (Cao et al., 2014; Moreno et al., 2020). The single time series model is to transform non-stationary series into stationary series and the time series must be continuous. Still, the data needs to be preprocessed before calculation and the calculation process of time series prediction model is complicated and not suitable in the situation of data loss. Usually, the machine learning prediction models consist of support vector machine (SVM), neural network and Gaussian process regression (Taghizadeh-Mehrjardi et al., 2021; Yousri et al., 2019). This study argues that although cloud load data has a large number of proliferation and complexity, and machine learning model can effectively predict cloud load resources with nonlinear changes.

In addition, Parand et al. (2021) argued that machine learning models had better generalization ability and mapping ability compared with the time series prediction model. Yang et al. (2014) designed a cloud service architecture and employed linear regression method to forecast the cloud load to improve the cloud computing scalability. The model prediction accuracy needs to be further improved due to the strong nonlinearity and time-varying of cloud load. Jiang et al. (2018) and Li et al. (2018) applied the ARMA model to predict cloud load. ARMA model has higher prediction accuracy but its parameter estimation is relatively complex compared with AR and MA models. Autoregressive integrated moving average (ARIMA) model is superior to ARMA model in predicting unstable time series. ARIMA introduces difference operation in the calculation process, and is widely used for cloud load forecasting (Calheiros et al., 2015; Barati and Sharifian, 2015). These studies are lacking on considering the applicability of time series models to the strong nonlinearity of

cloud load. This study employed ELM model with stronger nonlinear mapping ability to predict cloud load.

In this study, the extreme learning machine (ELM) model is employed to forecast the cloud load due to its stronger nonlinear mapping ability and generalization ability (de Franca et al., 2021; Choudhary et al., 2021). ELM model does not use gradient descent method to update the random parameters and reduces the training time compared with the traditional neural network models. The topology of the ELM model is simpler and unlike the long short term memory network (LSTM) model. The ELM algorithm has the advantages of low complexity and fast convergence compared with other similar algorithms. Prior studies on ELM have never ceased since the ELM algorithm was proposed (Chia et al., 2021). Liu et al. (2020) argued that the random parameters of ELM model affected the final forecasting accuracy. Therefore, this study proposes that the whale optimization algorithm (WOA) is improved and the WOA optimizer based on a hybrid strategy (HWOA) is proposed to determine the ELM model's hyper-parameters. The proposed HWOA-ELM cloud load predictive model is employed to predict the cloud load. The objectives of this study are as follows.

- Develop a new method on the basis of WOA optimizer.
- Use the proposed optimizer to address the influence of random parameters on Elm prediction results.
- Propose a new model to predict cloud load.

This study has four contributions as follows: (1) the HWOA optimizer is proposed and experimentally demonstrated to have a better optimization performance; (2) the proposed HWOA-ELM model is constructed to forecast the cloud load and is verified to achieve a better performance on the basis of evaluation indexes; (3) simulation experiments demonstrate that the number of testing samples and training samples has a great influence on the forecasting effect of the model; and (4) intelligent optimization algorithm is combined with machine learning model to provide an effective and accurate prediction to reinforce effectively the availability and economy in the cloud environment.

The rest of the study is structured as follows. Section 2 presents the modeling process of the cloud load prediction. Section 3 uses simulation experiments to verify the proposed predictive model. Section 4 presents the concluding remarks.

2. Literature review

Machine learning models attract increasing attentions. LSTM is applied to predict the fluctuating cloud load due to the good nonlinearity (Choudhary et al., 2021; Parand et al., 2021; Xu et al., 2017). For instance, Gupta et al. (2020) improved the LSTM to predict the online cloud load, and introduced the gradient descent method into the forecast model; however, the result ignores that the gradient descent method requires more iterations to correct the model parameters and requires longer training time. You et al. (2020) decomposed the cloud load time series to remove the sequence noise, and reconstructed the cloud load sequence. The denoised cloud load is predicted on the basis of the LSTM model. However, You et al. (2020) ignored to address the problem of signal loss during the denoising process. Kumar et al. (2021) proposed a self-directed cloud load prediction method, which combined a multilayer neural network with a developed heuristic optimization algorithm. The developed heuristic algorithm was applied to optimize the weights of the neural network (Khalilpourazari and Khalilpourazary, 2018; Rafique et al.,

2021). However, this method lacked to consider the influence of the number of training samples on the prediction results.

In addition, Zhao et al. (2018) used the improved SVM model with a better nonlinear mapping ability to forecast the cloud load and the cloud load time series was initially preprocessed through the chaotic analysis. The improved SVM model reliability is verified by simulation experiments (Calheiros et al., 2015; Mehrabi et al., 2021; Yang et al., 2014). SVM model has strong nonlinear mapping ability, but it is sensitive to the number of samples and only suitable for solving relatively small samples. Barati and Sharifian (2015) proposed the tuned support vector regression (SVR) and a hybrid model combined genetic algorithm with particle swarm algorithm. The combined intelligent algorithm is utilized to optimize the parameters of the SVR, but Barati and Sharifian (2015) ignored to analyze the sensitivity of SVR to kernel functions.

Moreover, neural network models are applied to forecast cloud load (Cao et al., 2014; Moreno et al., 2020; Tofighy et al., 2018). For instance, Chen et al. (2015) constructed a fuzzy neural network prediction model to forecast cloud resource requirements and proposed a novel method to determine the number of fuzzy rules. Demand characters of various users are firstly analyzed and then the predictive model is constructed according to the user requirements. Fuzzy neural network combines fuzzy logic reasoning with neural network, so it has strong robustness and fault tolerance. Cao et al. (2014) developed a novel ensemble model for dynamic load prediction considering that there was correlation among different resources. Tofighy et al. (2018) built a cloud resource management framework and used two filters to reduce the negative impact of outliers. The cloud load prediction model in this framework uses the Bayesian model based on probabilities, advantageous in terms of simplicity and speed. Xu et al. (2013) classified cloud load before prediction and then employed genetic algorithm to optimize Elman network to predict cloud load. The proposed method achieves better prediction results, but ignores the relatively high calculation cost.

In short, ELM with shorter training time and stronger mapping ability in this study is employed to forecast cloud load and the proposed HWOA optimizer is used to optimize the ELM random parameters (Chia et al., 2021; Liu et al., 2020; Xu et al., 2017). This study proposes the standard test functions with HWOA algorithm convergence performance and confirms the convergence performance outperforms other models. The proposed HWOA-ELM model needs to verify through three sets of simulation experiments to obtain better prediction cloud load results.

3. The model

3.1. Extreme learning machine model

Single hidden layer feedforward neural network such as back propagation (BP) neural network is used in various fields due to its excellent nonlinear mapping ability. There are several inherent shortcomings of feedforward neural network. Traditional neural network models adopt gradient descent method, thereby causing slow training speed and limiting the development of the model (He et al., 2020; Han et al., 2021). The ELM model is developed to have a faster training speed and less parameter setting compared with the existing network models to solve shortcomings of feedforward neural network (Alencar et al., 2016). The ELM model strong mapping ability and well generalization performance is to use in pattern recognition, fault recognition and time series prediction.

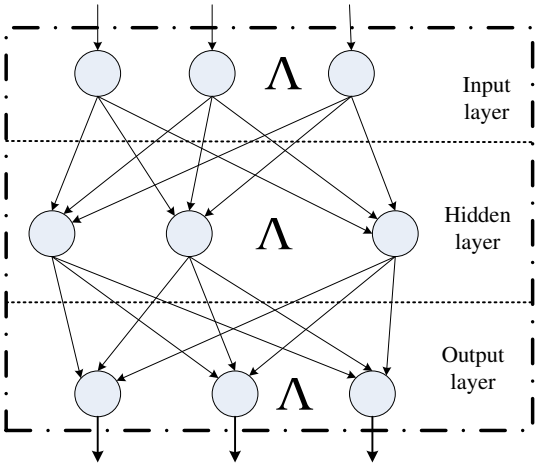


Figure 1. ELM model

Figure 1 indicated the ELM model consists of three layers of networks, the input, hidden and output layers. The three layers are connected by neurons. For ELM model, the number of neurons in each layer is *u*, *r*, and *e*. The weight coefficient between the hidden layer and the input layer is *S*. The weight coefficient between the hidden layer and the output layer is *P*, θ is the hidden layer neuron threshold (Huang et al., 2011; Ali et al., 2020; Wu et al., 2021).

$$\boldsymbol{S}=\begin{bmatrix} s_{11} & s_{12} & \Lambda & s_{1u} \\ s_{21} & s_{22} & \Lambda & s_{2u} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ s_{r1} & s_{r2} & \Lambda & s_{ru} \end{bmatrix}_{r\times u} \tag{1}$$

$$\boldsymbol{P}=\begin{bmatrix} p_{11} & p_{12} & \Lambda & p_{1e} \\ p_{21} & p_{22} & \Lambda & p_{2e} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ p_{r1} & p_{r2} & \Lambda & p_{re} \end{bmatrix}_{r\times e} \tag{2}$$

$$\boldsymbol{\theta}=\begin{bmatrix} \theta_1 \\ \theta_2 \\ \mathrm{M} \\ \theta_r \end{bmatrix}_{r\times 1} \tag{3}$$

The ELM model has M training sample sets. ***A*** is the input matrix and ***B*** is and output matrix (Li et al., 2019; Liu et al., 2020).

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & \Lambda & a_{1M} \\ a_{21} & a_{22} & \Lambda & a_{2M} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ a_{u1} & a_{u2} & \Lambda & a_{uM} \end{bmatrix}_{u\times M} \quad \boldsymbol{B}=\begin{bmatrix} b_{11} & b_{12} & \Lambda & b_{1M} \\ b_{21} & b_{22} & \Lambda & b_{2M} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ b_{e1} & b_{e2} & \Lambda & b_{eM} \end{bmatrix}_{e\times M} \tag{4}$$

Suppose that the activation function of ELM model is $y(\cdot)$ and the output is ***F***.

$$\boldsymbol{F}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\mathrm{K},\boldsymbol{f}_M] \tag{5}$$

$$\boldsymbol{f}_j = \begin{bmatrix} f_{1j} \\ f_{2j} \\ \mathrm{M} \\ f_{ej} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{r} p_{i1} y(\boldsymbol{s}_i \boldsymbol{a}_j + \theta_i) \\ \sum_{i=1}^{r} p_{i2} y(\boldsymbol{s}_i \boldsymbol{a}_j + \theta_i) \\ \mathrm{M} \\ \sum_{i=1}^{r} p_{ie} y(\boldsymbol{s}_i \boldsymbol{a}_j + \theta_i) \end{bmatrix} (j = 1, 2, \Lambda\ , M) \tag{6}$$

where $\boldsymbol{s}_i = [s_{i1}, s_{i2}, \Lambda\ , s_{iu}]$, $\boldsymbol{x}_i = [x_{1j}, x_{2j}, \Lambda\ , x_{uj}]^T$.

Equation (5) is expressed by equation (7).

$$\boldsymbol{F}^T = \boldsymbol{L}\boldsymbol{P} \tag{7}$$

$$\boldsymbol{L}(\boldsymbol{s}_1, \boldsymbol{s}_2, \Lambda\ , \boldsymbol{s}_r, \theta_1, \theta_2, \Lambda\ , \theta_r, \boldsymbol{a}_1, \boldsymbol{a}_2, \Lambda\ , \boldsymbol{a}_M) =$$

$$\begin{bmatrix} y(\boldsymbol{s}_1 + \boldsymbol{a}_1 + b_1) & y(\boldsymbol{s}_2 + \boldsymbol{a}_1 + b_2) & \Lambda & y(\boldsymbol{s}_r + \boldsymbol{a}_1 + b_r) \\ y(\boldsymbol{s}_1 + \boldsymbol{a}_2 + b_1) & y(\boldsymbol{s}_2 + \boldsymbol{a}_2 + b_2) & \Lambda & y(\boldsymbol{s}_r + \boldsymbol{a}_2 + b_r) \\ \mathrm{M} & \mathrm{M} & \Lambda & \mathrm{M} \\ y(\boldsymbol{s}_1 + \boldsymbol{a}_M + b_1) & y(\boldsymbol{s}_2 + \boldsymbol{a}_M + b_2) & \Lambda & y(\boldsymbol{s}_r + \boldsymbol{a}_M + b_r) \end{bmatrix} \tag{8}$$

where $\boldsymbol{L}$ is the hidden layer output matrix, $\boldsymbol{F}^T$ is the transposed matrix of $\boldsymbol{F}$.

