Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 6031129, 17 pages
https://doi.org/10.1155/2022/6031129

Research Article

Network Security Situation Prediction Model Based on EMD and ELPSO Optimized BiGRU Neural Network

Biao Zhang,[1] Mingqi Jia,[1] Jiazhong Xu,[1] Wanzhao Zhao ,[2] and Liwei Deng[1]

[1]*Heilongjiang Provincial Key Laboratory of Complex Intelligent System and Integration, School of Automation, Harbin University of Science and Technology, Harbin 150080, China*
[2]*Guangxi Agricultural Vocational and Technical University, Nanning 530005, Guangxi, China*

Correspondence should be addressed to Wanzhao Zhao; 18846824992@163.com

Received 7 September 2021; Revised 15 February 2022; Accepted 10 May 2022; Published 21 June 2022

Academic Editor: Wei Xiang

In order to improve the accuracy of network security situation prediction and the convergence speed of prediction algorithm, this paper proposes a combined prediction model (EMD-ELPSO-BiGRU) based on empirical mode decomposition (EMD) and improved particle swarm optimization (ELPSO) to optimize BiGRU neural network. Firstly, the network security situation data sequence is decomposed into a series of intrinsic mode function by EMD. Then, a particle swarm optimization algorithm (ELPSO) based on cooperative update of evolutionary state judgment and learning strategy is proposed to optimize the hyper-parameters of BiGRU neural network. Finally, a network security situation prediction model based on EMD-ELPSO-BiGRU is constructed to predict each intrinsic mode function, respectively, and the prediction results are superimposed to obtain the final network security situation prediction value. Simulation results show that ELPSO has better optimization performance, and EMD-ELPSO-BiGRU model has higher prediction accuracy and significantly improved convergence speed compared with other traditional prediction methods.

1. Introduction

With the rapid development of Internet technology, computer network has become an indispensable means of communication. However, there are various threats in the network environment. Although firewall, intrusion detection system, virus killing, and other technologies have been developed at present, these methods can only deal with the threats and cannot control the overall trend of the network well. Under this background, aiming at the problem of network security, researchers put forward network security situational awareness. Network security situation is a trend of network security situation. According to the change of network environment, network administrators can take measures to avoid network attacks or reduce the damage caused by network attacks. Network security situation prediction is an active defense mechanism [1], which first analyzes and understands the elements of current and past network situation, and then speculates the future network situation. Because the current situation of network security is reflected by the situation value obtained after situation assessment, and the situation value represents the network state value at every moment, the situation prediction problem is actually a time series prediction problem [2]. Because the trend of network security situation change is nonlinear and time-varying, many classical time series prediction methods are difficult to accurately find out the relationship between the current situation of network and the development trend, which leads to the inability to improve the prediction accuracy [3–5].

In the existing research on network security situation prediction, the methods used are mainly divided into three types: mathematical model-based, knowledge-based reasoning, and pattern recognition-based [6].

The method based on mathematical model is the first method applied in network security situation prediction. This method can comprehensively analyze various factors that may affect the change of network security state,

construct an evaluation function, and realize the mapping from the set composed of various situation factors to the network security situation space through mathematical expressions. Methods based on mathematical models include analytic hierarchy process, weight analysis, and time series analysis [7]. Wang and Hu [8, 9] predict the network security situation through time series analysis algorithm and analyze multiple historical situation values obtained by situation assessment algorithm in time series to realize the prediction of future network security state. However, because the sliding regression model based on time series requires the input series to meet the stationarity assumption, it cannot always guarantee high accuracy.

In network security situation assessment and prediction, knowledge-based reasoning method uses evidence theory, probability theory, and fuzzy theory to deal with uncertain information that may affect network security, and establishes corresponding assessment and prediction models based on expert knowledge and experience. Yang et al. [10] through the Bayesian algorithm based on probability theory have improved to form a dynamic Bayesian network model, then the prior probability is initialized and the posterior probability is adjusted by combining historical situation data with real-time situation data, and it is successfully applied to network security situation prediction. Yifan [11] proposed a risk assessment method based on Bayesian network, but it cannot be applied to large-scale network environment because of the high cost of calculating joint probability. Ruan [12] applies fuzzy reasoning to situation prediction, describes network security situation based on fuzzy sets, and combines Markov process. At the same time, genetic algorithm is introduced into fuzzy membership function, and fuzzy Markov chain is used to accurately predict network security situation.

Based on pattern recognition method, it is necessary to classify all possible situations of network security situation with the help of expert knowledge and experience or machine learning, and finally determine the network security status by calculating the correlation between training samples and measured data. The methods based on pattern recognition mainly include support vector machine, neural network, clustering analysis, grey relational analysis, rough set, and so on. Xiao et al. and Wang et al. [13, 14] all use the improved SVM method to predict the network security situation. Xiao et al. [13] optimize the parameters of SVM by particle swarm optimization (PSO) algorithm and propose a PSO-SVM network security situation prediction model, which finally accurately predicts the network security situation based on small sample data. On this basis, Wang et al. [14] reduce the influence of irregular disturbance by accumulating the original sequences and proves its superiority by comparing with PSO-SVM prediction model.

At present, machine learning has become a hot spot in solving nonlinear complex problems in various research fields. Neural network, which belongs to pattern recognition methods, is one of machine learning algorithms, which has been widely used in medical, financial, management, electrical, and other fields. A large number of researchers have also predicted the network security situation by neural network.

Compared with the traditional machine learning model, the deep learning model shows great potential in the field of network security situation prediction. Tao et al. and Zhang et al. [15, 16] study the network security situation prediction model based on BP neural network. Although BP neural network for network security situation prediction has a certain effect, BP neural network algorithm deficiencies lead to a lot of limitations. The characteristics of complex scenes and dynamic changes of network security situational awareness make the occurrence of network security events have great probability and suddenness. Li and Zhao [17] propose a network security situational prediction method based on LSTM. LSTM neural network is an improvement of recurrent neural network and has strong performance in processing time series data. Kurri et al. [18] propose a network traffic prediction method based on LSTM and introduces particle filter constraint algorithm to optimize network parameters. Aiming at the problem that the slow convergence speed affects the training cost in the training process of LSTM neural network, Li et al., Zhang et al., Liu et al., and Yang et al. [19–22] propose an intelligent optimization algorithm to improve the convergence speed of LSTM neural network model. Aiming at the problems of low prediction accuracy and low efficiency in traditional neural network, Chen et al. [23] propose a new prediction method of recurrent neural network based on gated recurrent unit. This method extracts information features from the original time series data and applies them to the depth RNN model for training and verification. After iteration and optimization, the trained model can obtain the accuracy of network security prediction. Wang et al. [24] propose a prediction method based on two-layer recurrent neural network LSTM and GRU. This method combines two improved recurrent neural networks. Although the prediction accuracy is improved, the complexity of the model is increased and the training time of the model is prolonged.

In order to solve the problems existing in traditional forecasting models, this paper proposes a combined forecasting model based on EMD and ELPSO optimized BiGRU neural network (EMD-ELPSO-BiGRU). Considering that multi-attribute security index data are used as the data support, the multi-attribute network security data are fused on the basis of BiGRU neural network, the network security situation data sequence is decomposed into a series of intrinsic mode function by empirical mode decomposition, and the super-parameters and network scale of the network are determined by improved particle swarm optimization (ELPSO) algorithm, which further improves the performance of the model. This model preserves the original network security data to a great extent, maximizes the correlation between mining data, and improves the prediction accuracy.

The rest of this paper is arranged as follows: the second section introduces the related basic algorithms involved in this paper, including empirical mode decomposition, BiGRU neural network, and conventional PSO; the third section introduces the particle swarm optimization algorithm based on cooperative update of evolutionary state judgment and learning strategy proposed in this paper; the

fourth section introduces the optimization of BiGRU neural network hyper-parameters based on ELPSO algorithm. The fifth section introduces the network security situation prediction model based on EMD-ELPSO-BiGRU; the sixth section discusses the experiment and results. The seventh section summarizes the work of this paper.

2. Correlation Basic Algorithm

2.1. Empirical Mode Decomposition. Empirical mode decomposition (EMD) [25] is a method to deal with nonlinear and nonstationary time-varying sequences. This method adaptively decomposes signals according to the time scale characteristics of data itself and is considered as a breakthrough in Fourier analysis and wavelet analysis based on stationary and linear assumptions. The screening process of EMD algorithm is to decompose complex time series data into a finite number of intrinsic mode function (IMF), and the IMF components obtained by decomposition contain the fluctuation information of the original data in different time scales.

For a given original time sequence sample data $x(t)$, firstly, the local maximum and minimum values on $x(t)$ are calculated, respectively, and the local maximum and minimum values are interpolated and fitted to obtain the upper and lower envelope network $x_{\max}(t)$ and $x_{\min}(t)$ sequence of the original data $x(t)$, and then calculate the mean value of the upper and lower envelope sequence to obtain the mean value sequence $m_1(t)$: x(t).

$$m_1(t) = \frac{x_{\max}(t) + x_{\min}(t)}{2}. \tag{1}$$

Subtract the mean sequence from the original sequence to get a new sequence $h_1^1(t)$ with low frequency removed:

$$h_1^1(t) = x(t) - m_1(t). \tag{2}$$

Generally speaking, $h_1^1(t)$ does not meet the conditions of the eigenmode function. At this time, $h_1^1(t)$ is used as the original sequence, and it is repeated k times until the average curve tends to zero. The judgment condition for marking $c_1(t) = h_1^k(t)$ and treating $c_1(t)$ as an IMF is

$$SD = \sum_{k=1}^{T} \frac{\left[h_1^{k-1}(t) - h_1^k(t)\right]^2}{\left[h_1^{k-1}(t)\right]^2}. \tag{3}$$

Here, SD is the sieving threshold, which is generally between 0.2 and 0.3. Subtract $c_1(t)$ from $x(t)$ to get the residual sequence $r_1(t) = x(t) - c_1(t)$ with the highest frequency components removed. The above screening process is repeated to obtain subsequent IMF components, until $c_n(t)$ is less than the predetermined error or $r_n(t)$ is a monotonic function, and the modal decomposition process is terminated. So far, the original sequence $x(t)$ can be represented by the n-order IMF component and the residual $r_n(t)$:

$$x(t) = \sum_{i=1}^{n} c_i(t) + r_n(t). \tag{4}$$

2.2. BiGRU Neural Network. Bidirectional gated recurrent unit (BiGRU) is a bidirectional gated-based recurrent neural network, which is composed of forward GRU and backward GRU [26]. GRU model is a variant of long short-term memory (LSTM [27]) network. Compared with LSTM, the network structure of GRU model is simpler, but the effect is basically the same as LSTM, which greatly reduces the time required for network training. The output of the current time step of the recurrent neural network is related to the output of the previous time step, which makes the recurrent neural network have memory and is suitable for processing sequence data. However, the traditional neural network only has short-term memory, which is not effective for long-distance dependence, and has the problem of gradient explosion or gradient disappearance. LSTM solves the above problems through gating mechanism and can learn long-span dependencies. The structure of LSTM neurons is shown in Figure 1.

