

Nonlinear Time Series Prediction of Fuzzy Inference System under Quantum Computing and New Self-evolving Interval Type II LSTM

Peng Mei*

Party School of the Central Committee of C.P.C, Beijing, China

Meipeng@ccps.gov.cn

Abstract. To enhance the prediction accuracy rate of the fuzzy inference system (FIS) in nonlinear time series, the Long Short-Term Memory (LSTM) network is integrated into the FIS to improve the ability of neural networks to process the fuzzy information. On this basis, the Quantum Echo State Network (QESN) is added and optimized by the particle algorithm to predict the nonlinear time series. The results indicate that the proposed new self-evolving interval type II LSTM (eit2FNN-LSTM) can realize single-point prediction; also, it can be used for granular-level prediction by combining with information particles, thereby achieving long-term prediction; besides, the QESN can predict the time series well, and its accurateness rate is higher than the benchmark algorithm. Therefore, the FIS based on quantum computing and the new eit2FNN-LSTM can effectively improve the prediction accuracy rate in the nonlinear time series, providing new ideas for prediction problems in other fields.

Keywords: Quantum computing, LSTM, Fuzzy inference, Time series prediction.

1. Introduction

With the continuous progress of electronic technology and networks, computers have become an indispensable tool in people's daily lives. Under various functions, the most basic computing unit in computers is various logic devices made of semiconductor materials, which convert all information in the real world into two signals, "0" and "1" for storage and operation [1, 2]. However, the development of computers also has a bottleneck period. At the same time, it is found in the microscopic world that there is a big difference in the existence mode and movement law of the material in the world and the familiar classic world; especially, the "quantum state (QS)" of the microscopic particle can exist simultaneously at multiple different positions or both "0" and "1" states; in addition, there may also be "quantum entanglement" between the quantum; despite the distance between two particles, they will affect each other at any time [3, 4]. Therefore, experts have begun exploring quantum physics actively, hoping to understand and investigate quantum computing as much as possible and have quantum computing replace classical computing to make predictions with relevant affairs.

Time series denotes data acquired on continuous time nodes that can reflect the fluctuation of an event or phenomenon with time. It is a collection of data that constitutes the time domain. In people's daily

lives, time series can be seen everywhere, such as weather data that fluctuates with time, people's consumption levels at different times, and the occurrence of peaks and troughs in stocks in the financial market; all of these are associated with time, and different situations may occur in a short time. Therefore, time series is critical in the natural and economic fields [5, 6].

Meanwhile, a large amount of data in reality can be modeled through time series; by looking for changing laws, things' future development and direction can be predicted. In the prediction process, past events of things may occur repeatedly, and the characteristics of these repeated occurrences can be seen statistically. There can be a connection between each thing, and this connection will show some statistical laws [7, 8]. Many real-life examples can be built into the corresponding event series prediction model and then predict the future development of things. However, the accuracy rate of many time series data is low, usually showing a fuzzy state, which will affect the prediction of the model. With the continuous progress of technology and the snowballing demand for the prediction of things, the accuracy rate of the time series needs to be improved, thereby improving the prediction performance. In the past time series prediction models, the fuzzy time model of fuzzy set theory can process fuzzy, incomplete, and uncertain data in time. Therefore, fuzzy information particles can also be considered an essential tool for solving long-term prediction problems [9].

In summary, information particles have a more significant advantage in the prediction of event series; thus, considering the advantages of quantum computing and Long Short-Term Memory (LSTM), the two are combined and applied to the nonlinear time series prediction of fuzzy inference system (FIS) to improve the prediction accurateness rate.

2. Related Works

Quantum computing is a disruptive new technology that may break traditional computing. Many experts and scholars consider it a brand-new computing technology, and it is hoped that it can gradually replace the classic computer in future development and play its role. Preskill (2018) pointed out that scale quantum technology in noise could perform tasks beyond the capabilities of current classic digital computers in the future; however, it would be affected by noise, which limited the dimension of quantum circuits that could be reliably implemented; through continuous explorations, it was expected to find a more accurate quantum gate for fully fault-tolerant quantum computing [10]. Linke et al. (2017) ran a series of algorithms on the two most advanced quantum position sub-computers under dissimilar technological platforms; one was that the openly reachable superconducting transmission tool had restricted connectivity, and another was a fully connected captive ion system. By comparing these two systems with different physical systems, it was understood that the utilization of more connected quantum algorithms and circuits benefited from a better connected qubit system. At the same time, the experiment showed that although quantum systems cannot replace classic computers, the critical factors of quantum computer expansion, such as the connectivity of qubits and the expressive power of gates, would be crucial for future quantum computers and their applications [11]. Gambetta et al. (2017) described the vital way to implement logical memory using superconducting qubits was to utilize the

rotating variety of the superficial code. By analyzing the current status of interconnected superconducting qubit network technology, it was learned that the combination of engineering solutions and quantum architecture laid the foundation for future scalable fault-tolerant quantum computers [12].

