Applied Intelligence (2021) 51:952–965
https://doi.org/10.1007/s10489-020-01840-y

Forecasting of e-commerce transaction volume using a hybrid of extreme learning machine and improved moth-flame optimization algorithm

Bo Zhang[1,2] · Runhua Tan[3] · Cheng-Jian Lin[4]

Published online: 10 September 2020

Abstract

The rapid development of e-commerce has resulted in optimization of the industrial structure of Chinese enterprises and has improved the Chinese economy. E-commerce transaction volume is an evaluation index used to determine the development level of e-commerce. This study proposed a model for forecasting e-commerce transaction volume. First, a hybrid moth–flame optimization algorithm (HMFO) was proposed. The convergence ability of the HMFO algorithm was analyzed on the basis of test functions. Second, using data provided by the China Internet Network Information Center, factors influencing e-commerce transaction volume were analyzed. The input variables of the e-commerce transaction volume prediction model were selected by analyzing correlation coefficients. Finally, a hybrid extreme learning machine and hybrid-strategy-based HMFO (ELM-HMFO) method was proposed to predict the volume of e-commerce transactions. The prediction results revealed that the root mean square error of the proposed ELM-HMFO model was smaller than 0.5, and the determination coefficient was 0.99, which indicated that the forecast e-commerce transaction volume was satisfactory. The proposed ELM-HMFO model can promote the industrial upgrading and development of e-commerce in China.

Keywords E-commerce transaction · Prediction · Machine learning · Extreme learning machine · Moth–flame optimization

1 Introduction

As a new business model of the twenty-first century, e-commerce has broad development prospects. E-commerce development has reached a new economic growth point [1, 2]. On November 11, 2019, Alibaba announced that the transaction volume of the Tmall shopping platform had reached RMB 268.4 billion. Furthermore, the transaction volume of the Jingdong Shopping Platform was reported to be RMB 204.4 billion. The number of people shopping online has increased sharply. According to statistics from the China Internet Network Information Center, by the end of 2018, China's Internet penetration rate was 59.6%, with 829 million Internet users. In 2018, the e-commerce transaction volume in China exceeded RMB 31 trillion. The growth rate of the e-commerce transaction volume in 2018 from the previous year was 8.5%. E-commerce, which is developing rapidly, promotes China's social development (intelligence, digitalization, and networking), which in turn promotes the transformation and upgrading of traditional industries in China. E-commerce transaction volume is an indicator of the development of e-commerce. The prediction of e-commerce transaction volume can provide reliable data support for the development of government and enterprise policies [3, 4].

To date, e-commerce development research has focused on two aspects, namely e-commerce index construction and transaction volume prediction [5, 6]. Two types of e-commerce transaction volume prediction model have been devised, namely the statistical regression model and machine learning model. Compared with statistical regression models, machine learning models exhibit strong adaptability to data and strong nonlinear fitting ability and are thus popular. Ji et al. [7] used the C-a-XGBoost model to predict e-

✉ Cheng-Jian Lin
cjlin@ncut.edu.tw

[1] School of Economics and Management, Tianjin University of Technology and Education, Tianjin 300222, China

[2] School of Economics and Management, Hebei University of Technology, Tianjin 300401, China

[3] National Engineering Research Center for Technological Innovation Method and Tool, Tianjin 300401, China

[4] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan

commerce transaction volume. This model combines a statistical regression model with a machine learning model. The linear part of the e-commerce transaction is predicted using the autoregressive integrated moving average (ARIMA) model, and the nonlinear part is predicted using the C-XGBoost model, and the final prediction result is weighted. Although the prediction accuracy of the model is high, the model is complex and its generalizability is low. Chang et al. [8] forecasted transaction volume by establishing a fuzzy neural network. In this method, k-means clustering is first used to cluster historical data. Then, sale volume is predicted using the fuzzy neural network. The k-means clustering method requires constant adjustment of the classification of samples, which is time consuming and affects the development of the model. Chen and Lu [9] proposed a hybrid model to predict e-commerce transaction volume. The hybrid model combines clustering technology with machine learning models. Compared with the single prediction model, the hybrid model had higher prediction accuracy. Di Pillo et al. [10] used the support vector machine (SVM) model to predict transaction value. Compared with the statistical regression ARIMA model, the SVM model has stronger regression ability for irregular data. However, the SVM model is more sensitive to the number of samples. When the number of samples is large, the predictive ability of the SVM model is limited. Zhang [11] developed the back propagation (BP) neural network model to predict e-commerce transaction volume. To escape from the local optimal value, the particle swarm optimization (PSO) algorithm was used to optimize the network's parameters. Li et al. [12] proposed a hybrid model to predict the e-commerce transaction volume of China. The model combines the nonlinear autoregressive natural network with ARIMA. The hybrid model obtained superior prediction results to the single prediction model.

The heuristic optimization algorithm can address high-dimension and multiple extremum problems in complex optimization and has been widely used in many fields, such as economic scheduling, model parameter optimization, and signal processing. Classical algorithms—such as the PSO, differential evolution, and genetic optimization algorithms—are widely used [13–15]. Latest algorithms include the ant colony, gray wolf, whale, cuckoo search, and ant lion algorithms [16–19]. These algorithms simulate animal foraging behavior, flight behavior, or crawling behavior. They focus on only one optimal solution during the iteration process, which limits their local development and global exploration abilities. The moth–flame optimization algorithm (MFO) is different from these algorithms. The MFO algorithm can track multiple optimal solutions simultaneously. Aziz et al. [20] applied the MFO algorithm to solve the optimal threshold determination problem in image segmentation. In the case of multilevel thresholds, the MFO algorithm is used to determine the optimal threshold. The results demonstrated that the MFO algorithm was superior to the whale algorithm. Li et al. [21] used the MFO algorithm to optimize the parameters of the least squares SVM model and predicted the annual power load. Zhang et al. [22] used the MFO algorithm for facial expression recognition and achieved excellent results.

The extreme learning machine (ELM) model is widely used because of its excellent nonlinear mapping ability, strong generalization ability, and simple iterative process. Liu et al. [23] forecast photovoltaic power generation by using the ELM model and employed an intelligent optimizer to enhance the regression ability of the ELM model. Li et al. [24] evaluated the aging of insulated gate bipolar transistor modules by using the ELM model. Baliarsingh and Vipsita [25] used the ELM model to classify cancers and achieved high classification accuracy. Because of the shallow structure of the ELM, even if many hidden nodes exist, the effect of using the ELM for feature learning is weak. To address this problem, Tang et al. [26] proposed a layered learning framework based on the ELM. Test experiments revealed that this method had high learning efficiency. Suresh et al. [27] used the ELM to measure the visual quality of JPEG encoded images. To improve the generalization performance of the ELM algorithm, the real-coded genetic algorithm and k-fold selection scheme were used to select the optimal input weights and deviation values. Wan et al. [28] used the ELM model to directly determine the optimal prediction interval for wind power generation. Experimental results demonstrated that the method had high efficiency and reliability. This method provided a novel general framework for probabilistic wind power forecasting, with high reliability and flexibility. Zhang et al. [29] used the ELM model to address the multiclass classification problem in the field of cancer diagnosis. The ELM model had high classification accuracy with reduced training time and implementation complexity.

Using data provided by the China Internet Network Information Center, this study established a forecast model of e-commerce transaction volume. First, to avoid the shortcoming of the MFO, which is that escaping from extreme values is difficult, the hybrid MFO (HMFO) was proposed. Second, the factors that affect e-commerce transaction volume were analyzed. The input of the e-commerce transaction volume forecast model was determined by the correlation coefficient. Finally, the ELM-HMFO hybrid improvement strategy model was used to predict the e-commerce transaction value of China. The main findings and contributions of this study are as follows:

(1) The HMFO algorithm was proposed and applied to e-commerce.

