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

Biao Zhang
Harbin University of Science and Technology

Shaopei Ji (✉️ 435323646@qq.com)
China Electronics Technology Group Corporation (China)

Jiazhong Xu
Harbin University of Science and Technology

Mingqi Jia
Harbin University of Science and Technology

Liwei Deng
Harbin University of Science and Technology

Research Article

Keywords: Network security situation prediction, Empirical mode decomposition, Collaborative update, BiGRU neural network, Prediction accuracy

DOI: https://doi.org/10.21203/rs.3.rs-778136/v1

License: ☕️ ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
Network Security Situation Prediction Model 
Based on EMD and ELPSO Optimized BiGRU Neural Network 

Biao Zhang¹, Shaopei Ji²*, Jiazhong Xu¹, Mingqi Jia¹, Liwei Deng¹ 
¹. School of Automation, Harbin University of Science and Technology, Harbin, China, 150080 
². No.30 Institute of CETC, Chengdu Sichuan 610041. (Corresponding author) E-mail: 435323646@qq.com

Abstract: The traditional network security situation prediction method depends on the accuracy of historical situation values, and there are correlations and differences in importance among various network security factors. To solve the above problems, a combined forecasting model based on Empirical Mode Decomposition and improved Particle Swarm Optimization (ELPSO) to optimize BiGRU neural network (EMD-ELPSO-BiGRU) is proposed. Firstly, the model decomposes the network security situation data sequence into a series of intrinsic modal components by empirical mode decomposition; Then, the prediction model of the BiGRU neural network is established for each modal component, and an improved Particle Swarm Optimization Algorithm (ELPSO) is proposed to optimize the super parameters of BiGRU neural network. Finally, the prediction results of each modal component are superimposed to obtain the final prediction value of the network security situation. In the experiment, on the one hand, ELPSO is compared with other particle swarm optimization algorithms, and the results show that ELPSO has better optimization performance; On the other hand, through simulation test and comparison between EMD-ELPSO-BiGRU and other traditional forecasting methods, the results show that the established combined forecasting model has higher forecasting accuracy.

Keywords: Network security situation prediction; Empirical mode decomposition; Collaborative update; BiGRU neural network; Prediction accuracy

1 Introduction

With the rapid popularization of the network in life, industry, and other fields, people's demand for the network is exploding, which is followed by endless network security problems. Various network security threats and complex network attacks mean to make it difficult for traditional anti-virus software, firewalls, and other protective means to take the initiative to defend. To cope with the severe form of network security, the network security situational awareness technology has received extensive attention since it was put forward [1-2]. As its core research category, Through the analysis and fusion of historical network data, the future security of the network can be predicted in advance within a period of time, various threats and attacks that may
occur in the future can be warned, network administrators can be helped to allocate network resources reasonably in advance, and corresponding measures can be taken to deal with various network risks.

With the continuous development of machine learning and the gradual deepening of artificial intelligence learning, the application of neural networks in the field of network security situation prediction has become a research hotspot in recent years. Compared with the existing prediction models, the neural network model has multi-features, high fault tolerance, self-organization ability, and self-study ability, has the strong nonlinear mapping and generalization ability for complex systems, and can directly construct the mapping relationship between multi-source heterogeneous data and situation values [3-5].

Liang W [6] et al. Proposed a network security situation prediction model based on the weighted Hidden Markov Model to solve the problem that existing methods cannot make full use of historical data to predict future changes. Firstly, multi-scale entropy information is used to solve the problem of training data, then the parameters of HMM transfer matrix are optimized, and finally, the autocorrelation coefficient is used to predict the future security situation based on the correlation between the features of historical data. Feng W et al. [7] proposed a new method of network security state prediction based on recurrent neural networks. Firstly, the internal and external information features are extracted from the original time series network data. Then the extracted features are applied to the depth RNN model for training and verification. After iteration and optimization, the accuracy of network security prediction by the state will be obtained through a well-trained model, and the model is robust to unstable network data. Experiments on the dataset show that compared with the traditional model, this method obtains more accurate and reliable prediction results. Xie B et al. [8] proposed a cloud security situation prediction model based on the evolutionary functional network for the dynamic uncertainty and prediction accuracy of security situation prediction in a complex cloud network environment. Firstly, this method combines evolutionary algorithm with the functional network, constructs evolutionary functional network model, and solves the problems of basis function selection and basis function coefficient correction of prediction model. Then, the random approximation algorithm is used to process and understand the cloud security situation elements, and the computational complexity of the prediction model is reduced by dimension reduction method. Finally, by constructing the credibility matrix of the uncertain influence relationship of security situation elements, the multivariate nonlinear regression algorithm is used to predict the cloud security situation. Simulation results show that compared with the BP model and RAN-RBF model, the prediction accuracy of this model is improved by 8.2% and 6.9%, and the convergence efficiency is improved by 12.3% and 10.8%, respectively. Hu G Y et al. [9] proposed a network security state prediction method based on the hidden Belief Rule Base model to solve the problem that the current methods cannot use mixed information containing qualitative knowledge and quantitative data. Firstly, an evolutionary strategy (CMA-ES) algorithm is developed, and then the parameters of the BRB model are optimized based on the CMA-ES algorithm. Experimental results show that, compared with other methods, the hidden BRB model and the improved CMA-ES algorithm can effectively predict the network security situation, and make full use of qualitative knowledge to improve the prediction accuracy. In order to solve the uncertainty conversion problem between qualitative concepts and quantitative values in situational awareness and improve the matching degree between data distribution and actual situation, Wu R F et al. [10] proposed an attitude
prediction method based on a multidimensional cloud model. Firstly, several cloud models are constructed according to the training historical data, and the corresponding key attributes are obtained. Then, the future situation value of the network is obtained by inverse operation of membership function. Experiments show that this method has good detection accuracy and predictable trend stability. In order to predict the development trend of network security more accurately, Lu H et al. [11] established a network security situation prediction model based on the Grey Wolf Optimization algorithm to optimize Support Vector Machine parameters. This method overcomes the problems of neural network training and local optimization, makes it more universal, and effectively improves the prediction effect of support vector machines. Simulation results show that the model improves the accuracy of prediction and shows the overall trend of the network security situation. Xiao J Q et al. [12] proposed a method of applying wavelet kernel function network to network security situation prediction based on the existing network security situation prediction model. Wavelet kernel function network combines the advantages of neural network and support vector machine, avoids the binding quadratic convex programming problem solved by Support Vector Machine, and introduces kernel function to avoid the size disaster problem of network scale. Experimental tests show that this method can accurately obtain the situation prediction results and has good situation prediction ability, which provides a new key for the prediction of the network security situation.