Zhao et al. (2017) analyzed the short-term prediction (STP) of traffic and pointed out that the emergence of a large number of traffic data and computing power in recent years had greatly improved the accurateness rate of traffic STP; then, a new LSTM network-based traffic prediction model was proposed. The comparison with the traditional prediction model found that the proposed model could consider the spatiotemporal correlation in the traffic system, which also proved that the proposed model had better performance [13]. Kong et al. (2017) believed that it was very challenging to predict the power load of a single energy user, which involved high volatility and uncertainty; then, a Recurrent Neural Network (RNN) model framework based on LSTM was proposed and tested on a set of public residential intelligent meter data. The results showed that the performance of the proposed LSTM method was superior to other algorithms, which further illustrated the method's effectiveness [14]. Based on the more advanced performance of the Fully Convolutional Network (FCN) in the task of classifying time series, Karim et al. (2017) put forward a time series categorization method for FCN with LSTM RNN sub-modules; the results suggested that this model could significantly improve the performance of FNN; at the same time, after improvements, it was found that the model could show good performance [15].

In summary, the previous explorations on quantum computing are more about analyzing its fault tolerance. Also, LSTM has been commonly utilized for traffic prediction and the establishment of prediction models. However, few studies have integrated the two. Therefore, considering the computational efficiency and fault tolerance of quantum computing, as well as the ability of LSTM to process more complex problems, the two are fused and applied to the nonlinear time series prediction of FIS, thereby providing decision support for decision-makers.

3. Method

3.1. LSTM Neural Network

LSTM network, which often processes time-series data, belongs to RNN. Compared with ordinary RNNs, LSTM can process more extended time series. This method has been well-researched in science and technology, such as image analysis, handwriting recognition, and disease prediction [16, 17].

There will be some gating structures in the LSTM structure diagram. The gates at the index position of each series generally include three types: forget gate, input gate, and output gate. Among them, the forget gate determines whether a thing is forgotten. The LSTM determines the probability of forgetting the hidden cell state of the previous layer. The hidden state of the previous series and the data of this series are passed through the Sigmoid activation function to attain the output value of the forget gate, which can be expressed by Equation (1):

$$f^{(t)} = \sigma(W_f h^{(t-1)} + U_f x^{(t)} + b_f) \quad (1)$$

where $f^{(t)}$ represents the output of the forget gate; σ represents the Sigmoid activation function; W_f , U_f , and b_f represent the coefficients and bias of the linear relationship; $h^{(t-1)}$ represents the hidden state of the previous series; $x^{(t)}$ represents the input of the data of this series.

The input gate of LSTM processes the input of the current series position. It is mainly composed of two parts. One part is to utilize the Sigmoid activation function to attain an output value, and the other is to utilize the tanh activation function to attain another output value. By multiplying the two output values, the state of the cell is updated, which can be represented by Equations (2) and (3):

$$i^{(t)} = \sigma(W_i h^{(t-1)} + U_i x^{(t)} + b_i) \quad (2)$$

$$a^{(t)} = \tanh(W_a h^{(t-1)} + U_a x^{(t)} + b_a) \quad (3)$$

where W_i , and W_a are coefficients and bias of linear relationship; represents the output value of the Sigmoid activation function, and represents the output value of the Tanh activation function.

Before analyzing the output gate, it is necessary to update the cell state. The results of the forget gate and the input gate will both affect the cell state. The cell state is updated by the sum of their products, which can be expressed by Equation (4):

$$C^{(t)} = C^{(t)} \otimes f^{(t)} + i^{(t)} \otimes a^{(t)} \quad (4)$$

where $C^{(t)}$ represents the state of the cell, $f^{(t)}$ represents the output value of the forget gate, and \otimes represents the Hadamard product.

The LSTM output gate can obtain two parts of output after the hidden cell state is updated, which can be expressed by Equations (5) and (6):

$$o^{(t)} = \sigma(W_o h^{(t-1)} + U_o x^{(t)} + b_o) \quad (5)$$

$$h^{(t)} = o^{(t)} \otimes \tanh(C^{(t)}) \quad (6)$$

3.2. Quantum Computing

Quantum computing can enhance the computing speed of neural networks. According to relevant literature, in terms of information processing, qubit neurons are much stronger than classic neuron models [18-20]. In quantum computing, the state of a qubit is a vector of a two-dimensional complex space. It has two polarization states, corresponding to 0 and 1 in a binary state. On this basis, state superposition can also be performed, which can be expressed by Equation (7):

$$|\phi\rangle = c_0|0\rangle + c_1|1\rangle \quad (7)$$

where c_0 and c_1 represent the amplitude of probability, that is, $|\phi\rangle$ either collapses to $|0\rangle$ with a probability of $|c_0|^2$, or $|1\rangle$ with a probability of $|c_1|^2$. Meanwhile, $|c_0|^2$ and $|c_1|^2$ satisfy the following conditions:

$$|c_0|^2 + |c_1|^2 = 1 \quad (8)$$

Quantum logic gates are the most basic in quantum computing. They consist of a single-bit revolving gate and a double-bit controlled-NOT (CNOT) gate. Quantum revolving gates can change the phase of qubits, and CNOT gates can perform XOR operations. To simplify calculations, complex numbers are usually used to represent QS. The probability of $|0\rangle$ is set to the real part, and the probability of $|1\rangle$ is set to the imaginary part. Therefore, the paraphrase of the QS can be expressed by Equation (9):

$$f(\theta) = e^{i\theta} = \cos \theta + i \cdot \sin \theta \quad (9)$$

where θ represents the phase, i is the imaginary unit $\sqrt{-1}$. For the revolving gate, the phase change operation of the complex plane QS can be described as Equation (10):

$$f(\theta_1 + \theta_2) = f(\theta_1) \cdot f(\theta_2) = e^{i(\theta_1 + \theta_2)} \quad (10)$$

where θ_1 and θ_2 represent different quantum phases. The CNOT gate performs the phase reverse operation and is controlled by the parameter γ , which can be described as Equation (11):

$$f\left(\frac{\pi}{2} \cdot \gamma - \theta\right) = \begin{cases} \sin \theta + i \cdot \cos \theta, \gamma = 1 \\ \cos \theta - i \cdot \sin \theta, \gamma = 0 \end{cases} \quad (11)$$

As shown in Equation (11), when γ takes 1, the QS realizes reverse rotation, and when it takes 0, it does not flip. The description and sentences of subsection. The description and sentences of subsection.

3.3. Fuzzy Neural Network (FNN)

The Quantum FNN integrates fuzzy theory and neural network, combining the two theories' advantages. Neural networks have adaptive solid learning capabilities and associative functions, which can apply related theories well. However, they cannot fully utilize some fuzzy information and existing empirical knowledge and have high requirements for sample data. Fuzzy inference can explain the reasoning process well and use expert knowledge well. It requires fewer data samples but too many manual interventions; therefore, the reasoning speed is slow, making adaptive learning difficult. Thus, the effective combination of the two can play a complementary role. The neural networks can improve the network computer learning ability, thereby improving the reasoning ability and providing free parameters that adjust the fuzzy rules for the fuzzy rule base.

In FNNs, fuzzy neurons are essential, mainly fuzzy, fuzzy logic, and defuzzified neurons. At the time of input, the antecedent of the network consists of fuzzy neurons. Its input data can be deterministic or fuzzy, and standardized values are output through operations such as training. The fuzzy logic neuron is in the middle part of the network and is used to realize the fuzzy logic function or fuzzy integration function. Thus, the fuzzy logic neuron is a multi-input single-output neuron. In the latter part of the FNN, the output value needs to be deblurred; therefore, the output value is specific. Its structure is shown in Figure 1:

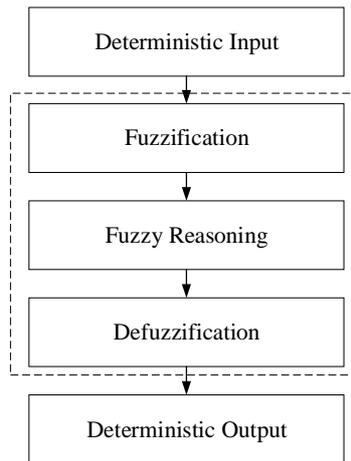


Figure 1. Structure of FNN.

A typical FNN consists of an input layer, a fuzzy layer, a fuzzy inference layer, a defuzzification layer, and an output layer. Its network structure is shown in Figure 2:

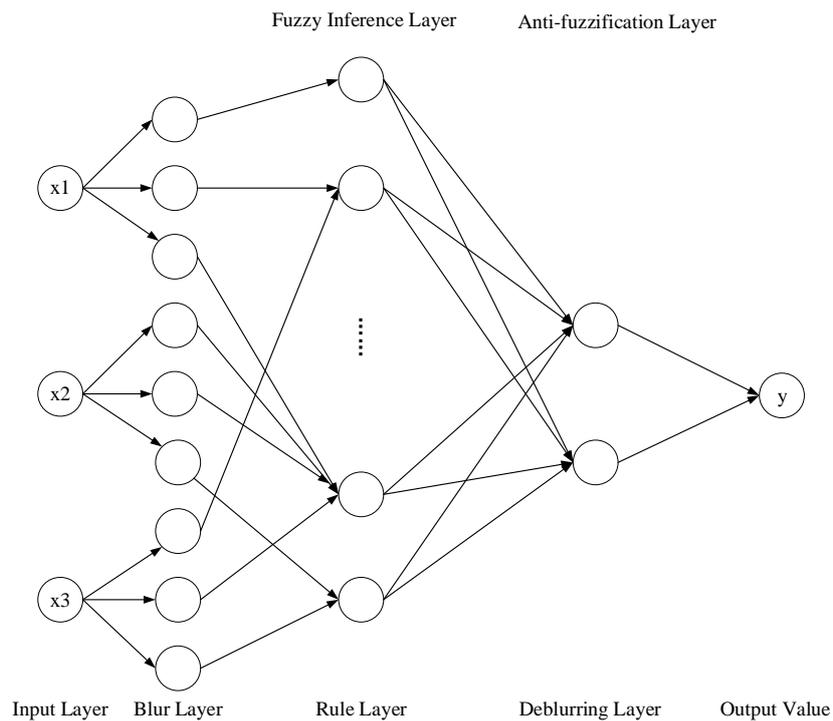


Figure 2. Structure of FNN.

