Research Article

Prediction of Chaotic Time Series Based on BEN-AGA Model

LiYun Su and **Fan Yang**

School of Sciences, Chongqing University of Technology, Chongqing 400054, China

Correspondence should be addressed to LiYun Su; cloudhopping@163.com

Received 21 October 2020; Revised 11 June 2021; Accepted 27 August 2021; Published 15 September 2021

Academic Editor: Nishant Malik

Copyright © 2021 LiYun Su and Fan Yang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the prediction problem of chaotic time series, this paper proposes a brain emotional network combined with an adaptive genetic algorithm (BEN-AGA) model to predict chaotic time series. First, we improve the emotional brain learning (BEL) model using the activation function to change the two linear structures the amygdala and the orbitofrontal cortex into the nonlinear structure, and then we establish the brain emotional network (BEN) model. The brain emotional network model has stronger nonlinear calculation ability and generalization ability. Next, we use the adaptive genetic algorithm to optimize the parameters of the brain emotional network model. The weights to be optimized in the model are coded as chromosomes. We design the dynamic crossover probability and mutation probability to control the crossover process and the mutation process, and the optimal parameters are selected through the fitness function to evaluate the chromosome. In this way, we increase the approximation capability of the model and increase the calculation speed of the model. Finally, we reconstruct the phase space of the observation sequence based on the short-term predictability of the chaotic time series; then we establish a brain emotional network model and optimize its parameters with an adaptive genetic algorithm and perform a single-step prediction on the optimized model to obtain the prediction error. The model proposed in this paper is applied to the prediction of Rossler chaotic time series and sunspot chaotic time series. The experimental results verify the effectiveness of the BEN-AGA model and show that this model has higher prediction accuracy and more stability than other methods.

1. Introduction

Determining the irregular movement in the system is called chaos. The chaotic time series is the manifestation of the discrete situation of chaos. The time series with chaotic characteristics generated by the chaotic model is the chaotic time series. The rich dynamics information is implied by the chaotic time series. It can be said that chaotic time series is a bridge from chaos theory to the real world and an important application field of chaos. The research and application of the prediction of chaotic time series are becoming more and more important. Choosing a single time series as the original data of time series prediction is a conventional way to study time series. A single time series presents a simple relationship between time and forecast. Observation of customs clearance shows that they are not a simple linear relationship. Therefore, when making time series forecasting, we pay more attention to the time series generated under nonlinear

systems, especially time series produced by chaotic systems. The theory of chaotic time series believes that the evolution of a single variable in a time series contains all the information about the long-term evolution of all variables in the system. Through the proposed phase space reconstruction theory by Packard, we can use any long-term evolutionary variable in the system to reduce the chaos of the whole system to study the chaotic behavior of the system [1]. Chaos theory provides new ideas for time series prediction. Chaotic time series are widely used in geomagnetic storm warning [2], short-term traffic flow prediction [3], sunspot data [4], prediction, and so on.

In recent years, scholars at home and abroad have proposed a variety of chaotic prediction models, such as: moving average model [5], autoregressive model [6], autoregressive moving average model [7], neural network (feedforward neural network, recurrent neural network, and constructional neural network) [8], and support vector

machine [9]. The moving average model, autoregressive model, and autoregressive moving average model are proven and mature models, which are widely used in linear time series forecasting. They have the advantages of high forecasting accuracy and short forecasting time, but for nonlinear time series forecasting accuracy not high, there are limitations in time series forecasting. A neural network is a powerful tool for nonlinear time series forecasting. It has the advantages of good nonlinear operation ability and high prediction accuracy. For example, Shen et al. [10] improved the extreme learning machine to predict the chaotic time series and verified the strong robustness of the proposed model. For example, Wang et al. [11] used convolution and short and long memory hybrid neural networks combined with genetic algorithm to predict the photovoltaic power generation, and the results showed that the method they adopted had better prediction performance for the photovoltaic power generation. However, because the neural network is limited by the number of training samples, there are problems of overfitting and falling into local optimal solutions, and the structure needs to be manually specified. Also with the increase of system complexity, the computing time increases significantly, which limits the applicability of neural networks. Support vector machines are also generally used for nonlinear time series. Its advantages are that it follows the principle of structural risk minimization, has better generalization performance, and can obtain global optimal solutions. For example, Guo [12] used a genetic algorithm to optimize support vector machines to carry out time series prediction and achieved higher prediction accuracy than before. Similarly, Yan et al. [13] used the support vector machine model optimized by a genetic algorithm to predict the short-term wind speed and obtained the predicted value of the short-term wind speed, which is of great practical significance. But it has disadvantages such as slow calculation speed; prediction accuracy is too dependent on the choice of parameters, and the model parameters do not have the disadvantage of a unified determination method. Therefore, it is necessary to further study new models with simple structure, good generalization performance, accuracy, and efficiency.

The brain emotional learning (BEL) model [9] is based on the structure of the limbic system of the mammalian brain to simulate the mechanism of the amygdala and the orbitofrontal cortex to generate emotions, consolidate memory, and avoid repeated learning. It has the advantages of simple structure, low calculation complexity, and fast calculation speed. Due to the advantages of the BEL model, it has a wide range of applications in time series forecasting, such as the prediction of geomagnetic storms and the forecast of short-term traffic flow. For example, Ying et al. [14] used the BEL model to predict chaotic time series and applied it to the field of geomagnetic alarm. For example, Yin and Tan [15] classified effective data through the BEL model and found that the BEL model had high classification accuracy and faster operation speed than the traditional algorithm. However, in the face of nonlinear time series, the prediction accuracy of the BEL model decreases, and the prediction results in the application depend on the setting of

the reward signal parameters, and there is no uniform setting standard for the reward signal parameters. In the past, researchers proposed different reward signal setting methods according to different problems. This approach limits the applicability of the BEL model. In order to improve the nonlinear time series prediction accuracy of the BEL model, the problem of reward signal setting is avoided. In this paper, we improved the BEL model based on the existing research results to build a brain emotional network (BEN) model by changing the linear amygdala and the orbitofrontal cortex structure in the model to a nonlinear structure with the activation function and weight, and weights are added to the output part to describe different effects. In order to improve the genetic algorithm in the face of nonlinear systems falling into local optimal solutions problem, we constructed an adaptive genetic algorithm with dynamic crossover probability and mutation probability to improve the flexibility of the genetic algorithm and enhance the global and local search capabilities. We combine adaptive genetic algorithm and brain emotional network model to build BEN-AGA model using adaptive genetic algorithm to optimize the parameters of the brain emotional network model. First, the weights to be optimized are encoded as chromosomes; then the crossover and mutation processes are controlled through dynamic crossover probability and dynamic mutation probability, and finally the optimal parameters are selected through the fitness function to evaluate the chromosome. The BEN-AGA model proposed in this paper is used in Rossler chaotic time series prediction and sunspot sequence prediction. The experimental results show that the BEN model has high prediction accuracy and fast calculation speed and can effectively reflect the trend of chaotic time series. Compared with other models, it has obvious superiority.