GRU network combines input gate and forgetting gate in LSTM, called update gate, which greatly reduces the time required for training the network. The structure of GRU neurons is shown in Figure 2.

In the GRU network, the update gate controls how many hidden states at the historical moment and candidate states at the current time are retained in the hidden state h_t at the current time. The function of the reset gate is to determine the degree of dependence between the candidate state h_t' at the current moment and the hidden state at the previous moment.

$$z_t = \sigma\left(w_z x_t + u_z h_{t-1} + b_z\right), \tag{5}$$

$$r_t = \sigma\left(w_r x_t + u_r h_{t-1} + b_r\right), \tag{6}$$

$$h_t' = \tanh\left(w_c x_t + u_c\left(r_t \odot h_{t-1}\right) + b_c\right), \tag{7}$$

$$h_t = z_t \odot h_{t-1} + \left(1 - z_t\right) \odot h_t', \tag{8}$$

$$y_t = \sigma\left(W_0 \cdot h_t\right), \tag{9}$$

x_t is the input at the current moment, h_{t-1} is the hidden state at the previous moment, h_t' is the candidate state at the current moment, h_t is the hidden state at the current moment, and y_t is the output at the current moment. Formula (8) is the calculation formula of the update gate, and formula (9) is the calculation formula of the reset gate.

In the GRU network, information can only be transmitted in one direction, but in practice, each input data may have a dependency on the input data before and after it. Using the BiGRU network through trainment data network in two directions makes the model more effective. The structure of BiGRU network is shown in Figure 3.

2.3. Conventional PSO. Particle swarm optimization (PSO) is an intelligent search algorithm that simulates the social behavior of bird groups [28] and searches the solution of the problem cooperatively through information sharing among individuals in the group. The specific mathematical

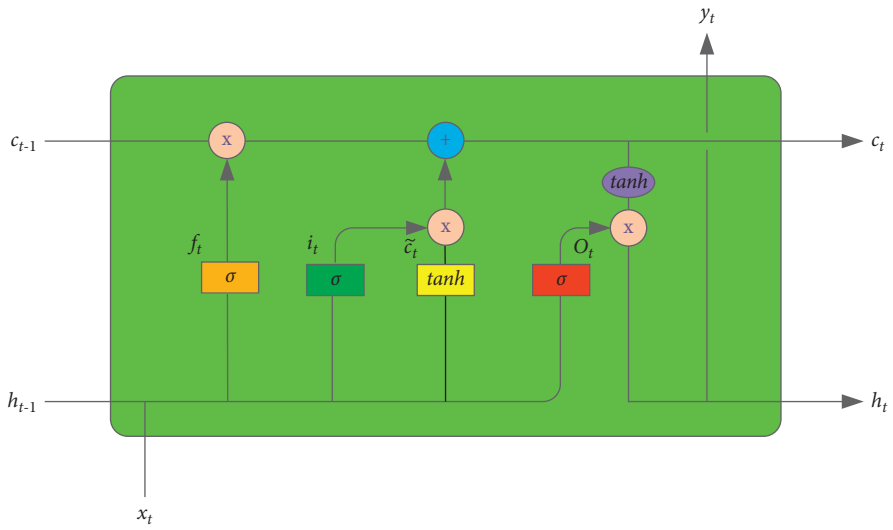

Figure 1: LSTM neuron structure diagram.

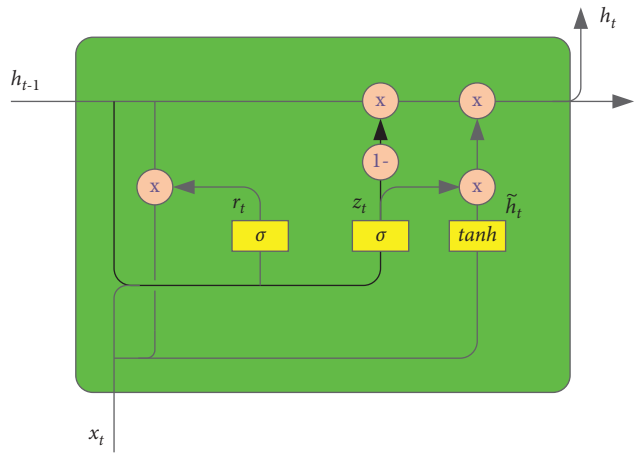

Figure 2: Structural diagram of GRU neurons.

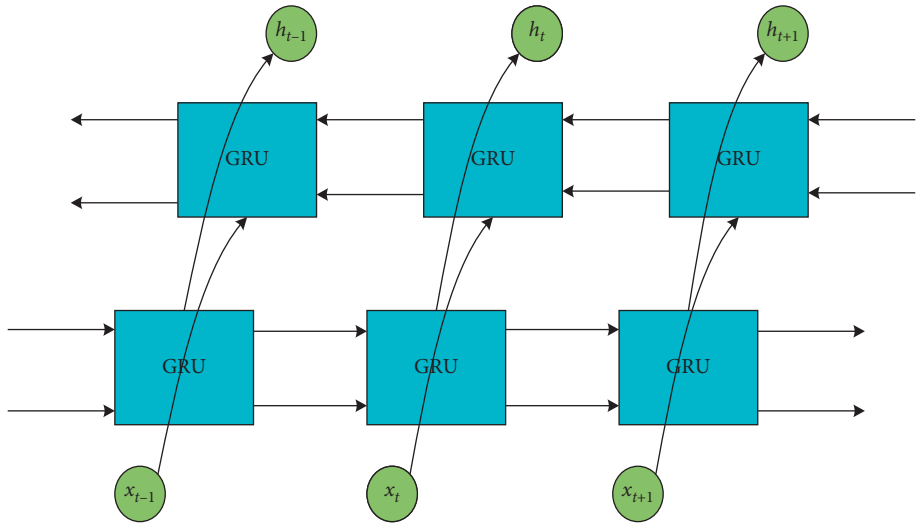

Figure 3: BiGRU network structure diagram.

description of the algorithm is assuming that the dimension of the target search space is D, the particle population size is N, $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ represents the position of the i-th particle in the D-dimensional search space, and $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ represents the velocity of the i-th particle, where $i = (1, 2, \ldots N)$. p_{best} represents the optimal position experienced by the i-th particle itself and g_{best} represents the optimal position experienced by the entire group. In the whole evolution process of the algorithm, each particle updates its own speed and position by continuously updating p_{best} and g_{best}, so as to find the best position of the particle when it reaches the optimal fitness value, which is the solution of the problem to be optimized. The particle velocity and position update formulas are

$$V_i^{t+1} = \omega V_i^t + c_1 r_1\left(p_{best} - x_i^t\right) + c_2 r_2\left(g_{best} - x_i^t\right), \tag{10}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1}. \tag{11}$$

Among them, ω is the inertia weight; c_1 and c_2 are learning factors, and usually the value is 2; r_1 and r_2 are random numbers distributed in [0, 1]; t is the current iteration number of the particle.

3. Particle Swarm Optimization Algorithm Based on Cooperative Update of Evolutionary State Judgment and Learning Strategy

PSO algorithm has the advantages of simple structure, few control parameters, outstanding global optimization ability, etc. It also has the characteristics of fast calculation speed, few parameters, and convenient implementation. However, the algorithm has some problems in the search process, such as premature convergence or falling into local optimum, which is mainly due to the loss of population diversity in the optimization process [29]. Keeping population diversity is an important measure to enhance the global search ability and avoid premature phenomenon. Therefore, in this paper, the learning strategy updating mechanism based on evolutionary state information decision is adopted in the iterative process of particle swarm optimization algorithm, and ELPSO is proposed.

Different from the traditional particle swarm optimization algorithm, ELPSO algorithm uses the information of population evolution to choose the appropriate learning strategy. When the evolutionary state is greater than the fixed threshold, the decision algorithm is in the convergence stage, and the full information learning strategy is adopted to update the speed and position of the information of the particles in the better neighborhood to speed up the convergence speed of the algorithm; when the evolution state is less than a fixed threshold, the decision algorithm is in the stage of jumping out of local optimum. The algorithm adopts local information learning strategy and updates the speed and position of local optimum and best neighborhood particles according to the information, so as to maintain the diversity of population and make the algorithm difficult to fall into local optimum.

3.1. Evolutionary State Analysis. In the iterative process of particle swarm optimization, the reduction of population diversity is the main reason why particle swarm optimization falls into local optimum. In view of this characteristic and the linear relationship between iteration times and population diversity, this paper defines the evolution factor E, and its calculation formula is

$$x_{\text{mean}}^d = \frac{1}{N}\sum_{i=1}^{N} x_i^d, \tag{12}$$

$$\text{div} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{d=1}^{D}\left(x_i^d - x_{\text{mean}}^d\right)\left(x_i^d - x_{\text{mean}}^d\right)}, \tag{13}$$

$$E = e^{-\text{div}/\text{div}_{\max}/k}\left(1 - \frac{t}{T}\right) \in [0, 1]. \tag{14}$$

In the formula, x_{mean}^d represents the average position between particles at the same latitude; N represents the total number of populations; D represents the particle dimension; k is used to adjust the sensitivity of the exponential function and is matched according to the initialization state of the population and the degree of population diversity; t is the current number of iterations; T is the maximum number of iterations; and div and $\text{div}_{\max}$ represent the values of the current population diversity and the maximum population diversity, respectively, when the two are equal, $E = 1$.