In particular, when y is infinitely divisible, $\boldsymbol{s}$ and θ is constant during the training process. Meanwhile, the only unknown parameter is $\boldsymbol{P}$. The parameter $\boldsymbol{P}$ is solved by finding the least squares solution in problem equation (9).

$$\min_{P} \left\| \boldsymbol{L}\boldsymbol{P} - \boldsymbol{F}^T \right\| \tag{9}$$

From problem eqaution (9), this study obtains $\hat{\boldsymbol{P}} = \boldsymbol{L}^{+}\boldsymbol{F}^T$, where $\boldsymbol{L}^{+}$ is the generalized inverse matrix.

3.2. Basic principles of whale optimizer

Mirjalili and Lewis (2016) proposed the whale optimization algorithm (WOA). WOA optimizer is a type of swarm intelligence optimizer, which simulates the hunting strategy of humpback whales (Mafarja et al., 2017; Chia et al., 2021). The process of WOA optimizer is divided into three stages: foraging and encircling, random search and bubble-net attacking (Tikhamarine et al., 2020; Aziz et al., 2017).

(1) Foraging and encircling stage

In this process, whale groups share the location information of prey with each other, and finally they surround the prey. This study assumes that the current optimal individual of the whale group is the target location (the optimal individual is closest to the prey), and other individuals update their locations and try to approach the optimal individual location (Emary et al., 2019; Yousri et al., 2019). The position updating equation of the whale is as follows:

$$\boldsymbol{pos}(t+1) = \boldsymbol{pos}^{\wedge}(t) - \boldsymbol{K} \bullet \boldsymbol{Q} \tag{10}$$

$$\boldsymbol{Q} = \left| \boldsymbol{J} \bullet \boldsymbol{pos}^{\wedge}(t) - \boldsymbol{pos}(t) \right| \tag{11}$$

where $\boldsymbol{pos}(t)$ is the current whale individual position, t is the current number of iterations, $\boldsymbol{pos}^{\wedge}(t)$ is the optimal whale individual position, $\boldsymbol{K}$ and $\boldsymbol{Q}$ are the coefficient vector.

$$\boldsymbol{K} = 2\boldsymbol{k} \bullet \boldsymbol{n} - \boldsymbol{k} \tag{12}$$

$$\boldsymbol{J} = 2\boldsymbol{n} \tag{13}$$

$$\boldsymbol{k} = 2 - (2t) / MAX_{itera} \tag{14}$$

where $\boldsymbol{n}$ is the random vector in [0, 1], $\boldsymbol{k}$ decreases from 2 to 0 in the iteration process.

(2) Random search stage

The current individual whale position is randomly selected as the optimal solution in this stage. In contrast to the prior process, the position of the whale is updated according to a

randomly selected whale instead of the optimal whale individual found so far. Adjusting the value of ***K*** makes other individuals move far away from the selected whale. The mathematical model is as follows (Mafarja et al., 2017):

$$\boldsymbol{pos}(t+1) = \boldsymbol{pos}_{rand}(t) - \boldsymbol{K} \bullet \boldsymbol{Q} \tag{15}$$

$$\boldsymbol{Q} = \left| \boldsymbol{J} \bullet \boldsymbol{pos}_{rand}(t) - \boldsymbol{pos}(t) \right| \tag{16}$$

where $\boldsymbol{pos}_{rand}(t)$ is the location of the random whale. The optimal individual is selected as the solution to achieve the local optimization of the WOA optimizer when $|\boldsymbol{K}| \le 1$. When $|\boldsymbol{K}| > 1$, the individual is randomly selected as the solution to achieve the global optimization of the WOA optimizer.

(3) Bubble-net attacking stage

In this stage, whales have two behaviors: shrinking encircling and spiraling. The encircling motion of the whale is formulized in equation (10), achieved by reducing ***k*** in equation (12). The spiral function is used to mimic the helix-shaped behavior of whales. The choice of the two behaviors is determined by the random number *v* to realize shrinking encircling and along a spiral-shaped path simultaneously. The location of individual whales is updated as follows:

$$\boldsymbol{pos}(t+1) = \begin{cases} \hat{\boldsymbol{pos}}(t) - \boldsymbol{K} \bullet \boldsymbol{Q} & v < 0.5 \\ \hat{\boldsymbol{pos}}(t) + \boldsymbol{C} \bullet e^{o*l} \bullet \cos(2\pi l) & v \ge 0.5 \end{cases} \tag{17}$$

where $\boldsymbol{C} = \left| \hat{\boldsymbol{pos}}(t) - \boldsymbol{pos}(t) \right|$ is the distance between the whale individual and the optimal individual obtained so far, *o* is the constant to define the spiral shape, $l \in [-1,\ 1]$.

3.3. The whale optimization algorithm based on hybrid strategy

The WOA optimizer has the advantage of its simplicity and fast calculation speed. However, when solving the high dimensionality problems, the WOA optimizer does not easily jump out of the minimum value, thus limiting the development of the WOA optimizer. A corresponding solution for this situation is to provide the improvements to enhance the local optimization and global optimizer optimization ability.

(1) Initialize the whale population using the Levy strategy

Population initialization has an important impact on the optimization process of the optimizer. A higher quality initial solution can speed up the optimizer's convergence. The Levy strategy is employed to initialize the population in this paper.

The random walk mode of Levy strategy expands the search range of the whale group, which avoids whale individual falling into local extremum effectively. Hence, it helps the WOA optimizer have better convergence performance (Edwards et al., 2007; Santos et al., 2021). The mathematical model of the Levy strategy is as follows:

The random step size of the Levy strategy proposed by Mantegna is as follows (Jensi et al., 2016; Hakli et al., 2014):

$$\lambda = \frac{\mu}{|\eta|^{\frac{1}{\nu}}} \ (0 < \nu < 2) \tag{18}$$

where λ is a random step size, μ and η follow a normal distribution as shown in equation (19).

$$\begin{cases} \mu \sim N(0, \delta_\mu^2) \\ \eta \sim N(0, \delta_\eta^2) \end{cases} \tag{19}$$

The definitions of δ_μ and δ_η are as follows:

$$\begin{cases} \delta_\mu = \left\{ \dfrac{\Gamma(1+v)\sin(\pi v/2)}{\Gamma[(1+v)/2]2^{(v-1)/2}} \right\}^{\frac{1}{v}} \\ \delta_\eta = 1 \end{cases} \tag{20}$$

where Γ(·) is a standard Gamma function.

The initialized position based on the Levy strategy is as follows:

$$\boldsymbol{pos}(t) = \boldsymbol{L}_b + (\boldsymbol{U}_b - \boldsymbol{L}_b) \oplus \boldsymbol{Levy}(v) \tag{21}$$

where $\boldsymbol{L}_b$ is the lower bound, $\boldsymbol{U}_b$ is the upper bound, *Levy*(*v*) is the random vector of the step size obeying the Levy distribution.

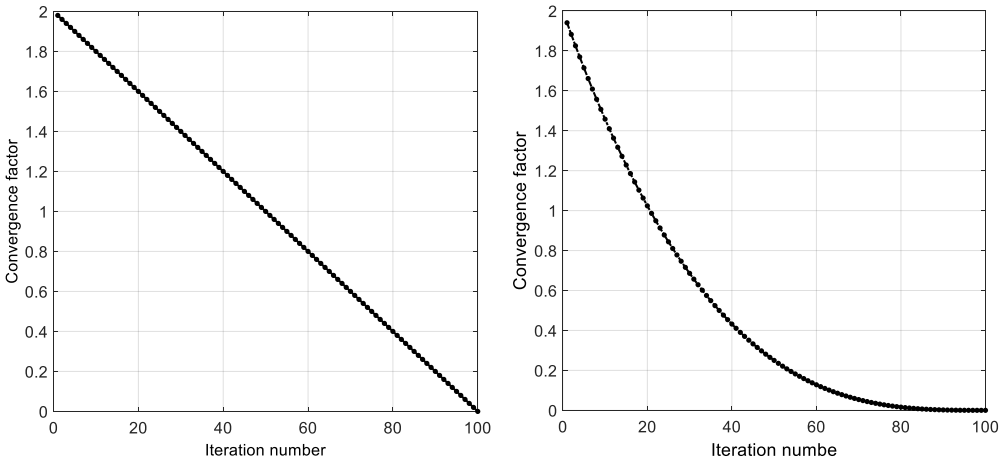
(2) Nonlinear convergence factor

Figure 2 (a) indicates the linear convergence factor adopted in equation (14). In the foraging and encircling stage, the convergence rate of WOA optimizer is the same in the early and late stages, which reduces the convergence rate of whales (Li et al., 2019). Hence, the nonlinear convergence factor is introduced into WOA optimizer to solve this problem. The nonlinear convergence factor is shown in equation (22).

$$a = 2 * (\frac{Max_{iter} - t}{Max_{iter}})^3 \tag{22}$$

where Max_{iter} is the maximum number of iterations.

Figure 2 (b) showed that the convergence factor decreases nonlinearly as the number of iterations increases. In the early stage of iteration, the attenuation speed of convergence factor is faster. The search step of whale group is larger to enhance the global optimization ability of WOA optimizer. In the late stage, the attenuation speed of convergence factor becomes slow, so that the search step of whale group is reduced to enhance the local search ability of WOA optimizer. The global and local search ability of whale group is more balanced through the nonlinear convergence factor.



(a) Linear convergence factor (b) Nonlinear convergence factor

Figure 2. Convergence factor

(3) Migration strategy

The migration strategy is introduced to the WOA optimizer, which increases the diversity of the whale group and enhances the ability of whales to jump out of the minimum. In terms of the migration frequency *Mig* (*Mig*=5), whales migrate to another place for food regularly. This regular migration is that whales have better ability to avoid falling into local minima and

the diversity of whales are enhanced. The updated location of whale groups after migration is as follows.

$$\boldsymbol{pos}(t+1) = B * \boldsymbol{pos}(t) + \boldsymbol{pos}(t) * randn(0, \sigma^2) \tag{23}$$

$$B = B_{\min} + (B_{\max} - B_{\min}) * \cos(2\pi \frac{t}{Max_{iter}}) \tag{24}$$

where B ($B_{\max}$=0.9, $B_{\min}$=0.4) is the convergence coefficient, *randn* follows the Gaussian distribution.

The pseudo code of the HWOA is as follows:

HWOA algorithm

Start
Initialize whale swarm parameters
Initialize the whale swarm according to *Levy* strategy
$\boldsymbol{pos}(t) = \boldsymbol{L}_b + (\boldsymbol{U}_b - \boldsymbol{L}_b) \oplus \boldsymbol{Levy}(v)$;
Calculation of whale individual fitness;
While ($t < Max_{iter}$)
If ($\mathrm{mod}(t,5) \neq 0$)
  If ($p < 0.5$)
    If ($|K| \leq 1$)
      The whale group is in the stage of foraging and encircling.
      Update location: $\boldsymbol{pos}(t+1) = \boldsymbol{pos}^{\wedge}(t) - \boldsymbol{K} \bullet \boldsymbol{Q}$;
    else if ($|K| > 1$)
      The whale group is in the random search stage.
      Update location: $\boldsymbol{pos}(t+1) = \boldsymbol{pos}_{rand}(t) - \boldsymbol{K} \bullet \boldsymbol{Q}$;
    end
  else if ($p \geq 0.5$)
      The whale population is in the Bubble-net attacking stage.
      Update location: $\boldsymbol{pos}(t+1) = \boldsymbol{pos}^{\wedge}(t) + \boldsymbol{C} \bullet e^{o*l} \bullet \cos(2\pi l)$;
  end
    Calculating fitness values;
else
    Whale migration: $\boldsymbol{pos}(t+1) = B * \boldsymbol{pos}(t) + \boldsymbol{pos}(t) * randn(0, \sigma^2)$;
end
  The optimal position is updated;
  t=t+1;
end

2.4. HWOA optimizer convergence test simulation

The convergence effect of the HWOA optimizer is verified in this section by benchmark testing functions. The specific expressions of the functions, range, as well as the optimal value are presented in Table 1. The optimal values of the four test functions are all 0.