(2) The ELM-HMFO model was developed to predict the e-commerce transaction volume in China and achieved excellent results.

(3) Through e-commerce transaction volume forecasting, the transaction trend of the market was determined, and decision information for enterprises was provided.

The paper is organized as follows. Section 2 introduces the modeling process of the e-commerce transaction volume prediction model. Section 3 analyzes the influencing factors and forecast results. The conclusions of this paper are presented in Section 4.

2 Model for forecasting e-commerce transaction volume

2.1 MFO

MFO, which is an intelligent heuristic algorithm, simulates the flight behavior of moth species [30]. The algorithm imitates the flight pattern of the moth at night. Moths mainly rely on moonlight to adjust their position, ensuring that the angle between their flight direction and the moonlight remains unchanged during the flight. Because the distance between the moon and moth is large, the moth flies in a relatively straight line. However, when the moth approaches artificial light sources, the flight path of the moth is disturbed from a straight line to a spiral path [31–33].

In the MFO algorithm, the target solution in the optimization process is the moth and the position of the moth represents the variable. By changing its position, a moth can fly in a multidimensional space. Let the position matrix of the moth be P_{OS} [21]:

$$P_{OS} = \begin{bmatrix} pos_{1,1} & pos_{1,2} & \cdots & pos_{1,D} \\ pos_{2,1} & pos_{2,2} & \cdots & pos_{2,D} \\ \vdots & \vdots & \vdots & \vdots \\ pos_{m,1} & pos_{m,2} & \cdots & pos_{m,D} \end{bmatrix} \tag{1}$$

where D is the variable dimension and m is the number of moths.

By inputting the position variable of the moth into the fitness function, the fitness function output is the objective value corresponding to the moth. The objective value corresponding to the moth is stored in the array Fit as follows:

$$P_{os}fit = \begin{bmatrix} P_{OS}fit_1 \\ P_{OS}fit_2 \\ \vdots \\ P_{OS}fit_m \end{bmatrix} \tag{2}$$

The moth requires the position information of the fire to update its flight path. The number of variables in the position matrix of the fire is the same as that of the moth. The position matrix of the fire is represented byFir as follows:

$$Fir = \begin{bmatrix} fir_{1,1} & fir_{1,2} & \cdots & fir_{1,D} \\ fir_{2,1} & fir_{2,2} & \cdots & fir_{2,D} \\ \vdots & \vdots & \vdots & \vdots \\ fir_{m,1} & fir_{m,2} & \cdots & fir_{m,D} \end{bmatrix} \tag{3}$$

Similarly, the position variable of the flame is input to the fitness function to obtain the fitness value corresponding to the flame. The fitness value of the flame is expressed by$Firfit$ as follows:

$$Firfit = \begin{bmatrix} Firfit_1 \\ Firfit_2 \\ \vdots \\ Firfit_m \end{bmatrix} \tag{4}$$

During the iterative process of the MFO algorithm, both the flame and moth positions are solutions to the optimization problem, but the positions of the moths and flames are updated differently. Moths are optimal individuals, searching for the optimal solution in their space. The flame is the optimal location in the space searched by the moth. The moth searches around the corresponding flame. During the iteration, the optimal position is updated using the fitness. The optimal position is used as the flame position in the next iteration. The positions of flames and moths are constantly updated in this search mechanism. The optimization process of the MFO algorithm consists of the following three parts:

$$MFO = (Init, FuncP, DecT) \tag{5}$$

$Init$ is the initialization function. The initial population position matrix and the corresponding fitness matrix of the MFO algorithm are obtained using$Init$.

$$Init : \varphi \rightarrow \{P_{OS}, P_{OS}fit\} \tag{6}$$

$FuncP$ represents the moth position update function. $FuncP$ receives the moth position matrix P_{OS} and returns the updated moth position matrix P_{OS}.

$$FuncP : P_{OS} \rightarrow P_{OS} \tag{7}$$

$DecT$ is the termination decision function. When the algorithm satisfies the termination condition, function $DecT$ returns T; when the algorithm does not satisfy the termination condition, function $DecT$ returns F.

$$DecT : P_{OS} \rightarrow \{T, F\} \tag{8}$$

The MFO algorithm uses the function $Init$ to generate the initial position and corresponding fitness. The program of function $Init$ is shown as follows:

```
Init function
for u=1:m
    for k=1:D
        Pos(u, k)=(U(u)-L(u))*rand+L(u);
        Posfit=fitness(Pos(u, k));
    end
end
```

where $U = [U_1, U_2, \ldots, U_m]$ and $L = [L_1, L_2, .., L_m]$ are the upper and lower bounds of the position variable, respectively.

Considering the influence of flames, the position of a moth during flight is updated as follows:

$$P_{OSi} = S\left(P_{OS_i,}Fir\right) \tag{9}$$

where P_{OS_I} is moth i, S is the spiral function, and Fir_j is flame j.

The spiral function is as follows:

$$S\left(P_{OS_i,}F_{ir_j}\right) = Dis_i \bullet e^{al} \bullet \cos(2\pi n) + Fir_j \tag{10}$$

where $l \in [-1, 1]$, a defines the constant of the spiral shape, and $Dis_i\left(Dis_i = \left|P_{OS_i,}Fir_j\right|\right)$ represents the distance between flame j and moth i.

The position of the moth relative to the flame is determined by the parameter l. The moth is as far from the fire as possible when $l = 1$ and as close to the fire as possible when $l = -1$. After each flame position is updated, the flame fitness values are sorted and the moth's position is updated using the sorted flames. Therefore, the position of the first moth in the population is always updated according to the best flame position, and the positions of the subsequent moths are updated according to the corresponding flame positions. This mechanism prevents moths from being attracted by the same flame, thereby expanding the search range of the moths.

If a fixed number of moths search m positions, the local search capability of the MFO algorithm may be poor. The MFO algorithm uses an adaptive mechanism to adjust the number of flames. During the iteration, the number of flames is decreased gradually. However, it is necessary to ensure that the number of fires and moths are consistent in the iteration process. At the end of the iteration, a moth's position is updated according to the best flame position [34, 35]. The specific adjustment method is as follows:

$$Fir_{num} = round\left(Init_{num} - t^{*}\frac{Init_{num} - 1}{T}\right) \tag{11}$$

where t is the current number of iterations, $round$ is the rounding function, $Init_{num}$ is the number of first-generation flames, and T is the total number of iterations.

Based on the aforementioned principles, the operation process of the MFO algorithm during the iterations is as follows:

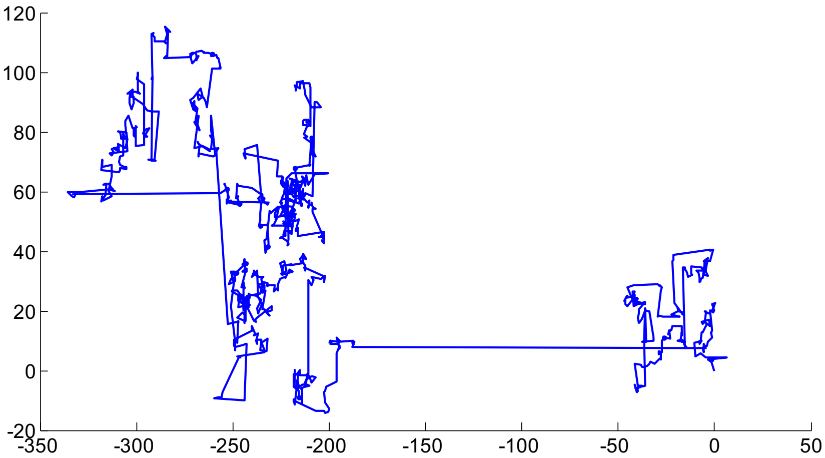
(1) Initialize the position of the moth and calculate the corresponding objective value.
(2) The optimal position of the moth is considered the initial fire position.
(3) Update moth and fire locations.
(4) Update the moth position according to the fire position.
(5) Change the number of flames on the basis of the number of iterations.
(6) Determine whether to terminate the iteration.