3.2. Neighborhood Selection Strategy. In the iterative process, according to the coding characteristics of particle swarm optimization, the Hamming distance between each particle and other particles is calculated, and they are sorted. According to the sorting results, the neighbors of a given particle with a specified number are obtained.

$$\begin{aligned} D_{ij} &= H\left(x_i, x_j\right), \\ j &= 1, \ldots, N;\ j \neq i, \end{aligned} \tag{15}$$

$$S = \text{sort}(D), \tag{16}$$

$$\text{Neighbors} = S(1{:}\ T). \tag{17}$$

In the formula, D_{ij} represents the Hamming distance between the i-th particle and the j-th particle in the population; H is the function to calculate the Hamming distance; S is the set of sorting results; $Neighbors$ represents the current neighborhood particle set; and T is the specified number of neighbors.

3.3. Full Information Learning Strategies. In order to improve the particle optimization problem, ELPSO algorithm adopts full information learning strategy to ensure the optimization ability and convergence performance. In the iterative process of the ELPSO algorithm, particle i obtains information from neighboring particles with better fitness value and at the same time avoids the influence of bad

neighboring particles. The neighboring particles with better fitness value have greater influence on particle i. Based on the above ideas, the ELPSO algorithm adopts a full information learning strategy, and its speed and position update expressions are

$$V_i^d = X\left(V_i^d + \frac{\varphi}{k_i}\sum_{m=1}^{k_i} r_m \cdot \frac{f(p_{im})}{\text{sum}_i} \cdot \left(p_{im}^d - x_i^d\right)\right), \tag{18}$$

$$\text{sum}_i = \sum_{m=1}^{k_i} f(p_{im}), \tag{19}$$

$$S\left(V_i^d\right) = 1 - \text{sqrt}\left(\left(1 - V_i^d\right)^2\right), \tag{20}$$

According to the relevant literature, using the convergence coefficient X and the acceleration coefficient φ to adjust the particle velocity, the algorithm performance is better, where $X = 0.729$, $\varphi = 4.1$; k_i is the number of particles in the better neighborhood of particle i; i_m is the m-th better neighborhood particle of particle i; p_{im} is the position of the particle i_m; $f(p_{im})$ is the fitness value of the particle; sum (i) is the sum of the fitness values of the particles in the better neighborhood; and r_m represents a number uniformly distributed between [0,1]. Equation (20) is the particle position update formula. V_i^d is the particle velocity value, and $S(V_i^d)$ is the probability value of the velocity mapping. If the probability value is greater than the random number rand, the particle position vector takes its own complement; otherwise it remains unchanged.

3.4. Local Information Learning Strategies. In particle swarm optimization, particle i updates speed and position according to the information of local optimal and optimal neighbor particles, is less affected by other particles, and can move more freely in the search space, which is conducive to maintaining population diversity. ELPSO adopts the local information learning strategy, and its speed and position update expressions are

$$V_i^d = X\left[V_i^d + \frac{\varphi}{2}r_1\left(p_i^d - x_i^d\right) + \frac{\varphi}{2}r_2\left(p_{i_{nb}}^d - x_i^d\right)\right]. \tag{21}$$

In the formula, p_i^d is the local optimal position of the particle i; $p_{i_{nb}}^d$ is the optimal position of the neighborhood of particle i; and r_1 and r_2 represent the numbers evenly distributed between [0, 1].

3.5. ELPSO Algorithm Flow. The ideal particle swarm optimization algorithm should not fall into local optimum while ensuring fast convergence speed, which is difficult to achieve by using a single learning strategy. Therefore, ELPSO algorithm adopts different learning strategies to solve complex optimization problems in different evolutionary states. Aiming at the problem that particle swarm optimization is premature and easy to fall into local optimum, the iterative process of particle swarm optimization is divided into two stages: jumping out of local optimum and converging. At the same time, the evolution state is divided. If the evolution factor $E < 0.7$, it is judged that the algorithm is in the stage of jumping out of local optimum, which shows that the population diversity is poor. Local information learning strategy should be selected to ensure that particles can move more freely in the search space to maintain the population diversity; if the evolutionary factor $E > 0.7$ or $E = 0.7$, the decision algorithm is in the convergence stage, which shows that the population diversity is good. All-information learning strategy should be selected to ensure that particles get information from neighborhood particles with better fitness value to accelerate convergence. The specific steps of ELPSO are

Step 1. Population initialization. Set particle population size, learning rate factor, iteration times, and search space dimension.

Step 2. Evolution state determination. Calculate the evolution factor E; if $E < 0.7$, it is judged that the algorithm is in the stage of jumping out of local optimum; if $E \geq 0.7$, the decision algorithm is in convergence stage.

Step 3. Particle velocity update. If the algorithm is in the stage of jumping out of local optimum, the particle velocity is updated by formula (21); if the algorithm is in the convergence stage, the particle velocity is updated by equations (18) and (19).

Step 4. Update particle position. The particle position is updated by equation (20).

Step 5. Repeat steps 2 to 5 until the termination condition is met.

Step 6. satisfies the termination condition (reaching the maximum iteration times), outputs the optimal value, and obtains the corresponding objective function value, and the algorithm ends.

In ELPSO algorithm, evolutionary state judgment is the key to balance convergence and jump out of local optimum. The optimization mechanism of particle swarm optimization algorithm in which evolutionary state judgment and learning strategy are updated cooperatively is shown in Figure 4.

4. Optimization of Hyper-Parameters of BiGRU Neural Network Based on ELPSO Algorithm

When ELPSO algorithm is used to optimize BiGRU network, this paper uses supervised learning to train the model in the training stage of BiGRU network and takes the mean square error function as the loss function of the model.

Its mathematical definition is as follows:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2. \tag{22}$$

where N is the number of training samples, y_i is the actual value, and $\hat{y}_i$ is the model prediction value.

The training data of BiGRU neural network involve the setting of several super-parameters: the number of neurons

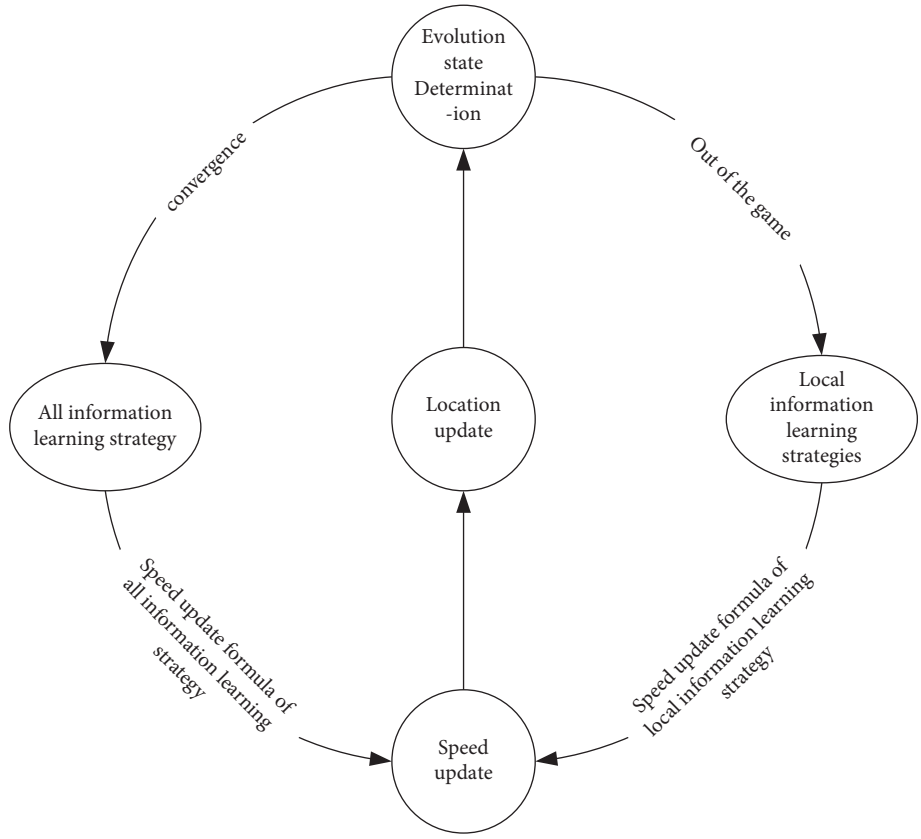

Figure 4: Algorithm optimization mechanism.

m, the time step T, and the batch size. The number of neurons determines the fitting degree of neural network, and the time step and batch size directly affect the training results of the model. In practical application, different super-parameter settings corresponding to different data sets will affect the prediction accuracy. In this paper, ELPSO is used to optimize these super-parameters, and according to the input data, the neural network structure and training mode are adaptively optimized to obtain the optimal combination of model parameters. The specific steps are as follows:

Step 1: initializes the parameters of the algorithm, and determine the population size, iteration times, inertia weight, and the change interval of the learning factor.

Step 2: randomly generates a three-dimensional population particle (M, T, batch size) and initializes the position and velocity of the particle, and the dimension of the particle is the parameter to be optimized.

Step 3: takes formula (22) as the fitness function of the particle. The smaller the fitness function, the smaller the loss function of the model, and the better the parameter combination obtained by the particle.

Step 4: updates the velocity and position of particles.

Step 5: stops when the number of iterations is reached or the fitness function of particles tends to be stable, and the particles at the best position of the population are the optimal parameter combination obtained this time; otherwise, turn to Step 4 to continue iteration.

The flowchart of using ELPSO algorithm to solve the optimal parameter combination of BiGRU model is shown in Figure 5.

5. Network Security Situation Prediction Model Based on EMD-ELPSO-BiGRU

In order to analyze the characteristics of network security situation change in detail, this paper proposes a combined prediction model (EMD-ELPSO-BiGRU) based on empirical mode decomposition and improved particle swarm optimization (ELPSO) to optimize BiGRU neural network. Firstly, the network security situation sequence is stabilized by variational empirical mode decomposition, which is decomposed into a series of different modal components to reduce the complexity of the network security situation sequence; then, BiGRU neural network optimized based on ELPSO algorithm is used to predict each modal component; finally, the prediction results of each modal component of the network security situation sequence are superimposed to obtain the network security situation prediction value. The network security situation prediction process is shown in Figure 6.