In order to better deal with the influence of the relationship between network security factors on the network security situation, this paper proposes a combined forecasting model (EMD-ELPSO-BiGRU) based on Empirical Mode Decomposition and improved Particle Swarm Optimization (ELPSO) to optimize the BiGRU neural network. Considering that multi-attribute security index data is used as data support in this paper, Based on the BiGRU model, multi-attribute network security data are fused, and the network security situation data sequence is decomposed into a series of intrinsic modal functions by using empirical mode decomposition, and the optimized Particle Swarm Optimization algorithm is used to determine the network super parameters and network scale. Further, improve the performance of the model. The model retains the original network security data to a great extent and maximizes the correlation between mining data, thus improving the prediction accuracy. This paper mainly includes the following points:

1. Predict the future network security situation by using the correlation and difference of different network security index data in the time dimension, fully mine the information contained in the time series and consider the dependence between different index data.
2. In order to further improve the performance and reduce the prediction error, the improved PSO is used to optimize the super parameters of the neural network.

The rest of this paper is arranged as follows: in the second section, the concept of empirical mode decomposition is briefly introduced; In the third and fourth sections, the method proposed in this paper is introduced in detail, from the overall design to each part. The fifth section discusses the experiment and results. Finally, the sixth section summarizes the work of this paper.

2 Empirical mode decomposition

Empirical Mode Decomposition [13] is a method to deal with nonlinear and non-stationary time-varying sequences. This method adaptively decomposes signals according to the time scale
characteristics of the data itself. It is considered as a breakthrough to Fourier analysis and wavelet analysis based on stationary and linear assumptions. The screening process of the EMD algorithm is to decompose complex time series data into a finite number of Intrinsic Mode Functions, and the decomposed IMF components contain the fluctuation information of the original data in different time scales.

For a given original time series sample data, firstly calculate the local maximum value and the minimum value on \( x(t) \), respectively carry out interpolation fitting on the local maximum value and the minimum value to obtain the upper and lower packet network \( x_{\text{max}}(t) \) and \( x_{\text{min}}(t) \) sequences of the original data \( x(t) \), and then calculate the upper and lower envelope sequences The average value of the sequence obtains the average value sequence \( m_1(t) \):

\[
m_1(t) = \frac{x_{\text{max}}(t) + x_{\text{min}}(t)}{2}
\]

The mean value sequence is subtracted from the original sequence to obtain a new sequence \( h_1^1(t) \) with low frequency removed:

\[
h_1^1(t) = x(t) - m_1(t)
\]

Where \( h_1^1(t) \) does not satisfy the condition of the eigenmode function. At this time, \( h_1^1(t) \) is taken as the original sequence, repeated \( k \) times until the average curve tends to zero, and \( c_1(t) = h_1^1(t) \) is recorded and \( c_1(t) \) is regarded as an IMF. Judgment conditions are

\[
SD = \sum_{k=1}^{\infty} \left[ \frac{[h_1^{k-1}(t) - h_1^1(t)]^2}{[h_1^1(t)]^2} \right]
\]

Where SD is the screening threshold, and the general value is between 0.2 and 0.3. \( c_1(t) \) is subtracted from \( x(t) \) to obtain a residual sequence \( r_1(t) = x(t) - c_1(t) \) from which the highest frequency component is removed. The above screening process is repeated to obtain subsequent IMF components until \( c_n(t) \) is less than a predetermined error or \( r_n(t) \) is a monotone function, and the modal decomposition process is terminated. At this point, that original sequence \( x(t) \) may be represented by an IMF component of order \( n \) and a residual \( r_n(t) \):

\[
x(t) = \sum_{i=1}^{n} c_i(t) + r_n(t)
\]

3 Optimization of Super Parameters of BiGRU Neural Network Based on ELPSO

3.1 Improved Particle Swarm Optimization

3.1.1 Regular PSO

Particle Swarm Optimization is an intelligent search algorithm that simulates the social behavior of bird groups \(^{[14]}\). It searches for the solution of the problem cooperatively through
information sharing among individuals in the group. The specific mathematical description of the algorithm is as follows: 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, \( V_i = V_{i1}, V_{i2}, \ldots, V_{iD} \) represents the velocity of the \( i \)-th particle, where \( i = 1, 2, \ldots, N \). \( p_{\text{best}} \) represents the optimal position experienced by the \( i \)-th particle itself, and \( g_{\text{best}} \) represents the optimal position experienced by the whole group. In the whole evolution process of the algorithm, each particle updates its own speed and position by constantly updating \( p_{\text{best}} \) and \( g_{\text{best}} \), so as to find the best position of the particle when reaching the optimal fitness value, that is, the solution of the problem to be optimized. The particle velocity and position update formula are:

\[
V_{i}^{t+1} = \omega V_{i}^{t} + c_1 r_1 (p_{\text{best}} - X_{i}^{t}) + c_2 r_2 (g_{\text{best}} - X_{i}^{t}) \tag{5}
\]

\[
X_{i}^{t+1} = X_{i}^{t} + V_{i}^{t+1} \tag{6}
\]

Where \( \omega \) is inertia weight; \( c_1 \) and \( c_2 \) are learning factors, usually taking a value of 2; \( r_1 \) and \( r_2 \) are random numbers distributed in \([0, 1]\); \( t \) is the current number of iterations of the particle.