2. Chaotic Time Series Prediction

The observed signal obtained in the application of time series forecasting is often a single time series, such as $\{y(t), t = 1, 2, \dots, n\}$. However, there is no simple linear relationship between the generally predicted object and the observed signal, such as time and power consumption. Therefore, the chaotic time series generated under the nonlinear system, especially the chaotic system, is very important when predicting the time series. The theory of chaotic time series tells us that the long-term evolution of a single variable in the time series contains all the information about the long-term evolution of the entire system.

2.1. Phase Space Reconstruction. The state of the system at a certain moment is called the phase, and the space formed by all phases is called the phase space, which is the geometric space that determines the state of the system. The phase space of a chaotic system is generally high dimensional. The original single-variable time series is actually the final reflection of the interaction of many influencing factors, which contains the traces of the influence of all the variables participating in the influence. Packard proposed the theory

of phase space reconstruction to study chaotic systems (chaotic behavior of the system) by using a single long-term evolution time series in the system. Phase space reconstruction is to embed the time series into a higher dimensional phase space so that the information contained in the original time series can be fully revealed. Phase space reconstruction is a very important step in processing chaotic time series. The quality of phase space reconstruction directly affects the prediction results. The theory put forward by Takens [16] is very instructive in phase space reconstruction. The theorem holds that the development and

change of any influencing factor in the system are determined by the development and change process of other influencing factors related to it. The principle is as follows: for a time series $\{x(t), t = 1, 2, 3, \dots, n\}$, where n is the length of the time series. Reconstructed phase space is as follows:

$$X(t) = \{x(t), x(t - \tau), \dots, x(t - (m - 1)\tau)\}, \quad (1)$$

$$t = 1 + (m - 1)\tau, 2 + (m - 1)\tau, \dots, n.$$

Then the phase space state can be expressed as follows:

$$X = [X(1 + (m - 1)\tau), X(2 + (m - 1)\tau), \dots, X(n)]',$$

$$\begin{pmatrix} X(1 + (m - 1)\tau) \\ X(2 + (m - 1)\tau) \\ \dots \\ X(n) \end{pmatrix} = \begin{bmatrix} x(1 + (m - 1)\tau) & x(1 + (m - 2)\tau) & \dots & x(1) \\ x(2 + (m - 1)\tau) & x(2 + (m - 2)\tau) & \dots & x(2) \\ \dots & \dots & \dots & \dots \\ x(n) & x(n - \tau) & \dots & x(n - (m - 1)\tau) \end{bmatrix}, \quad (2)$$

where $X(t)$ is a point in the phase space, m is the embedding dimension, and τ is the delay time.

It is obvious from the above formula that the determination of embedding dimension and delay time is the key to phase space reconstruction. A good embedding dimension and delay time can make the reconstructed phase space preserve the mutual influence of the original data without generating redundant information and noise. Selecting feature vectors in the reconstructed phase space for calculation and prediction can greatly improve the reliability and accuracy of prediction. In this paper, the complex autocorrelation method [17] is used to solve the delay time τ , and the method of Cao [18] is used to solve the embedding dimension m .

2.2. Single-Step Prediction Model Construction. For the observation signal $\{y(t), t = 1, 2, \dots, n\}$, according to Section 2.1, a phase point in phase space can be expressed as $Y(t) = (y(t), y(t - \tau), \dots, y(t - (m - 1)\tau))$, where $t = n_1, n_1 + 1, \dots, n$, $n_1 = 1 + (m - 1)\tau$. According to the Takens theorem [19], for each point in the reconstructed phase space, there is a smooth mapping $f: R^m \rightarrow R$, such that $y(t + 1) = f(Y(t))$, $t = n_1, n_1 + 1, \dots, n - 1$. As long as f is found or the approximate mapping f' of f is found, the next observation signal can be single-step predicted. Therefore, the approximate mapping f' of the observed signal $y(t)$ after phase space reconstruction is established as follows:

$$y(t + 1) \approx f'(Y(t)), \quad t = n_1, n_1 + 1, \dots, n - 1. \quad (3)$$

The single-step prediction value obtained is $\hat{f}'(Y(t))$, so the corresponding prediction error $e(t + 1)$ is as follows:

$$e(t + 1) = y(t + 1) - \hat{f}'(Y(t)), \quad t = n_1, n_1 + 1, \dots, n - 1. \quad (4)$$

Equation (3) is the single-step prediction model we need, and equation (4) is the prediction error of the model. From equation (3), the selection of approximate mapping f' will directly affect the prediction effect, so choosing a good mapping f' is essential.

3. Brain Emotional Network (BEN) Model

3.1. Brain Emotional Model. The brain emotional learning (BEL) model is inspired by neurophysiology and proposed by Buzug and Pfister [20]; it has many extensive applications in the prediction of chaotic time series. The three key parts of the emotional learning model are the amygdala, orbitofrontal cortex, and reward signals. The amygdala is an important organization that controls emotional responses. It can receive different sensory stimuli, generate emotions, form memories, and consolidate memories. The information from the outside world is transformed into nerve stimulation through the human senses and reaches the amygdala through long and short reflex arcs to become emotions. In the short reflex arc, sensory stimuli reach the thalamus and are directly transmitted to the amygdala; in the long reflex arc, sensory stimuli pass through the thalamus to reach vision the cortex is processed and transferred to the amygdala. The orbitofrontal cortex can also receive sensory stimulation through long and short reflex arcs, but it is mainly used to adjust the amygdala and assist the amygdala to judge emotions. The amygdala and the orbitofrontal cortex are the main parts of the model. Emotional learning occurs in two main parts. The amygdala is responsible for generating emotions and output according to stimuli, and the orbitofrontal cortex is responsible for regulating the learning of the amygdala, avoiding over- and underlearning. Reward signals as emotions naturally produced by the brain can directly act on the amygdala and orbitofrontal cortex. The model is shown in Figure 1.

FIGURE 1: Brain emotional learning (BEL) model.