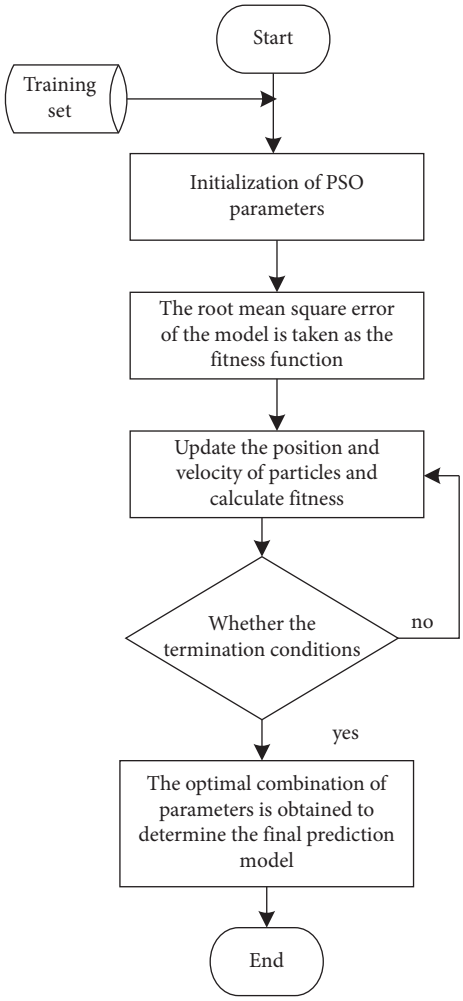

Figure 5: Flowchart of ELPSO optimizing BiGRU.

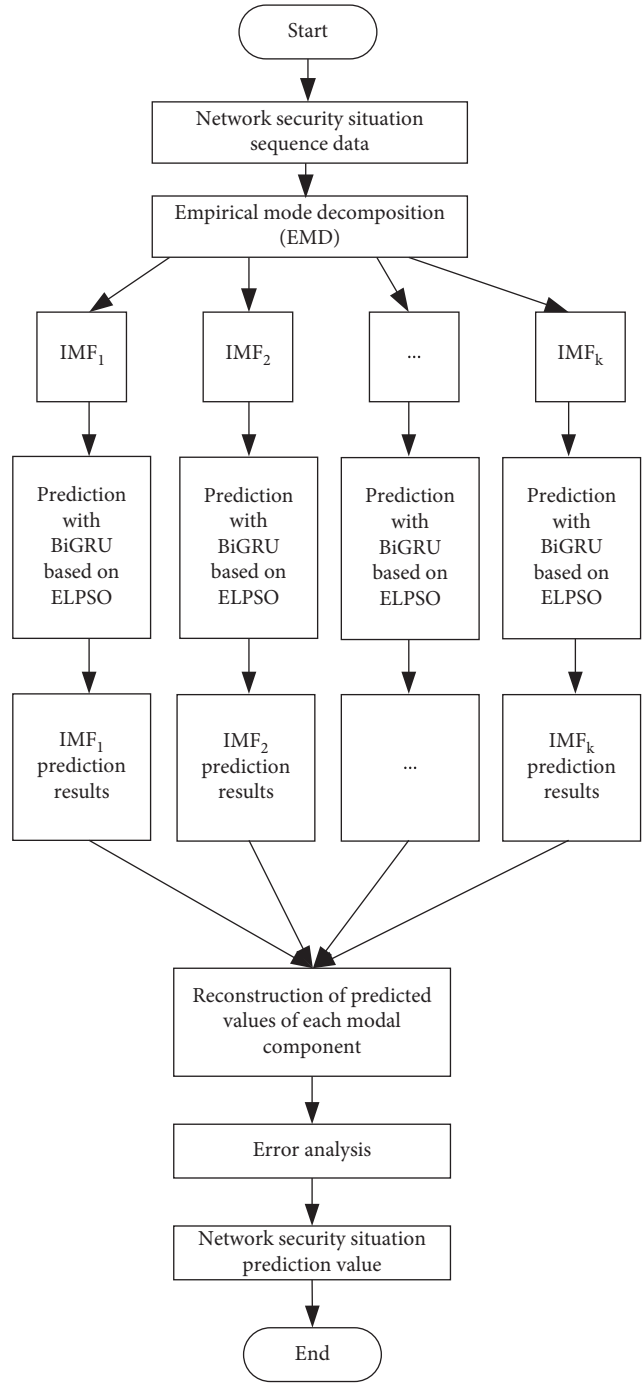

Figure 6: Flowchart of network security situation prediction based on EMD-ELPSO-BiGRU.

6. Experiences and Discussion

6.1. Performance Evaluation of ELPSO Algorithm

6.1.1. Benchmark Function. In order to test the effectiveness of ELPSO proposed in this paper, conventional particle swarm optimization (PSO) [30], improved particle swarm optimization (MPSO) [31], quantum particle swarm optimization (QPSO) [32], IAP-PSO [33], EIW-PSO [34], CLPSO [35], and SRPSO [36] are selected for comparative experiments on 12 benchmark functions. The mathematical expressions for the 12 test functions are shown below [37].

(1) Sphere function

$$f_1(x) = \sum_{i=1}^{N} x_i^2. \tag{23}$$

(2) Schwefel function

$$f_2(x) = \max\left\{\left|x_i\right|, 1 \le i \le n\right\}. \tag{24}$$

(3) Schwefel function

$$f_3(x) = \sum_{i=1}^{n} \left|x_i\right| + \prod_{i=1}^{n} \left|x_i\right|. \tag{25}$$

(4) Step function

$$f_4(x) = \sum_{i=1}^{n} \left(x_i + 0.5\right)^2. \tag{26}$$

(5) Schaffer function

$$f_5(x) = \sum_{i=1}^{n} \left(x_i^2 + x_{i+1}^2\right)^{0.25} \left[\sin^2\left(50\left(x_i^2 + x_{i+1}^2\right)^{0.1}\right) + 1.0\right]. \quad (27)$$

(6) Rastrigin function

$$f_6(x) = \sum_{i=1}^{N} \left(x_i^2 - 100\cos\left(2\pi x_i\right) + 10\right). \quad (28)$$

(7) Griewank function

$$f_7(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1. \quad (29)$$

(8) Ackley function

$$f_8(x) = -20\exp\left(-0.2 \times \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos\left(2\pi x_i\right)\right) + 20 + e. \quad (30)$$

(9) Schaffer function

$$f_9(x) = 0.5 + \frac{\left(\sin^2\sqrt{x_1^2 + x_2^2} - 0.5\right)}{\left(1 + 0.001\left(x_1^2 + x_2^2\right)\right)^2}. \quad (31)$$

(10) Branin function

$$f_{10}(x) = \left(x_2 - \frac{5.1}{4\pi}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10. \quad (32)$$

(11) Six-hump camel back function

$$f_{11}(x) = 4x_1^2 - 2.1x_1^4 + \frac{x_1^6}{3} + x_1x_2 - 4x_2^2 + 4x_2^4. \quad (33)$$

(12) Goldstein price function

$$f_{12}(x) = \left[1 + \left(x_1 + x_2 + 1\right)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \times \left[30 + \left(2x_1 - 3x_2\right)^2\left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]. \quad (34)$$

6.1.2. Analysis of Simulation Results. In the experiment, different PSO algorithms set the same population size $N = 40$, the maximum number of iterations is $T_{\max} = 500$, the learning factor $c_1 = c_2 = 2$, and other parameter settings are consistent with the original literature; in the ELPSO algorithm, $w_{\max} = 0.9$, $w_{\min} = 0.4$, $\sigma = 0.1$.

In order to test the performance of the algorithm, the experiments were divided into three groups, the dimensions of the algorithm were set to 10, 30, and 50, and the four algorithms were run independently for 50 times. The mean value (MEAN) of the test results of each algorithm is shown in Tables 1–3.

From the comparison results of Tables 1–3, it can be obtained that on the 12 test functions, compared with other algorithms, the ELPSO algorithm has further improved the optimization effect of the test function and has better stability; whether in low or high dimensions, the ELPSO algorithm can find better results in unimodal, multimodal, and combined functions.

6.1.3. T Test and Friedman Test. In order to further clarify whether there are significant differences between algorithms, this paper introduces T test [38] and Friedman test [39] to test the performance of 8 algorithms on 12 test functions from a statistical point of view. The experimental results are shown in Table 4. The T test results show that the performance difference between ELPSO algorithm and other algorithms is obvious; compared with PSO, ELPSO has better performance in 9 test functions, and there is no difference in 3 test functions. Compared with MPSO, ELPSO has 7 better functions, 3 no difference, and 1 worse; compared with QPSO, four functions of ELPSO are better, six have no difference, and two are worse; compared with IAP-PSO, ELPSO has better 8 functions and no difference in 4 functions. Compared with EIW-PSO, ELPSO has 7 better functions, 4 no difference, and 1 worse; compared with CLPSO, ELPSO has 3 better functions, 7 no difference, and 2 worse functions; compared with SRPSO, ELPSO has better 7 functions and no difference in 5 functions. The Friedman test results of 8 algorithms show that the rank mean of ELPSO algorithm is the smallest, and the performance of ELPSO algorithm is the best among the 8 algorithms. Combining the two test results, we can see that the performance of ELPSO algorithm is better than other algorithms, where "+" indicates that ELPSO algorithm is superior to other algorithms, "=" indicates that there is no obvious difference between algorithms, "−" indicates that ELPSO algorithm is inferior to other algorithms, and $w/t/l$ indicates the statistical number of these three comparison results, respectively.

6.1.4. Wilcoxon Rank Test. Referring to the data statistics and analysis methods in reference [40], Wilcoxon rank test with significance level of 0.05 is used to judge the performance of the algorithm. Among them, "+," "−," and "≈," respectively, indicate that the results of ELPSO algorithm are

Table 1: Test results of 8 algorithms ($D = 10$).