Table 1. Benchmark testing functions

Function	Range	Optimum
$f_1 = \sum_{j=1}^{D} x_j^2$	$-100 \leq x \leq 100$	0

$f_2 = \prod_{j=1}^{D} \lvert x_j \rvert + \sum_{j=1}^{D} \lvert x_j \rvert$	$-10 \le x \le 10$	0
$f_3 = \sum_{j=1}^{D} j * x_j^4 + random[0,1)$	$-1.28 \le x \le 1.28$	0
$f_4 = \sum_{j=1}^{D} (x_j^2 - 10 * \cos(2 \times \pi \times x_j)) + 10 * D$	$-5.12 \le x \le 5.12$	0

Ant lion optimizer (ALO) through imitating ant lion's behavior of catching. Askarzadeh (2016) developed crow search algorithm (CSA) by analyzing crow's intelligent behavior. WOA, ALO, CSA and HWOA are tested respectively by four benchmark testing functions. Each simulation is performed on the unified platform (Intel Core i5 processor, 8GB RAM, Windows 10 and MATLAB R2016a) to test the convergence performance of these optimizers. The test dimension of the benchmark testing functions is 30, and each function tests respectively each optimizer 15 times. The number of iterations of optimizer is 500 and the population of optimizer is 30. Table 2 showed the optimization results of four optimizers.

Table 2. Optimization results

Optimization function	Algorithm	Worst optimization value	Best optimization value	Average optimization value
f_1	ALO	0.53e-03	2.53e-04	1.30e-03
	CSA	13.77	3.27	8.04
	WOA	2.26e-74	2.57e-87	1.50e-75
	HWOA	0	0	0
f_2	ALO	124.71	2.51	50.26
	CSA	4.96	1.69	3.54
	WOA	3.78e-48	1.59e-55	2.54e-49
	HWOA	3.14e-216	9.42e-231	2.54e-217
f_3	ALO	0.43	0.16	0.26
	CSA	0.06	0.03	0.04
	WOA	0.82e-02	2.03e-04	2.90e-02
	HWOA	5.96e-04	6.58e-06	1.43e-04
f_4	ALO	120.38	50.74	84.97
	CSA	56.41	17.62	28.67
	WOA	1.13e-13	0	7.57e-15
	HWOA	0	0	0

Table 2 revealed that the convergence results of the HWOA optimizer were more competitive than those of other optimizers. The test results demonstrated that the HWOA optimizer had high convergence accuracy and a strong ability to avoid local extremes. For f_1 and f_4, the proposed HWOA converged to 0. f_1 is a unimodal function, and f_4 is a multimodal function. Unimodal and multimodal functions respectively test the convergence performance and the ability to avoid local extremum. For f_2, HWOA optimizer converged to

9.42e-231, which was better than WOA, ALO and CSA algorithms. For f_3, HWOA algorithm converged to 6.58e-06, which was smaller than WOA, ALO and CSA optimizers. HWOA did not converge to 0 for f_2 and f_3, but the convergence accuracy was more satisfactory than WOA, ALO and CSA optimizers' results.

The HWOA optimizer nonlinear convergence factor enhances the global and local searching ability of the whale group. At the same time, the HWOA optimizer periodic migration strategy increases the whale population diversity and makes HWOA optimizer more minimum jumping out capable. HWOA optimizer outperforms WOA optimizer.

Figure 3 presented that the HWOA optimizer had a strong convergence performance for the four test functions. The HWOA optimizer convergence curve is reached the optimal value faster compared with WOA optimizer. The optimizer had higher convergence efficiency compared with WOA optimizer. Therefore, this study adopts the proposed HWOA optimizer with stronger convergence performance to reduce the influence of random parameters on the prediction effect of the ELM model.

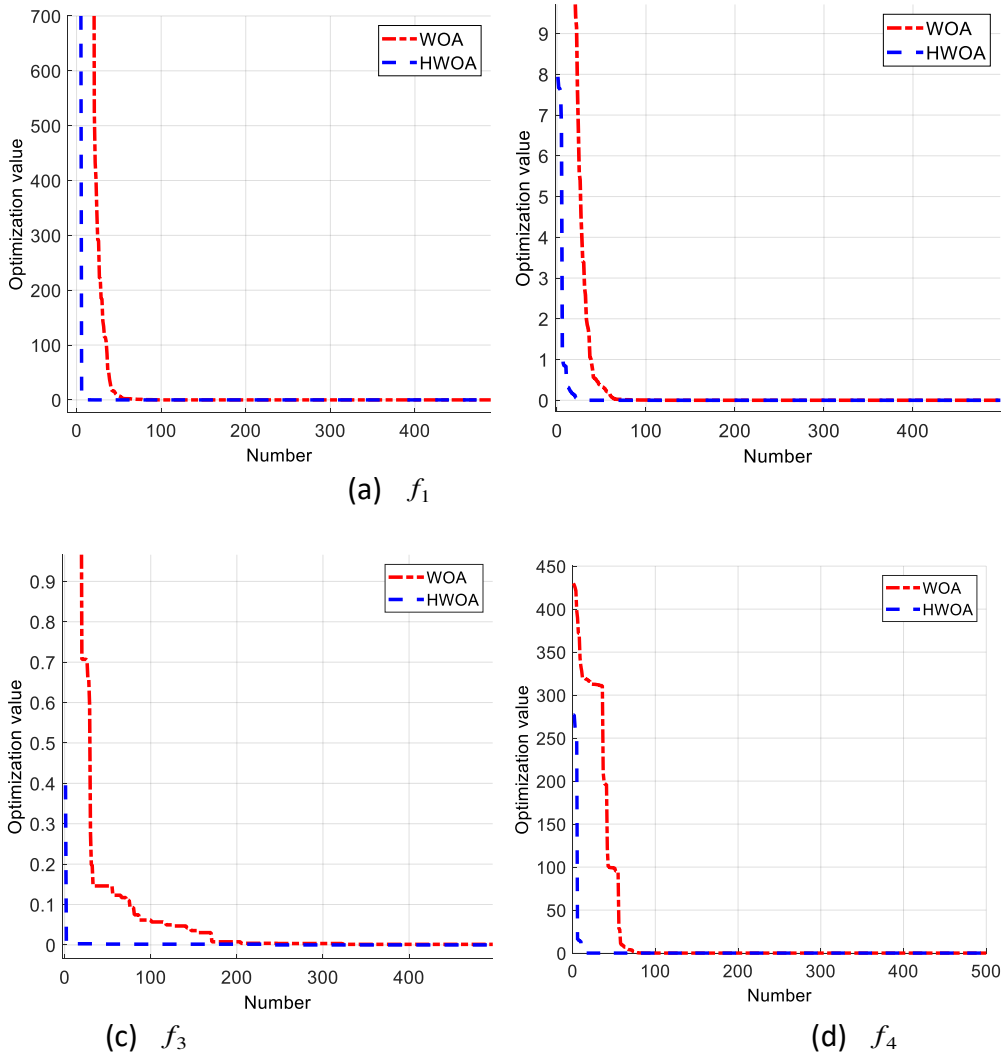


Figure 3. Convergence curves

4. Results

This section presents HWOA-ELM cloud load forecasting model and simulation experiments.

4.1 HWOA-ELM cloud load forecasting model

The threshold and weight have a great influence on the forecasting results, but they all are randomly selected in the ELM model. The HWOA is first used to optimize the ELM model hyper-parameters, and then the cloud load is predicted by the proposed predictive model.

The cloud load prediction process of HWOA-ELM model is described as follows:

(1) Divide cloud load data to determine the training set and test set of cloud load forecast model;
(2) Normalize the cloud load data set;
(3) Initialize the HWOA optimizer parameters;
(4) Train the cloud load prediction model with cloud load training set;
(5) Enter the optimal hyper-parameter into the ELM model;
(6) Predict the cloud load by cloud load forecast model with cloud load test set; and
(7) Analyze the cloud load prediction results.
(8)

The flow diagram of the cloud load is as follows:

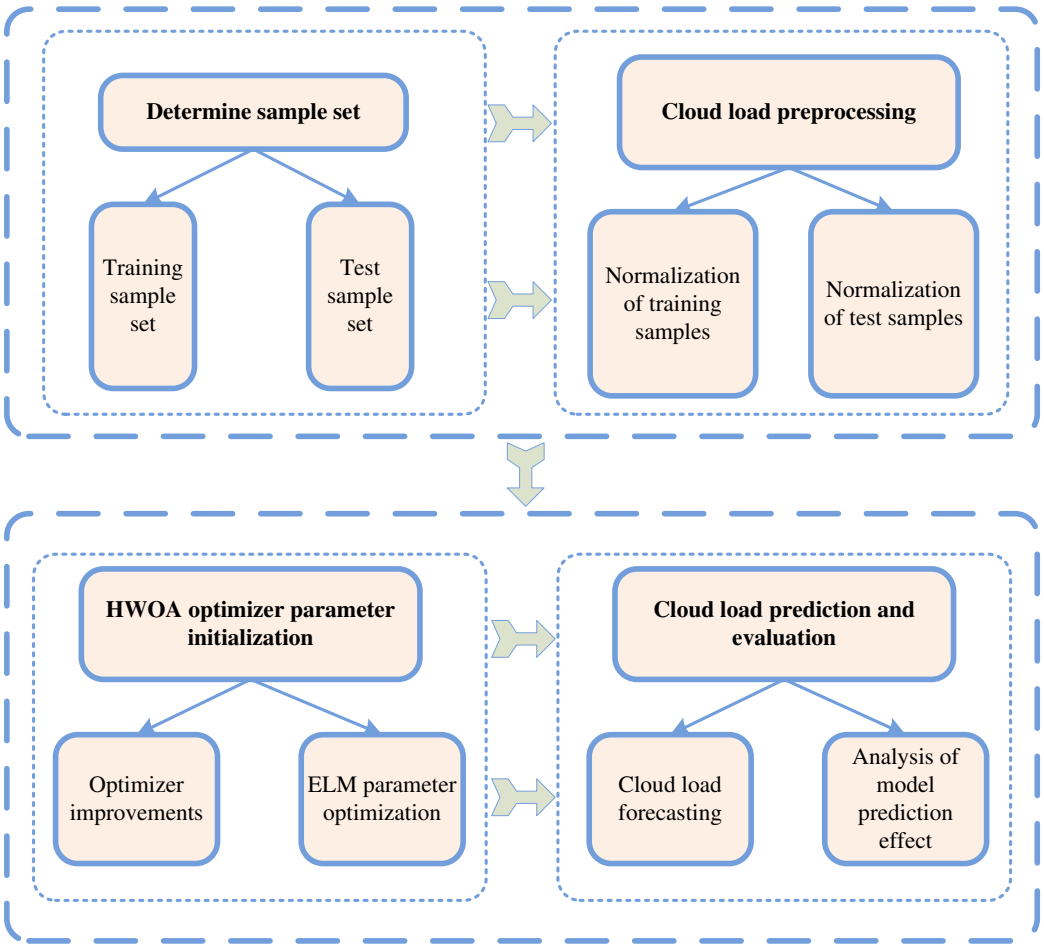


Figure 4. Cloud load forecasting process

The prediction results are analyzed by root mean square error (RMSE), mean absolute percentage error (MAPE) and decision coefficient (r^2). RMSE reflects the degree that the predicted value deviates from the actual value. The size of r^2 determines the explanatory

ability of the independent variable to the dependent variable, the explanatory ability is stronger and the better goodness of fit while the closer r^2 approaches to 1

$$RMSE = \sqrt{\frac{1}{Num}\sum_{j=1}^{Num}(\hat{q_j} - q_j)^2} \tag{25}$$

$$MAPE = \frac{100}{Num}\sum_{j=1}^{Num}\left|\frac{\hat{q_j} - q_j}{q_j}\right| \tag{26}$$

where *Num* is the total number of samples, *q*^ is the predicted sample, *q* is the actual sample.

$$r^2 = \frac{(Num\sum q\hat{}\cdot q - \sum q\hat{}\sum q)^2}{(Num\sum (q\hat{})^2 - \sum (q\hat{})^2)\cdot(Num\sum (q)^2 - \sum (q)^2)} \tag{27}$$

3.2 Simulation experiment and data analysis

The data is provided by a cloud service provider in Tianjin, which contains a total of 300 continuous cloud load data. Figure 5 illustrates that the cloud load data has a strong nonlinearity and its fluctuation is relatively large, posing a great challenge to the prediction accuracy and stability for cloud load forecasting.