When the number of nodes or weights is changed, or when choosing different membership functions of the fuzzy system, the architecture can be adjusted to realize the system function better.

The concept of information particle was proposed in 1979, which can be expressed by Equation (12):

$$g \stackrel{\Delta}{=} (X \text{ is } G) \text{ is } \lambda \quad (12)$$

where X represents the variable on the universe U , and G is the convex fuzzy subset of the universe U , which is presented by the membership function, indicating that X belongs to the possible probability of the fuzzy subset G . The purpose of using information particles is to simplify complex problems, thereby removing some irrelevant and redundant information. Different forms of information particles will be obtained when the numerical series in the time series is processed with different granulation functions.

4. Experimental Process

4.1. Model Construction

The LSTM RNN is added to the type II fuzzy neural system. The long-term dependency relationship in the series data is processed further through the gate mechanism. Furthermore, the inference structure of a new self-evolving interval type II LSTM FNN (eit2FNN-LSTM) that can be used to synchronize and recognize nonlinear dynamic systems is obtained. The model mainly finds the functional relationship between the input and output values. In this system, there are multiple inputs and outputs. Based on the basic structure, it is assumed that the eit2FNN-LSTM model consists of n input nodes, m output nodes, and six layers. The first layer is the input layer with n nodes; each node represents an input feature value, and the input data will be sent to the fuzzy layer after input. The second layer is the fuzzification layer. Each node in this layer uses the interval type II Gaussian membership function to fuzzify the input value, thereby achieving fuzzification.

In this layer, the j -th input variable $x_{t,j}$ and the k -th rule can be indicated by the bounded interval of the upper membership function and the lower membership function, as shown in Equations (13) and (14):

$$\varphi_{k,j}^{up}(t) = \begin{cases} \exp\left\{-\frac{1}{2}\left(\frac{(x_{t,j}-\mu_{k,j}^l)}{\sigma_{k,j}}\right)^2\right\}, x_{t,j} < \mu_{k,j}^l \\ 1, \mu_{k,j}^l \leq x_{t,j} \leq \mu_{k,j}^r \\ \exp\left\{-\frac{1}{2}\left(\frac{(x_{t,j}-\mu_{k,j}^r)}{\sigma_{k,j}}\right)^2\right\}, x_{t,j} > \mu_{k,j}^r \end{cases} \quad (13)$$

$$\varphi_{k,j}^{lo}(t) = \begin{cases} \exp\left\{-\frac{1}{2}\left(\frac{(x_{t,j}-\mu_{k,j}^l)}{\sigma_{k,j}}\right)^2\right\}, x_{t,j} > \frac{(\mu_{k,j}^l+\mu_{k,j}^r)}{2} \\ \exp\left\{-\frac{1}{2}\left(\frac{(x_{t,j}-\mu_{k,j}^r)}{\sigma_{k,j}}\right)^2\right\}, x_{t,j} \leq \frac{(\mu_{k,j}^l+\mu_{k,j}^r)}{2} \end{cases} \quad (14)$$

where $\varphi_{k,j}^{up}(t)$ denotes the upper membership function, $\varphi_{k,j}^{lo}(t)$ denotes the lower membership function, $\mu_{k,j} \in [\mu_{k,j}^l, \mu_{k,j}^r]$ denotes the center of the j -th characteristic of the k -th rule, and $\vec{\sigma}_{k,j} = [\sigma_{k,1}, \sigma_{k,2}, \dots, \sigma_{k,n}]$ denotes the width of the k -th rule.

The third layer is the rule layer. This layer consists of K nodes, and each node corresponds to the output interval of each rule. The output of the upper layer performs a fuzzy operation through the

algebraic product of the membership function of the rule predecessor. Equations (15) and (16) can obtain the output interval of this layer, and the output is an interval of type I:

$$F_k^{lo}(t) = \prod_{j=1}^n \varphi_{k,j}^{lo}(t) \quad (15)$$

$$F_k^{up}(t) = \prod_{j=1}^n \varphi_{k,j}^{up}(t) \quad (16)$$

The interval can be expressed as: $[F_k^{lo}(t), F_k^{up}(t)], k = 1, 2, \dots, K$

The fourth layer is the interval reduction layer consisting of K nodes, and each node decreases the interval type I fuzzy geometry to type I fuzzy number. Such decrease is implemented by Equation (17):

$$F_k(t) = (1 - \alpha_k)F_k^{lo}(t) + \alpha_k F_k^{up}(t), k = 1, 2, \dots, K \quad (17)$$

where α_k represents the adaptive weight of the uncertainty of the k -th rule, the value range is $[0, 1]$, and $F_k(t)$ represents the output value.