Benchmark function	PSO	MPSO	QPSO	IAP-PSO	EIW-PSO	CLPSO	SRPSO	ELPSO
f_1	$2.36E+02$	$1.30E+03$	$7.76E+02$	$2.70E+02$	$3.56E+03$	$4.02E+02$	$8.88E+02$	$0.00E+00$
f_2	$3.04E+05$	$2.54E+06$	$6.31E+06$	$1.28E+06$	$1.16E+07$	$5.00E+06$	$2.57E+06$	$1.25E+05$
f_3	$1.08E+03$	$1.92E+03$	$1.57E+03$	$2.81E+02$	$6.42E+02$	$2.01E+03$	$1.20E+03$	$2.24E+02$
f_4	$6.01E-01$	$1.89E+00$	$1.35E+00$	$2.45E-01$	$4.14E+00$	$1.86E+00$	$1.16E+00$	$2.08E-01$
f_5	$9.28E+01$	$1.49E+02$	$1.17E+02$	$1.51E+01$	$6.72E+00$	$5.25E+01$	$3.18E+01$	$3.00E+00$
f_6	$8.97E+01$	$1.57E+02$	$1.21E+02$	$1.45E+01$	$2.03E+01$	$6.13E+01$	$4.56E+01$	$1.42E+01$
f_7	$7.96E+00$	$2.08E+01$	$1.26E+01$	$2.63E+00$	$9.06E-00$	$2.99E+02$	$5.24E+01$	$4.72E-01$
f_8	$3.61E+02$	$5.57E+02$	$4.72E+02$	$5.28E+01$	$8.11E+01$	$5.19E+02$	$3.99E+02$	$2.00E+02$
f_9	$1.37E+03$	$1.91E+03$	$1.65E+03$	$3.15E+02$	$1.90E+03$	$7.88E+02$	$3.06E+02$	$1.60E+02$
f_{10}	$1.22E+03$	$2.05E+03$	$1.55E+03$	$9.21E+02$	$1.91E+03$	$1.38E+03$	$2.76E+02$	$2.03E+02$
f_{11}	$5.96E+02$	$1.99E+03$	$1.32E+03$	$3.55E+02$	$4.49E+02$	$1.69E+03$	$1.04E+03$	$2.84E+02$
f_{12}	$4.13E+02$	$1.77E+03$	$8.19E+02$	$3.53E+02$	$3.87E+02$	$2.33E+03$	$1.03E+03$	$2.03E+02$

Table 2: Test results of 8 algorithms ($D = 30$).

Benchmark function	PSO	MPSO	QPSO	IAP-PSO	EIW-PSO	CLPSO	SRPSO	ELPSO
f_1	$6.72E+03$	$1.23E+04$	$9.99E+03$	$1.43E+03$	$8.82E+03$	$9.19E+03$	$8.97E+03$	$1.69E+01$
f_2	$8.44E+07$	$2.60E+08$	$1.52E+08$	$4.01E+07$	$5.20E+07$	$6.52E+07$	$5.82E+07$	$3.69E+06$
f_3	$6.54E+03$	$8.45E+03$	$7.87E+03$	$4.01E+02$	$5.13E+03$	$6.46E+03$	$5.86E+03$	$3.74E+02$
f_4	$2.41E+00$	$3.85E+00$	$3.05E+00$	$4.05E-01$	$1.50E+00$	$3.34E+00$	$2.49E+00$	$3.36E-01$
f_5	$6.29E+02$	$8.32E+02$	$7.06E+02$	$7.66E+01$	$2.33E+02$	$2.78E+02$	$2.54E+02$	$4.92E+01$
f_6	$6.14E+02$	$7.95E+02$	$6.99E+02$	$5.48E+01$	$2.60E+02$	$3.76E+02$	$3.26E+02$	$2.49E+01$
f_7	$3.88E+02$	$2.38E+03$	$1.45E+03$	$5.36E+02$	$1.56E+04$	$1.64E+04$	$1.58E+04$	$1.53E+02$
f_8	$6.80E+03$	$8.52E+03$	$7.92E+03$	$3.24E+02$	$6.36E+03$	$8.21E+03$	$7.47E+03$	$4.16E+02$
f_9	$2.50E+01$	$3.71E+05$	$1.81E+04$	$6.70E+03$	$1.61E+03$	$8.03E+03$	$5.37E+02$	$2.24E+01$
f_{10}	$2.49E+02$	$2.85E+03$	$8.19E+02$	$5.41E+02$	$2.74E+02$	$1.65E+03$	$5.16E+02$	$2.15E+02$
f_{11}	$1.89E+03$	$6.01E+03$	$3.83E+03$	$1.05E+03$	$3.76E+03$	$7.64E+03$	$5.66E+03$	$8.50E+02$
f_{12}	$4.02E+03$	$7.44E+03$	$5.45E+03$	$8.73E+02$	$3.93E+03$	$7.58E+03$	$5.71E+03$	$8.71E+02$

Table 3: Test results of 8 algorithms ($D = 50$).

Benchmark function	PSO	MPSO	QPSO	IAP-PSO	EIW-PSO	CLPSO	SRPSO	ELPSO
f_1	$2.84E+04$	$3.84E+04$	$3.32E+04$	$2.54E+03$	$1.29E+04$	$1.63E+04$	$1.40E+04$	$1.88E+01$
f_2	$3.44E+08$	$6.21E+08$	$4.82E+08$	$7.03E+07$	$1.58E+08$	$2.93E+08$	$2.15E+08$	$1.00E+07$
f_3	$1.38E+04$	$1.58E+04$	$1.49E+04$	$4.96E+02$	$1.16E+04$	$1.32E+04$	$1.24E+04$	$4.20E+02$
f_4	$3.49E+00$	$4.58E+00$	$4.03E+00$	$2.54E+00$	$4.34E+00$	$3.62E+00$	$4.83E-01$	$2.80E-01$
f_5	$1.53E+03$	$2.58E+03$	$2.15E+03$	$2.81E+02$	$2.88E+02$	$3.93E+02$	$3.48E+02$	$2.46E+01$
f_6	$1.61E+03$	$2.37E+03$	$1.98E+03$	$2.32E+02$	$6.17E+02$	$7.64E+02$	$7.01E+02$	$3.93E+01$
f_7	$1.30E+04$	$5.29E+04$	$3.17E+04$	$1.06E+04$	$2.24E+04$	$3.81E+04$	$2.79E+04$	$1.96E+01$
f_8	$2.90E+03$	$5.19E+03$	$3.96E+03$	$8.14E+02$	$3.08E+03$	$3.32E+03$	$3.21E+03$	$2.97E+02$
f_9	$2.43E+04$	$9.88E+05$	$1.08E+05$	$2.47E+05$	$1.30E+02$	$1.32E+05$	$7.84E+03$	$5.69E+01$
f_{10}	$8.21E+02$	$3.79E+03$	$2.05E+03$	$7.09E+02$	$4.81E+03$	$2.04E+03$	$9.86E+02$	$6.31E+02$
f_{11}	$6.56E+03$	$1.16E+04$	$8.74E+03$	$8.49E+03$	$1.37E+04$	$1.06E+04$	$1.17E+03$	$1.42E+03$
f_{12}	$7.24E+03$	$1.41E+04$	$1.07E+04$	$1.79E+03$	$9.00E+03$	$1.68E+04$	$1.18E+04$	$1.71E+03$

Table 4: T test and Friedman test results of 8 algorithms.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	$w/t/l$	Rank mean
PSO	+	+	+	+	+	+	+	+	=	=	=	+	9/3/0	3.17
MPSO	+	+	+	=	+	=	+	=	+	=	−	+	7/4/1	3.08
QPSO	+	+	+	=	+	=	=	=	=	−	−	=	4/6/2	1.96
IAP-PSO	=	+	+	+	+	+	+	+	=	=	=	+	5/4/1	2.51
EIW-PSO	+	=	+	+	+	=	+	=	+	=	−	+	8/3/2	2.87
CLPSO	+	+	=	=	+	=	=	=	=	−	−	=	6/4/3	2.69
SRPSO	+	+	+	+	=	+	=	+	=	=	=	+	7/2/9	1.87
ELPSO														1.79

better than, worse than, and equivalent to the test results of corresponding algorithms.

From the Wilcoxon results in Table 5, when $\alpha = 0.05$, the ELPSO algorithm has obtained obvious advantages compared with the comparison algorithm in the test function. It can be seen that compared with other algorithms, the ELPSO algorithm has outstanding advantages in solving high-dimensional problems.

6.1.5. Average Number of Iterations at Specified Precision. In order to comprehensively analyze the performance of the algorithm, this section gives 8 algorithms to test 12 benchmark functions under the specified precision of 10^{-10}, the dimension is 30, and the average number of iterations for each algorithm runs independently for 50 times. The results are shown in Table 6.

From the experimental results in Table 6, it can be seen that the PSO algorithm only achieves the specified accuracy on 3 test functions, and the MPSO algorithm achieves the specified accuracy on 10 functions. However, QPSO, IAP-PSO, EIW-PSO, CLPSO, SRPSO, and ELPSO achieve the specified accuracy in all test functions. And compared with other algorithms, the ELPSO algorithm can achieve the specified accuracy with the least number of iterations, and the average number of iterations is between 11 and 34. This shows that the convergence speed of the ELPSO algorithm has obvious advantages and high optimization performance, which further shows that the ELPSO algorithm has the characteristics of fast convergence speed.

6.2. Simulation Analysis of Network Security Situation Prediction

6.2.1. Selection of Network Security Situation Data. In this paper, the weekly data of security situation released by the National Internet Emergency Center are used as the experimental basis [41]. The National Internet Emergency Center is a network security technology coordination organization in Chinese mainland, which mainly processes the national security incidents statistically, evaluates the network security status, and publishes security information on a weekly, monthly, and annual basis. The dynamic weekly report mainly evaluates the basic situation of network security with five security indicators, including the number of hosts infected with network viruses in China, the total number of tampered websites in China, the total number of backdoor websites implanted in China, the number of phishing pages of domestic websites, and the number of new information security vulnerabilities. In this paper, 120 safety data from the 31st issue of 2017 to the 45th issue of 2019 are selected as experimental basis to verify the superiority of this method. The evaluation method of reference [42] is cited to quantify the original data, and the network security situation value of 120 weeks is obtained. The specific quantification model is shown in Figure 7.

6.2.2. Experimental Data and Its Preprocessing. In this paper, the data of the first 101 weeks are selected as the training set and the data of the last 18 weeks as the test set according to the time sequence. The time window is set as the time step of the recurrent neural network, and the prediction time is one week. Because of the complexity and randomness of the network environment and the great difference of the dimensions of situation values, the activation function of the neural network used in this paper is extremely sensitive to whether the input data are within [−1, 1]. Therefore, standardizing the data can accelerate the convergence speed and improve the prediction accuracy of the neural network. The input data are processed by data normalization, and the specific calculation formula is as follows [43].