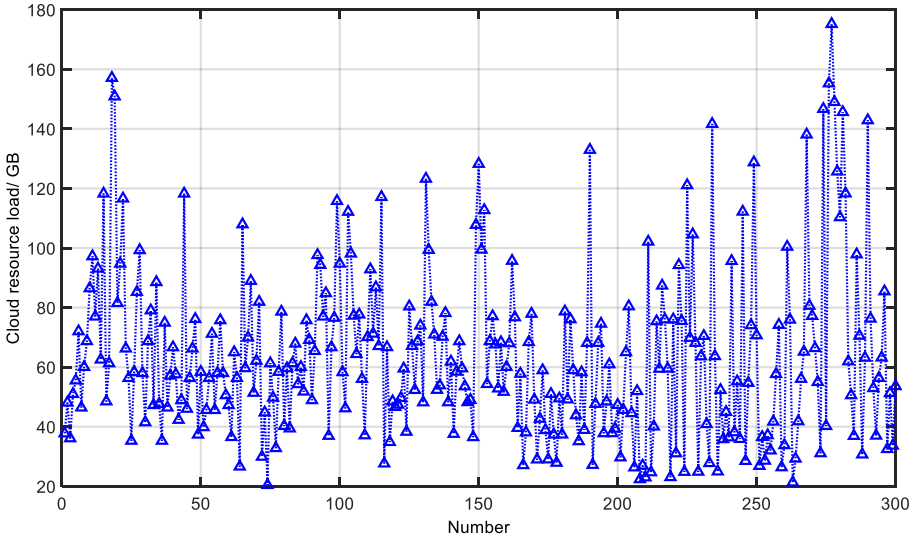
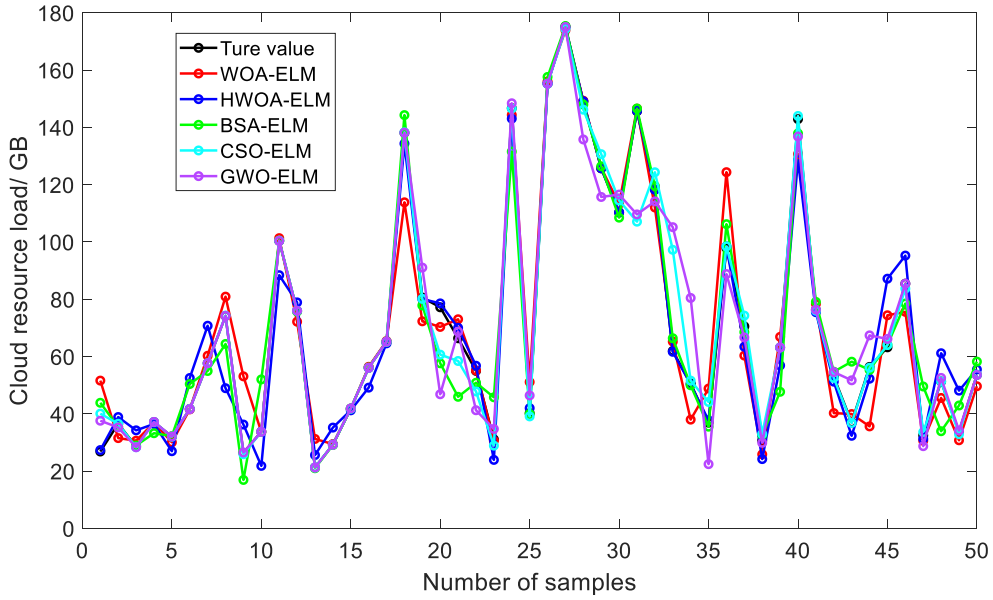


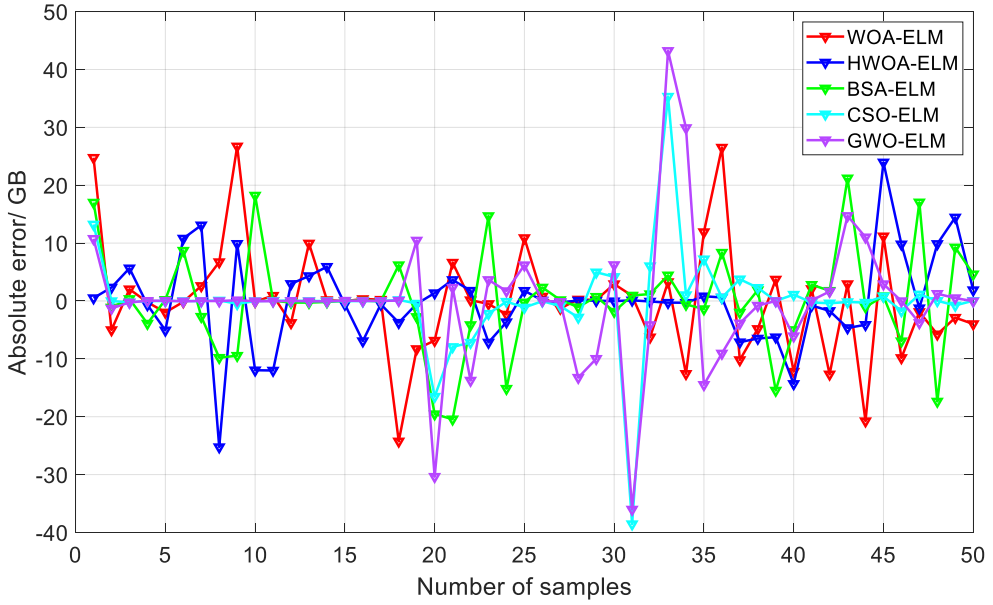
Figure 5. Cloud load data

In this section, three simulation experiments are performed for testing the forecasting effect of the proposed predictive model. (1) the number of training sets is 250 and the number of testing sets is 50; (2) the number of training sets is 200 and the number of testing sets is 100; (3) the number of training sets is 150 and the number of testing sets is 150. The purpose of three simulation tests is to assess the forecasting stability of proposed predictive model and the influence of sample number on the prediction effect of the model. Meanwhile, the forecasting results are compared with those of the WOA-ELM model.

The WOA-ELM and HWOA-ELM models are trained respectively using 250 sample sets, and make their respective prediction for 50 cloud load data. The forecasting results of the models are shown in Figure 6.



(a) Forecasting results



(b) Forecasting error

Figure 6. Forecasting curves and errors of simulation experiment 1

Figure 6 (a) presented that the cloud load prediction curves of five predictive models fit the changing trend of actual values, but the fitting degree of prediction curves of different models and real value curve is different. At the beginning of the forecast, the cloud load prediction model was the closest to the actual value. The cloud load predicted by WOA-ELM, BSA-ELM, CSO-ELM and GWO-ELM models had large deviations from the actual cloud load. The forecasting errors are depicted in Figure 6 (b). For the 5th to 30th sample points, the CSO-ELM and GWO-ELM models showed strong predictive performance, but for the 30th to 35th sample points, the AE of the GWO-ELM and CSO-ELM models exceeded 30, indicating that the model's prediction stability was low. The AE of the BSA-ELM and CSO-ELM models stabilized in the interval [-20, 20]. The AE of the HWOA-ELM model was controlled in the interval [-10, 10]. The forecasting error fluctuation of proposed predictive model was more stable compared with the other forecast models, further illustrating that the HWOA-ELM

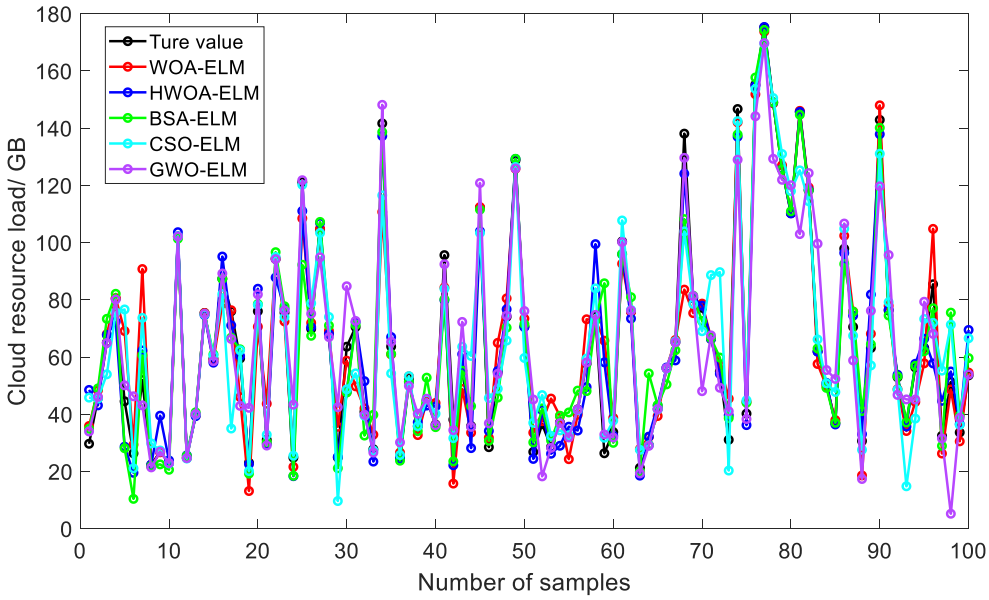
model was more suitable for cloud load data and had higher prediction accuracy. Table 3 lists the predictive results of five models.

Table 3. Analysis of prediction results in simulation experiment 1

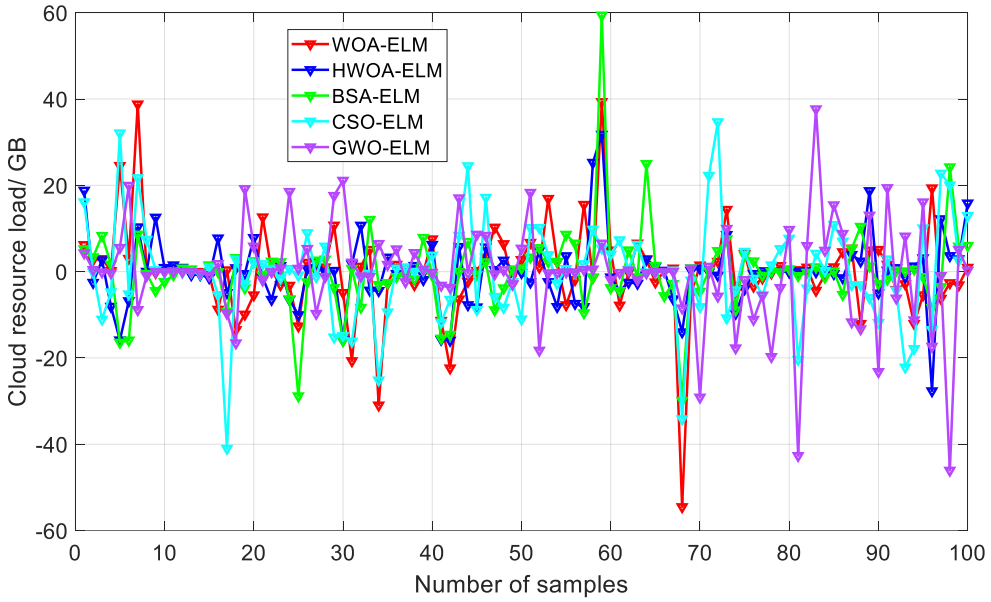
Models	AE interval	MAPE/%	RMSE	r^2/%
WOA-ELM	[-24.25, 26.71]	12.89	9.63	94.38
HWOA-ELM	[-25.24, 23.95]	10.29	7.67	96.43
BSA-ELM	[-20.43, 21.20]	12.03	8.71	95.40
CSO-ELM	[-38.55, 35.31]	11.11	8.33	95.80
GWO-ELM	[-36.02, 43.22]	13.57	11.39	92.13

Table 3 revealed that the proposed predictive model obtained a more competitive prediction effect on 50 cloud load data than other forecasting models. In proposed HWOA-ELM model, he AE interval was smaller than the other four models. The MAPE value was the smallest, which was 2.6%, 1.74%, 0.82%, 3.28% smaller than WOA-ELM, BSA-ELM, CSO-ELM and GWO-ELM models. The RMSE was also more competitive compared with the other four models. The RMSE value was 7.67, which was 1.96, 1.04, 0.66, 3.72 smaller than WOA-ELM, BSA-ELM, CSO-ELM and GWO-ELM models. RMSE and MAPE reflected the prediction error. The r^2 value of the proposed model was higher than the other four models, which indicated that proposed model obtains a more satisfactory fitting effect. The r^2 of the proposed predictive model was 2.05% higher than WOA-ELM model, indicating that the model had a strong explanatory power for cloud load.