In Figure 1, the sensory input signal is X ; the reward signal is R_{ew} ; A_{th} is the maximum sensory input signal; A is the output value of the amygdala; O is the output value of the orbitofrontal cortex; v is the weight between the sensory cortex and the amygdala; w is the weight between the sensory cortex and the orbitofrontal cortex; and the output value is recorded as T . It can be seen that the amygdala receives X , A_{th} , and R_{ew} ; the orbitofrontal cortex receives X and R_{ew} , and the output of the emotional learning model is responded by sensory input signals. The emotional learning process is the weight adjustment process of the emotional learning model. According to the definition of the brain emotional learning model [20], the weight adjustment law of the amygdala and orbitofrontal cortex is as follows:

$$\Delta v = \alpha \left[X \cdot \max \left(0, R_{ew} - \sum_i A_i \right) \right], \quad (5)$$

$$\Delta w = \beta \left[X \cdot \sum_i (O_i - R_{ew}) \right], \quad (6)$$

where Δv represents the weight change rate of the amygdala; Δw represents the weight change rate of the orbitofrontal cortex; A_i and O_i represent the output nodes of the amygdala and orbitofrontal cortex, respectively; and α and β represent the learning rate of the amygdala and orbitofrontal cortex, respectively. Observing the above two formulas, we can see that reward signal R_{ew} plays an important role in the adjustment of the weight of the amygdala and orbitofrontal cortex. Therefore, the prediction result of the model depends on the choice of reward signal, and there is no unified standard for the choice of reward signal, which makes the model the versatility is not high. The general expression of the model is as follows:

$$T = A - O = \sum_{i=1}^m A_i \cdot v_i + A_{th} \cdot v_{m+1} - \sum_{i=1}^m O_i \cdot w_i. \quad (7)$$

3.2. BEN Model Establishment. The emotional learning model has been widely used in short-time traffic flow prediction, geomagnetic storm warning, and other fields and has achieved certain effects in the application. However, there are still problems such as low generality of the model and further improvement of prediction accuracy due to the lack of a unified standard for reward signals. Considering the improvement of the affective learning model, we proposed the BEN network model to improve the prediction accuracy and avoid the problem of setting reward signals, so that the established BEN network model has a stronger universality in the application of related fields.

The amygdala and the orbitofrontal cortex are the core of the emotional learning model. Sensory signal input is processed by the amygdala and the orbitofrontal cortex and then interacts to become the emotional output. The processing of the amygdala and the orbitofrontal cortex uses a simple linear weighting function to describe the emotional learning process, but it is more realistic to use a nonlinear model to describe the emotional learning process; because the amygdala and the orbitofrontal cortex are simple linear weighting functions, we regard them as two simple neural networks and act on the activation functions on the two neural networks to change the simple linear structure into a nonlinear structure. Considering the influence of the amygdala and the orbitofrontal cortex in the process of emotional learning is not equal, the basis of the activation function, they are given output weights to describe the influence in the process of emotional learning, so as to establish the BEN model. The BEN model has stronger nonlinear computing capabilities and better model structure and can better describe the emotional learning process. The prediction results of the model are too dependent on the reward signal R_{ew} , and the reward signal R_{ew} is no unified standard. So reward signal R_{ew} is initialized randomly, and the adaptive genetic algorithm is used to iteratively search for the optimal reward signal R_{ew} to avoid artificially setting reward signals to affect the accuracy of prediction results.

FIGURE 2: BEN model.

The structure of the established BEN model is shown in Figure 2.

Here, the two neural networks of the orbitofrontal cortex and the amygdala are net1 and net2, respectively, and linear addition is used internally, and the linear output is selected as the output. The final output is the sum of the linear weights of the two neural networks.

In the BEN model, the original signal becomes the sensory input signal after phase space reconstruction:

$$X_i = \{x_1, x_2, x_3, \dots, x_m\}, \quad (8)$$

where m is the phase space reconstruction dimension also the number of sensory input signals.

The maximum value of sensory input signal X_i directly acts on the output result as follows:

$$A_{th} = \max(X_i). \quad (9)$$

The current amygdala and the orbitofrontal cortex are regarded as two neural networks, net1 and net2, respectively. For each sensory input signal X_i , net1 and net2 have corresponding sensory input signals. The acceptance points are A_i, O_i , so

$$\text{Net1: } A_i = x_i \cdot v_i, \quad i = 1, 2, 3, \dots, m, \quad (10)$$

$$\text{Net2: } O_i = x_i \cdot w_i, \quad i = 1, 2, 3, \dots, m, \quad (11)$$

where v_i represents the weight between the nodes of net1 and w_i represents the weight between the nodes of net2.

So the internal output T_a of net1 can be linearly expressed as follows:

$$T_a = \sum_{i=1}^m A_i + b_a = \sum_{i=1}^m x_i \cdot v_i + b_a, \quad (12)$$

where b_a is a random item with an initial value randomly give.

The internal output T_o of net2 can be linearly expressed as follows:

$$T_o = \sum_{i=1}^m O_i + b_o = \sum_{i=1}^m x_i \cdot w_i + b_o, \quad (13)$$

where b_o is a random item with an initial value randomly give.

So the entire output T can be expressed linearly as follows:

$$\begin{aligned} T &= N_1 T_a - N_2 T_o + w_{m+1} A_{th} \\ &= N_1 \left(\sum_{i=1}^m A_i + b_a \right) - N_2 \left(\sum_{i=1}^m O_i + b_o \right) + w_{m+1} A_{th} \\ &= N_1 \left(\sum_{i=1}^m x_i \cdot v_i + b_a \right) - N_2 \left(\sum_{i=1}^m x_i \cdot w_i + b_o \right) + w_{m+1} A_{th}, \end{aligned} \quad (14)$$

where N_1 and N_2 represent the output weight of the amygdala and orbitofrontal cortex, respectively.

From equation (14), the current model is a linear relationship. We improve the model to strengthen the model's nonlinear calculation ability and generalization ability to better describe the chaotic time series. For the output of the entire model, we still adopt the linear function model. The output function of the model is a linear function and is marked as $g(\cdot)$. The hidden layer functions in net1 and net2 are also linear functions, marked as $f_1(\cdot), f_2(\cdot)$, respectively. In the two separate neural networks, net1 and net2, activation function $h(x)$ is used to make the network nonlinear and improve generalization ability. After the activation function is applied, the output of the hidden layer function of net1 and net2 is $T_1 = h(f_1(\cdot)), T_2 = h(f_2(\cdot))$, respectively. The hidden layer takes the activation function

sigmoid, and the expression is $h(x) = 1/(1 + e^{-x})$. So the expression of the whole model can be written as follows:

$$T = g(T_1, T_2, A_{th}), \quad (15)$$

where w and v are the weight vector from the layer to the hidden layer, x is the sensory input signal vector. It can be seen from the model that the output linear function is $g(\cdot) = T_1 - T_2 + A_{th}$. So equation (15) can be written as follows:

$$\begin{aligned} T &= N_1 h(f_1(\cdot)) - N_2 h(f_2(\cdot)) + w_{m+1} A_{th} \\ &= N_1 h(v \cdot x) - N_2 h(w \cdot x) + w_{m+1} A_{th}. \end{aligned} \quad (16)$$