### 3.1.2 Improved particle swarm optimization algorithm

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, fewer parameters, and convenient implementation. However, the algorithm has the problem of premature convergence or local optimization in the search process, the main reason is that the population loses diversity in the optimization process \([15]\). Maintaining population diversity is an important measure to enhance the global search ability of the algorithm and avoid premature phenomena. Therefore, this paper adopts the learning strategy update mechanism based on evolutionary state information judgment in the iterative process of particle swarm optimization algorithm and proposes a particle swarm optimization algorithm based on a cooperative update of evolutionary state judgment and learning strategy (ELPSO).

Different from the traditional particle swarm optimization algorithm, the ELPSO algorithm uses the information of population evolution state to select the appropriate learning strategy. When the evolution state is greater than the fixed threshold, it is judged that the algorithm is in the convergence stage, and the full information learning strategy is adopted to update the information speed and position 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 algorithm is judged to be in the stage of jumping out of the local optimal. The algorithm adopts a local information learning strategy to maintain the diversity of the population according to the information update speed and position of the local optimal and optimal neighborhood particles so that the algorithm is not easy to fall into the local optimal.
3.1.2.1 Analysis of evolution status

In the iterative process of particle swarm optimization, the decrease 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:

\[
x_{\text{mean}}^d = \frac{1}{N} \sum_{i=1}^{N} x_i^d
\]

\[
div = \frac{1}{N} \sum_{i=1}^{N} \left( \sum_{d=1}^{D} (x_i^d - x_{\text{mean}}^d)(x_j^d - x_{\text{mean}}^d) \right)
\]

\[
E = e^{-\frac{div}{div_{\text{max}}}} \left( 1 - \frac{t}{T} \right) \in [0,1]
\]

Where: \( x_{\text{mean}}^d \) represents the average position between particles at the same latitude; \( N \) represent that total number of the population; \( D \) represent a particle dimension; \( k \) is used to adjust the sensitivity degree of the exponential function, and \( k \) is matched with it according to the population initialization state and the population diversity degree; \( t \) is the number of current iterations; \( T \) is the maximum number of iterations \( div \), and \( div_{\text{max}} \) represent the current population diversity and the maximum population diversity respectively. When the two are equal \( E = 1 \).

3.1.2.2 Neighborhood Selection Strategy

In the iterative process, the Hamming distance between each particle and other particles is calculated according to the coding characteristics of the particle swarm optimization algorithm, and it is sorted, and the specified number of neighbors of a given particle is obtained according to the sorting results.

\[
D_{ij} = H(x_i, x_j) \quad j = 1, ..., N; \ j \neq i
\]

\[
S = \text{sort}(D)
\]

\[
\text{Neighbors} = S(1:T)
\]

Where: \( D_{ij} \) represents the Hamming distance between the \( i \)-th particle and the \( j \)-th particle in the population; \( H \) is a function of calculating Hamming distance; \( S \) is the set of the sorting results; \( \text{Neighbors} \) represents the current set of neighborhood particles; \( T \) is that specify the number of neighbors.
3.1.2.3 All-information learning strategy

In order to improve the particle optimization problem, the ELPSO algorithm adopts a full information learning strategy to ensure optimization ability and convergence performance. In the iterative process of the ELPSO algorithm, particle $i$ gets information from neighborhood particles with better fitness values and avoid being affected by bad neighborhood particles. The neighborhood particles with better fitness value have a greater influence on the particle $i$. Based on the above ideas, the ELPSO algorithm adopts a full information learning strategy, and its speed and position update expressions are:

$$V_i^d = X(V_i^d + \frac{\phi}{k_i} \sum_{m=1}^{k_i} r_m \cdot \frac{f(p_{im})}{\text{sum}_i} \cdot (p_{im}^d - x_i^d)) \quad (13)$$

$$\text{sum}_i = \sum_{m=1}^{k_i} f(p_{im}) \quad (14)$$

$$S(V_i^d) = 1 - \sqrt{1 - (V_i^d)^2} \quad (15)$$

According to the related literature, the convergence coefficient $X$ and acceleration coefficient $\phi$ are used to adjust the particle velocity, and the algorithm has better performance [9, 17], where $X = 0.729$ $\phi = 4.1 k_i$, is the number of particles in the better neighborhood of particle $i$; $im$ is the $m$-th better neighborhood particle of a particle $i$; $p_{im}$ is the position of the particle $im$, $f(p_{im})$ is the fitness value of, and $\text{sum}_i$ is the sum of the fitness values of the particles in the better neighborhood; $r_m$ represents a number evenly distributed between $[0, 1]$.

Equation (15) is the particle position update formula. $V_i^d$ W52 is the particle velocity value and $S(V_i^d)$ is the probability value of velocity mapping. If the probability value is greater than the random number rand, the particle position vector takes its own complement set, otherwise, it will remain unchanged.

3.1.2.4 Local information learning strategies

In Particle Swarm Optimization, particle $i$ update speed and position according to the information of local optimal and optimal neighborhood particles and are less affected by other particles, so they can move more freely in the search space, which is beneficial to maintain population diversity. ELPSO adopts local information learning strategy, and its speed and position update expressions are as follows:

$$V_i^d = X[V_i^d + \frac{\phi}{2} r_1 (p_i^d - x_i^d) + \frac{\phi}{2} r_2 (p_{im}^d - x_i^d)] \quad (16)$$

Where: $p_i^d$ is the local optimal position of particle $i$; $p_{im}^d$ is the optimal position of the neighborhood of particle $i$; $r_1$ and $r_2$ represent numbers evenly distributed between $[0, 1]$. 
3.1.2.5 ELPSO algorithm flow

An ideal particle swarm optimization algorithm should not fall into local optimality while ensuring fast convergence speed. It is difficult to achieve this state by using a single learning strategy. Therefore, the ELPSO algorithm uses different learning strategies to solve complex optimization problems in different evolution states. Aiming at the problems of premature and easy to fall into local optimum in particle swarm optimization algorithm, The iterative process of particle swarm optimization is divided into two stages: jumping out of local optimum and convergence, and the evolution state is divided at the same time. If the evolution factor \( E < 0.7 \), the algorithm is judged to be in the stage of jumping out of local optimum, indicating 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 evolution factor \( E > 0.7 \) or \( E=0.7 \), the judgment algorithm is in the convergence stage, indicating that the population diversity is better. The all-information learning strategy should be selected to ensure that particles obtain information from neighboring particles with better fitness values 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 \( E \), and if \( E \) is less than 0.7, it is judged that the algorithm is in the stage of jumping out of local optimum; If \( E >= 0.7 \), the decision algorithm is in the 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 Equation (16); If the algorithm is in the convergence stage, the particle velocity is updated by equations (13) and (14).