The fifth layer is the loop layer. As the loop structure of the system, LSTM consists of $K * m$ nodes. Each node is a loop rule node, which creates an inner feedback hoop. When the step size is t , the output of the loop rule depends on the output of the current rule and the output value of the node at the previous instant, as shown in Equation (17) to Equation (22):

$$f_{k,q}(t) = \sigma(W_{fx}^{k,q} \cdot F_k(t) + W_{fh}^{k,q} \cdot h_{k,q}(t-1) + b_f^{k,q}) \quad (18)$$

$$i_{k,q}(t) = \sigma(W_{ix}^{k,q} \cdot F_k(t) + W_{ih}^{k,q} \cdot h_{k,q}(t-1) + b_i^{k,q}) \quad (19)$$

$$ct_{k,q}(t) = \tanh(W_{cx}^{k,q} \cdot F_k(t) + W_{ch}^{k,q} \cdot h_{k,q}(t-1) + b_{ct}^{k,q}) \quad (20)$$

$$o_{k,q}(t) = \sigma(W_{ox}^{k,q} \cdot F_k(t) + W_{oh}^{k,q} \cdot h_{k,q}(t-1) + b_o^{k,q}) \quad (21)$$

$$c_{k,q}(t) = f_{k,q}(t) \circ c_{k,q}(t-1) + i_{k,q}(t) \circ ct_{k,q}(t) \quad (22)$$

$$h_{k,q}(t) = o_{k,q}(t) \circ \tanh(c_{k,q}(t)) \quad (23)$$

where $h_{k,q}(t)$ denotes the output value of the loop rule, $F_k(t)$ denotes the output value of the current rule, $h_{k,q}(t-1)$ denotes the output value of the loop node at the previous time, $[W_{fx}^{k,q}, W_{fh}^{k,q}, b_f^{k,q}]$ denotes the weight matrix (WM) and offset term of the forget gate $f_{k,q}$ of the k -th circulation rule, $[W_{ix}^{k,q}, W_{ih}^{k,q}, b_i^{k,q}]$ represents the WM and offset term of the input gate $i_{k,q}$ of the k -th circulation rule, $[W_{cx}^{k,q}, W_{ch}^{k,q}, b_c^{k,q}]$ represents the WM and offset term of the unit state $ct_{k,q}$ of the k -th circulation rule; $[W_{ox}^{k,q}, W_{oh}^{k,q}, b_o^{k,q}]$ represents the WM and offset term of the output gate $o_{k,q}$ of the k -th cycle rule. Meanwhile, in the above equations, $\sigma(z) = 1/(1 + e^{-z})$ and $\tanh(z) = (e^z - e^{-z})/(e^z + e^{-z})$ are the sigmoid activation function and the tanh activation function, respectively.

Finally, there is the output layer, where each node represents an output variable. The output variable of the calculation network is obtained through the weighted mean defuzzification operation. The output value can be expressed by Equation (24):

$$y_q(t) = \frac{\sum_{k=1}^K w_{k,q} h_{k,q}(t)}{\sum_{p=1}^K h_{k,q}(t)}, q = 1, 2, \dots, m \quad (24)$$

where $w_{k,q}$ represents the connection of the k -th cycle rule and the q -th output weight.

4.2. Model learning

The established model structure is carried out mainly through offline and online learning. For offline learning, the network structure will be fixed after training, but due to the concept drift, the series data distribution will change with time. Therefore, offline learning is unsuitable for streaming data in practical applications. On the contrary, online learning can adapt the network structure according to the continuous changes in data and data characteristics, which is more appropriate for processing time-series data. Therefore, considering the problem of adaptive adjustment and reducing the computational burden as much as possible, the online learning method is adopted for training. The learning process of the model structure is shown in Figure 3:

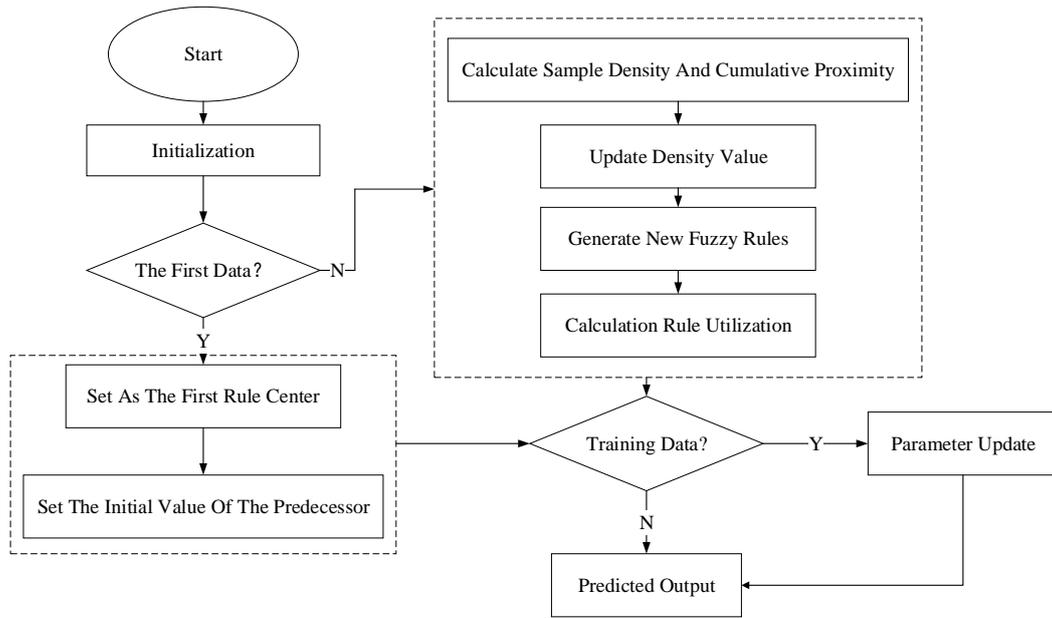


Figure 3. Learning flowchart of the model structure.

On this basis, parameter learning is also carried out. In the case of a single output, errors should be as few as possible, and the error function is shown in Equation (25):

$$E(t) = \frac{1}{2} \sum_q [y_q^d(t) - y_q(t)]^2 \quad (25)$$

where $y_q^d(t)$ represents the q -th actual output value of the t -th sample and $y_q(t)$ represents the q -th predicted output value of the t -th sample. In the network, updating the output value in the network

structure is also necessary before calculating the predicted output value. Therefore, the subsequent parameter update can be achieved by the following equation:

$$w_{k,q}(t+1) = w_{k,q}(t) - \xi \frac{\partial E(t)}{\partial w_{k,q}} \quad (26)$$

where: ξ represents the learning rate.

The gradient corresponding to the network weights between the fifth and sixth layers can be expressed by Equation (27):

$$\frac{\partial E(t)}{\partial w_{k,q}} = \frac{\partial E(t)}{\partial y_q(t)} \frac{\partial y_q(t)}{\partial w_{k,q}} = (y_q^d(t) - y_q(t)) \cdot h_{k,q}(t) \quad (27)$$

where $\delta_{k,q}$ represents the error term at time t . Therefore, the error term corresponding to each layer in the system can be expressed by Equation (28) to Equation (31):

$$\delta_o^{k,q} = \frac{\partial E(t)}{\partial net_o^{k,q}(t)} \quad (28)$$

$$\delta_f^{k,q} = \frac{\partial E(t)}{\partial net_f^{k,q}(t)} \quad (29)$$

$$\delta_i^{k,q} = \frac{\partial E(t)}{\partial net_i^{k,q}(t)} \quad (30)$$

$$\delta_{ct}^{k,q} = \frac{\partial E(t)}{\partial net_{ct}^{k,q}(t)} \quad (31)$$

Therefore, the derivation of the error term passed to any time T can be expressed by Equation (32):

$$\delta_{k,q}(T) = \prod_{j=T}^{t-1} \delta_o^{k,q}(j) W_{oh}^{k,q} + \delta_f^{k,q}(j) W_{fh}^{k,q} + \delta_i^{k,q}(j) W_{ih}^{k,q} + \delta_{ct}^{k,q}(j) W_{ct}^{k,q} \quad (32)$$

4.3. Particle swarm optimized Quantum Echo State Network (QESN)

In QESN, the neurons in different dynamic storage layers are qubit neurons; therefore, the model consists of the output, quantum, and output layers. The data input by the model will change from the natural state to the QS when passing through the input layer. The input value at this time can be expressed by Equation (33):

$$z_c = f\left(\frac{\pi}{2} \cdot x_c + \theta_c\right) \quad (33)$$

where x_c represents the input variable, f represents the mapping relationship; furthermore, the real number is converted into a QS, and θ_c represents the phase parameter, which will be primed before training.

When using particle swarm to optimize QESN, it is mainly to find the best value of δ . The optimized model is shown in Figure 4:

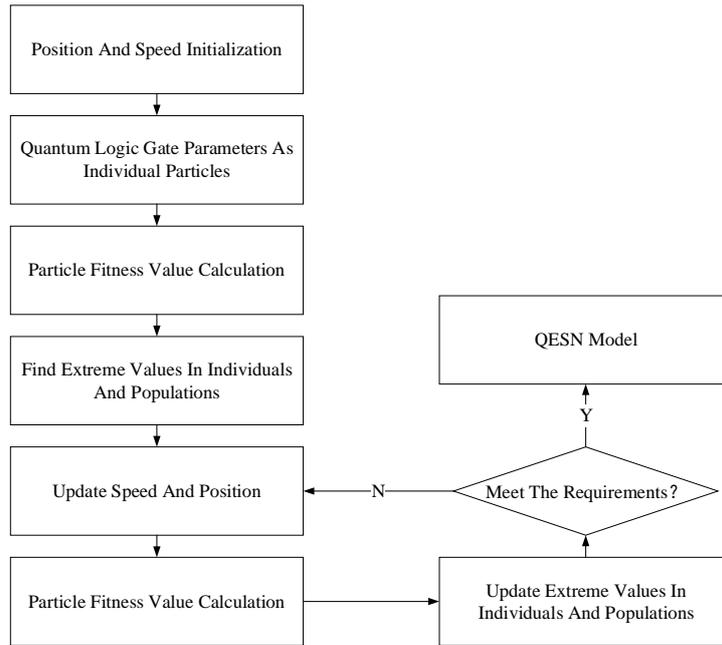


Figure 4. Process of particle swarm optimization.