Since our activation function chooses sigmoid as the activation function, so

$$T = \frac{N_1}{1 + e^{-vx}} - \frac{N_2}{1 + e^{-wx}} + w_{m+1} A_{th}. \quad (17)$$

Equation (17) can be written in detail as follows:

$$\begin{aligned} T &= \frac{N_1}{1 + e^{-(vx+b_a)}} - \frac{N_2}{1 + e^{-(wx+b_o)}} + w_{m+1} A_{th} \\ &= \frac{N_1}{1 + e^{-(\sum_{i=1}^m X_i \cdot v_i + b_a)}} - \frac{N_2}{1 + e^{-(\sum_{i=1}^m X_i \cdot w_i + b_o)}} + w_{m+1} A_{th}. \end{aligned} \quad (18)$$

Equation (18) is the established BEN model. The input of the BEN model is $X_i = \{x_1, x_2, x_3, \dots, x_m\}$; the output of the BEN model is T ; and the parameters of the BEN model are w, v, b_a, b_o, N_1, N_2 . We consider using evolutionary algorithms to optimize the parameters of the network model. Common evolutionary algorithms are genetic algorithm (GA) and differential evolution (DE) algorithm. Comparing the genetic algorithm with the differential evolution algorithm, in an iterative way, the genetic algorithm is the parent generation of new children, and the differential evolution algorithm is the parent generation of its own evolution. In the selection of the genetic algorithm through probability to eliminate the inferior, differential evolution algorithm is the absolute elimination of the inferior. In terms of the difference between the core of the algorithm, the core of the genetic algorithm is crossover operation, while the differential evolution algorithm is mutation operation. The robustness and convergence speed of genetic algorithm are lower than that of differential evolution algorithm, but it has stronger global search ability. Considering that the BEN network constructed in this paper is to study the prediction of chaotic time series, the genetic algorithm with stronger global search ability is adopted as the learning algorithm to optimize the parameters of the network model.

4. BEN Model Combined with Adaptive Genetic Algorithm (BEN-AGA Model)

4.1. Adaptive Genetic Algorithm. Genetic algorithm is a kind of randomized search method evolved from the evolutionary laws of the biological world (survival of the fittest, genetic mechanism of survival of the fittest). It was first proposed by

Professor J. Holland in the United States in 1975 [21]. Its main feature is to directly operate on structural objects without the limitation of derivation and function continuity; it has inherent implicit parallelism and better overall optimum search capability: using probabilistic optimization methods, it can automatically obtain and guide the optimized search space and adaptively adjust the search direction, and no definite rules are required. These properties of the genetic algorithm have been widely used in combinatorial optimization [21], machine learning [22], signal processing [19], adaptive control [23], and other fields. It is a key technology in modern intelligent computing.

The entire process of genetic algorithm is divided into chromosome encoding [24], fitness value calculation [25], selection operation [12], crossover operation [13], mutation operation [26], and the last three operations are collectively referred to as the genetic operator. The process of the genetic algorithm is shown in Figure 3.

4.2. Adaptive Genetic Operator.

- (1) In the selection operation, roulette and the optimal individual retention strategy are used to select the mating group. In the roulette method, the probability of each chromosome being selected is proportional to its fitness value. The fitness value of the i -th chromosome is fit_i ; then the individual selection probability p_i is expressed as follows:

$$\begin{cases} fit_i = \frac{k}{F(ch_i)}, \\ p_i = \frac{fit_i}{\sum_{i=1}^{pop} fit_i}, \end{cases} \quad (19)$$

where pop is the population size, k is the correlation coefficient, and $F(ch_i)$ is the fitness value calculated by the fitness function based on the current chromosome.

- (2) We use the maximum fitness value of the group and the median of the group fitness to describe the dynamic adaptive probability

In the crossover operation, the dynamic crossover probability p_{cross} is designed as follows:

$$p_{cross} = \begin{cases} p_{cross \max} - \frac{p_{cross \min} (fit - fit_{mid})}{fit_{\max} - fit_{mid}}, & fit \geq fit_{mid}, \\ p_{cross \max}, & fit < fit_{mid}, \end{cases} \quad (20)$$

where $p_{cross \max}$ and $p_{cross \min}$ are the preset maximum crossover probability and minimum crossover probability, respectively; $fit_{\max}$ and fit_{mid} are the group's maximum fitness value and the group's median of fitness value, respectively; and fit is the larger fitness value among the parents of the current

FIGURE 3: Genetic algorithm process.

parent. Using arithmetic crossover method, the following formula is obtained:

$$\begin{cases} X'_1 = rX_1 + (1-r)X_2, \\ X'_2 = rX_2 + (1-r)X_1, \end{cases} \quad (21)$$

where X_1 and X_2 represent the chromosomes of the two parents, X'_1 and X'_2 represent the chromosomes of the two offspring, and r is a random number between 0 and 1.

In the mutation operation, the adaptive mutation probability p_{mut} is designed as follows:

$$p_{\text{mut}} = \begin{cases} p_{\text{mut max}} - \frac{p_{\text{mut min}}(\text{fit}' - \text{fit}_{\text{mid}})}{\text{fit}_{\text{max}} - \text{fit}_{\text{mid}}}, & \text{fit}' \geq \text{fit}_{\text{mid}}, \\ p_{\text{mut max}}, & \text{fit}' < \text{fit}_{\text{mid}}, \end{cases} \quad (22)$$

where $p_{\text{mut max}}$ and $p_{\text{mut min}}$ are the preset maximum mutation probability and minimum mutation probability, respectively; fit_{max} and fit_{mid} are the maximum fitness value of the group and the median of the group fitness, respectively; and fit' is the fitness value of the individual who needs to mutate.

4.3. Combination of BEN Model and Adaptive Genetic Algorithm. In order to further improve the approximation ability of the BEN model and increase the calculation speed of the BEN model, the BEN model and the adaptive genetic algorithm are combined to construct a brain emotional network adaptive genetic algorithm (BEN-AGA) model.