2.2 MFO algorithm based on the hybrid strategy (HMFO)

A limitation of the traditional MFO algorithm is its poor ability to escape from a local solution. At the end of iteration as the number of moths and flames decrease the local development capability of the MFO algorithm is limited, which results in moths adopting spiral flight to approach the flame. The spiral flight results in large value space of the flame and also increases the flight time of the moth. To solve this limitation of the MFO algorithm, the hybrid strategy is used.

First, the Levy flight strategy is used to improve the moth update strategy. The convergence ability of the MFO algorithm is limited because the numbers of flames and moths decrease in later iterations. The local convergence ability of the MFO algorithm can be strengthened using the short-distance walk characteristics of the Levy flight strategy. Because of the variability of the long-distance jumping direction of the Levy flight strategy, the ability of the MFO algorithm to determine the best solution in the iterative process can be enhanced. The ability of the algorithm to avoid a local solution and determine the best solution can be improved using the long- and short-distance search method of the Levy flight strategy. The random walk trajectory of the Levy flight strategy is depicted in Fig. 1.

Fig. 1 Levy strategy path



The random search path of Levy flight is depicted as follows:

$$Levy \sim \frac{\omega}{|\rho|^{\frac{1}{\alpha}}} (1 < \alpha < 3) \tag{12}$$

where $\alpha = 1.5$; ω and ρ follow a normal distribution.

$$\begin{cases} \omega \sim N\left(0, \sigma_\omega^2\right) \\ \rho \sim N\left(0, \sigma_\rho^2\right) \end{cases} \tag{13}$$

The normal distribution values are as follows:

$$\begin{cases} \sigma_\omega = \left[\frac{\Gamma(1+\alpha)^* \sin\left(\frac{\alpha}{2}\pi\right)}{\Gamma\left(\frac{1+\alpha}{2}\right)^* \alpha^* 2^{\frac{1+\alpha}{2}}}\right]^{\frac{1}{\alpha}} \\ \sigma_\rho = 1 \end{cases} \tag{14}$$

The moth's position in flight is updated according to the Levy strategy as follows:

$$P_{OS_i}(t+1) = P_{OS_i}(t) + \zeta \oplus Levy(\alpha) \tag{15}$$

where ζ is the random step size and $\oplus$ is point multiplication.

In the MFO algorithm, moths search through spiral flight. To improve the global searching ability of moths in early flight and the local searching ability in subsequent iterations, the sine coefficient is introduced. The optimization efficiency of the moth is improved by the sine coefficient as follows:

$$SC = SC_{min} + (SC_{max} - SC_{min})^* \sin\left(\frac{t}{T}\pi\right) \tag{16}$$

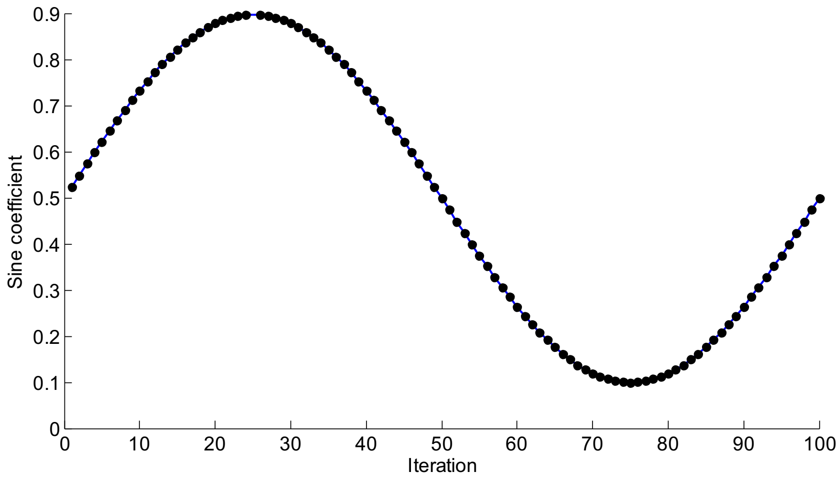
where SC_{min} and SC_{max} are the minimum and maximum values of the sine coefficient, respectively. The curve of the sine coefficient is depicted in Fig. 2.

The curve depicted in Fig. 2 changes according to the sine law, which causes the MFO algorithm to start from local search at the beginning of the iteration. As the coefficient increases, the MFO algorithm turns to global search and then turns to local search in the final stage. This coefficient strengthens the convergence capability of the MFO algorithm.

2.3 Algorithm performance comparison

Standard functions were used to test the feasibility of the HMFO algorithm. These standard functions consisted of three

Fig. 2 Sine coefficient



unimodal functions and three multimodal functions. The MFO, PSO, and ant lion optimizer (ALO) algorithms were compared. The ALO algorithm imitates the hunting behavior of an ant lion [36]. The expressions, variable value ranges, and optimal solutions of the unimodal function and multimodal functions are listed in Tables 1 and 2 [37, 38].

The HMFO, MFO, ALO, and PSO algorithms were tested 15 times with six test functions. The maximum number of iterations was 500. The population size for HMFO, MFO, ALO and PSO was 30. The same test platform was used during all simulations. The platform parameters were as follows: i5 processor, 16 GB of memory, 512 GB hard disk capacity, and MATLAB 2016a simulation software. Table 3 depicts the test data for the algorithms.

For the unimodal test functions, the average test result of the HMFO algorithm was the best. The convergence results of the MFO and ALO algorithms were similar. The test result of the PSO algorithm was poor. Similarly, for the convergence interval obtained using 15 repeated tests, the convergence interval of the HMFO algorithm was the smallest. The PSO algorithm exhibited the largest convergence interval. The difference between the convergence intervals of the MFO and ALO algorithms was small. Therefore, for the unimodal test function, the test results for the HMFO algorithm were superior; the convergence effects of the MFO and ALO algorithms were similar but inferior to that of the HMFO algorithm. The PSO algorithm exhibited poor performance.

For the multimodal functions, the HMFO algorithm exhibited strong convergence ability. For the $S4$ and $S5$ functions, the HMFO algorithm converged to 0. For the $S6$ function, although HMFO did not converge to 0, its convergence effect was more accurate than those of the other three algorithms. For the $S4$ and $S5$ functions, the convergence results of the MFO, ALO, and PSO algorithms were similar. For the $S6$ function, the average convergence accuracy of the MFO and ALO algorithms was higher than that of the PSO algorithm. For the multimodal test functions, the convergence result of the HMFO algorithm was more accurate than those of the other three algorithms. The convergence effects of the PSO, ALO, and MFO algorithms were similar.

Table 1 Unimodal functions

S	Dimension	Variable value range	Optimal solution
$S_1 = \sum_{i=1}^{D} u_i^2$	30	[−100, 100]	0
$S_2 \sum_{i=1}^{D} \lvert u_i \rvert + \prod_{i=1}^{D} \lvert u_i \rvert$	30	[−10, 10]	0
$S_3 = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} u_j \right)^2$	30	[−100, 100]	0

From the convergence data presented in Table 3, the HMFO algorithm was found to have strong convergence ability for both unimodal and multimodal functions. The MFO, ALO, and PSO algorithms exhibited poor ability to avoid the local solution. For the multimodal functions, the test results of the three algorithms were imprecise. Through the sine coefficient and Levy flight, the HMFO algorithm exhibited strong local and global convergence capabilities. Especially for multimodal functions, the HMFO algorithm exhibited strong convergence ability.