To further verify the feasibility of the proposed model, cloud loads of different test samples were selected to test the proposed model. The HWOA-ELM, WOA-ELM, BSA-ELM, CSO-ELM and GWO-ELM models were trained through using 200 cloud load and 100 cloud load was selected as testing samples. Figure 7 depicted the prediction curves and AE errors of the five models.



(a) Forecasting results



(b) Forecasting error

Figure 7. Forecasting curves and errors of simulation experiment 2

The prediction curves of WOA-ELM, BSA-ELM, CSO-ELM, GWO-ELM and the proposed predictive model for 100 cloud load are presented in Figure 7(a), and Figure 7(b) shows their forecasting errors. The cloud load forecast curves of the five models reflected the fluctuation trend of actual cloud load. It is found by comparing Figure 6 and Figure 7 that the forecasting errors of both models all increase to some extent. Figure 7 illustrated that the prediction cloud load model was still closer to the real cloud load, and its cloud load prediction error was obviously smaller than WOA-ELM model. The proposed model revealed a strong prediction stability. The maximum cloud load AE value of WOA-ELM, GWO-ELM and BSA-ELM models exceeded 40. The AE values of the HWOA-ELM model were controlled within the interval [-20, 20], and the AE fluctuation of the proposed model was more stable compared with the other four models. The proposed model got a satisfactory cloud load predictive result. Table 4 presented the results of the prediction evaluation in the simulation experiment 2.

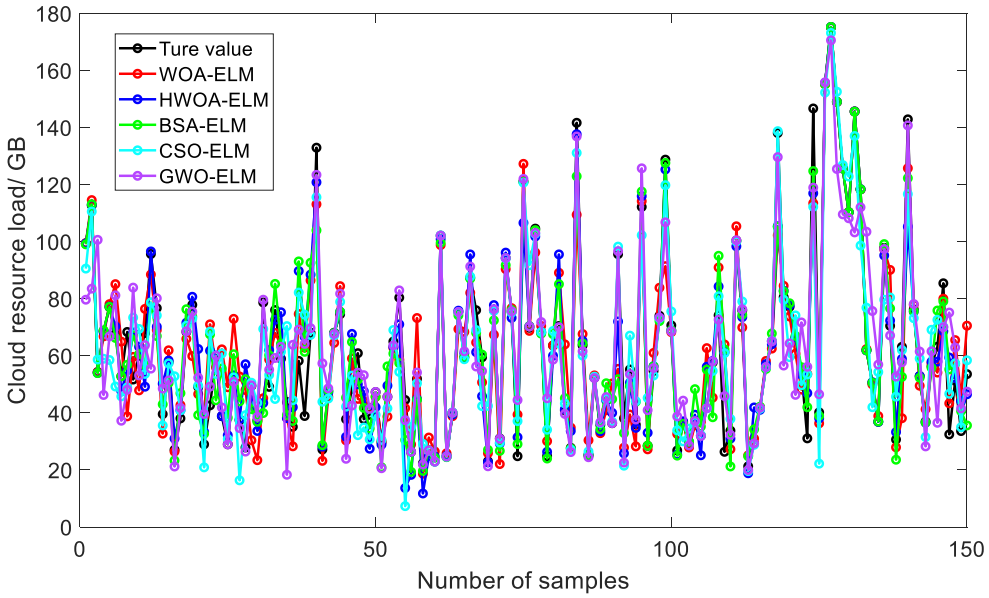
Table 4. Analysis of prediction results in simulation experiment 2

Models	AE interval	MAPE/%	RMSE	r^2/%
WOA-ELM	[-54.51, 39.35]	13.04	10.90	90.94
HWOA-ELM	[-27.66, 31.78]	10.70	8.03	95.08
BSA-ELM	[-29.87, 59.41]	12.43	9.72	92.79
CSO-ELM	[-40.98, 34.68]	14.93	11.60	89.74
GWO-ELM	[-46.06, 37.67]	14.06	11.75	89.47

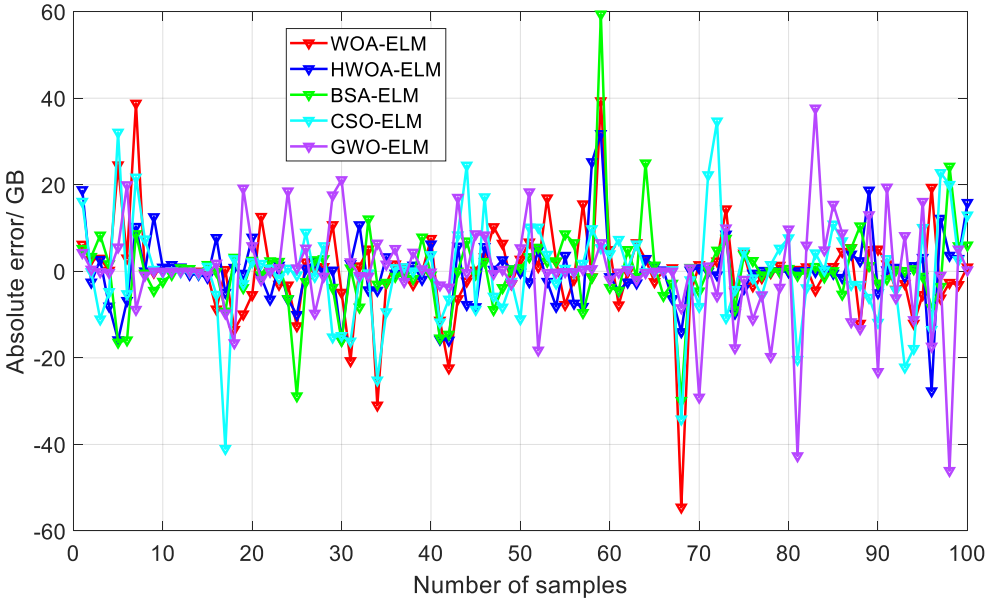
Table 4 presented the evaluation results of 5 predictive models. The proposed HWO-ELM obtained a competitive cloud load prediction evaluation result. The MAPE and RMSE values of the HWOA-ELM model were 10.70 and 8.03, respectively. The cloud load prediction error of the proposed model was smaller compared with the other four models. The MAPE is 2.34%, 1.73%, 4.23% and 3.36% smaller than WOA-ELM, BSA-ELM, CSO-ELM and GWO-ELM models. The RMSE was 2.87, 1.69, 3.57 and 3.72 smaller than WOA-ELM,

BSA-ELM, CSO-ELM and GWO-ELM models. Meanwhile, the proposed model had a satisfactory fitting effect, r^2 was 95.08%, which was higher than the other four forecast models. The above analysis demonstrated that the prediction stability and the proposed model accuracy were more competitive than those of the other four forecast models.

The 150 cloud load samples are employed to train the proposed model, and 150 cloud load samples were selected as the test set to test the proposed model. Meanwhile, WOA-ELM, BSA-ELM, CSO-ELM and GWO-ELM models are used as comparative models to analyze the cloud load forecasting performance of the proposed model. Figure 8 depicts the cloud load forecasting curves and errors.



(a) Forecasting results



(b) Forecasting error

Figure 8. Forecasting curves and errors of simulation experiment 3

Figure 8(a) depicts the cloud load predictive curves of the five forecast models, and Figure (b) shows the cloud load AE of each model. For different cloud loads, the prediction

effects of the predictive models were different, and the cloud load forecasting curves of the model fitted the change trend of the actual cloud load. For most cloud loads, the proposed model presented a satisfactory fitting effect. The AE value fluctuated more smoothly compared with other four models. The maximum AE of WOA-ELM, CSO-ELM, BSA-ELM and GWO-ELM models exceeded 40, indicating that the model predictive stability was low. However, the proposed model reveals high predictive stability. The evaluation indicators were used to evaluate the different models predictive performance, as presented in Figure 5.

Table 5. Analysis of prediction results in simulation experiment 3

Models	AE interval	MAPE/%	RMSE	r^2/%
WOA-ELM	[-40.18, 37.61]	17.51	12.83	84.31
HWOA-ELM	[-37.61, 34.89]	13.09	10.82	88.83
BSA-ELM	[-43.79, 35.92]	13.51	11.14	88.16
CSO-ELM	[-37.19, 35.52]	17.35	12.20	85.80
GWO-ELM	[-42.37, 46.32]	16.58	13.48	82.67

Table 5 depicted that the fitting effect of each model was worse as the cloud load sample set increased, and the r^2 value decreased below 90%. The AE interval was [-37.61, 34.89]. The AE interval of the HWOA model had the smallest fluctuation range compared with the other four models. The AE interval of GWO-ELM model had the largest fluctuation, which was [-42.37, 46.32]. The fitting effect of the model became worse to a certain extent as the number of testing samples increased. The r^2 of five models maintained above 90%, but the r^2 in Table 5 was reduced to below 90% compared the data in Tables 3 and 4. The interpretative capability became worse. However, the proposed model kept the competitive fitting effect with the decrease of training samples and increase of testing samples. In HWOA-ELM model, the MAPE and RMSE are obtained more competitive and the MAPE was 4.42%, 0.42%, 4.26% and 3.49% smaller than WOA-ELM, BSA-ELM, CSO-ELM and GWO-ELM models. The RMSE was 2.01, 0.32, 1.38 and 2.66 smaller than WOA-ELM, BSA-ELM, CSO-ELM and GWO-ELM models. The proposed model obtained lower cloud load forecasting errors and higher r^2 value from the simulation experiment 3.

The proposed HWOA-ELM model performance is verified through three simulation experiments. Experimental results show that the proposed model has high predictive stability and accuracy, and provides a new solution for predicting cloud load.

4. Concluding Remarks

This study improves cloud load forecasting accuracy. Internet of Things services have been continually moved to the cloud besides the vast growth of internet traffic with the rapid development of Internet technology. An accurate prediction is considered as the basis and premise for management and decision-making. Cloud load prediction accuracy directly affects enterprise costs and quality of service. Accurate cloud load prediction is necessary to reasonably arrange the cloud server and improves the quality of cloud service (Liu et al., 2020). An effective method is foremost for solving any practical problems. Under this circumstance, the WOA based on a HWOA-ELM is proposed to predict cloud load, possessing small prediction error and strong fitting ability. The WOA optimizer has been improved due to its inherent limitations, and then the improved HWOA optimizer is used to optimize the hyper-parameters of the ELM model. The proposed model is formulated to forecast the

cloud load to combine the improved WOA optimizer with machine learning model (Parand et al., 2021). The proposed cloud load prediction approach is verified through multiple experiments, revealing that the proposed model obtains competitive results. The contributions of his study are as follows.