After the combination, the adaptive genetic algorithm is used to optimize the parameters of the BEN model, so that the calculation of the model is more accurate and efficient, and the problem of artificially setting reward signals that affect the prediction accuracy is avoided. The combination is as follows:

The parameters in the BEN model are α, β, c_1, c_2 , where α is the hidden layer parameter and β is the output layer weight. First, the parameters in the BEN model are chromosome-coded to correspond to the genetic operator of the adaptive genetic algorithm. The parameter encoding chromosome of the BEN model is as follows:

$$X = [\alpha_1, \alpha_2, \dots, \alpha_{m+1}, c_1, \beta_1, \dots, \beta_m, c_2], \quad (23)$$

where $\alpha_1, \alpha_2, \dots, \alpha_{m+1}$ and $\beta_1, \dots, \beta_m$ represent the weights between the neural nodes in the hidden layer, c_1 and c_2 are the thresholds between neurons, and m is the number of sensory signals.

Then the corresponding fitness function is established to guide the search process of the adaptive genetic algorithm. Good or bad fitness function design influences the convergence speed and prediction accuracy of the algorithm. Generally speaking, the fitness function is converted from the objective function. For the BEN-AGA model, the prediction error is naturally considered as the objective function. The smaller the error and the better the parameters of the corresponding chromosome allocation also show that the BEN-AGA model selected the better the parameters. Therefore, the corresponding fitness function is as follows:

$$F(X) = \frac{1}{n} \sum (y_l - \hat{y}_l)^2, \quad (24)$$

where y_l is the predicted value under the l input mode, $\hat{y}_l$ represents the expected output value under the l input mode, and $F(X)$ represents the average prediction error when chromosome X is the model parameter. Finally, according to the chromosome and fitness function established corresponding to the BEN-AGA model, the optimal chromosome (the adaptive genetic algorithm chromosome with optimal fitness) is selected by adaptive genetic operation, which is the best corresponding to the BEN-AGA model parameter combination. Then it is updated to the BEN-AGA model, and the prediction result is got through single-step prediction.

4.4. Implementation Steps of BEN-AGA Model. BEN-AGA model prediction of chaotic time series needs to construct a BEN-AGA model first, input variables, and then obtain all weight parameters in the BEN-AGA model through an adaptive genetic algorithm and update the weight parameters to the BEN-AGA model. Forecast through the updated BEN model. The flow chart is shown in Figure 4.

As the picture shows, the specific steps are as follows:

Step 1: To normalize the chaotic time series and reconstruct the phase space to determine the embedding dimension and delay time

FIGURE 4: The process of combining BEN model and adaptive genetic algorithm.

Step 2: To establish a BEN-AGA model and initialize its parameters ω randomly

Step 3: To encode the required weights and initialize the chromosomes in the population

Step 4: To input observed values into the Ben-AGA model, the fitness evaluation was carried out, and the optimal chromosome was selected by the adaptive genetic algorithm

Step 5: To iterate to the maximum population algebra, confirm the optimal chromosome, and update the parameter ω of the BEN-AGA model

Step 6: To obtain the predicted value of the chaotic time series by single-step prediction through the BEN-AGA model of the confirmed parameters

5. Simulation

5.1. Simulation Analysis of Rossler Chaotic Time Series

5.1.1. Parameter Setting. The Rossler system is an accurate model for studying chaotic systems. It is a three-dimensional first-order ordinary differential equation. The equation is expressed as follows:

$$\begin{cases} \dot{x} = -x - z, \\ \dot{y} = x + ay, \\ \dot{z} = b + z(x - c), \end{cases} \quad (25)$$

where x , y , and z are time functions, select the values of each parameter in the equation as follows: $A = 10$, $b = 8/3$, $c = 28$, and initial values of the equation are selected as $x(0) = 1$,

$Y(0) = 1$, $z(0) = 1$ and the selection sampling time is $t = 0.01$. The fourth-order Runge-Kutta method is used to generate 10,000 data points, and the first component x is the chaotic time series, which is recorded as $x(t)$. The first 2,000 points are discarded to ensure that the system is completely chaotic; 4,000 points are taken as training samples, and the last 4,000 points are used as test samples.

According to the sample data generated by Rossler chaotic time series, phase space reconstruction is performed, and the required embedding dimension and delay time are calculated. The delay time τ obtained by the complex autocorrelation method [17] is 7; the embedding dimension obtained by the method of Brown [18] is 6. According to the second subsection of this article, the single-step prediction method is used to set the prediction step size $\eta = 1$. A six-dimensional vector of type is constructed as follows:

$$\{x(t), x(t-7), x(t-14), x(t-21), x(t-28), x(t-35)\}. \quad (26)$$

As the input of the BEN-AGA model, and the output of the model is represented by a one-dimensional vector as $x(t+1)$.

From the input data as a six-dimensional vector, it can be determined that the number of input nodes of the BEN model is 6; the output node is 1; the hidden layer node is determined by cross-validation as 12; and the initial network weights are random numbers between $[-1, 1]$. Next, the parameters of the adaptive genetic algorithm are set. The chromosome length is determined by the number of BEN network parameters. In order to get the best performance of the genetic algorithm, the hyperparameters are worth studying. There are four key parameters in genetic algorithm: population size, population iteration number, crossover probability, and mutation probability. In this paper, full factor analysis is used to obtain a reasonable combination of four genetic algorithm parameters. The hierarchy of the four parameters is designed as follows: two levels for population size $\in \{50, 100\}$; two levels for Iterations Number $\in \{50, 100\}$; three levels for Crossover probability $\in \{0.8, 0.3, 0.75, 0.3, 0.9, 0.25\}$; and three levels for mutation probability $\in \{0.01, 0.001, 0.1, 0.01, 0.02, 0.009\}$. A full factorial analysis needs $2^2 \cdot 3^2 = 36$ experiments. Some experimental results are shown in Table 1.

The reasonable parameter combination finally selected is shown in Table 2.

5.1.2. Simulation Result. This article compares and analyzes the prediction results of the BEL [9] model, the BEN-AGA model, and the brain emotional network combined with the genetic algorithm (BEN-GA) model. Figures 5–7 show the prediction results and errors of the chaotic time series of the sample data of the BEL, BEN-GA, and BEN-AGA models.

From Figure 5, the BEL model can roughly predict the trend of the chaotic time series, but the deviation of the prediction at the peak and fluctuation of the curve is too large, and the error range is within $[-2.5, 2.5]$. Figure 6 shows that when using the BEN model and genetic algorithm to predict chaotic time series, there is a significant

TABLE 1: Parameter sensitivity results table.

NO	Population size	Iterations number	Crossover probability	Mutation probability	MSE
1	50	50	0.8 0.3	0.01 0.001	0.3244
2	50	50	0.8 0.3	0.1 0.01	0.3146
3	50	100	0.75 0.3	0.02 0.009	0.2369
4	100	100	0.8 0.3	0.01 0.001	0.1247
5	100	50	0.9 0.25	0.02 0.009	0.3078

TABLE 2: Adaptive genetic algorithm parameter setting table.