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \tag{35}$$

where x and x' are the data before processing, and $\min(x)$ and $\max(x)$ are the minimum and maximum values in the data set. Therefore, the normalized network security situation value is shown in Figure 8.

6.2.3. Model Metrics and Evaluation Indicators. In this paper, two measurement methods are selected to evaluate the proposed prediction model: mean absolute error and root mean square error. The specific formula is defined as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right|, \tag{36}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}, \tag{37}$$

where N is the number of training samples, y_i is the actual value, and $\hat{y}_i$ is the predicted value.

After preprocessing the network security situation data, the ELPSO algorithm can be used to obtain the optimal combination of model parameters. Initialize ELPSO: the population size of the PSO algorithm is 5, the evolution times are 40, and the dimension of each particle is 3, which, respectively, represent the parameters to be optimized—the number of encoder neurons, the number of prediction network neurons, the time step T, and the batch size. For simplicity, the number of encoder neurons is set to be equal to the number of prediction network neurons. The maximum value of learning factors c_1 and c_2 is 2.5, the minimum value is 0.5, and the weight factor w is 0.8.

6.2.4. Optimal Parameter Selection of the Model. Figure 9 shows the training results of ELPSO algorithm optimizing BiGRU neural network. The number of neurons, time step size, and batch size gradually converge to the optimal value with the update of the algorithm. As can be seen from Figure 9, the number of neurons finally converges to 21, the batch size of model training data is 1, and the optimal time step is 6. So far, the best super-parameters are

Table 5: Wilcoxon rank test results of 8 algorithms.

Dimension	ELPSO VS	p-value	+	≈	−	$\alpha = 0.05$
10	PSO	0.005173	10	0	0	Yes
	MPSO	0.007182	10	0	0	Yes
	QPSO	0.006253	9	1	0	Yes
	IAP-PSO	0.004179	10	0	0	Yes
	EIW-PSO	0.005187	10	1	0	Yes
	CLPSO	0.005462	10	0	0	Yes
	SRPSO	0.005383	10	0	0	Yes
30	PSO	0.005875	10	0	0	Yes
	MPSO	0.006217	10	1	0	Yes
	QPSO	0.006348	10	0	0	Yes
	IAP-PSO	0.005981	10	0	0	Yes
	EIW-PSO	0.006312	10	0	0	Yes
	CLPSO	0.005819	10	1	0	Yes
	SRPSO	0.006416	10	0	0	Yes
50	PSO	0.005349	10	0	0	Yes
	MPSO	0.007381	10	0	0	Yes
	QPSO	0.006945	10	0	0	Yes
	IAP-PSO	0.005946	10	0	0	Yes
	EIW-PSO	0.006218	10	1	0	Yes
	CLPSO	0.006325	10	0	0	Yes
	SRPSO	0.005419	10	0	0	Yes

Table 6: Average number of iterations under the specified accuracy.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}
PSO	—	243	—	—	—	—	—	—	227	—	204	—
MPSO	164	150	213	94	—	113	—	208	186	94	125	194
QPSO	62	18	102	22	179	61	62	98	151	42	149	83
IAP-PSO	87	54	115	96	49	117	103	115	147	98	99	96
EIW-PSO	114	71	79	83	65	49	58	92	83	30	74	60
CLPSO	79	85	105	92	71	72	64	84	101	53	59	86
SRPSO	34	64	86	78	48	63	55	71	45	44	39	67
ELPSO	11	12	16	14	34	9	10	16	21	19	24	13

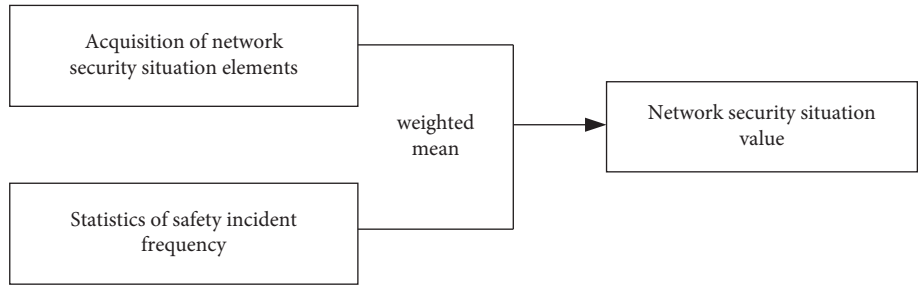

Figure 7: Quantitative model of network security situation assessment.

obtained to modify the model structure of BiGRU neural network and obtain the best parameter combination.

6.2.5. Analysis of Simulation Experiment Results. In order to evaluate the performance of the proposed model in network security situation prediction, comparative experiments are carried out with traditional machine learning methods and deep learning methods, including BP [44], LSTM [45], BiGRU [46], and ELPSO-BiGRU models. The experimental environment of this paper is Windows 10 operating system, and Keras deep learning framework is used for model training and testing in Python3.7 environment, hardware configuration: 64-bit operating system with Inter (R) Core (TM) i5-8500 CPU 3.00 GHZ processor.

(1) EMD Decomposition of Experimental Data. Firstly, EMD is carried out on the network security situation data sequence, the number of modal components is adaptively obtained in the recursive process, and five intrinsic modal functions and a residual component R are obtained, as shown in Figure 10. According to the characteristics of modal components after

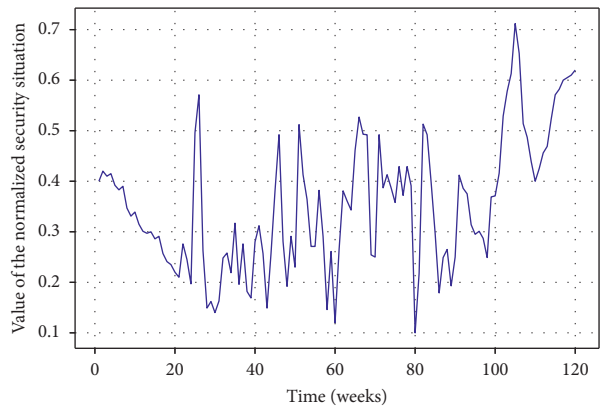

Figure 8: Normalized network security situation time series.

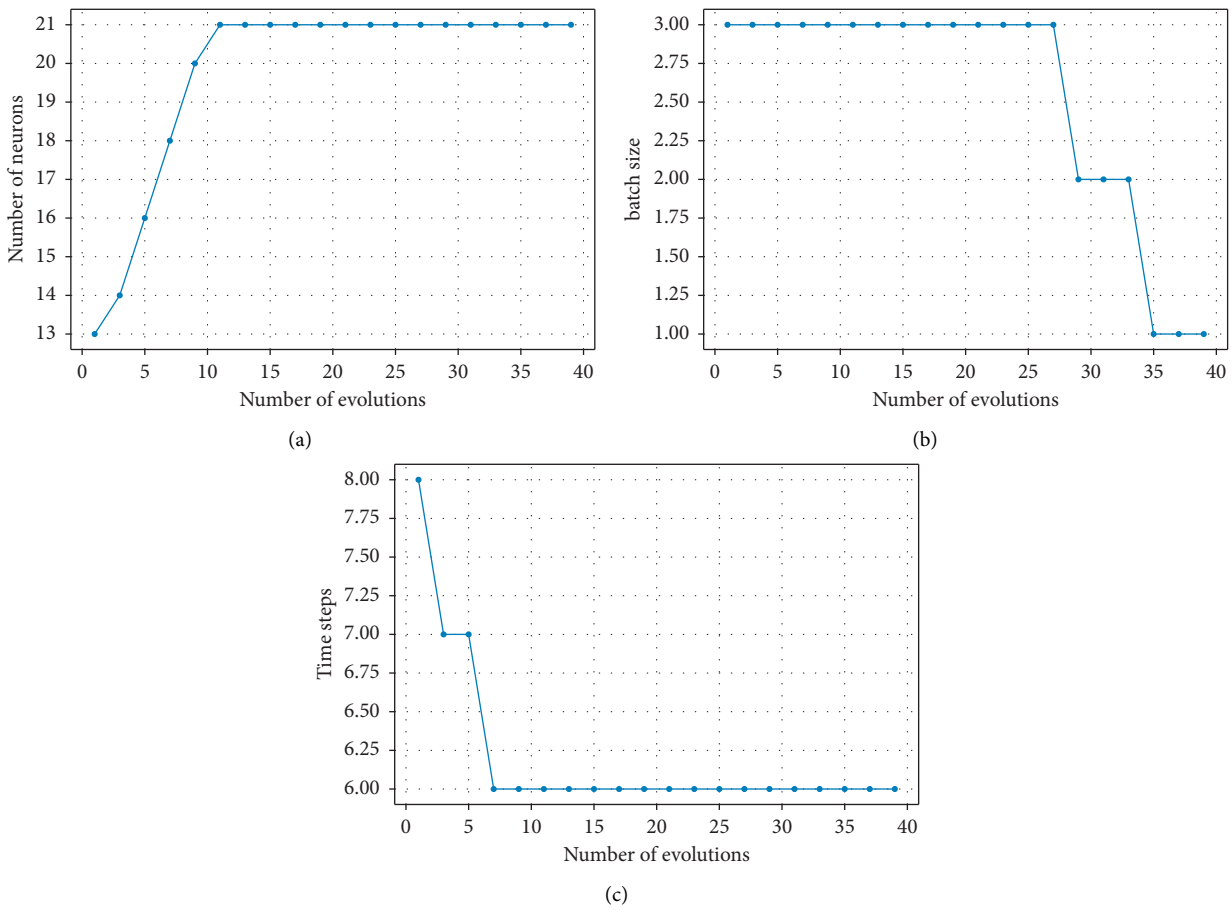

Figure 9: ELPSO optimizations—the parameters of BiGRU. (a) Number of neurons. (b) Batch size. (c) Time steps.

decomposition of network security situation data sequence, it is generally believed that high frequency components reflect the random influence of network security situation; some lower frequency components also have strong sinusoidal fluctuation characteristics, which can be considered as periodic components of network security situation data series; the low frequency part is the trend item of network security situation, which can clearly show the long-term trend of network security.