In the figure above, the original particle position and velocity are set randomly.

5. Results and Discussion

Here, the Mackey-Glass time series is utilized as an example for analysis. The time-lag differential equation is used to generate a time series, as shown in Equation (34):

$$\frac{X(t)}{dt} = \frac{ax(t-\tau)}{1+x^\gamma} - \beta x(t) \quad (34)$$

where $a = 0.2, \beta = 0.1, \gamma = 10$. If $X(t) = 0, X(0) = 1.2, \tau = 16$, the time series corresponding to 1197 data will be obtained, as shown in Figure 5:

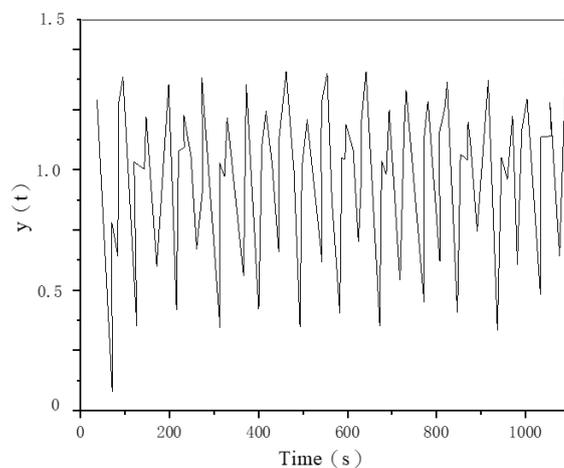


Figure 5. The time series.

When processing the original series, the fuzzy time unit corresponding to the K line is $T=2$, and 598 K lines are obtained, which are divided into 46 groups further, with 46 corresponding information particles. The 30 information particles are used as the training set, and the rest 16 are used as the test set. Therefore, the predicted results are shown in Figure 6:

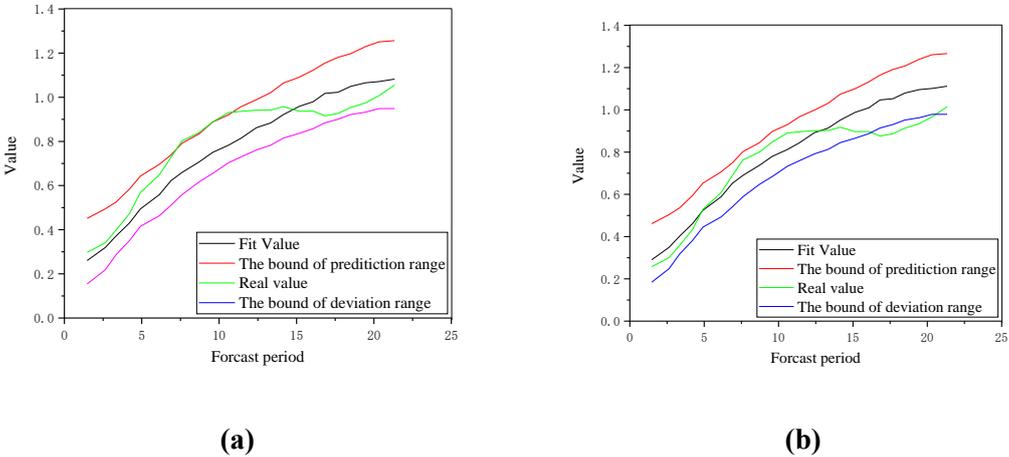


Figure 6. The prediction results.

As shown in the above figure, Figure 6 (a) is the training set, and Figure 6 (b) is the test set. When making predictions, the range of the test set is narrower, and the prediction results are closer; therefore, the prediction accuracy rate is also higher. On this basis, the predictions of different methods are compared, and the results in Figure 7 are obtained:

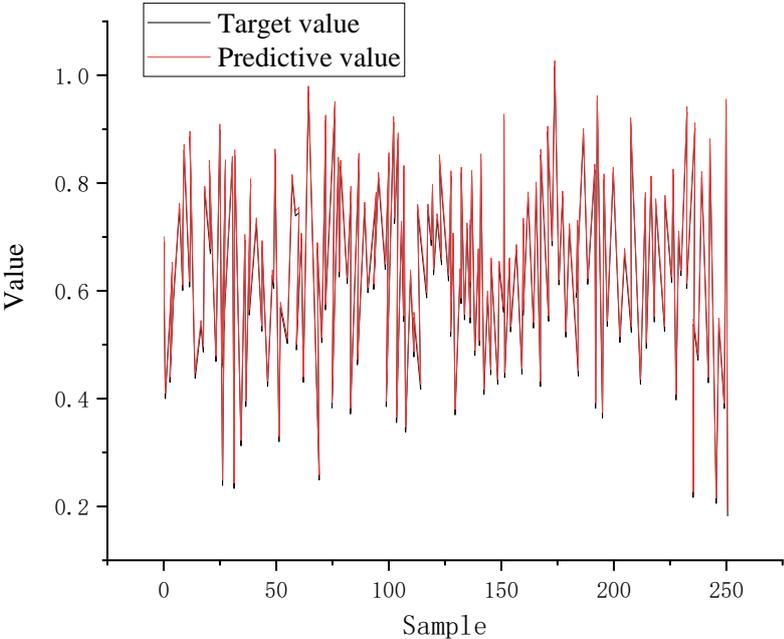


Figure 7. The prediction results.