Population size	Iterations number	Chromosome length	Crossover probability	Mutation probability
100	100	15	$P_{c\max} = 0.8P_{c\min} = 0.3$	$P_{c\max} = 0.01P_{c\min} = 0.001$

FIGURE 5: BEL model chaotic time series prediction chart.

improvement. The forecast deviation at peak and fluctuation points is significantly reduced, and the error range is also reduced to $[-0.5, 0.5]$, but the forecast curve still cannot clearly reflect. The trend of chaotic time series is caused by the genetic algorithm falling into a local optimal solution in the face of nonlinear model optimization. From Figure 7, the BEN model combined with the adaptive genetic algorithm for prediction, it is obvious that the prediction curve can fully reflect the prediction trend of the chaotic time series, which fits the observation curve best, the prediction deviation between peak and fluctuation is small, and the error range is again reduced $[-0.25, 0.25]$. In summary, the

prediction effect of the BEN-AGA model is better than the other two models, and it has higher prediction accuracy.

Because the BEN-AGA model has a smaller error in the prediction results, the optimization ability of the adaptive genetic algorithm is higher than that of the genetic algorithm, and the convergence speed is faster. As shown in Figure 8, the adaptive genetic algorithm fitness value and genetic algorithm fitness value iterative trajectory are presented.

It can be seen from Figure 8 that the fitness value of the adaptive genetic algorithm has converged when iterated to 50 generations, and the fitness value is 0.0052; when the

FIGURE 6: BEN-GA model chaotic time series prediction chart.

genetic algorithm is iterated to 88 generations, the fitness value has converged, and the fitness value is 0.0061; the adaptive genetic algorithm has a higher convergence speed and better fitness value, so the BEN-AGA model has a faster iteration speed, higher prediction efficiency, and higher accuracy.

5.1.3. Error Analysis. In order to better evaluate the effectiveness of the three models, two error evaluation indicators are introduced, namely the mean square error (MSE) and the mean absolute percentage error (MAPE) as follows:

$$e_{\text{MAPE}} = \frac{\sum_{t=1}^N |y(t) - \hat{y}(t)|/y(t)}{N} \times 100\%, \quad (27)$$

$$e_{\text{MSE}} = \frac{\sum_{t=1}^N (p(t) - y(t))^2}{N}, \quad (28)$$

where $y(t)$ is the actual observation, $\hat{y}(t)$ is the predicted value, and N is the number of data. The error evaluation results of the three models are shown in Figure 9.

Figure 9 lists the statistical comparison results of the mean square error (MSE) and mean absolute percentage error (MAPE) of the BEL model, the BEL-GA model, and the BEL-AGA model under different sample sizes. From the figure, it can be clearly seen that the BEL model performs poorly under each different sample size, and the prediction error is much larger than the BEN-AGA and the BEN-GA models. The minimum mean square error is 0.4784 at the sample size of 4,000, and the minimum average absolute percentage error is also obtained at the sample size of 4,000 as 35.34%. The model's prediction error is not stable due to

the sample size. The effect of the BEN-GA model is better than that of the BEL model. The minimum mean square error is 0.1074, and the minimum average absolute percentage error is 15.63%. The prediction error is relatively stable and is less affected by the number of samples. The BEN-AGA model performed best. The minimum mean square error of 0.0460 was obtained at the sample size of 1,000, and the minimum mean absolute percentage error of 11.59% was obtained at the sample size of 4,000. The stability of the prediction error is not affected by the sample size.

As shown in Figure 10, the error curve of the BEN-AGA model fluctuates in a small range near the zero point in the prediction of different samples without much change, indicating that the prediction effect is good, and the prediction trend can be accurately restored, will not produce excessive errors at extreme values. The error curve of the BEL model fluctuates violently and changes significantly, indicating that the prediction effect is average, and the prediction trend cannot be restored well, and it is easy to cause large errors in over- or underfitting at extreme values. The BEN-GA model is between the two models and shows a certain fluctuation near the zero point, which can roughly restore the forecast trend.

In summary, compared with the BEL and the BEN-GA models, the BEN-AGA model has higher prediction accuracy and smaller prediction error and can more accurately reflect the prediction trend.

5.2. Prediction Analysis of Sunspot Series

5.2.1. Parameter Setting. In order to further verify and explain the effectiveness of the BEN-AGA model and apply it

FIGURE 7: Chaotic time series prediction diagram of BEN-AGA model.

FIGURE 8: Adaptive genetic algorithm and genetic algorithm fitness value iteration trajectory graph.

to the prediction of sunspot chaotic time series, we select annual sunspot data from 1,700 to 1,998 as a sample and select the annual sunspot number index as our sample observations. To normalize the observations of the number of sunspots, we selected each year and then reconstruct the phase space. The delay time τ obtained by the complex autocorrelation method [17] is 3; the embedded dimension obtained by the method of Brown [18] is 4. After phase space

reconstruction, 280 sets of data are generated: the first 180 sets of data are selected as training samples, and the last 100 sets of data are used as test samples.

After phase space reconstruction, the input data is a four-dimensional vector. It can be determined that the number of input nodes of the BEN model is 4; the output node is 1; the hidden layer node is determined by cross-validation as 12; and the initial network weights are random

FIGURE 9: Model error graph under different sample sizes.

FIGURE 10: Model error curves of different sample sizes.

numbers in $[-1, 1]$. Next, the parameters of the adaptive genetic algorithm are set. The chromosome length is determined by the number of BEN network parameters as 11. Other parameters are determined as follows through the method of tenfold cross-validation and experience: the population size is 100, the maximum evolutionary algebra is 50, the maximum crossover probability is $p_{c\max} = 0.7$, the minimum crossover probability is $p_{c\min} = 0.1$, the maximum mutation probability is $p_{m\max} = 0.1$, and the minimum mutation probability is $p_{m\min} = 0.003$. And the initial coding of the chromosome is performed, and the coding range is $[-1, 1]$. After BEN-AGA model training and prediction, the obtained data are denormalized to obtain the prediction result of sunspots.

5.2.2. Sunspot Series Prediction Result. In the sunspot sequence prediction, Figure 11 shows the prediction curve and prediction error curve of 180 training data. It can be seen from Figure 11(a) that the red predicted curve can fit the blue actual observation curve very well, indicating that the BEN-AGA model can describe the changing trend of sunspot number well. From the error curve of Figure 11(b), the prediction error of each point fluctuates in a small range near the zero point, indicating that the prediction error of the BEN-AGA model is small.