(2) Comparison of Prediction Accuracy. In order to evaluate the prediction ability of each model as a whole, the final two errors of different models are calculated, and the results are shown in Table 7. In order to increase the fairness of comparison, this paper carries out many experiments on all prediction models to take the average value. According to the average absolute error and root mean square error selected in this paper to measure the accuracy of the prediction results, the two evaluation indicators, respectively, represent the deviation between the

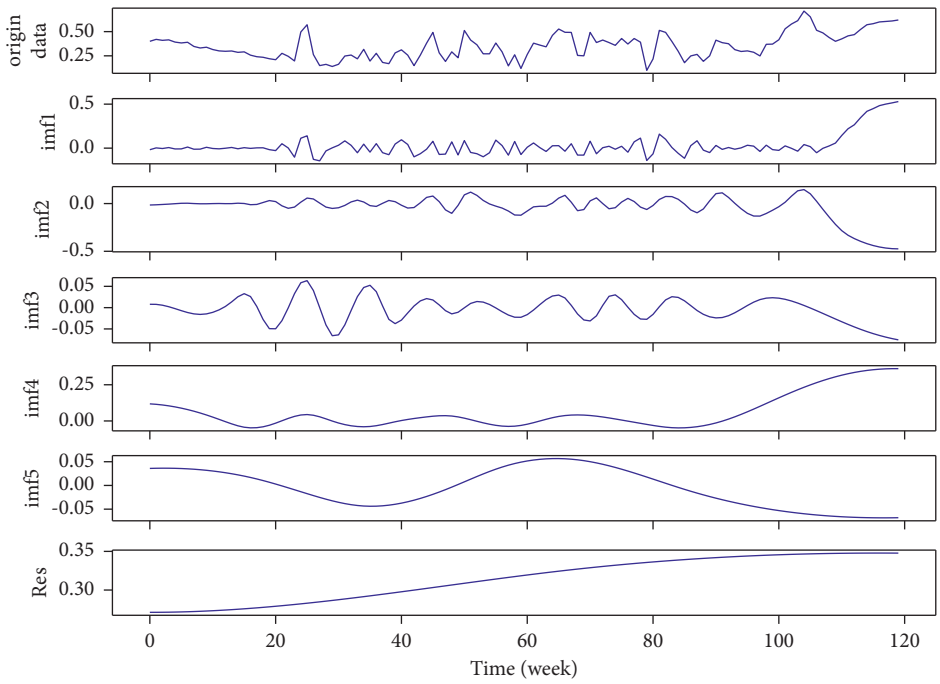

Figure 10: EMD results.

Table 7: Comparison of prediction error indexes of model prediction error index comparison.

Prediction model	MAE	RMSE
BP	0.0745	0.0895
LSTM	0.0174	0.0186
BiGRU	0.0148	0.0156
ELPSO-BiGRU	0.0082	0.0108
EMD-ELPSO-BiGRU	0.0032	0.0054

predicted value and the real value and the fitting accuracy. The smaller the value, the better the prediction effect. As can be seen from Table 7, the EMD-ELPSO-BiGRU model has greater advantages than other models in overall error. Compared with the ELPSO-BiGRU model, the error is reduced by 60.9%, compared with the BiGRU model, the error is reduced by 78.3%, and compared with the prediction model of the BP neural network, the error is reduced by 97.8%, indicating that the EMD-ELPSO-BiGRU model is effective for the prediction of network security situation data.

The results in Table 8 can further prove that the EMD-ELPSO-BiGRU model can obtain good prediction results at most time points. Table 8 shows the absolute errors of different prediction models at each time point during prediction. It can be seen that the absolute errors of this method are all controlled within 0.004 and most of the errors are one order of magnitude lower than 0.004, with higher prediction accuracy than other models.

Figure 11 shows the comparison of prediction accuracy between EMD-ELPSO-BiGRU and basic prediction model models such as BiGRU, LSTM, and BP, and BiGRU neural network optimized based on ELPSO algorithm. It can be seen intuitively from the figure that all prediction models have a certain prediction ability, but the prediction value of the EMD-ELPSO-BiGRU model has the highest fitting degree with the real value and almost coincides with the real value at each prediction point.

(3) Prediction Time Comparison. The evaluation criteria of time series prediction not only depend on the accuracy of prediction, but also depend on the accuracy under different prediction durations. In this paper, the prediction accuracy of different prediction models under different prediction duration is compared, and the results are shown in Figure 12. It can be seen that all models have the smallest error in single-step prediction. Under the same prediction time, EMD-ELPSO-BiGRU model has better prediction ability. With the increase of prediction time, the prediction error gradually increases and then changes stably, and the model has certain robustness.

(4) Convergence Analysis. In the previous section, the complexity of training different models once was compared. Figure 13 shows the change of training error of the model with the number of iterations. It can be observed from the figure that the method in this paper has significant advantages in convergence speed and convergence accuracy, which shows that the model can learn data well.

Table 8: Comparison of prediction absolute of different prediction models at each time point.

Serial number	BP	LSTM	BiGRU	ELPSO-BiGRU	EMD-ELPSO-BiGRU
1	0.01014	0.09067	0.0056	0.00266	0.00069
2	0.00791	0.06355	0.00547	0.00211	0.00254
3	0.00803	0.08235	0.00707	0.00213	0.00290
4	0.01153	0.08235	0.00707	0.00213	0.00116
5	0.01041	0.10887	0.00496	0.00377	0.00037
6	0.01006	0.10584	0.0058	0.00443	0.00017
7	0.01043	0.02663	0.00674	0.00209	0.00051
8	0.00904	0.03808	0.00634	0.00223	0.00071
9	0.00026	0.00825	0.01207	0.00047	0.00205
10	0.00849	0.05474	0.00904	0.00075	0.00014
11	0.00268	0.01946	0.00955	0.00883	0.00331
12	0.00733	0.0477	0.01169	0.02419	0.00343
13	0.00483	0.09163	0.00756	0.00587	0.00276
14	0.00918	0.15478	0.0063	0.00414	0.00157
15	0.00989	0.12663	0.00424	0.00485	0.00211
16	0.00909	0.03466	0.00768	0.00258	0.00045
17	0.00875	0.0201	0.00778	0.00035	0.00107
18	0.00116	0.07084	0.01005	0.01816	0.03832

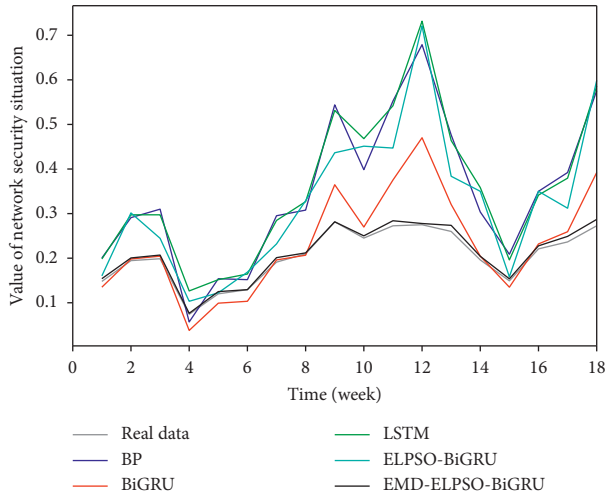

Figure 11: Comparison of situation prediction of different prediction models.

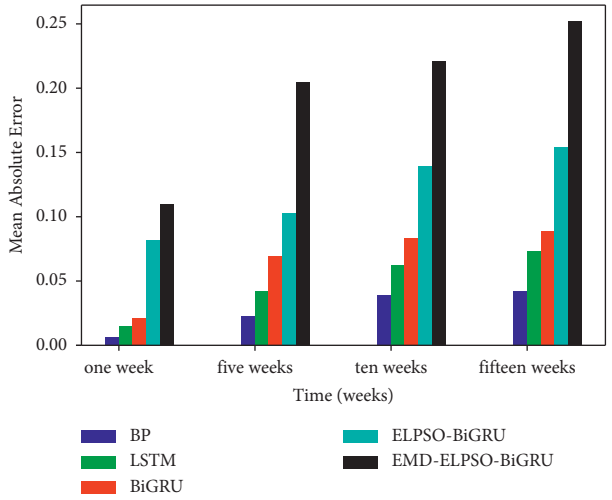

Figure 12: Comparison of mean absolute error under a different forecast time.

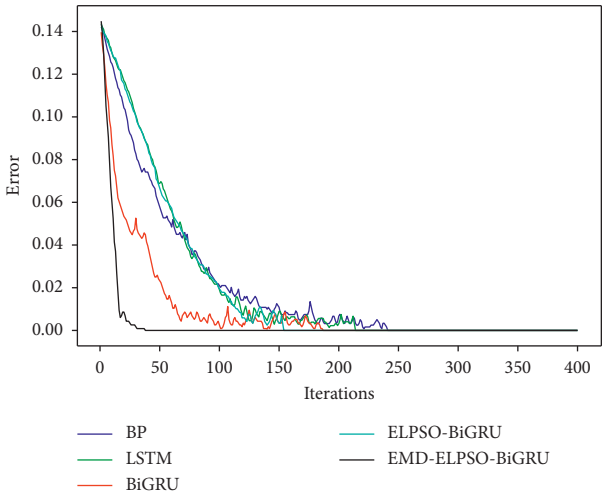

Figure 13: Curve of training error changing with the number of iterations.

7. Convention

In this paper, a combined prediction model of network security situation based on the EMD-ELPSO-BiGRU model is established for network security situation data series. Firstly, the network security situation data are decomposed by the EMD algorithm, and the BiGRU neural network based on ELPSO optimization is used to predict. In the experiment, firstly, the paper compares the proposed ELPSO algorithm with PSO and QPSO to optimize the benchmark function; then, the EMD-ELPSO-BiGRU, BP, LSTM, BiGRU, and ELPSO-BiGRU models are used to predict the network security situation, and the following conclusions can be drawn:

(1) ELPSO algorithm adopts full information learning strategy in the convergence stage based on evolutionary state judgment, which has faster convergence speed than other algorithms; in the stage of jumping

out of local optimum based on evolutionary state judgment, local information learning strategy is adopted to effectively avoid the algorithm falling into local optimum by maintaining population diversity.

(2) Empirical mode decomposition decomposes the network security situation sequence thoroughly, which can reduce the nonstationarity of the data sequence. When the data after empirical mode decomposition are predicted by neural network, the network has higher prediction accuracy and generalization ability.