The ALO algorithm required the longest execution time. The HMFO and MFO algorithms required the least execution time. The execution times of the HMFO and MFO algorithms were nearly identical, indicating that use of the HMFO algorithm does not considerably increase the calculation cost.

2.4 Elm

ELM is an improved feed-forward neural network. As a novel single-hidden-layer neural network, the ELM model is applied in various fields, such as data mining, fault classification, and life prediction. The most crucial feature of the ELM model is that the model's weights and thresholds are initialized randomly [39–41]. The weights and thresholds need not be updated during the learning process, which speeds up calculation. The ELM model is simple, has strong generalizability, and learns quickly. Thus, the ELM model solves the complex structure and adjustable parameter problems of the traditional model. The topology of the ELM model is depicted in Fig. 3.

The ELM model consists of a three-layer network (hidden, input, and output layers), as depicted in Fig. 3. The hidden and input layers are connected by neurons, as are the hidden and output layers. The number of neurons in the hidden, output, and input layers is herein denoted m, k, and n, respectively [42, 43].The input matrix is A, and the output matrix is B. The total number of samples is N.

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,N} \end{bmatrix}_{n \times N} B = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,N} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ b_{k,1} & b_{k,2} & \cdots & b_{k,n} \end{bmatrix}_{k \times N} \tag{17}$$

Suppose C is the weight between the input and hidden layers and D is the weight between the output and hidden layers.

Table 2 Multimodal functions

S	Dimension	Variable value range	Optimal solution
$S_4=\sum_{i=1}^{D}u_i^2-10^{*}\cos(2\pi u_i)+10^{*}D$	30	[−5.12, 5.12]	0
$S_5=\frac{1}{4000}\sum_{i=1}^{D}u_i^2-\prod_{i=1}^{D}\cos\left(\frac{u_i}{\sqrt{i}}\right)+1$	30	[−600, 600]	0
$S_6=-20^{*}e^{-0.2^{*}\sqrt{\frac{1}{n}\sum_{i=1}^{D}u_i^2}}+e-e^{\frac{1}{n}\sum_{i=1}^{D}\mathrm{COS}(2\pi u_i)}+20$	30	[−32, 32]	0

$$C=\begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{i,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n} \end{bmatrix}_{m\times n} D = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,k} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ d_{m,1} & d_{m,2} & \cdots & d_{m,k} \end{bmatrix}_{m\times k} \tag{18}$$

The hidden layer threshold of the ELM model is as follows:

$$\beta=[\beta_1,\beta_2,\ldots,\beta_m]^T \tag{19}$$

When the matrix A is input to the ELM model, the prediction output matrix Q is obtained:

$$Q=\underset{k\times N}{[q_1,q_2,\ldots,q_N]}$$

$$q_j=\begin{bmatrix} q_{1j} \\ q_{2j} \\ \vdots \\ q_{kj} \end{bmatrix}_{k\times 1}=\begin{bmatrix} \sum_{i=1}^{m} d_{i1}E(c_ia_j+\beta_i) \\ \sum_{i=1}^{m} d_{i2}E(c_ia_j+\beta_i) \\ \vdots \\ \sum_{i=1}^{m} d_{k1}E(c_ia_j+\beta_i) \end{bmatrix}_{k\times 1} \tag{20}$$

where E (a) is the activation function and has infinite differentiability.

Equation 20 can be represented as follows:

$$FD=Q^T$$

$$F=\begin{bmatrix} E(c_1a_1+\beta_1) & E(c_2a_1+\beta_2)\cdots E(c_ka_1+\beta_k) \\ E(c_1a_2+\beta_1) & E(c_2a_2+\beta_2)\cdots E(c_ka_2+\beta_k) \\ \vdots\ \vdots & \vdots\ \vdots \\ E(c_1a_N+\beta_1) & E(c_2a_N+\beta_2)\ \ \cdots E(c_ka_2+\beta_k) \end{bmatrix}_{N\times k} \tag{21}$$

where F represents the output matrix of the hidden layer and Q^T represents the transpose matrix.

In the ELM model, C and β are determined randomly. The connection weight D can be calculated using the least square solution.

$$\frac{min}{}D\left\|FD-Q^T\right\| \tag{22}$$

The least square solution is as follows:

$$D=F^{+}Q^T \tag{23}$$

where F^{+} is the generalized inverse matrix.

2.5 E-commerce transaction volume forecast using the ELM-HMFO model

The deviation value and input weight of the ELM model were randomly determined. Although the number of setting parameters during the iteration was reduced, random parameters increased the number of neurons in the network, which reduced the resource utilization of the model. Similarly, random parameters could reduce the prediction stability of the model, causing a large regression error of the ELM model. To improve the forecast stability and effect of the ELM model, the HMFO algorithm was used to solve the random parameters problem.

The convergence analysis in Section 2.3 revealed that the HMFO algorithm had strong convergence ability and exhibited an excellent convergence effect for multimodal functions. By combining the HMFO algorithm with the ELM model, the ELM-HMFO model was established to predict e-commerce transaction volume. The prediction process of the ELM-HMFO model was as follows:

1. Divide the sample and determine the input and output of the model.
2. Initialize the population of the HMFO algorithm and the parameters of the ELM model.
3. Normalize the sample data.
4. Use the training set to train the e-commerce transaction volume prediction model.
5. Search the optimal parameters using the HMFO algorithm.
6. Input the optimal parameters to the ELM model.
7. Use the ELM-HMFO model to predict the e-commerce transaction volume.

Table 3 Test results

S	Algorithm	Average convergence value	Convergence interval	Average execution time (s)
S_1	HMFO	2.41e-42	[2.13e-52, 3.61e-41]	0.10
	MFO	3.27e-13	[2.64e-15, 3.01e-12]	0.09
	ALO	7.75e-9	[3.03e-9, 2.23e-8]	5.01
	PSO	2.08e-1	[1.22e-5, 2.15]	1.81
S_2	HMFO	3.61e-25	[1.22e-27, 3.13e-24]	0.11
	MFO	6.64e-9	[1.36e-9, 2.19e-8]	0.11
	ALO	7.75e-9	[3.03e-9, 2.23e-8]	4.91
	PSO	1.23	[3.05e-2, 2.78]	1.62
S_3	HMFO	8.98e-35	[1.47e-42, 9.54e-34]	0.27
	MFO	5.52e-2	[1.20e-3, 2.26e-1]	0.26
	ALO	3.71e-2	[1.22e-4, 2.29e-1]	5.11
	PSO	10.75	[5.07e-2, 96.48]	4.37
S_4	HMFO	0	–	0.13
	MFO	23.09	[7.95, 50.81]	0.13
	ALO	14.56	[4.97, 38.80]	5.24
	PSO	19.76	[3.97, 26.86]	2.12
S_5	HMFO	0	–	0.13
	MFO	1.84e-1	[7.38e-2, 4.01e-1]	0.13
	ALO	2.26e-1	[1.01e-1, 5.21e-1]	4.97
	PSO	3.21e-1	[1.34e-1, 6.10e-1]	2.09
S_6	HMFO	8.88e-16	–	0.14
	MFO	7.70e-2	[2.42e-8, 1.15]	0.14
	ALO	4.75e-1	[2.16e-5, 2.01]	5.02
	PSO	2.71	[3.40e-3, 4.16]	2.29

(8) Normalize the forecast results of e-commerce transaction volume and analyze the forecast effect.