As shown in the above results, the proposed new self-evolution interval type II LSTM based on quantum computing can reduce the generalization error in MG time series prediction. Therefore, this method can make better predictions, improving the accuracy rate.

In summary, the eit2FNN-LSTM method can realize single-point detection and be used for granular-level prediction when combined with information particles, achieving long-term time series prediction. In the meantime, when the particle algorithm is used to optimize the QESN, it is found that the algorithm can effectively enhance the prediction accuracy rate of the model.

6. Conclusion

The LSTM network and FIS are fused. It is expected that the effective fusion of LSTM, quantum computing, and FIS can improve the ability of the prediction model to process fuzzy information and improve the prediction accuracy rate of nonlinear time series. However, only the Mackey-Glass hybrid time series is analyzed, which may be accidental. In the subsequent exploration, more time series will be predicted to validate the usefulness of the proposed model. Therefore, the FIS based on quantum computing and the new eit2FNN-LSTM can provide more ideas for the in-depth study of single-input value prediction, multi-input value prediction, and online detection series data in nonlinear time series prediction.

References

- [1] Cao Y, Romero J, Olson J P, et al. Quantum chemistry in the age of quantum computing. *Chemical reviews*, 2019, 119(19), pp. 10856-10915.
- [2] Bruzewicz C D, Chiaverini J, McConnell R, et al. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 2019, 6(2), pp. 021314.
- [3] Killoran N, Izaac J, Quesada N, et al. Strawberry fields: A software platform for photonic quantum computing. *Quantum*, 2019, 3, pp. 129.
- [4] Gyongyosi L, Imre S. A survey on quantum computing technology. *Computer Science Review*, 2019, 31, pp. 51-71.
- [5] Gao K, Jo S B, Shi X, et al. Over 12% efficiency nonfullerene all-small-molecule organic solar cells with sequentially evolved multilength scale morphologies. *Advanced Materials*, 2019, 31(12), pp. 1807842.
- [6] Liu L, Ding L, Rovere M, et al. A cellular complex of BACE1 and γ -secretase sequentially generates A β from its full-length precursor. *Journal of Cell Biology*, 2019, 218(2), pp. 644-663.
- [7] Wu Y, Yuan M, Dong S, et al. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, 2018, 275, pp. 167-179.
- [8] Sherstinsky A. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 2020, 404, pp. 132306.
- [9] Kim H Y, Won C H. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications*, 2018, 103, pp. 25-37.
- [10] Preskill J. Quantum Computing in the NISQ era and beyond. *Quantum*, 2018, 2, pp. 79.
- [11] Linke N M, Maslov D, Roetteler M, et al. Experimental comparison of two quantum computing architectures. *Proceedings of the National Academy of Sciences*, 2017, 114(13), pp. 3305-3310.
- [12] Gambetta J M, Chow J M, Steffen M. Building logical qubits in a superconducting quantum computing system. *npj Quantum Information*, 2017, 3(1), pp. 1-7.
- [13] Zhao Z, Chen W, Wu X, et al. LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 2017, 11(2), pp. 68-75.
- [14] Kong W, Dong Z Y, Jia Y, et al. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid*, 2017, 10(1), pp. 841-851.
- [15] Karim F, Majumdar S, Darabi H, et al. LSTM fully convolutional networks for time series classification. *IEEE access*, 2017, 6, pp. 1662-1669.

- [16] Reimers N, Gurevych I. Reporting score distributions makes a difference: Performance study of lstm-networks for series tagging. arXiv preprint arXiv:1707.09861, 2017.
- [17] Ullah A, Ahmad J, Muhammad K, et al. Action recognition in video series using deep bi-directional LSTM with CNN features. *IEEE Access*, 2017, 6, pp. 1155-1166.
- [18] Kim J, El-Khamy M, Lee J. Residual LSTM: Design of a deep recurrent architecture for distant speech recognition. arXiv preprint arXiv:1701.03360, 2017.
- [19] Yu W, He X, Yang Z, et al. Sequentially responsive biomimetic nanoparticles with optimal size in combination with checkpoint blockade for cascade synergetic treatment of breast cancer and lung metastasis. *Biomaterials*, 2019, 217, pp. 119309.
- [20] Bediako J K, Sarkar A K, Lin S, et al. Characterization of the residual biochemical components of sequentially extracted banana peel biomasses and their environmental remediation applications. *Waste management*, 2019, 89, pp. 141-153.