Step 4 Particle position update. Use Equation (15) to update the particle position.

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

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

In the ELPSO algorithm, evolutionary state determination is the key to balance convergence and jump out of local optimum. The optimization mechanism of particle swarm optimization algorithm with evolutionary state determination and learning strategy updated cooperatively is shown in Fig. 1.
3.2 BiGRU neural network

Bidirectional Gated Recurrent Unit network is a bidirectional gating-based cyclic neural network, which is formed by combining forward GRU and backward GRU \[16\]. The GRU model is a variant of the long-term and short-term memory network LSTM \[17\]. LSTM solves the problems of gradient disappearance in the process of backpropagation in traditional RNN. The structure of LSTM neurons is shown in Fig. 2.

However, GRU keeps the LSTM effect and has a simpler structure, fewer parameters, and better convergence. The Gated Recurrent Unit (GRU) consists of two gates, reset gate \( R \) and
update gate $Z$. These two gating mechanisms can save the information in the long-term sequence without being cleared with time or removed because it is irrelevant to the prediction. Its structure is shown in Fig. 3.

![Fig. 3 Structure diagram of GRU neurons](image)

The calculation process of the GRU model is as follows:

\[
r_t = \sigma(W_r \cdot [h_{t-1}, x_t])
\]

(17)

\[
z_t = \sigma(W_i \cdot [h_{t-1}, x_t])
\]

(18)

\[
b_{1t} = \tanh(W_c \cdot [r_t \cdot h_{t-1}, x_t])
\]

(19)

\[
h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot b_{1t}
\]

(20)

The state $h_{t-1}$ transmitted at the last moment and the input $x_t$ of the current node are used to obtain two gating states, The Sigmoid function can convert the data into a value in the range of $0 \sim 1$, thus serving as a gate signal. After obtaining the gate signal, the reset gate $R$ is used to obtain the "reset" data $h_{t-1}' = h_{t-1} \cdot r$, then $h_{t-1}'$ is spliced with the input $x_t$, and then the data is scaled to the range of $-1 \sim 1$ through a tanh function. After that, the update gate $Z$ carries out two steps of forgetting and memorizing at the same time. The update expression is shown in Formula (20), where $Z$ ranges from 0 to 1, and the closer the gate signal is to 1, the more state information is brought in at the previous moment.

When a Bidirectional Gated Recurrent Unit models a text sequence, at each moment, the input will simultaneously provide two GRU with opposite directions, while the output is jointly determined by the two unidirectional GRU. The specific structure of BiGRU is shown in Fig. 4.
The specific calculation process is as follows:

\[ h_t = GRU(h_{t-1}, U_t) \]  
\[ \hat{h}_t = GRU(\hat{h}_{t-1}, U_t) \]  
\[ h_t = w_h \hat{h}_t + v_h \hat{h}_{t-1} + b_t \]

In the above formula, the concealment state of the forward GRU output at time \( t \) is \( h_t \), the concealment state of the reverse GRU output is \( \hat{h}_t \), and the concealment states \( h_t, w_t, \) and \( v_t \) of the BiGRU output are weight matrices; \( U_t \) is the GRU input at time \( t \); \( b_t \) is the bias vector.

### 3.3 Optimization of BiGRU Super Parameters by ELPSO Algorithm

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

Its mathematical definition is as follows:

\[ MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \]  

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 the BiGRU neural network involves the setting of multiple super parameters: the number of neurons \( m \), the time step \( T \), and the batch size. The number of neurons determines the fitting degree of the neural network, and the time step and batch size directly affect the results of model training. In practical application, different data sets correspond to different super parameter settings, which will affect the prediction accuracy\(^{18-19}\). In this paper, ELPSO is used to optimize these super parameters, and the neural network structure and training mode are adaptively optimized according to the input data to obtain the optimal combination of model.
parameters. The specific steps are as follows:

Step 1 initializes the relevant parameters of the algorithm and determines the change intervals of population size, iteration times, inertia weight, and learning factors.

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

Step 3 uses Formula (24) 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 Update the velocity and position of the particles.

Step 5 stops when the iteration times are reached or the fitness function of the particles tends to be stable, and the particles in the best position of the group are the optimal parameter combination obtained this time, otherwise, turn to step 4 to continue iteration.

The flow chart of using the ELPSO algorithm to solve the optimal parameter combination of the BiGRU model is shown in the following Fig. 5.

![Fig. 5 Flow chart of ELPSO optimization of BiGRU super parameters](image)
4 EMD-ELPSO-BiGRU network security situation prediction model

In order to analyze the characteristics of network security situation changes in detail, this paper proposes a combined forecasting model (EMD-ELPSO-BiGRU) based on empirical mode decomposition and improved Particle Swarm Optimization (ELPSO) to optimize BiGRU neural network. Firstly, the variational empirical mode decomposition is used to stabilize the network security situation sequence and decompose it into a series of different modal components to reduce the complexity of the network security situation sequence. Then, the BiGRU neural network sub-model is established for each modal component, and the parameters of the sub-model are optimized by ELPSO algorithm. Finally, the prediction results of each modal component of the network security situation sequence are superimposed to obtain the prediction value of the network security situation. The network security situation prediction process is shown in Fig. 6.
Network security situation
sequence data

Empirical mode decomposition
(EMD)

IMF\(_1\)

IMF\(_2\)

\ldots

IMF\(_k\)