The flowchart of the e-commerce transaction volume prediction model is depicted in Fig. 4.

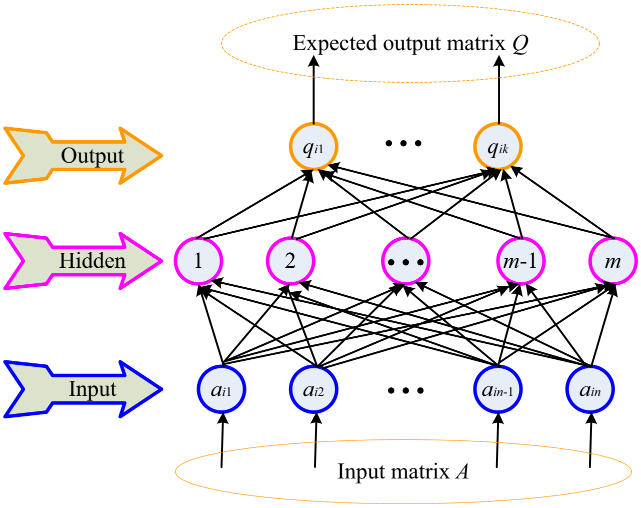


Fig. 3 Topology of the ELM model

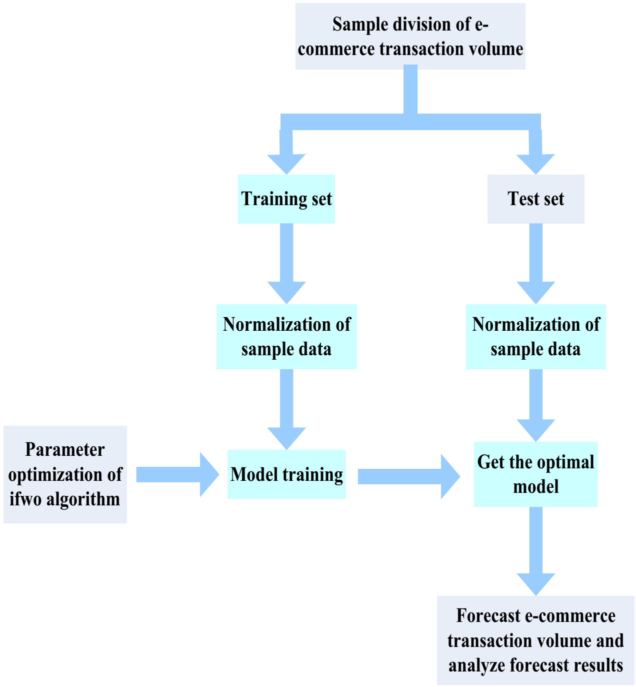

Sample division of e-commerce transaction volume
Training set
Test set
Normalization of sample data
Normalization of sample data
Parameter optimization of ifwo algorithm
Model training
Get the optimal model
Forecast e-commerce transaction volume and analyze forecast results

Fig. 4 E-commerce transaction volume forecast flow

3 Data analysis and simulation experiment

3.1 Analysis of e-commerce transaction volume

The e-commerce transaction system is a complex and dynamic system. To predict e-commerce transaction volume, the dynamics, nonlinearity, and volatility of e-commerce data should be considered. The data used in this study were obtained from the China Internet Network Information Center. We present statistics on the e-commerce data from 2004 to 2019. The e-commerce transaction volume is depicted in Fig. 5.

Figure 5 reveals that the e-commerce transaction volume increased every year, but the growth was nonlinear. Before 2011, the growth rate of the e-commerce transaction volume of China was slow, with trading volume less than RMB 10 trillion; after 2012, the e-commerce transaction volume grew rapidly and exceeded RMB 34 trillion by the end of 2019. Factors such as the number of CN domain names, number of Internet users, number of websites, Internet penetration, export bandwidth, and mobile phone penetration affect the e-commerce transaction volume of China. This study mainly analyzed the following factors influencing the e-commerce transaction volume:

(1) The number of Internet users directly affects Internet demand and e-commerce transaction volume.

(2) Websites are the platform for e-commerce and directly affect the scale of e-commerce transaction volume.

(3) CN domain names are China's top-level domain names and critical for promoting the use of the Internet. The greater the number of CN domain names, the more Internet information service companies there are. The CN domain name to a certain extent affects the volume of e-commerce transactions.

(4) The Internet penetration rate is similar to the number of Internet users. A higher Internet penetration rate indicates a higher integration degree of the Internet information platform and is also a factor affecting e-commerce transaction volume.

(5) Export bandwidth affects the information communication between countries. It also affects users' Internet access quality and experience. High export bandwidth promotes the development of e-commerce platforms, which affects e-commerce transactions.

The correlation coefficient C was used to reflect the degree of correlation between the five factors and e-commerce transaction volume. The correlation coefficient C was calculated as follows:

$$C = \frac{m\sum op - \sum o\sum p}{\sqrt{m\sum o^2 - \left(\sum o\right)^2}\sqrt{m\sum p^2 - \left(\sum p\right)^2}} \tag{24}$$

where m is the number of variables and $C \in [0, 1]$. When C is 0, two variables are unrelated; when C is 1, two variables are strongly correlated.

The coefficients of correlation between influencing factors and transaction value are presented in Table 4.

Table 4 indicates that the number of netizens, number of websites, and export bandwidth were strongly correlated with e-commerce transaction volume. The correlation coefficient C was higher than 0.9 for these factors. The correlation between export bandwidth and e-commerce transaction volume, with a correlation coefficient of 0.98, was the strongest. The correlation between e-commerce transaction volume and CN domain names was the weakest, with a correlation coefficient of 0.87.

To reduce the amount of calculation required using the e-commerce transaction volume forecast model, the input of the model was determined from the correlation coefficients. Table 4 reveals that the coefficients of correlation of e-commerce transaction value with number of websites, number of Internet users, Internet penetration rate, and export bandwidth were high. Therefore, we used the number of websites, network penetration rate, export bandwidth, and number of Internet users as input variables for the model. The e-commerce transaction volume was used as the model's output variable.

Fig. 5 E-commerce turnover

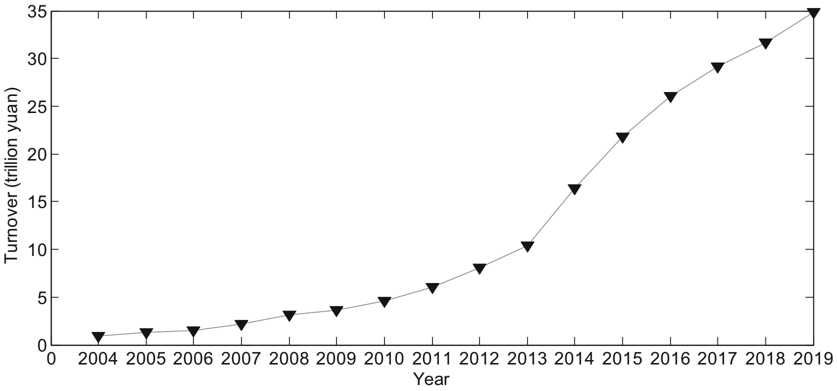


Table 4 Correlation analysis

Factor	C
Number of Internet users	0.91
Number of Websites	0.92
Number of CN domain names	0.87
Internet penetration	0.89
Export bandwidth	0.98