Figure 12 is the prediction curve and prediction error curve of 100 test data. It can be seen from Figures 12(a) that the red prediction curve can fit the blue actual observation curve well; correspondingly, it can be seen from Figure 12(b) that the prediction error of each point is in a small range near the zero point. Fluctuations illustrate the effectiveness of the BEN-AGA model in predicting sunspot sequences.

Based on Figures 11 and 12, both the training data and the test data prediction curve can fit the observation curve well, indicating that the BEN-AGA model can well reflect the changing trend of the sunspot sequence, and the error curves are all in fluctuations near the zero point indicate that the BEN-AGA model has high prediction accuracy. In order to further illustrate the fitting ability of the BEN-AGA model, the correlation coefficient between the predicted value of the model and the observed value is calculated. The correlation coefficients on the training set and the test set are 0.8986 and 0.8918, respectively; it shows that the BEN-AGA model has a good fitting performance to the sunspot sequence.

In order to compare and verify the performance of the BEN-AGA model in terms of prediction accuracy, stability, and speed, under the same experimental parameter settings, the BEL [9] model, BEL-GA [14], and BEN-GA model are used for sunspots Sequence for single-step prediction. Each model was run independently 20 times, and the average value of M_{se} , M_{ad} , and M_{ape} , and running time was obtained by statistics. Among them, M_{se} , M_{ad} , and M_{ape} are used as prediction accuracy indicators, and stability uses a variance of M_{ape} as indicators. The results are shown in Table 3.

It can be seen from Table 3 that in terms of accuracy indicators M_{se} , M_{ad} , and M_{ape} , the statistical values of the BEN-AGA model compared with other models are all the smallest, indicating that the BEN-AGA model has the

highest prediction accuracy. In terms of time complexity, the BEL model has the minimum complexity, while the other three models are consistent. In terms of running time, the running time of the BEL model is the shortest. Because the iterative calculation of the adaptive genetic algorithm is added to the BEN-AGA model, the running time is slightly longer than the basic BEL model, but the running time is not much different; from the perspective of stability, the variance of M_{ape} of the BEN-AGA model is the smallest, so the BEN-AGA model has better stability than other models. In summary, the accuracy and stability of the BEN-AGA model in the prediction of the sunspot number sequence are better than other models, and the calculation time is also better than most other models. Therefore, the BEN-AGA model is in the prediction of the sunspot sequence. It is effective and superior to other comparison models.

5.3. Comparison of Performance of Different Models. In order to compare the prediction performance of the model proposed in this paper, it is respectively compared with the deep learning network model, SVM model, LM-BP neural network, MLP-BP neural network, basic BEL model [14], and Dendritic neuron model [27, 28]. Rossler chaotic system is selected to produce the original chaotic sequence, single-step projections for chaotic sequence. The parameter settings of the model were set according to the experiment above. Each model was run independently for 25 times, and the average values of MSE, MAD, MAPE, and running time were obtained by statistical analysis of the running results. MSE, MAD, MAPE as the prediction accuracy index, and the stability is measured by the variance of MAPE. Different models of the chaotic sequence forecast results are shown in Table 4.

Table 4 shows the statistical indicators of BEN model, LM-BP neural network model, BEL basic model, SVM model, and deep learning network. We found that the MSE, MAD, and MAPE of the BEN model were all smaller than that of the MLP-BP neural network model, LM-BP neural network model, BEL basic model, and SVM model. This shows that the prediction accuracy of the BEN model is very good, higher than that of common neural network models, but it is still insufficient compared with deep learning network. From the perspective of prediction stability, the BEN model has a very small variance of MAPE, indicating that its prediction stability is very good. Compared with other models, the prediction stability of the BEN model is lower than that of the deep learning network and better than other common neural network models. In terms of running time, the running time of the BEL model is far less than that of other models, and the running time of the BEN model is also smaller than that of other models. However, the running time of the BEN model is longer than the BEL model because it performs iterative calculation of the genetic algorithm on the basis of BEL model. In the face of chaotic time series prediction, The BEN model is superior to the traditional network model in terms of accuracy, stability, and running time. The prediction accuracy is slightly lower than the deep learning network and dendritic neuron model. The deep learning network and dendritic

FIGURE 11: The prediction curve and prediction error curve of training samples.

FIGURE 12: The prediction curve and prediction error curve of the test sample.

TABLE 3: Single-step prediction results of sunspot sequences of different models.

Model	M_{se}	M_{ad}	M_{ape}	Time complexity	Running time/s	Variance of MAPE
BEL	12.2187	7.4592	0.6338	$O(n)$	30.6143	0.3426
BEL-GA	6.7455	2.4573	0.2827	$O(n^2)$	56.7258	0.1153
BEN-GA	3.1317	0.8475	0.0715	$O(n^2)$	62.4631	0.0287
BEN-AGA	1.8872	0.8356	0.0494	$O(n^2)$	48.6551	0.0139

TABLE 4: Comparison of performance of different models.

Model	Precision index			Variance of MAPE	Elapsed time
	MSE	MAD	MAPE		
Deep learning network	0.0012	1.5367	0.0631	0.0023	22.3458
MLP-BP	1.4368	7.9446	0.8373	0.7426	27.6714
LM-BP	0.7612	4.0634	0.6378	0.5253	12.6425
BEL	0.2086	3.1268	0.1165	0.0287	1.5643
SVM [14]	—	7.5467	—	—	—
BEN-AGA	0.0413	2.1536	0.0847	0.0112	10.7658
Dendritic neuron model	0.0037	1.6432	0.0746	0.0031	11.2496

neuron model have better predictive performance. But BEN model has a faster running time and basically the same prediction stability. Therefore, BEN model has great potential in chaotic time series prediction and has broad prospects in geomagnetic early warning, short-term traffic flow prediction, and other aspects.

6. Conclusion

Based on the research of the existing brain emotional learning (BEL) model, this paper proposes the BEN model by changing the linear structure into a nonlinear structure and giving output weights to describe the influence of different structures. Then we further propose the Ben-AGA model through the combination of adaptive genetic algorithm and BEN model. The weights to be optimized are encoded as chromosomes; the crossover process and the mutation process are controlled by the dynamic crossover probability and dynamic mutation probability of the design; and the chromosome is evaluated by the fitness function to select the best parameters. Compared with the original model, the BEN-AGA model has stronger approximation ability, calculation ability, higher prediction accuracy, and stability. The model is applied to Rossler chaotic time series prediction and sunspot sequence prediction, and the experimental results show that the BEN-AGA model proposed in this paper can effectively predict the sunspot sequence. Compared with the model before improvement, it has the advantages of high prediction accuracy, fast convergence, and high stability. The experimental results objectively verify the effectiveness and rationality of the improved method and the established model. By comparing the prediction results of chaotic time series with those of other models, the BEN model has better prediction accuracy, less operation time, and stronger stability than the traditional neural network model. Compared with the deep learning network, the prediction accuracy is still slightly lower, but the model structure is simpler, and the calculation time is shorter. In summary, the BEN model has the advantages of high prediction accuracy, strong stability, and less running time, and the model is simple and easy to be applied in the actual prediction system. It has great potential and broad prospects for application in geomagnetic early warning and short-term traffic flow prediction.