EMD-ELPSO-BiGRU
prediction model

EMD-ELPSO-BiGRU
prediction model

EMD-ELPSO-BiGRU
prediction model

EMD-ELPSO-BiGRU
prediction model

IMF\(_1\)
prediction
results

IMF\(_2\)
prediction
results

\ldots

IMF\(_k\)
prediction
results

Reconstruction of predicted
values of each modal
component

Error analysis

Network security situation
prediction value

End

Fig. 6 Flow chart of network security situation prediction based on EMD-ELPSO-BiGRU
5 Experiences and Discussion

5.1 Performance Evaluation of ELPSO Algorithm

5.1.1 Benchmark Function

In order to test the effectiveness of the ELPSO algorithm proposed in this paper, this paper selects 12 test functions [20] to carry out comparative experiments with conventional Particle Swarm Optimization algorithm, improved Particle Swarm Optimization algorithm (MPSO) [21], and Quantum Particle Swarm Optimization algorithm (QPSO) [22]. The mathematical expressions of the 12 test functions are as follows [23].

1) Sphere function
\[ f_1(x) = \sum_{i=1}^{N} x_i^2 \]  
(25)

2) Schwefel function
\[ f_2(x) = \max \{ |x_i|, 1 \leq i \leq n \} \]  
(26)

3) Schwefel function
\[ f_3(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \]  
(27)

4) Step function
\[ f_4(x) = \sum_{i=1}^{n} (|x_i + 0.5|) \]  
(28)

5) Schaffers function
\[ f_5(x) = \sum_{i=1}^{N} (x_i^2 + x_{i+1}^2)^{0.25} \left[ \sin^2 \left( 50 \left( x_i^2 + x_{i+1}^2 \right)^{0.1} \right) + 1.0 \right] \]  
(29)

6) Rastrigin function
\[ f_6(x) = \sum_{i=1}^{N} (x_i^2 - 100 \cos(2\pi x_i) + 10) \]  
(30)

7) Griewank function
\[ f_7(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1 \]  
(31)
(8) Ackely function
\[ f_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e \]  
(32)

(9) Schaffer function
\[ f_9(x) = 0.5 + \frac{(\sin^2\sqrt{x_1^2 + x_2^2} - 0.5)}{(1 + 0.001(x_1^2 + x_2^2))^2} \]  
(33)

(10) Branin function
\[ f_{10}(x) = \left(x_2 - \frac{5.1}{4\pi}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10(1 - \frac{1}{8\pi}\cos x_1) + 10 \]  
(34)

(11) Six-Hump Camel-back function
\[ f_{11}(x) = 4x_1^2 - 2.1x_1^4 + \frac{x_1^6}{3} + x_1x_2 - 4x_2^2 + 4x_2^4 \]  
(35)

(12) Goldstein-price function
\[ f_{12}(x) = [1 + (x_1 + x_2 + 1)\frac{1}{2}(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \]  
(36)

Among the 12 test functions, the theoretical optimal values \( f_1 - f_9 \) are all 0, and the theoretical optimal values of \( f_{10}, f_{11} \) and \( f_{12} \) are 0.3979, -1.0316, and 3, respectively.

5.1.2 Analysis of simulation results

In the experiment, different PSO algorithms set the same population size \( N=40 \), the maximum iteration times are \( T_{\text{max}}=500 \), and the learning factor is \( c_1 = c_2 = 2 \). Other parameters are consistent with the original literature. In ELPSO algorithm: \( w_{\text{max}} = 0.9 \), \( w_{\text{min}} = 0.4 \), \( \sigma = 0.1 \); In the \( \text{betarnd}(a,b) \) function: \( a = 1 \), \( b = 2 \).

In order to test the performance of the algorithm, the experiment is divided into three groups. The dimensions in the algorithm are set to 2, 30, and 50. The four algorithms run independently 50 times respectively. The comparison results of each algorithm on MEAN and Standard Deviation are shown in Tables 1, 2, and 3. Fig. 2 shows the fitness value curves of some test functions.

Where \( f_9 - f_{12} \) is a 2-dimensional test function.

|               | \( f_9 \)   | \( f_{10} \) | \( f_{11} \) | \( f_{12} \) |
|---------------|-------------|-------------|-------------|-------------|
| MEAN          | 3.69 E-03   | 3.98 E-01   | 3.36 E-16   | 3.98 E-16   | 3.00 E+00   | 1.37 E-15   |
| STD           | 4.76 E-03   | 1.73 E-01   | 1.03 E+00   | 1.91 E-05   | 3.00 E+00   | 2.20 E-04   |
|               | 2.89 E-01   | 6.52 E-06   | -1.03 E+00  | 2.05 E-06   | 3.00 E+00   | 1.46 E-04   |
|               | 0.00 E+00   | 0.00 E+00   | 3.98 E-01   | 1.08 E-04   | 1.63 E-05   | 3.00 E+00   | 1.10 E-03   |

From the experimental results in Table 1, it can be seen that for the two 2-dimensional
functions $f_{11} - f_{12}$, the four algorithms can find the theoretical optimal value, which shows that there is no difference in the optimization accuracy of the algorithms. However, for the Schaffer function, which is relatively complex and difficult to find the theoretical optimal value, ELPSO and QPSO algorithms can find the theoretical optimal value. From the comparison results of 2-dimensional test functions, it can be seen that the optimization performance advantage of the ELPSO algorithm is not obvious.