- The WOA optimizer based on the hybrid strategy is proposed and results that the optimization performance of proposed HWOA optimizer is better than WOA, ALO and CSA due to the improved HWOA optimizer is tested with benchmark testing functions and the proposed optimizer converges to 0 for f_1 and f_4.
- The proposed model prediction performance is tested by three simulation experiments and compares with four models. For simulation experiment 1, the MAPE value of HWOA-ELM model is the smallest, which is 2.6%, 1.74%, 0.82%, 3.28% smaller than WOA-ELM, BSA-ELM, CSO-ELM and GWO-ELM models. The RMSE obtained by HWOA-ELM model is more competitive compared with other four models. The HWOA-ELM model RMSE value is 7.67, which is 1.96, 1.04, 0.66, 3.72 smaller than WOA-ELM, BSA-ELM, CSO-ELM and GWO-ELM models. The prediction error is relatively smaller and the fitting effect is better.
- This study demonstrates that the number of testing samples and training samples have a great influence on the model forecasting effect. In simulation experiment 1 and 2, the r^2 of the proposed model maintains above 90%. The r^2 values of forecast models are reduced to less than 90% in simulation experiment 3. The RMSE of the two models increases as the number of testing samples grows.
- This study combines intelligent optimization algorithm with machine learning model to establish the cloud load predictive model. The proposed model has strong nonlinear mapping and generalization abilities to have positive significance for improving the resource optimization efficiency in the cloud environment.

The results indicate that the proposed predictive model has achieved better performance. There are several limitations in this study. First, the convergence ability of the HWOA optimizer needs to be further improved. Second, the generalization ability of the proposed model needs to be further enhanced. Future study might be to integrate data mining under the precondition of minimum computational complexity before the prediction, which is not only a complicated issue but an important strategy realizes a more efficient, operational, customized cloud resource management and optimization.

Author Contributions: Conceptualization, H. P. and W.-S. W.; methodology, H. P. and W.-S. W.; software, L.-L. L. and M.-L. T.; formal analysis, L.-L. L. and M.-L. T.; data curation, H.-P., W.-S. W. and M.-L. T.; writing—original draft preparation, H. P., W.-S. W., M.-L. T. and L.-L. L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (Grant No. 51475136) and Guangdong science and technology plan project of China (Grant No. 2016B030305007).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alencar, A. S. C.; Neto, A. R. R.; Gomes, J. P. P. A new pruning method for extreme learning machines via genetic algorithms. Applied Soft Computing. 2016, 44, 101-107.

2. Ali, S.; Li, J.; Pei, Y.; Aslam, M. S.; Shaukat, Z.; Azeem, M. An Effective and Improved CNN-ELM Classifier for Handwritten Digits Recognition and Classification. Symmetry-Basel. 2020, 12(10).
3. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. Computers & Structures. 2016, 169(C), 1-12.
4. Aziz, M. A. E.; Eweesc, A. A.; Hassanien, A. E. Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation. Expert Systems with Applications. 2017, 83, 242-256.
5. Barati, M.; Sharifian, S. A hybrid heuristic-based tuned support vector regression model for cloud load prediction. Journal of Supercomputing. 2015, 71(11), 4235-4259.
6. Calheiros, R. N.; Masoumi, E.; Ranjan, R.; Buyya, R. Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS. IEEE Transactions on Cloud Computing. 2015, 3(4), 449-458.
7. Cao, J.; Fu, J. W.; Li, M. L.; Chen, J. J. CPU load prediction for cloud environment based on a dynamic ensemble model. Software-Practice & Experience. 2014, 44(7), 793-804.
8. Chen, Z. J.; Zhu, Y. C.; Di, Y. Q.; Feng, S. C. Self-Adaptive Prediction of Cloud Resource Demands Using Ensemble Model and Subtractive-Fuzzy Clustering Based Fuzzy Neural Network. Computational Intelligence and Neuroscience. 2015, 919805
9. Chia, M. Y.; Huang, Y. F.; Koo, C. H. Swarm-based optimization as stochastic training strategy for estimation of reference evapotranspiration using extreme learning machine. Agricultural Water Management. 2021, 243.
10. Choudhary, R.; Shukla, S. A clustering based ensemble of weighted kernelized extreme learning machine for class imbalance learning. Expert Systems with Applications. 2021, 164, 114041
11. de Franca; F. O.; de Lima, M. Z. Interaction-transformation symbolic regression with extreme learning machine. Neurocomputing. 2021, 423, 609-619.
12. Edwards, A. M.; Phillips, R. A.; Watkins, N. W.; Freeman, M. P.; Murphy, E. J.; Afanasyev, V.; Buldyrev, S. V.; Da Luz, M. G. E.; Raposo, E. P.; Stanley, H. E.; Viswanathan, G. M. Revisiting Levy flight search patterns of wandering albatrosses, bumblebees and deer. Nature. 2007, 449(7165), 1044-1048.
13. Emary, E.; Zawbaa, H. M.; Sharawi, M. Impact of Levy flight on modern meta-heuristic optimizers. Applied Soft Computing. 2019, 75, 775-789.
14. Gupta, S.; Dileep, A. D.; Gonsalves, T. A. Online Sparse BLSTM Models for Resource Usage Prediction in Cloud Datacentres. Ieee Transactions on Network and Service Management. 2020, 17(4), 2335-2349.
15. Hakli, H.; Uguz, H. A novel particle swarm optimization algorithm with Levy flight. Applied Soft Computing. 2014, 23, 333-345.
16. Han, S.; Zhu, K.; Wang, R. Improvement of evolution process of dandelion algorithm with extreme learning machine for global optimization problems. Expert Systems with Applications. 2021, 163.
17. He, W.; Xie, Y.; Lu, H.; Wang, M.; Chen, H. Predicting Coronary Atherosclerotic Heart Disease: An Extreme Learning Machine with Improved Salp Swarm Algorithm. Symmetry-Basel. 2020, 12(10).
18. Huang, G. B.; Wang, D. H.; Lan, Y. Extreme learning machines: a survey. International Journal of Machine Learning and Cybernetics. 2011, 2(2), 107-122.

19. Jensi, R.; Jiji, G. W. An enhanced particle swarm optimization with levy flight for global optimization. Applied Soft Computing. 2016, 43, 248-261.
20. Jiang, H.; Haihong, E.; Song, M. Multi-prediction based scheduling for hybrid workloads in the cloud data center. Cluster Computing-the Journal of Networks Software Tools and Applications. 2018, 21(3), 1607-1622.
21. Juszczyk, M.; Lesniak, A. Modelling Construction Site Cost Index Based on Neural Network Ensembles. Symmetry-Basel. 2019, 11(3), 18.
22. Kim, D.-Y.; Jeong, Y.-S.; Kim, S. Data-Filtering System to Avoid Total Data Distortion in IoT Networking. *Symmetry* 2017, *9*, 16.
23. Kumar, J.; Singh, A. K.; Buyya, R. Self directed learning based workload forecasting model for cloud resource management. Information Sciences. 2021, 543, 345-366.
24. Li, C. B.; Zheng, X. S.; Yang, Z. K.; Kuang, L. Predicting Short-Term Electricity Demand by Combining the Advantages of ARMA and XGBoost in Fog Computing Environment. Wireless Communications & Mobile Computing. 2018, 18.
25. Li, L. L.; Liu, Z. F.; Tseng, M. L.; Chiu, A. S. F. Enhancing the Lithium-ion battery life predictability using a hybrid method. Applied Soft Computing. 2019, 74, 110-121.
26. Li, L. L.; Sun, J.; Tseng, M. L.; Li, Z. G. Extreme learning machine optimized by whale optimization algorithm using insulated gate bipolar transistor module aging degree evaluation. Expert Systems with Applications. 2019, 127, 58-67.
27. Liu, Z. F.; Li, L. L.; Tseng, M. L.; Lim, M. K. Prediction short-term photovoltaic power using improved chicken swarm optimizer- Extreme learning machine model. Journal of Cleaner Production. 2020, 248, 119272.
28. Mafarja, M.; Mirjalili, S. Whale optimization approaches for wrapper feature selection. Applied Soft Computing. 2017, 62, 441-453.
29. Mafarja, M. M.; Mirjalili, S. Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. Neurocomputing. 2017, 260, 302-312.
30. Mahmud, M. S. A.; Abidin, M. S. Z.; Buyamin, S.; Emmanuel, A. A.; Hasan, H. S. Multi-objective Route Planning for Underwater Cleaning Robot in Water Reservoir Tank. Journal of Intelligent & Robotic Systems. 2021, 101, 9
31. Meenakshi, A.; Sirmathi, H.; Ruth, J. A. Cloud n computing-based resource provisioning using k-means clustering and GWO prioritization. Soft Computing. 2019, 23, 10781-10791.
32. Mehrabi, M.; Giacaman, N.; Sinnen, O. Unified programming concepts for unobtrusive integration of cloud-based and local parallel computing. Future Generation Computer Systems-the International Journal of Escience. 2021, 115, 700-719.
33. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. Advances in Engineering Software. 2016, 95, 51-67.
34. Moreno, S. R.; Mariani, V. C.; Coelho, L. d. S. Hybrid multi-stage decomposition with parametric model applied to wind speed forecasting in Brazilian Northeast. Renewable Energy. 2021, 164, 1508-1526.
35. Parand, K.; Aghaei, A. A.; Jani, M.; Ghodsi, A. A new approach to the numerical solution of Fredholm integral equations using least squares-support vector regression. Mathematics and Computers in Simulation. 2021, 180, 114-128.
36. Rafique, A.; Van Landuyt, D.; Beni, E. H.; Lagaisse, B.; Joosen, W. CryptDICE: Distributed data protection system for secure cloud data storage and computation. Information Systems. 2021, 96.

37. Ros, S.; Caminero, A. C.; Hernandez, R.; Robles-Gomez, A.; Tobarra, L. Cloud-based architecture for web applications with load forecasting mechanism: a use case on the e-learning services of a distant university. Journal of Supercomputing. 2014, 68(3), 1556-1578.
38. Santos, M. A. F. d.; Nobre, F. D.; Curado, E. M. F. Monitoring Levy-process crossovers. Communications in Nonlinear Science and Numerical Simulation. 2021, 92.
39. Khalilpourazari, s., Khalilpourazary, S. SCWOA: an efficient hybrid algorithm for parameter optimization of multi-pass milling process, Journal of Industrial and Production Engineering. 2018 35:3, 135-147
40. Safavi, M.; Siuki, A. K.; Hashemi, S. R. New optimization methods for designing rain stations network using new neural network, election, and whale optimization algorithms by combining the Kriging method. Environmental Monitoring and Assessment. 2021, 193(1).
41. Taghizadeh-Mehrjardi, R.; Schmidt, K.; Toomanian, N.; Heung, B.; Behrens, T.; Mosavi, A.; Scholten, T. Improving the spatial prediction of soil salinity in arid regions using wavelet transformation and support vector regression models. Geoderma. 2021, 383.
42. Tikhamarine, Y.; Malik, A.; Pandey, K.; Sammen, S. S.; Souag-Gamane, D.; Heddam, S.; Kisi, O. Monthly evapotranspiration estimation using optimal climatic parameters: efficacy of hybrid support vector regression integrated with whale optimization algorithm. Environmental Monitoring and Assessment. 2020, 192(11).
43. Tofighy, S.; Rahmanian, A. A.; Ghobaei-Arani, M. An ensemble CPU load prediction algorithm using a Bayesian information criterion and smooth filters in a cloud computing environment. Software: Practice & Experience. 2018, 48, 2257-2277.
44. Wu, T.; Xue, W.; Wang, H.; Chung, C. Y.; Wang, G.; Peng, J.; Yang, Q. Extreme Learning Machine-Based State Reconstruction for Automatic Attack Filtering in Cyber Physical Power System. IEEE Transactions on Industrial Informatics. 2021, 17(3), 1892-1904.
45. Xu, D. Y.; Yang, S. L.; Liu, R. P. A mixture of HMM, GA, and Elman network for load prediction in cloud-oriented data centers. Journal of Zhejiang University Science C: Computers & Electronics. 2013, 14(11), 845-858.
46. Xu, J.; Liu, X.; Ma, M.; Liu, A.; Wang, T.; Huang, C. Intelligent Aggregation Based on Content Routing Scheme for Cloud Computing. Symmetry. 2017, 9, 221.
47. Yang, J. Q.; Liu, C. C.; Shang, Y. L.; Cheng, B.; Mao, Z. X.; Liu, C. H.; Niu L. S.; Chen, J. L. A cost-aware auto-scaling approach using the workload prediction in service clouds. Information Systems Frontiers. 2014, 16(1), 7-18.
48. Ye, K. Key Feature Recognition Algorithm of Network Intrusion Signal Based on Neural Network and Support Vector Machine. Symmetry-Basel. 2019, 11(3).
49. You, D.; Lin, W.; Shi, F.; Li, J.; Qi, D.; Fong, S. A novel approach for CPU load prediction of cloud server combining denoising and error correction. Computing. 2020.
50. Yousri, D.; Allam, D.; Eteiba, M. B. Chaotic whale optimizer variants for parameters estimation of the chaotic behavior in Permanent Magnet Synchronous Motor. Applied Soft Computing. 2019, 74, 479-503.
51. Zhao, L. Load forecasting model of cloud computing resources based on support vector machine. Journal of Nanjing University of Science and Technology. 2018, 42(6), 687-692.