(3) Compared with traditional BP neural network, LSTM, BiGRU, and ELPSO-GRU, EMD-ELPSO-BiGRU model improves the prediction accuracy of network security situation prediction.

(4) The EMD-ELPSO-BiGRU prediction model proposed in this paper is universal, which is not only suitable for network security situation prediction, but also suitable for ship motion posture prediction and stock price prediction.

In the follow-up research, we will focus on the combination of deep learning models such as BiGRU and swarm intelligence algorithms such as PSO and GA to further enhance the effect of deep learning models such as LSTM in practical application.

Data Availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The paper was supported by the joint fund for enterprise innovation and development of National Natural Science Foundation of China (no. U19B2021) and The Education Science Project of the Junior Teacher in the Education Department of Fujian Province (JAT160532).

References

[1] Y. B. Leau and S. Manickam, "Network Security Situation Prediction: A Review and Discussion," in *Proceedings of the international conference on soft computing, intelligence systems, and information technology*, pp. 424–435, Springer, Berlin, Heidelberg, March 2015.

[2] X. Su, Z. Dong, and L. Sun, "Xu kuikui Network security situation prediction method based on enhanced LSTM," *Computer Technology and Development*, vol. 31, no. 07, pp. 127–133, 2021.

[3] W. Sun, "Sun Huan Research on network security situation prediction technology," *Computer technology and development*, vol. 29, no. 04, pp. 100–104, 2019.

[4] H. Yang, "Zhang Xugao Network security situation prediction based on self correction coefficient smoothing method," *Journal of communications*, vol. 41, no. 05, pp. 196–204, 2020.

[5] Q. Hu, C. Li, and Y. Lu, "Song Yafei Network security situation prediction method based on hierarchical optimization confidence rule base," *Computer Engineering*, vol. 46, no. 12, pp. 127–133, 2020.

[6] H. Yu, X. Yang, and L. Wang, "Network Security Situation Prediction Based on Combining Associated Entropy and Deep Recurrent Neural Network," *Transactions on Emerging Telecommunications Technologies*, vol. 12, Article ID E4164, 2020.

[7] R. Xue, P. Tang, and S. Fang, "Prediction of computer network security situation based on association rules mining," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 2794889, 9 pages, 2022.

[8] G. Wang, "Comparative study on different neural networks for network security situation prediction," *Security and Privacy*, vol. 4, no. 1, pp. 138–147, 2021.

[9] J. Hu, D. Ma, C. Liu, Z. Shi, H. Yan, and C. Hu, "Network security situation prediction based on MR-SVM," *IEEE Access*, vol. 7, pp. 130937–130945, 2019.

[10] H. Yang, L. Zhang, X. Zhang, and J. Zhang, "An adaptive IoT network security situation prediction model," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 371–381, 2022.

[11] Z. Yifan, "Application of machine learning in network security situational awareness," *IEEE*, in *Proceedings of the 2021 world conference on computing and communication technologies (wccct)*, pp. 39–46, Dalian, China, June 2021.

[12] Z. Ruan, "Network security prediction method based on kubernetes," *Journal of physics: conference series. Journal of Physics: Conference Series*, vol. 2010, no. 1, Article ID 012109, 2021.

[13] B. Xiao, Y. Lan, H. Zhao, X. Wu, and W. Liu, "An ISM-based analysis method on the influencing factors of network security situation (NSS)," *Journal of Interconnection Networks*, vol. 45, Article ID 2143029, 2022.

[14] Y. Wang, A. Smahi, H. Zhang, and H. Li, "Towards double defense network security based on multi-identifier network architecture," *Sensors*, vol. 22, no. 3, p. 747, 2022.

[15] X. Tao, K. Kong, F. Zhao, S. Cheng, and S. Wang, "An efficient method for network security situation assessment," *International Journal of Distributed Sensor Networks*, vol. 16, no. 11, 2020.

[16] R. Zhang, M. Liu, Q. Zhang, and Z. Cai, "A Network Security Situation Prediction Algorithm Based on BP Neural Network Optimized by SOA," in *Proceedings of the international conference on artistic intelligence and security*, pp. 417–427, Springer, Hohhot, China, September 2020.

[17] S. Li and D. Zhao, "A LSTM-based method for comparison and evaluation of network security situation," in *Proceedings of the 2019 18th ieee international conference on trust, security and privacy in computing and communications/13th ieee international conference on big data science and engineering*, pp. 723–728, TrustCom/BigDataSE). IEEE, Rotorua, New Zealand, August 2019.

[18] V. Kurri, V. Raja, and P. Prakasam, "Cellular traffic prediction on blockchain-based mobile networks using LSTM model in 4G LTE network," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1088–1105, 2021.

[19] S. Li, D. Zhao, and Q. Li, "A framework for predicting network security situation based on the improved LSTM," *EAI Endorsed Transactions on Collaborative Computing*, vol. 4, no. 13, Article ID 165278, 2020.

[20] H. Zhang, C. Kang, and Y. Xiao, "Research on network security situation awareness based on the LSTM-DT model," *Sensors*, vol. 21, no. 14, p. 4788, 2021.

[21] D. Liu, J. Cheng, Z. Yuan et al., "Prediction methods for energy internet security situation based on hybrid neural network [C]//IOP Conference Series: earth and Environmental Science," *IOP Conference Series: Earth and Environmental Science*, vol. 645, no. 1, Article ID 012085, 2021.

[22] H. Yang, R. Zeng, G. Xu, and L. Zhang, "A network security situation assessment method based on adversarial deep learning," *Applied Soft Computing*, vol. 102, Article ID 107096, 2021.

[23] L. Chen, M. Zheng, Z. Liu, F. Chen, K. Zhou, and B. Liu, "SAE + Bi-GRU Based Security Situation Prediction for Smart Grid," in *Proceedings of the international conference on emerging internetworking, data & web technologies*, pp. 21–30, Springer, Okayama, Japan, October 2022.

[24] B. Wang, W. Kong, H. Guan, and N. N. Xiong, "Air quality forecasting based on gated recurrent long short term memory model in internet of things," *IEEE Access*, vol. 7, pp. 69524–69534, 2019.

[25] A. O. Boudraa and J. C. Cexus, "EMD-based signal filtering," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 6, pp. 2196–2202, 2007.

[26] R. Ortega-Bueno, P. Rosso, and J. E. M. Pagola, "UO UPV2 at HAHA 2019: BiGRU neural network informed with linguistic features for humor recognition," in *Proceedings of the iberian languages evaluation forum (iberlef 2019)*, CEUR-WS, Bilbao, Spain, 2019.

[27] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[28] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, 2015.

[29] S. Sennan, S. Ramasubbareddy, S. Balasubramaniyam, A. Nayyar, M. Abouhawwash, and N. A. Hikal, "T2FL-PSO: type-2 fuzzy logic-based particle swarm optimization algorithm used to maximize the lifetime of internet of things," *IEEE Access*, vol. 9, pp. 63966–63979, 2021.

[30] M. A. M. De Oca, T. Stutzle, M. Birattari, and M. Dorigo, "Frankenstein's PSO: a composite particle swarm optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1120–1132, 2009.

[31] D. Tian and Z. Shi, "MPSO: m," *Swarm and Evolutionary Computation*, vol. 41, pp. 49–68, 2018.

[32] S. N. Omkar, R. Khandelwal, T. V. S. Ananth, G. Narayana Naik, and S. Gopalakrishnan, "Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures," *Expert Systems with Applications*, vol. 36, no. 8, pp. 11312–11322, 2009.

[33] Q. Zhang, "Wei Yachen Particle swarm optimization algorithm with independent adaptive parameter adjustment," *J] Computer Science and Exploration*, vol. 14, no. 04, pp. 637–648, 2020.

[34] H. B. Dong, D. J. Li, and X. P. Zhang, "A partial swarm optimization algorithm for dynamically adjusting inertia weights," *Computer Science*, vol. 45, no. 2, pp. 98–102 +139, 2018.

[35] X. Zhang, D. X. Zou, and P. Xiao, "Adaptive simplified particle swarm optimization and its application," *Computer Engineering and Applications*, vol. 55, no. 8, pp. 250–263, 2019.

[36] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[37] B. O. Arani, P. Mirzabeygi, and M. S. Panahi, "An improved PSO algorithm with a regional diversity-preservation scheme and enhanced exploration-exploitation balance," *Swarm and Evolutionary Computation*, vol. 11, pp. 1–15, 2013.

[38] T. K. Kim, "T test as a parametric statistic," *Korean Journal of Anesthesiology*, vol. 68, no. 6, p. 540, 2015.

[39] C. Nguyen, S. DiVerdi, A. Hertzmann, and F. Liu, "Depth Conflict Reduction for Stereo Vr Video Interfaces," in *Proceedings of the 2018 chi conference on human factors in computing systems*, pp. 1–9, Montreal Canada, Apirl 2018.

[40] J. Derrac, S. García, F. Herrera, G. Salvador, and D. Molina, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[41] https://www.cert.org.cn/publish/main/index.html.

[42] W. F. Jiang, *Research on Network Security Situation Prediction Based on Multi-Model Weight Extraction and Fusion*, Lanzhou University of Technology, Lanzhou, Gansu, China, 2016.

[43] D. Xu, S. Ji, Y. Meng, and Z. Zhang, "A Software Reliability Prediction Algorithm Based on MHPSO-BP Neural Network," in *Proceedings of the 2017 global conference on mechanics and civil engineering (gcmce 2017)*, pp. 47–53, Atlantis Press, Guangzhou, China, June 2017.

[44] X. Li, H. Chen, and B. Ariann, "Computer network security evaluation model based on neural network," *Journal of Intelligent and Fuzzy Systems*, vol. 37, no. 1, pp. 71–78, 2019.

[45] L. Shang, W. Zhao, J. Zhang, Q. Fu, Q. Zhao, and Y. Yang, "Network Security Situation Prediction Based on Long Short-Term Memory Network," in *Proceedings of the 2019 20th asia-pacific network operations and management symposium (apnoms)*, pp. 1–4, IEEE, Matsue, Japan, September 2019.

[46] W. Feng, Y. Wu, and Y. Fan, "A new method for the prediction of network security situations based on current neutral network with gated current unit," *International Journal of Intelligent Computing and Cybernetics*, vol. 13, 2018.