Data Availability

The simulation experiment data uses the fourth-order Runge–Kutta method to solve the Rossler chaotic system. The experiment used the annual sunspot data from 1,700 to 1,998 as the sample. The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Chongqing Natural Science Foundation of China (Grant no. cstc2018jcyjAX0464).

Supplementary Materials

Supplementary Material 1 is the result of the parameter sensitivity experiment, and all the results have been presented in the form of a table. (*Supplementary Materials*)

References

- [1] H. Kantz and T. Schrieffer, *Nonlinear Time Series Analysis*, Cambridge University Press, Cambridge, UK, 2003.
- [2] T. Babaie, R. Karimizandi, and C. Lucas, “Learning based brain emotional intelligence as a new aspect for development of an alarm system,” *Soft Computing*, vol. 12, no. 9, pp. 857–873, 2008.
- [3] H.-J. Yang and X. Hu, “Wavelet neural network with improved genetic algorithm for traffic flow time series prediction,” *Optik*, vol. 127, no. 19, pp. 8103–8110, 2016.
- [4] Z. Tian, “Combined kernel least squares support vector machines for chaotic time series prediction,” *Acta Physica Sinica*, vol. 63, no. 16, pp. 70–80, 2014.
- [5] A. C. Baker and C. Toque, “A review of the potential for radium from luminising activities to migrate in the environment,” *Journal of Radiological Protection: Official Journal of the Society for Radiological Protection*, vol. 25, pp. 127–140, 2005.
- [6] M. Sangiorgio and F. Dercole, “Robustness of LSTM neural networks for multi-step forecasting of chaotic time series,” *Chaos, Solitons & Fractals*, vol. 139, Article ID 110045, 2020.

- [7] H. Akaike, "Fitting autoregressive models for prediction," *Annals of the Institute of Statistical Mathematics*, vol. 21, no. 1, pp. 243–247, 1969.
- [8] L. Zhang and Y. Zhang, "Infinite depth neural network method for big data analysis," *Journal of Computer Research and Development*, vol. 53, no. 1, pp. 68–79, 2016.
- [9] X. G. Zhang, "Introduction to statistical learning theory and support vector machines," *Acta Automatica Sinica*, vol. 1, 2000.
- [10] L. Shen, J. Chen, Z. Zeng, and J. Jin, "Modeling and forecasting of chaotic time series based on robust extreme learning machine," *Acta Physica Sinica*, vol. 67, no. 3, pp. 32–42, 2018.
- [11] C. Wang, "Prediction of photovoltaic power generation based on genetic algorithm optimized convolutional long and short memory hybrid neural network model," *Acta Physica Sinica*, vol. 69, no. 10, pp. 149–155, 2020.
- [12] C. Guo, "Optimization of support vector machine time series forecast model based on genetic algorithm," *Chinese Journal of Scientific Instrument*, vol. 9, pp. 1080–1084, 2006.
- [13] X. Yan, R. Gong, and Q. Zhang, "Application of SVM optimized by genetic algorithm in short-term wind speed prediction," *Power System Protection and Control*, vol. 44, no. 9, pp. 38–42, 2016.
- [14] M. Ying, "Chaotic time series prediction based on brain emotion learning model and adaptive genetic algorithm," *Acta Physica Sinica*, vol. 67, no. 8, pp. 20–31, 2018.
- [15] M. Yin and G. Tan, "Effective data classification based on improved brain emotion learning algorithm," *Journal of Central South University*, vol. 45, no. 11, pp. 2663–2670, 2017.
- [16] R. Legenstein and W. Maass, "Edge of chaos and prediction of computational performance for neural circuit models," *Neural Networks*, vol. 20, no. 3, pp. 323–334, 2007.
- [17] L. Cao, "Practical method for determining the minimum embedding dimension of a scalar time series," *Physica D: Nonlinear Phenomena*, vol. 110, no. 1, 1997.
- [18] R. Brown, P. Bryant, and H. D. Abarbanel, "Computing the Lyapunov spectrum of a dynamical system from an observed time series," *Physical Review A, Atomic, Molecular, and Optical Physics*, vol. 43, pp. 2787–2806, 1991.
- [19] X. R. Li and Y. G. Guo, "Dam damage identification on the base of optmic neural network weight by genetic algorithm," *Technology for Earthquake Disaster Prevention*, vol. 2, pp. 189–196, 2008.
- [20] T. Buzug and G. Pfister, "Optimal delay time and embedding dimension for delay-time coordinates by analysis of the global static and local dynamical behavior of strange attractors," *Physical Review A*, vol. 45, no. 10, pp. 7073–7084, 1992.
- [21] D. E. Holmes and L. C. Jain, *Innovations in Machine Learning*, Springer, Berlin, Heidelberg, 2006.
- [22] Y. Y. Yu, Y. Chen, and T.-Y. Li, "Improved genetic algorithm for solving TSP," *Control and Decision*, vol. 29, no. 8, pp. 1483–1488, 2014.
- [23] S. Aslan and D. Karaboga, "A genetic artificial bee colony algorithm for signal reconstruction based big data optimization," *Applied Soft Computing*, vol. 88, Article ID 106053, 2020.
- [24] H. Liu, "Research on a neural network optimization algorithm based on improved genetic algorithm," *Chinese Journal of Scientific Instrument*, vol. 37, no. 7, pp. 1573–1580, 2016.
- [25] S. Levasseur, Y. Malecot, M. Boulon, and E. Flavigny, "Statistical inverse analysis based on genetic algorithm and principal component analysis: applications to excavation problems and pressuremeter tests," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 34, no. 5, pp. 471–491, 2010.
- [26] J. Zhang, J.-P. Ou, and R.-H. Zhan, "Automatic target recognition of moving target based on empirical mode decomposition and genetic algorithm support vector machine," *Journal of Central South University*, vol. 22, no. 4, pp. 1389–1396, 2015.
- [27] H. He, S. Gao, T. Jin, S. Sato, and X. Zhang, "A seasonal-trend decomposition-based dendritic neuron model for financial time series prediction," *Applied Soft Computing*, vol. 108, Article ID 107488, 2021.
- [28] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, 2019.