Figures

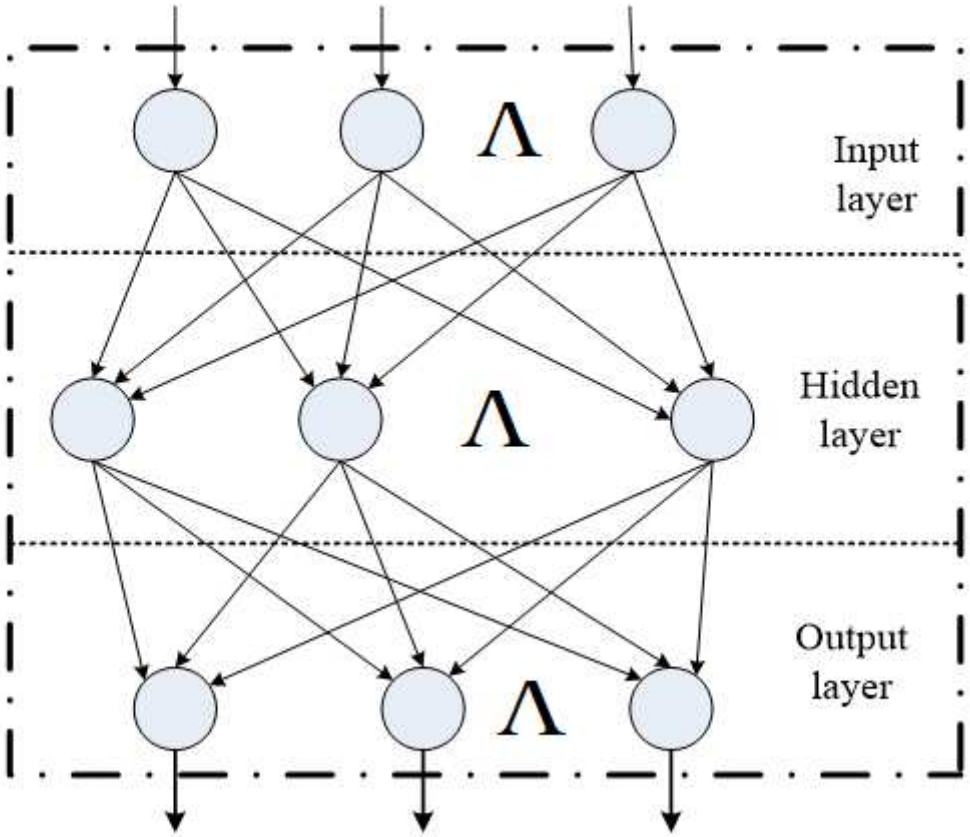


Figure 1

ELM model

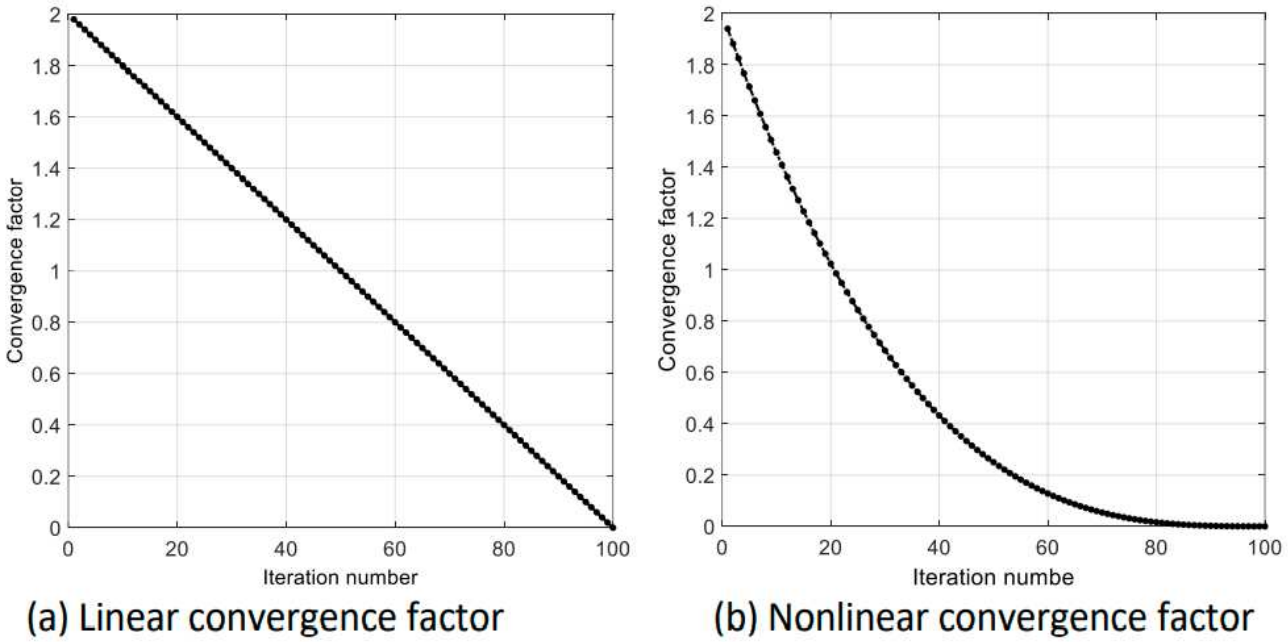


Figure 2

Convergence factor

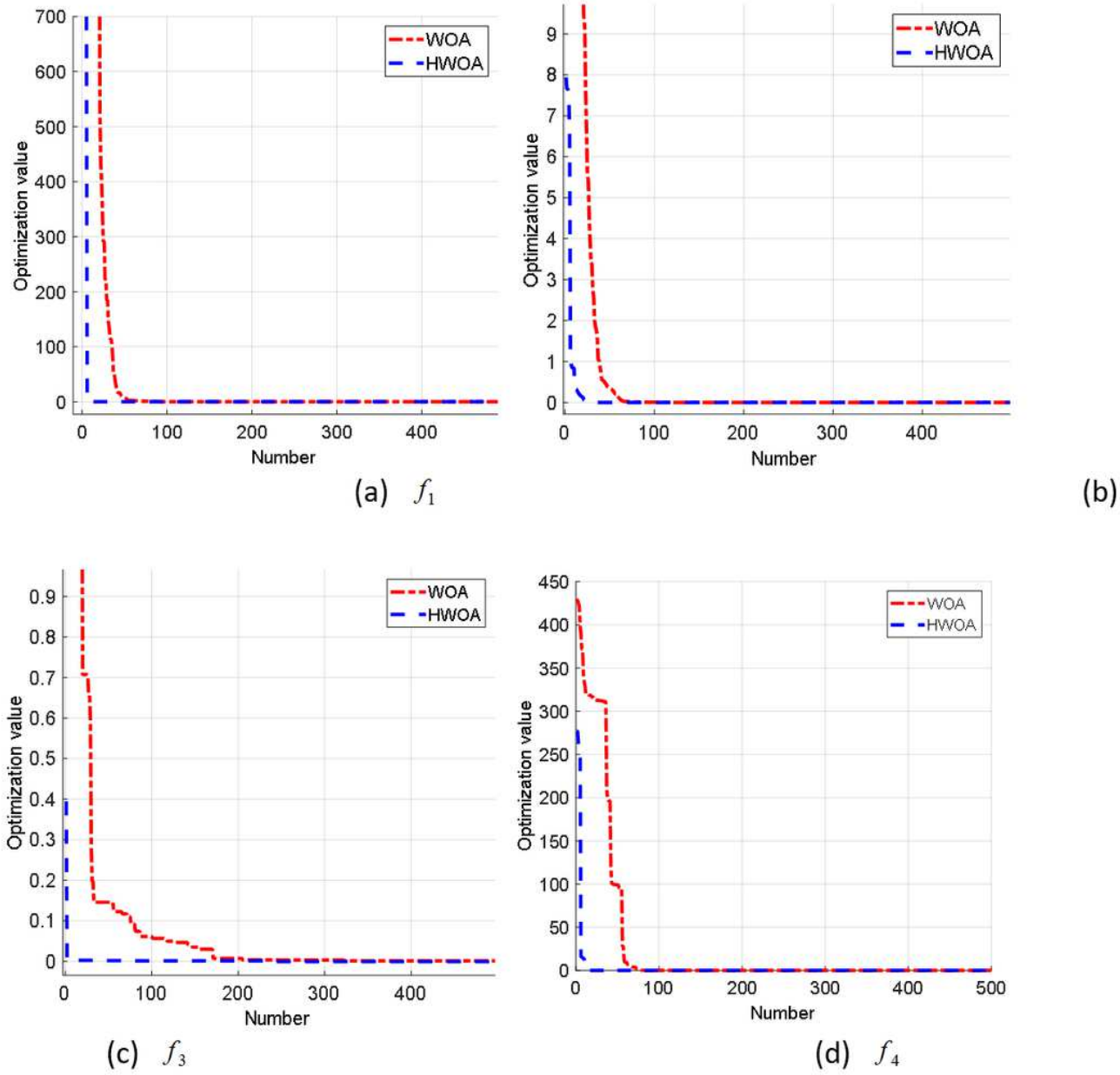


Figure 3

Convergence curves

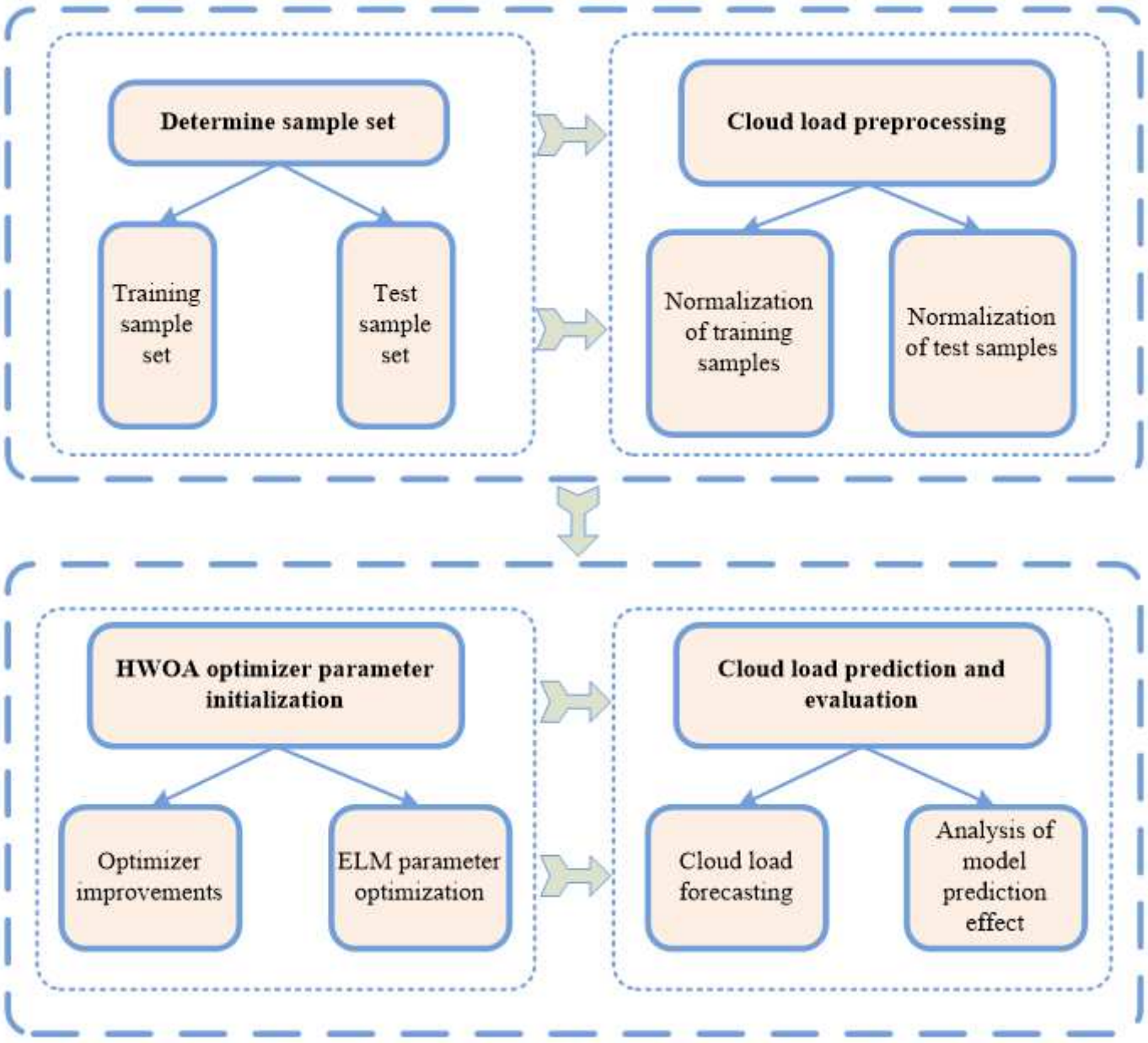


Figure 4

Cloud load forecasting process

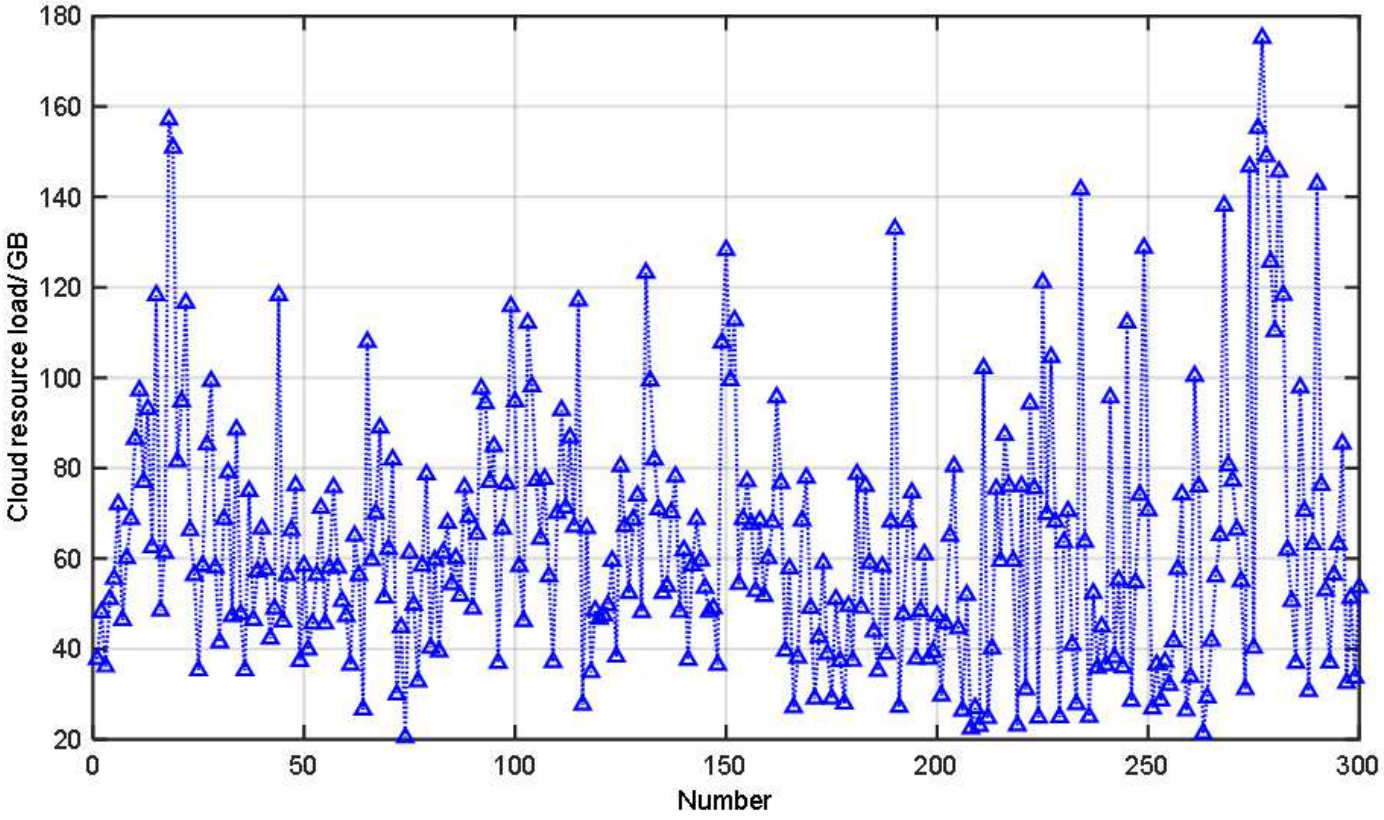


Figure 5