Table 2 Experimental results of four algorithms (D = 30)

|       | $f_1$       | $f_2$       | $f_3$       | $f_4$       |
|-------|-------------|-------------|-------------|-------------|
| MEAN  | STD         | MEAN        | STD         | MEAN        | STD         |
| PSO   | 6.43 E-01   | 3.11 E-01   | 2.50 E-29   | 7.46 E-29   | 3.67 E+00   | 1.51 E+00   | 5.98 E+01   | 3.17 E+01   |
| MPSO  | 5.49 E-33   | 3.32 E-32   | 9.48 E-21   | 3.04 E-20   | 2.48 E-20   | 4.65 E-20   | 0.00 E+00   | 0.00 E+00   |
| QPSO  | 1.09 E-94   | 2.40 E-94   | 4.48 E-128  | 2.87 E-127  | 1.47 E-48   | 3.09 E-48   | 0.00 E+00   | 0.00 E+00   |
| ELPSO | 0.00 E+00   | 0.00 E+00   | 3.56 E-230  | 0.00 E+00   | 6.19 E-224  | 0.00 E+00   | 0.00 E+00   | 0.00 E+00   |

Table 3 Experimental results of four algorithms (D = 50)

|       | $f_1$       | $f_2$       | $f_3$       | $f_4$       |
|-------|-------------|-------------|-------------|-------------|
| MEAN  | STD         | MEAN        | STD         | MEAN        | STD         |
| PSO   | 9.32 E+01   | 8.95 E+00   | 5.10 E+01   | 1.10 E+01   | 3.22 E+01   | 5.06 E+00   | 4.18 E+00   | 1.03 E+00   |
| MPSO  | 1.57 E-08   | 1.78 E-08   | 0.00 E+00   | 0.00 E+00   | 9.75 E-01   | 6.24 E-01   | 8.88 E-16   | 0.00 E+00   |
| QPSO  | 1.93 E-25   | 7.08 E-25   | 0.00 E+00   | 0.00 E+00   | 0.00 E+00   | 0.00 E+00   | 8.88 E-16   | 0.00 E+00   |
| ELPSO | 0.00 E+00   | 0.00 E+00   | 0.00 E+00   | 0.00 E+00   | 0.00 E+00   | 0.00 E+00   | 8.88 E-16   | 0.00 E+00   |

In order to further compare the differences in the optimization performance of the algorithms, the eight test functions $f_1 - f_8$ are compared in different dimensions. From the comparison results in Table 2 and Table 3, it can be obtained that the ELPSO algorithm has further improved the optimization effect of test functions compared with other algorithms on 8 test functions. For unimodal problem $f_1 - f_3$ s and multimodal problem $f_4$ s, the average value of the ELPSO algorithm is smaller than the other three algorithms, which shows that ELPSO has better optimization accuracy. Especially for the unimodal function $f_1$, which is generally used to test the
convergence speed of the algorithm, the ELPSO algorithm can find the theoretical optimal solution 0, and as can be seen from Fig. 7 (a), the ELPSO algorithm can converge to the optimal solution with only a few iterations. At the same time, for the nonlinear multimodal function $f_6 - f_8$, MPSO, QPSO, and ELPSO have achieved the same optimization accuracy on functions $f_6$ and $f_8$, and both MPSO and ELPSO can find the theoretical optimal solution 0 on function $f_6 - f_7$. However, as can be seen from Fig. 7, the ELPSO algorithm requires fewer iterations, so it has a faster convergence speed. As can also be seen from the results of Table 2 and Table 3, The standard deviation of the ELPSO algorithm is smaller than the other three algorithms, so the algorithm has a better stability. Therefore, compared with the other three algorithms, the ELPSO algorithm can have high optimization accuracy and fast convergence speed on the unimodal or multimodal function problems (especially the test function problem with higher dimensions).

Fig.7 Convergence curves of different test functions

5.1.3 T-Test and Friendman test

As seen from Table 1, 2, and 3, QPSO and ELPSO algorithms have the same optimization accuracy in 8 out of 12 test functions. In order to further clarify whether there are significant differences between the algorithms, this paper discusses the performance of the algorithms from a statistical point of view; T-test [24] and Friendman test [25] are introduced to test the performance of the four algorithms on 12 test functions; The experimental results are shown in Table 4; T-test results show that the performance difference between PSO algorithm and ELPSO algorithm is
obvious; Compared with MPSO algorithm, ELPSO has better performance on 7 test functions, 4 have no difference, and 1 is worse; Compared with QPSO, ELPSO has 4 better functions, 6 have no difference, and 2 are worse. Friendman test shows that the better the performance of the algorithm, the smaller the rank mean. According to the Friendman test results of the four algorithms, the rank means the value of the ELPSO algorithm is the smallest, and the performance of the ELPSO algorithm is the best among the four algorithms. Synthesizing the two test results, it can be seen that the performance of the ELPSO algorithm is better than the other three algorithms. Among them, "+" indicates that the ELPSO algorithm is superior to other algorithms, "=" indicates that there is no obvious difference between algorithms, "-" indicates inferior to other algorithms, and $w/l/l$ indicates the statistical number of these three comparison results respectively.

### Table 4 Results of T-test and Friendman test for four algorithms

|       | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $w/l/l$ | Rank mean |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|---------|---------|---------|---------|-----------|
| PSO   | +     | +     | +     | +     | +     | +     | =     | =     | =     | =       | =       | +       | 9/3/0    | 3.17      |
| MPSO  | +     | +     | +     | =     | =     | +     | =     | =     | =     | -       | +       | 7/4/1    | 3.08     |
| QPSO  | +     | +     | +     | =     | =     | =     | =     | =     | =     | -       | -       | =       | 4/6/2    | 1.96      |
| ELPSO |       |       |       |       |       |       |       |       |       |         |         | 1.79     |           |

### 5.1.4 Average number of iterations at specified precision

From the experimental results in Section 5.1.2, Tables 2, 3, and 4 show the comparison of experimental results on the optimization accuracy of the algorithms under the specified number of iterations. In order to comprehensively analyze the performance of the algorithms, this section shows 4 algorithms to test 8 test functions ($f_1 - f_6$) under the specified accuracy $10^{-10}$ with a dimension of 30. The average number of iterations for each algorithm to run 50 times independently is shown in Table 5.