3.2 E-commerce transaction volume forecast

The e-commerce transaction data published on the website of the China Internet Information Center were used as the test and training data of the prediction model. The number of Internet users, number of websites, network penetration rate, and export bandwidth from 2004 to 2019 were considered as the input samples of the model. The e-commerce transaction volume from 2004 to 2019 was considered the output samples of the model. The ELM-HMFO model was used to predict the e-commerce transaction volume. The ELM-MFO model and SVM model were selected as the comparison models. The relative error (*AE*), root mean square error (*RMSE*), and decision coefficient (R^2) were employed to analyze the prediction results:

$$AE = \frac{p-a}{a} \times 100\% \tag{25}$$

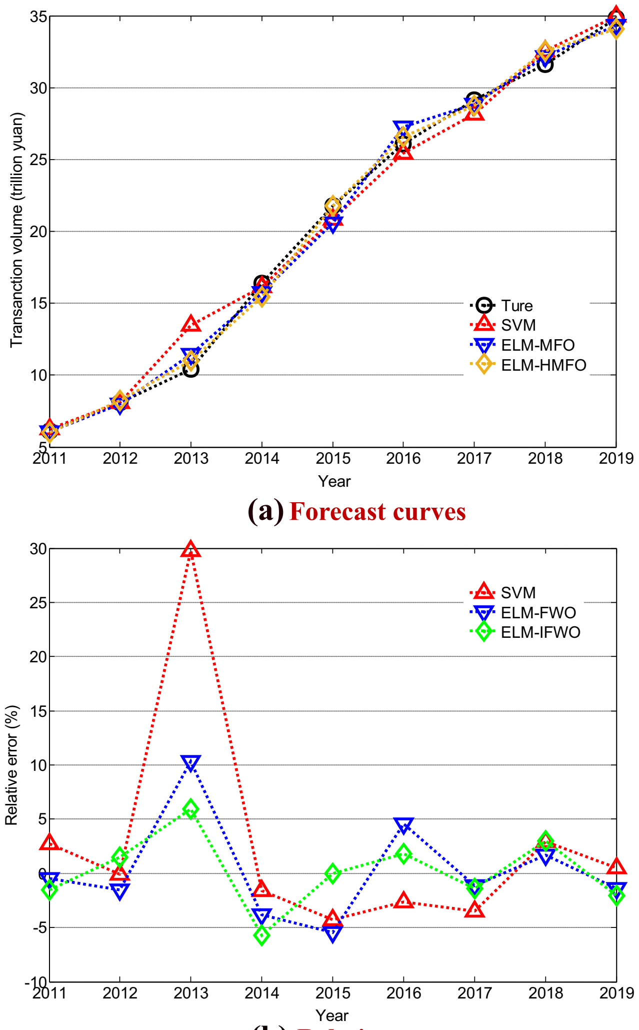
$$RMSE = \sqrt{\frac{1}{S}\sum_{i=1}^{s}(p_i-a_i)^2} \tag{26}$$

$$R^2 = 1-\frac{\sum_{i=1}^{s}(p_i-a_i)^2}{\sum_{i=1}^{s}\left(a_i-\overline{p}\right)^2} \tag{27}$$

where a is the actual transaction volume, p is the regression transaction volume, and $\overline{p}$ is the regression average transaction volume.

First, the number of Internet users, number of websites, network penetration rate, and export bandwidth from 2004 to 2010 were considered as the training input samples, and the e-commerce transaction volumes from 2004 to 2010 were used as the training output samples of the prediction model. The export bandwidth, number of websites, network penetration rate, and number of Internet users from 2011 to 2019 were used as the test input samples of the model, and the e-commerce transaction volume from 2011 to 2019 was used as the test output samples of the prediction model. The ELM-HMFO, SVM, and ELM-MFO models were used to forecast the e-commerce transaction volume from 2011 to 2019. The regression curve of each model is depicted in Fig. 6.

Figure 6(a) shows the regression of the SVM, ELM-MFO, and ELM-HMFO models. Figure 6(b) depicts the relative error curves. The regression curves reflect the upward trend of the real curve in e-commerce transaction volume from 2011 to 2019. Comparing the error curves, the *AE* errors of the three models were within [−6%, 6%]. The fluctuation of the *AE* curve of the ELM-HMFO model was the smallest, indicating that the ELM-HMFO model obtained superior predictions. On the basis of the overall prediction curves of the three models, the *RMSE* and R^2 of the models were calculated. Table 5 lists the fitting effects of the SVM, ELM-MFO, and ELM-HMFO models.



(a) Forecast curves

(b) Relative error

Fig. 6 Forecast curves of the e-commerce transactions from 2011 to 2019

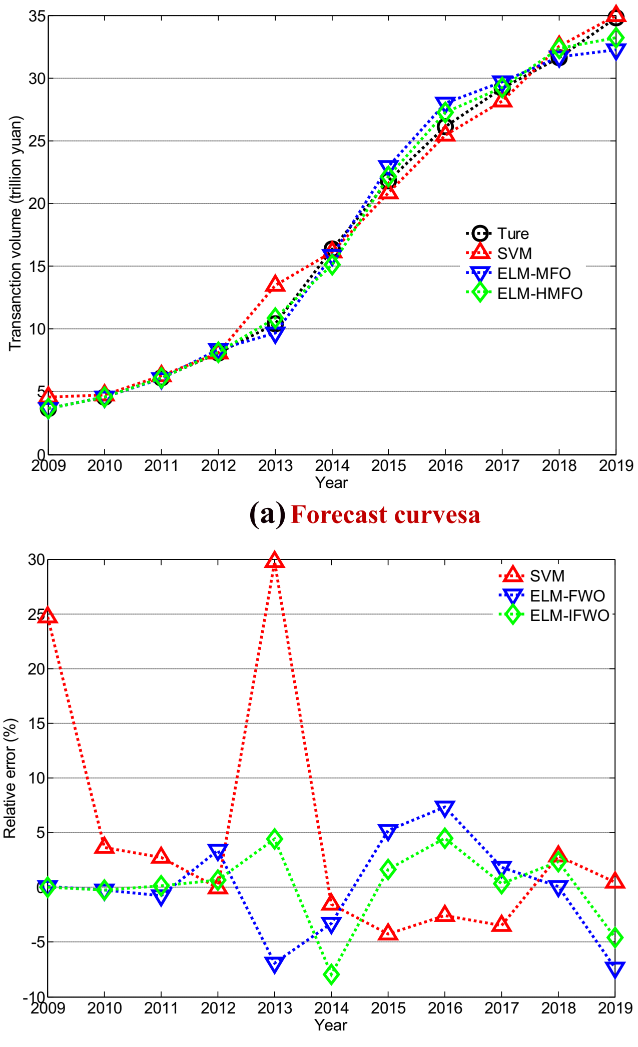
Table 5 Analysis of the forecast results of e-commerce transaction volume from 2011 to 2019

Model	*AE* interval	*RMSE*	R^2	Execution time (s)
SVM	[−4.26%, 29.74%]	1.19	0.98	1.21
ELM-MFO	[−5.48%, 10.31%]	0.74	0.98	9.48
ELM-HMFO	[−5.71%, 5.94%]	0.58	0.99	9.70

The *AE* interval of the ELM-HMFO model was the smallest, which indicated superior prediction stability. The minimum *RMSE* of the ELM-HMFO model was 0.58, which was 51.26% smaller than that of the SVM model and 21.62% smaller than that of the ELM-MFO model. The ELM-HMFO model exhibited the smallest prediction error. The R^2 of the SVM, ELM-MFO, and ELM-HMFO models was higher than 0.98, indicating that the regression fit was close. Figure 6(a) reveals that the regression results reflect the fluctuation in the real value of the e-commerce transaction volume. The execution time of the SVM algorithm was the shortest. The execution times of the ELM-MFO and ELM-HMFO models were similar, because the MFO and the HMFO algorithms form part of the ELM-MFO and ELM-HMFO models, respectively. During the training process, the two algorithms optimized the random parameters of the ELM model. Therefore, the execution times of the ELM-MFO and ELM-HMFO models were longer than that of the SVM model.