Cloud load data

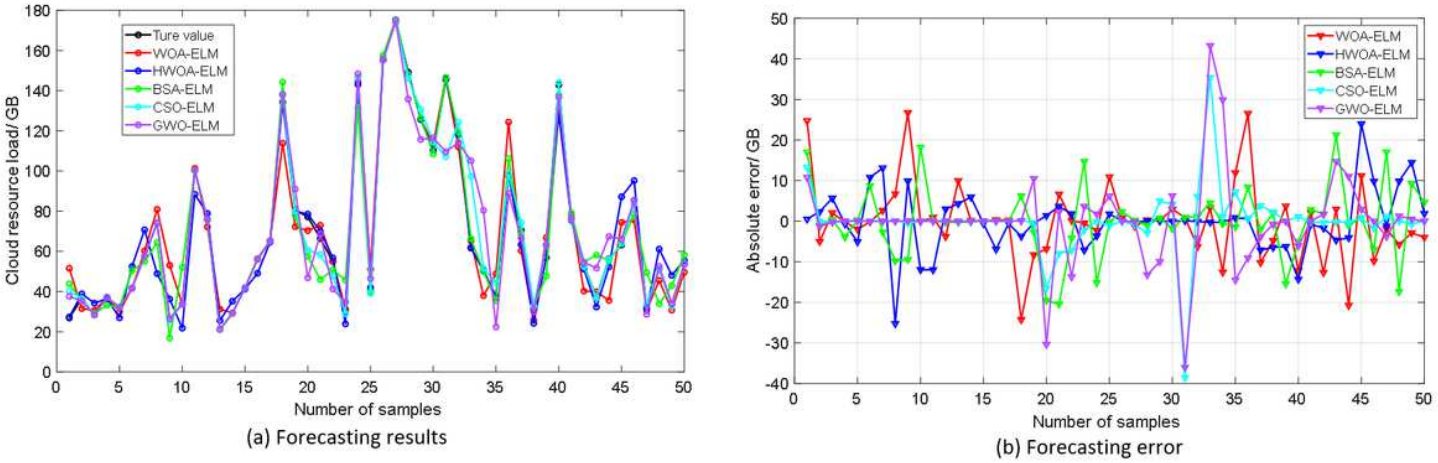


Figure 6

Forecasting curves and errors of simulation experiment 1

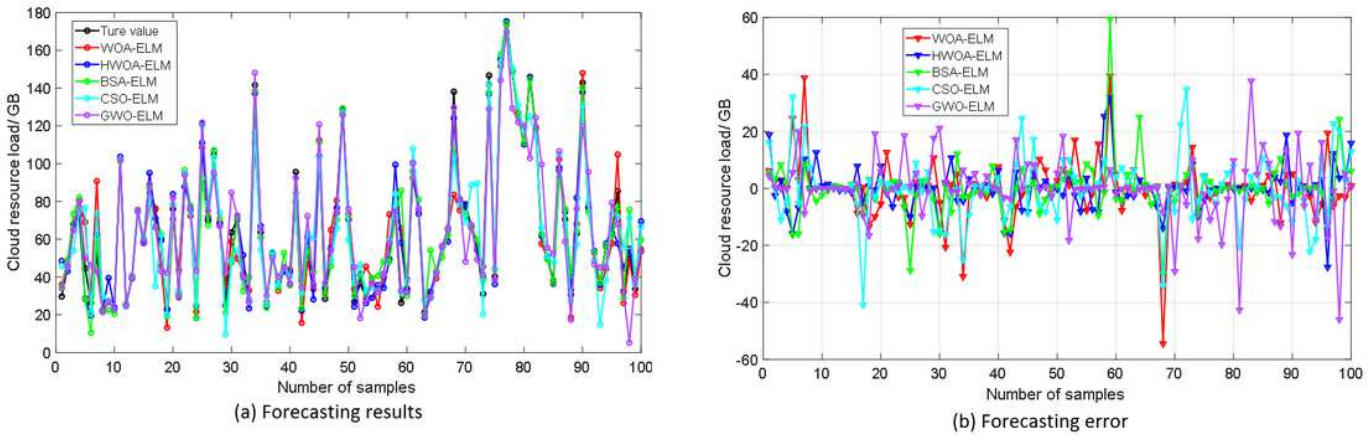


Figure 7

Forecasting curves and errors of simulation experiment 2

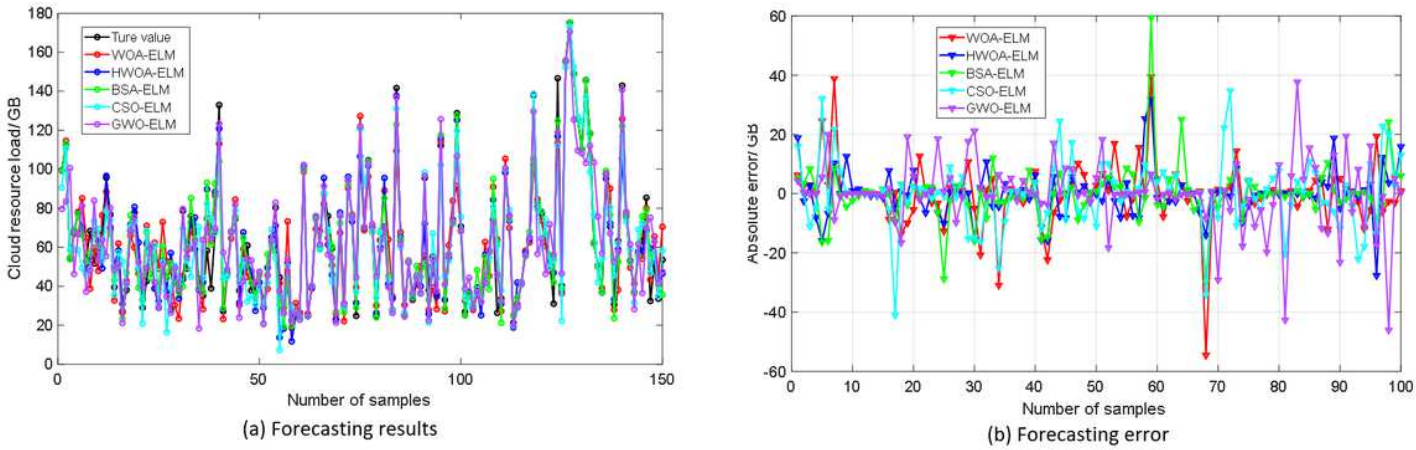


Figure 8

Forecasting curves and errors of simulation experiment 3