### Table 5 Average number of iterations under the specified accuracy

|       | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| PSO   | /     | 243   | /     | /     | /     | /     | /     | /     |
| MPSO  | 164   | 150   | 213   | 94    | /     | 113   | /     | 208   |
| QPSO  | 62    | 18    | 102   | 22    | 179   | 61    | 62    | 98    |
| ELPSO | 11    | 12    | 16    | 4     | 34    | 9     | 10    | 16    |

As can be seen from the experimental results in Table 5, The PSO algorithm only achieves the specified accuracy on one test function, The MPSO algorithm achieves the specified accuracy in 6 of the 8 functions, The QPSO and ELPSO algorithms achieve the specified accuracy on all test functions. And compared with the other three algorithms, The ELPSO algorithm can achieve the specified accuracy with the least number of iterations, and the average number of iterations is between 4 and 34, which shows that the ELPSO algorithm has obvious advantages in convergence speed and high optimization performance, which further shows that the ELPSO algorithm has the characteristics of fast convergence speed.
5.2 Simulation Analysis of Network Security Situation Prediction

5.2.1 Selection of Network Security Situation Data

This paper uses the weekly report data of security situations released by the National Internet Emergency Center as the experimental basis. The National Internet Emergency Center is a network security technology coordination organization in mainland China. It mainly carries out statistical processing of security incidents across the country and evaluates the network security status, and releases security information on a weekly, monthly and annual basis. Dynamic Weekly 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 websites implanted with backdoors in China, the number of counterfeit pages of domestic websites and the number of new information security vulnerabilities. In this paper, the safety data of 120 issues from the 31st issue of 2017 to the 45th issue of 2019 are selected as the experimental basis to verify the superiority of this method. Citing the evaluation method of the document [26] to quantify the original data, the network security situation value of 120 weeks is obtained, and the specific quantitative model is shown in Fig. 8.

![Fig.8 Quantitative model of network security situation assessment](image)

5.2.2 Experimental data and its pretreatment

In this paper, the data of the first 101 weeks are selected as the training set and the data of the last 18 weeks are selected as the test set according to the time sequence. The time window is set as the time step of the cyclic neural network, and the prediction time is one week. Due to the complexity and randomness of the network environment and the large difference in the dimensions of situation values, the activation function of the neural network used in this paper is extremely sensitive to whether the value of the input data is within [-1, 1]. Therefore, standardizing the data can accelerate the convergence speed and improve the prediction accuracy of the neural network. The input data is processed by data normalization, and the specific calculation formula is as follows [27].

\[
x' = \frac{x - \min(x)}{\max(x) - \min(x)}
\]  

(37)
Where \( x \) and \( x' \) are data before processing, and \( \min(x) \) \( \max(x) \) are minimum and maximum values in the data set. Therefore, the normalized network security situation value is shown in Fig. 9.

![Fig. 9 Normalized Network Security Situation Time Series](image)

### 5.2.3 Model measurement and evaluation index

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:

\[
\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} | y_i - \hat{y}_i | \tag{38}
\]

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{39}
\]

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 \( c_1, c_2 \) and is 2.5, the minimum value is 0.5, and the weight factor \( w \) is 0.8.
5.2.4 Selection of optimal model parameters

Fig. 10 shows the training results of the ELPSO algorithm optimizing the BiGRU neural network. The number of neurons, time step size, and batch size gradually converge to the optimal value as the algorithm is updated. As can be seen from Fig. 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. Thus, the super parameter with the best performance is obtained to correct the model structure of the BiGRU neural network and obtain the optimal parameter combination.
5.2.5 Analysis of simulation results

In order to evaluate the performance of the model proposed in this paper in network security situation prediction, comparative experiments are carried out with traditional machine learning methods and depth learning methods, including BP [28], LSTM [29], BiGRU, 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 Python 3.7 environment. The hardware configuration is the 64-bit operating system, and the processor is Inter (R) Core (TM) i5-8500 CPU 3.00 GHz.

5.2.5.1 EMD decomposition of experimental data

Firstly, EMD decomposition 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 one residual component R are obtained, as shown in Fig. 11. Judging from the modal component characteristics of the network security situation data sequence after decomposition, it is generally believed that the high-frequency component reflects the random influence part of the 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.
5.2.5.2 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 6. 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 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 6, 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.

| 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 |

The results in Table 7 can further prove that the EMD-ELPSO-BiGRU model can obtain good prediction results at most time points. Table 7 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.
Table 7 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         |

Fig. 12 shows a comparison of prediction value accuracy between EMD-ELPSO-BiGRU and basic prediction models such as BiGRU, LSTM, 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.

![Comparison of situation prediction of different prediction models](image)

Fig.12 Comparison of situation prediction of different prediction models
5.2.5.3 Forecast Time Comparison

Because the evaluation standard of time series prediction depends not only on the accuracy of prediction but also on the accuracy under different prediction periods. In this paper, the prediction accuracy of different prediction models under different prediction periods is compared, and the results are shown in Fig. 13. It can be seen that all models have the smallest error in single-step prediction. Under the same prediction time, the EMD-ELPSO-BiGRU model has better prediction ability. With the increase of prediction time, the prediction error gradually increases and then changes stably. The model has certain robustness.

![Fig.13 Comparison of mean absolute error under different forecast time](image)

5.2.5.4 Convergence analysis

The previous section compared the complexity of training different models once. Fig. 14 shows the change of training error of the models with the number of iterations. It can be observed from the figure that the method in this paper has obvious advantages in convergence speed and convergence accuracy, which shows that the model can learn data well.

![Fig.14 Curve of training error changing with the number of iterations](image)
6 Conclusion

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 is 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 in benchmark function optimization experiment; Then, EMD-ELPSO-BiGRU and BP, LSTM, BiGRU, ELPSO-BiGRU models are used for network security situation prediction experiments, and the following conclusions can be drawn:

The main results are as follows:

1) ELPSO algorithm adopts full information learning strategy in the convergence stage based on evolutionary state decision, which has faster convergence speed than other algorithms; and adopts local information learning strategy in the jump out of local optimal stage based on evolutionary state decision, which effectively avoids the algorithm from falling into local optimal by maintaining population diversity.

2) Empirical mode decomposition can decompose the network security situation sequence thoroughly, which can reduce the non-stationarity of the data sequence. When the data after empirical mode decomposition is predicted by using the neural network, the network has higher prediction accuracy and generalization ability;

3) Compared with traditional BP neural network, LSTM, BiGRU, and ELPSO-GRU, the 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, stock price prediction, etc.