The number of Internet users, number of websites, network penetration rate, and export bandwidth from 2004 to 2008 were used as the training input samples of the model. The e-commerce transaction volume from 2004 to 2008 was used as the training output samples. The number of Internet users, number of websites, network penetration rate, and export bandwidth from 2009 to 2019 were used as the test input samples of the model. The e-commerce transaction volume from 2009 to 2019 was used as the test output samples. The ELM-HMFO, SVM, and ELM-MFO models were used to forecast the e-commerce transaction volume from 2009 to 2019. The regression curves are depicted in Fig. 7.

The predicted values of the e-commerce transaction volume obtained using the SVM, ELM-MFO, and ELM-HMFO models are illustrated in Fig. 7(a). The relative error curves are shown in Fig. 7(b). The predicted values of the ELM-HMFO and ELM-MFO models in Fig. 7(a) are close to the true values. Comparing the forecast errors in Fig. 7(b), the *AE* errors of the ELM-HMFO and ELM-MFO models were within 8%. The maximum *AE* error of the SVM model exceeded 25%. The analysis results of the ELM-HMFO, ELM-MFO, and SVM models for the e-commerce transaction volume from 2009 to 2019 are presented in Table 6.



(a) Forecast curvesa

(b) Relative error

Fig. 7 Forecast curves of the e-commerce transaction volume from 2009 to 2019

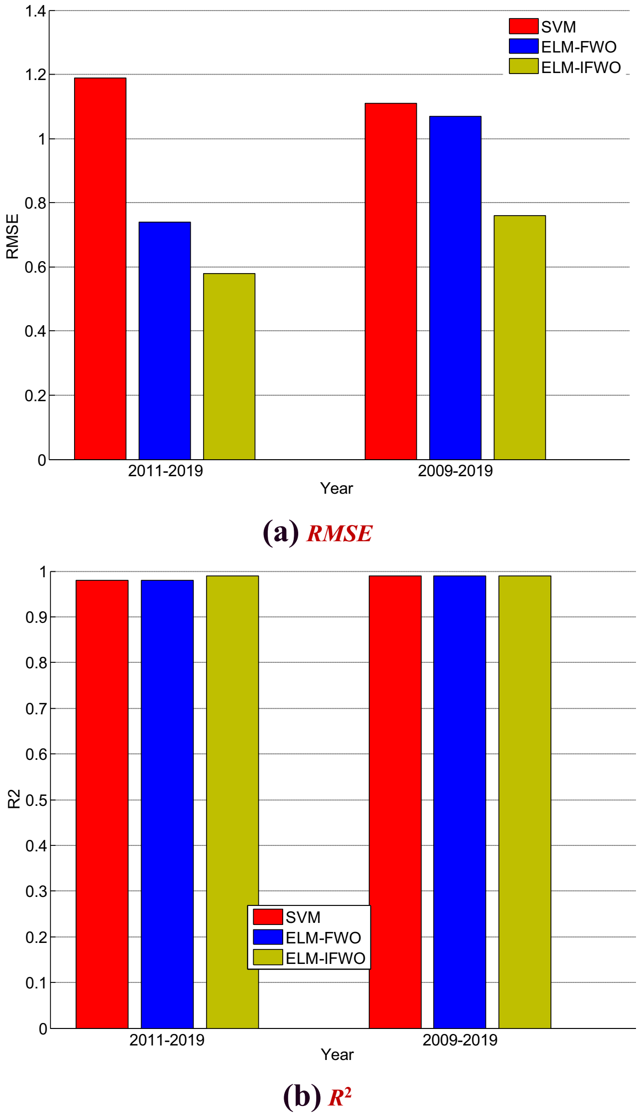
The *AE* interval of the ELM-HMFO model was [−7.92%, 4.48%], which indicated that the prediction stability was excellent. The *AE* interval of the SVM model was [−4.28%, 29.74%]. The maximum *AE* of the SVM model was 29%, which indicated that the prediction error was relatively large. The minimum *RMSE* of the ELM-HMFO model was 0.76, which was 31.53% smaller than that of the SVM model and 28.97% smaller than that of the ELM-MFO model. The R^2 of the SVM, ELM-MFO, and ELM-HMFO models was higher than 0.99, indicating that the regression results reflected the trend in actual values. Similarly, the execution time of the three algorithms was similar to that in the first group of simulations. The SVM model had the shortest execution time. The execution time of ELM-MFO and ELM-HMFO models was within 10 s.

Figure 8 depicts the regression effect analysis of the e-commerce transaction volume predicted using the SVM, ELM-MFO, and ELM-HMFO models.

The *RMSE* and R^2 of the three models for the e-commerce transaction volume from 2011 to 2019 and 2011 to 2019 are

Table 6 Analysis of the forecast results of e-commerce transactions from 2009 to 2019

Model	*AE* interval	*RMSE*	R^2	Execution time (s)
SVM	[−4.28%, 29.74%]	1.11	0.99	1.03
ELM-MFO	[−7.36%, 7.33%]	1.07	0.99	9.71
ELM-HMFO	[−7.92%, 4.48%]	0.76	0.99	9.57



(a) ***RMSE***

(b) ***R***^{2}

Fig. 8 Prediction and evaluation

depicted in Fig. 8. The prediction accuracy of the ELM-HMFO model was high, and its *RMSE* was lower than 0.8. The fitting effects of the three models were similar, and the R^2 of the three models was higher than 0.98. The comprehensive analysis showed that the fitting effect and regression error of the ELM-HMFO model were satisfactory.

4 Conclusion

As an emerging industry, e-commerce has driven industrial transformation and promoted the development of the manufacturing, logistics, and service fields. E-commerce has become a driving force behind China's economic development. E-commerce transaction volume reflects the development level of e-commerce. The government can consider the trend in e-commerce transaction volume when formulating planning policies, and the development trend of e-commerce transaction volume can provide data support for enterprise investment decisions. Therefore, analyzing the development trend of e-commerce transaction volume is necessary. For this reason, this paper proposes an efficient ELM-HMFO model for predicting e-commerce transaction volume, and the model achieved favorable prediction results. The main conclusions and contributions obtained in this study are as follows:

(1) The novel HMFO algorithm was proposed. The performance of MFO algorithm was improved by introducing a sine coefficient and the Levy strategy.

(2) For the multipeak test functions S_4 and S_5, the HMFO algorithm converged to the optimal value 0, which demonstrated strong optimization ability.

(3) The ELM-HMFO model was proposed to accurately forecast the e-commerce transaction volume. The *RMSE* of the ELM-HMFO model was 51.26% smaller than that of the SVM model and 21.62% smaller than that of the ELM-MFO model for the e-commerce transaction volume from 2011 to 2019; the *RMSE* of ELM-HMFO model was 31.53% smaller than that of the SVM model and 28.97% smaller than that of the ELM-MFO model for the e-commerce transaction volume from 2009 to 2019.

(4) The ELM-HMFO model can be used by the government to formulate planning policies and assist the investment decisions of enterprises. The model can play a critical role in evaluating the development trend of e-commerce.

The prediction results for e-commerce transaction volume and obtained using the HMFO and ELM-HMFO models were compared, showing that the proposed model exhibited higher prediction accuracy. However, the proposed model also has certain limitations. In future research, the MFO algorithm should be enhanced to reduce the execution time and improve convergence accuracy. The proposed model is simple. In future research, we will focus on developing a hybrid model with higher prediction accuracy.

Acknowledgments This study was supported by the Ministry of Education Research of Industry–University cooperation and Cooperative Education Action Project of China [Project No. 201702051010].

References

1. Chen LF (2019) Green certification, e-commerce, and low-carbon economy for international tourist hotels. Environ Sci Pollut Res 26(18):17965–17973
2. Geng, RB; Wang, SC; Chen, X; Song, DY; Yu, J (2020). Content marketing in e-commerce platforms in the internet celebrity economy. Ind Manag Data Syst, 22

3. Yang, ZZ; Yu, S; Lian, F (2020). Online shopping versus in-store shopping and its implications for urbanization in China: based on the shopping behaviors of students relocated to a remote campus. Environ Dev Sustain, 21
4. Wang XT, Wang H (2019) A study on sustaining corporate innovation with E-commerce in China. Sustainability. 11(23):16
5. Khouja M, Liu X (2020) A Retailer's decision to join a promotional event of an E-commerce platform. Int J Electron Commer 24(2): 184–210
6. Cao LL (2014) Business model transformation in moving to a Cross-Channel retail strategy: a case study. Int J Electron Commer 18(4):69–95
7. Ji SW, Wang XJ, Zhao WP, Guo D (2019) An application of a three-stage XGBoost-based model to sales forecasting of a cross-border E-commerce Enterprise. Math Probl Eng 2019:15
8. Chang PC, Liu CH, Fan CY (**2009**) Data clustering and fuzzy neural network for sales forecasting: A case study in printed circuit board industry. *k* 22(5):344–355
9. Chen IF, Lu CJ (2017) Sales forecasting by combining clustering and machine-learning techniques for computer retailing. Neural Comput Applic 28(9):2633–2647
10. Di Pillo G, Latorre V, Lucidi S, Procacci E (2016) An application of support vector machines to sales forecasting under promotions. *4or-a Quarterly Journal of Oper Res* 14(3):309–325
11. Zhang YZ (2019) Application of improved BP neural network based on e-commerce supply chain network data in the forecast of aquatic product export volume. Cogn Syst Res 57:228–235
12. Li, MB; Ji, SW; Liu, G (2018). Forecasting of Chinese E-commerce sales: an empirical comparison of ARIMA, nonlinear autoregressive neural network, and a combined ARIMA-NARNN model. Math Probl Eng, 12
13. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8(3):256–279
14. Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31
15. Jones G, Willett P, Glen RC, Leach AR, Taylor R (1997) Development and validation of a genetic algorithm for flexible docking. J Mol Biol 267(3):727–748
16. Dorigo M, Blum C (2005) Ant colony optimization theory: a survey. Theor Comput Sci 344(2–3):243–278
17. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
18. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
19. Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35
20. Abd El Aziz M, Eweesc AA, Hassanien AE (**2017**) Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation. *Expert Systems with Applications* 83:242–256
21. Li CB, Li SK, Liu YQ (2016) A least squares support vector machine model optimized by moth-flame optimization algorithm for annual power load forecasting. Appl Intell 45(4):1166–1178
22. Zhang L, Mistry K, Neoh SC, Lim CP (2016) Intelligent facial emotion recognition using moth-firefly optimization. Knowl-Based Syst 111:248–267
23. Liu ZF, Li LL, Tseng ML, Lim MK (2020) Prediction short-term photovoltaic power using improved chicken swarm optimizer - extreme learning machine model. J Clean Prod 248:14
24. Li LL, Sun J, Tseng ML, Li ZG (2019) Extreme learning machine optimized by whale optimization algorithm using insulated gate bipolar transistor module aging degree evaluation. Expert Syst Appl 127:58–67
25. Baliarsingh SK, Vipsita S (2020) Chaotic emperor penguin optimised extreme learning machine for microarray cancer classification. IET Syst Biol 14(2):85–95
26. Tang JX, Deng CW, Huang GB (2016) Extreme learning machine for multilayer perceptron. Ieee Transactions on Neural Networks and Learning Systems 27(4):809–821
27. Suresh S, Babu RV, Kim HJ (2009) No-reference image quality assessment using modified extreme learning machine classifier. Appl Soft Comput 9(2):541–552
28. Wan C, Xu Z, Pinson P, Dong ZY, Wong KP (2014) Optimal prediction intervals of wind power generation. IEEE Trans Power Syst 29(3):1166–1174
29. Zhang RX, Huang GB, Sundararajan N, Saratchandran P (2007) Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis. Ieee-Acm Transactions on Computational Biology and Bioinformatics 4(3): 485–495
30. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
31. Allam D, Yousri DA, Eteiba MB (2016) Parameters extraction of the three diode model for the multi-crystalline solar cell/module using moth-flame optimization algorithm. Energy Convers Manag 123:535–548
32. Yildiz BS, Yildiz AR (2017) Moth-flame optimization algorithm to determine optimal machining parameters in manufacturing processes. Materials Testing 59(5):425–429
33. Mei RNS, Sulaiman MH, Mustaffa Z, Daniyal H (2017) Optimal reactive power dispatch solution by loss minimization using moth-flame optimization technique. Appl Soft Comput 59:210–222
34. Khalilpourazari S, Khalilpourazary S (2019) An efficient hybrid algorithm based on water cycle and moth-flame optimization algorithms for solving numerical and constrained engineering optimization problems. Soft Comput 23(5):1699–1722
35. Savsani V, Tawhid MA (2017) Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems. Eng Appl Artif Intell 63:20–32
36. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
37. Li LL, Liu ZF, Tseng ML, Chiu ASF (2019) Enhancing the Lithium-ion battery life predictability using a hybrid method. Appl Soft Comput 74:110–121
38. Li LL, Wen SY, Tseng ML, Wang CS (2019) Renewable energy prediction: a novel short-term prediction model of photovoltaic output power. J Clean Prod 228:359–375
39. Huang GB (2014) An insight into extreme learning machines: random neurons. Random Features and Kernels Cognitive Computation 6(3):376–390
40. Huang GB, Ding XJ, Zhou HM (2010) Optimization method based extreme learning machine for classification. Neurocomputing. 74(1–3):155–163
41. Huang GB, Zhou HM, Ding XJ, Zhang R (2012) Extreme learning machine for regression and multiclass classification. Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics 42(2):513–529
42. Huang GB, Wang DH, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Learn Cybern 2(2):107–122
43. Huang GB, Chen L (2007) Convex incremental extreme learning machine. Neurocomputing. 70(16–18):3056–3062

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Bo Zhang received his B.S. degree in 2004 from Tianjin University, received his M.S. degree in 2009 from Hebei University of Technology, received his Ph.D. degree in 2013 from Hebei University of Technology. Now he is a lecturer in Tianjin University of Technology and Education. His main research interests include electronic commerce, technological innovation, etc.

Runhua Tan received his B.S. degree in 1982 and M.S. degree in 1984 from Hebei University of Technology, received his Ph.D. degree in 1998 from Zhejiang University. Now he is Professor in Hebei University of Technology and Director of the National Engineering Research Center for Technological Innovation Method and Tool. His main research interests include C-TRIZ / TRIZ, technological innovation process and method, innovative design, customized design, etc.

Cheng-Jian Lin received his Ph.D. degree in electrical and control engineering from National Chiao-Tung University, Taiwan, in 1996. Currently, he is a Chair Professor of the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung City, Taiwan. His current research interests are machine learning, pattern recognition, intelligent control, image processing, and evolutionary robots. (cjlin@ncut.edu.tw)