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 analysed during the current study are available from the corresponding author on reasonable request.

References

[1]. Gutzwiller R, Dykstra J, Payne B. Gaps and Opportunities in Situational Awareness for Cybersecurity[J]. Digital Threats: Research and Practice, 2020, 1(3): 1-6.

[2]. Kokkonen T. Architecture for the cyber security situational awareness system[M]/Internet of things, smart spaces, and next generation networks and systems. Springer, Cham, 2016: 294-302.

[3]. Yang G R, Wang X J. Artificial neural networks for neuroscientists: A primer[J]. Neuron, 2020, 107(6): 1048-1070.

[4]. Jordan M I , Mitchell T M . Machine learning: Trends, perspectives, and prospects[J].
[5]. Dral P O. Quantum chemistry in the age of machine learning[J]. The journal of physical chemistry letters, 2020, 11(6): 2336-2347.

[6]. Liang W, Chen Z, Yan X, et al. Multiscale entropy-based weighted hidden Markov network security situation prediction model[C]/2017 IEEE International Congress on Internet of Things (ICIOT). IEEE, 2017: 97-104.

[7]. Feng W, Wu Y, Fan Y. A new method for the prediction of network security situations based on recurrent neural network with gated recurrent unit[J]. International Journal of Intelligent Computing and Cybernetics, 2018.

[8]. Xie B, Zhao G, Chao M, et al. A prediction model of cloud security situation based on evolutionary functional network[J]. Peer-to-Peer Networking and Applications, 2020: 1-15.

[9]. Hu G Y, Zhou Z J, Zhang B C, et al. A method for predicting the network security situation based on hidden BRB model and revised CMA-ES algorithm[J]. Applied Soft Computing, 2016, 48: 404-418.

[10]. Wu R F, Chen G L. Research of network security situation prediction based on multidimensional cloud model[C]/2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. IEEE, 2012: 409-414.

[11]. Lu H, Zhang G, Shen Y. Cyber Security Situation Prediction Model Based on GWO-SVM[C]/International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. Springer, Cham, 2019: 162-171.

[12]. Xiao J Q. Prediction of networks security situation based on wavelet kernel function network[C]/Applied Mechanics and Materials. Trans Tech Publications Ltd, 2012, 127: 53-58.

[13]. Boudraa A O, Cexus J C. EMD-based signal filtering[J]. IEEE transactions on instrumentation and measurement, 2007, 56(6): 2196-2202.

[14]. Marini F, Walczak B. Particle swarm optimization (PSO). A tutorial[J]. Chemometrics and Intelligent Laboratory Systems, 2015, 149: 153-165.

[15]. Sennan S, Ramasubbareddy S, Balasubramaniyam S, et al. T2FL-PSO: Type-2 Fuzzy Logic-based Particle Swarm Optimization Algorithm used to Maximize the Lifetime of Internet of Things[J]. IEEE Access, 2021, 9: 63966-63979.

[16]. Ortega-Bueno R, Rosso P, Pagola J E M. UO UPV2 at HAHA 2019: BiGRU neural network informed with linguistic features for humor recognition[C]/Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEUR-WS, Bilbao, Spain (9 2019). 2019.

[17]. Greff K, Srivastava R K, Koutnik J, et al. LSTM: A search space odyssey[J]. IEEE transactions on neural networks and learning systems, 2016, 28(10): 2222-2232.

[18]. SRIVASTAVA D, SINGH Y, SAHOO A. Auto tuning of RNN hyper-parameters using cuckoo search algorithm[C]/Proc. of the12th International Conference on Contemporary Computing, 2019: 1-5.

[19]. GUO X, STEIN B V. A new approach towards the combined algorithm selection and hyper-parameter optimization problem[C]/Proc. of the IEEE Symposium Series on Computational Intelligence, 2019: 2042-2049.

[20]. De Oca M A M, Stutzle T, Birattari M, et al. Frankenstein's PSO: a composite particle swarm optimization algorithm[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5):
1120-1132.

[21] Tian D, Shi Z. MPSO: Modified particle swarm optimization and its applications[J]. Swarm and evolutionary computation, 2018, 41: 49-68.

[22] Omkar S N, Khandelwal R, Ananth T V S, et al. Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures[J]. Expert Systems with Applications, 2009, 36(8): 11312-11322.

[23] Arani B O, Mirzabeygi P, Panahi M S. An improved PSO algorithm with a territorial diversity-preserving scheme and enhanced exploration–exploitation balance[J]. Swarm and Evolutionary Computation, 2013, 11: 1-15.

[24] Kim T K. T test as a parametric statistic[J]. Korean journal of anesthesiology, 2015, 68(6): 540.

[25] Nguyen C, DiVerdi S, Hertzmann A, et al. Depth conflict reduction for stereo vr video interfaces[C]//Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. 2018: 1-9.

[26] JIANG W F. Research on network security situation prediction based on multi-model weight extraction and fusion [D]. Lanzhou: Lanzhou University of Technology, 2016.

[27] Xu D, Ji S, Meng Y, et al. A Software Reliability Prediction Algorithm Based on MHPSO-BP Neural Network[C]//2017 Global Conference on Mechanics and Civil Engineering (GCMCE 2017). Atlantis Press, 2017: 47-53.

[28] Li X, Chen H, Ariann B. Computer network security evaluation model based on neural network[J]. Journal of Intelligent & Fuzzy Systems, 2019, 37(1): 71-78.

[29] Shang L, Zhao W, Zhang J, et al. Network Security Situation Prediction Based on Long Short-Term Memory Network[C]//2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE, 2019: 1-4.

Acknowledgement

The paper is supported by Heilongjiang Provincial Key Laboratory of Complex Intelligent System and Integration. This project was partially funded by the major science and technology projects of Sichuan Province (No.2017GZDZX0002)

Author contributions

ZB run the analysis and wrote the paper. JSP conceptualized the problem, run the analysis and wrote the paper. XJZ, JMQ and DLW conceptualized the problem and wrote the paper. All authors read and approved the final manuscript.

Competing interests

The authors declare no conflict of interest.